IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 July 2024, accepted 10 August 2024, date of publication 14 August 2024, date of current version 23 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3443517

APPLIED RESEARCH

Chip Analysis for Tool Wear Monitoring in Machining: A Deep Learning Approach

ATIQ UR REHMAN[1], **TAHIRA SALWA RABBI NISHAT**[1], **MOBYEN UDDIN AHMED**[1], **SHAHINA BEGUM**[1], **AND ABHISHEK RANJAN**[2]

[1]Artificial Intelligence and Intelligent Systems Research Group, School of Innovation, Design and Engineering, Mälardalen University, 722 20 Västerås, Sweden
[2]Research and Development Team, SECO Tools AB, 737 30 Västmanland, Sweden

Corresponding author: Mobyen Uddin Ahmed (mobyen.uddin.ahmed@mdu.se)

This work was supported in part by the DIGital Twins for Industrial COGnitive Systems (DIGICOGS) through the Industry 4.0 and Artificial Intelligence funded by Vinnova Diarienr and the Innovation Program Process Industrial IT and Automation (PiiA) under Grant 2019-0532, in part by the Cognitive Predictive Maintenance and Quality Assurance using Explainable Ai and Machine Learning (CPMXai) under Project 2021-03679, and in part by Sweden's Innovation Agency (VINNOVA) and Mälardalen University.

ABSTRACT Recent strides in integrating artificial intelligence (AI) with production systems align with the trend towards highly automated manufacturing, demanding smarter machinery. This dovetails with the overarching vision of Industry 4.0, moving beyond conventional models towards employing AI for real-time modeling of production processes, enabling adaptable and learning-enabled models. This study focuses on leveraging cutting-edge deep learning techniques to monitor and classify tool wear using authentic image data from machining processes. Various deep learning algorithms, including CNN, AlexNet, EfficientNetB0, MobileNetV2, CoAtNet-0, and ResNet18, are explored for monitoring and measuring wear through images of machining chips. The collected images of machining chips are categorized as 'Accepted', 'Unaccepted', and 'Optimal'. Due to imbalanced datasets, the study investigates two distinct strategies: upsampling and downsampling. The study also aimes to enhance sensitivity for a specific minority class to meet industrial requirements. The study showed that upsampling enhanced accuracy and almost fulfilled the stated requirements, whereas downsampling did not achieve the desired outcomes. The study evaluates and compares the effectiveness of recently introduced deep learning algorithms with other CNN-based architectures in classifying tool wear states in real-world scenarios. It sheds light on the challenges faced by the machining industry, particularly the prevalent issue of class imbalance in real-world machining data. The observed results indicate that ResNet18 and AlexNet outperform other algorithms, achieving a weighted average accuracy of 96% for both multiclass and binary classification problems when considering upsampled datasets. Consequently, the study concludes that both ResNet18 and AlexNet demonstrate adaptability to class imbalances, generalization to real-world machining scenarios, and competitive accuracy.

INDEX TERMS Deep learning, industry 4.0, machining, neural networks, predictive maintenance, tool wear.

I. INTRODUCTION

In the world of machining, tools gradually wear out over time until they reach a critical point and fail [1], [2]. This gradual increase in tool wear determines how long a tool will last before needing to be replaced [3]. As wear increases, it creates more friction and force [4], leading to vibrations and power increases [5], [6]. If tools are not replaced on time, it can harm the machine [7].

To prevent sudden tool failures, it's crucial to monitor wear from the start [8]. Additionally, increased wear affects the quality of the finished product, like surface smoothness and accuracy [9]. Understanding how wear works is vital during machining, as it greatly impacts the final result [7]. It also affects other factors, like chip production and surface hardness [2].

Therefore, a reliable system is needed to detect tool wear early on to avoid unexpected downtime or working with a worn-out tool that is not able to produce the desired results [10]. Online monitoring systems that sense

The associate editor coordinating the review of this manuscript and approving it for publication was Yu Liu.

increasing wear can help make the most of a tool's lifespan [11].

Tool wear monitoring systems fall into two groups: direct and indirect methods. Direct methods measure wear using techniques like electric resistance, optics, or imaging, but they have limitations in certain situation [12]. These limitations include material dependency, limited applicability to certain materials, surface finish and contaminants, sensitivity to environmental conditions, and complexity in multi-tool environments, among others. Indirect methods gather data during cutting operations that give clues about tool wear. While they don't directly measure wear, they can provide valuable information [12]. For example, they might use temperature, cutting force, vibration, surface roughness, or other sensing measurements [13], [14]. However, implementing these methods can be complex and costly due to the installation of extra equipment.

Recently, there's been a growing interest in using artificial intelligence (AI) alongside production systems [15]. This aligns with a trend towards highly automated production, which requires more intelligent machines [16]. These advancements are driving us towards the development of Industry 4.0 [17]. Unlike traditional models, which can be quite simplified and evaluative [18], there's a shift towards using AI for real-time modeling of production processes [19]. This allows for models that can adapt and learn as they are used.

Among the recent develpments on AI algorithms, a notable work is done by Dai et al [20]. In their study on CoAtNet, they delve into the fusion of convolution and attention in machine learning, examining it from two fundamental angles: generalization and model capacity. They find that convolutional layers exhibit superior generalization, converging swiftly due to their robust prior inductive bias. On the other hand, attention layers boast higher model capacity, especially benefiting from larger datasets. By combining both convolutional and attention layers, a synergistic enhancement in both generalization and capacity is achieved. Yet, a critical challenge they faced was effectively combining convolutional and attention layers to strike a balance between accuracy and efficiency. In response, they made two significant observations. Firstly, they noted that the widely used depthwise convolution can be seamlessly integrated into attention layers through a straightforward relative attention mechanism. Secondly, by appropriately stacking convolutional and attention layers, remarkable gains in both generalization and capacity were achieved. Building upon these insights, they put forth a straightforward yet highly effective network architecture called CoAtNet, harnessing the strengths of both Convolutional Neural Networks (ConvNets) and Transformers.

The dataset used for this work is hugely imbalanced and categorized with three classes: 'Accepted', 'Unaccepted', and 'Optimal' and it has the highest instances of the 'Accepted' class, following by 'Unaccepted', and 'Optimal' classes. This work focuses on advanced deep learning techniques to monitor and classify tool wear using authentic image data from machining processes. Specifically, it explores the performance of various deep learning algorithms, including CNN, AlexNet, EfficientNetB0, MobileNetV2, CoAtNet-0, and ResNet18, in classifying tool wear states with imbalanced data and opting for result that is sensitive to a minor class. The overall goal is to find a technique that provides good accuracy and precise response, especially with the 'Unaccepted' labelled class. The study also addresses the challenges of class imbalance in real-world machining data by investigating upsampling and downsampling strategies.

The main contributions of this research are:

1) Evaluation of the effectiveness of state-of-the-art deep learning algorithms in classifying tool wear.
2) Comparison of different deep learning models in handling class imbalance in real-world machining data.
3) Demonstration of the adaptability and generalization capabilities of ResNet18 and AlexNet, achieving high accuracy in both multiclass and binary classification problems.

The rest of the article is organised as follows: Literature review (Section II), Methodology (Section III), Experimental Setup(Section IV), Performance evaluation (Section V), and Conclusions (Section VI).

II. LITERATURE REVIEW

A. TOOL WEAR MONITORING USING AI

The integration of Artificial Intelligence (AI) techniques with precision cutting force measurements has become a focal point in current research. This combination enhances the processing and analysis of acquired force signals through various AI methodologies. Among these, neural networks are prominently featured in studies related to tool wear monitoring. For instance, Wang and Cui [21] employed an auto-associative neural network approach, while Freyer et al. [22] utilized two distinct strategies based on orthogonal and unidirectional force components, both incorporating time-delay neural networks. Results indicate comparable accuracy in monitoring tool conditions during the turning process. Additionally, Kuram and Ozcelik [23]applied regression analysis and fuzzy logic to estimate flank wear, surface roughness, and cutting forces during micromilling, showing successful application for reliable process output estimation. Wang et al. [24] introduced a complex learning system using hidden Markov models, radius basis functions, and support vector machines, achieving an accuracy rate above 99% for real-time monitoring during titanium alloy milling. Another widely adopted approach for tool condition monitoring involves Adaptive Neuro-Fuzzy Inference Systems (ANFIS) utilizing the Takagi Sugeno fuzzy inference model. Xu et al. [25] developed an intelligent ANFIS model, incorporating an improved particle swarm optimization method for tool wear assessment during milling. Experimental results demonstrated superior prediction accuracy compared to other intelligent approaches. McParland et al. [26] applied cutting force data from turning medical-grade CoCrMo alloy to estimate tool wear based on a Bayesian hierarchical Gaussian process model. The Bayesian approach significantly improved tool wear estimation accuracy, even for non-linear wear rates, making it applicable for online tool life prediction.

B. PROGRESS IN DEEP LEARNING

Since the emergence of AlexNet, Convolutional Neural Networks (ConvNets) have dominated computer vision [27]. Meanwhile, inspired by the success of self-attention models like Transformers in natural language processing, efforts have been made to integrate attention mechanisms into computer vision. Recently, in a study, Dosovitskiy et al. [28] the Vision Transformer (ViT) demonstrated competitive performance on ImageNet-1K using primarily vanilla Transformer layers. When pre-trained on the extensive JFT-300M dataset, ViT achieved comparable results to state-of-the-art ConvNets, hinting at the potentially greater scalability of Transformer models. However, ViT's performance lags behind ConvNets in low-data scenarios. Even with additional pre-training, its accuracy on ImageNet is notably lower than ConvNets of similar size [29]. This suggests that vanilla Transformer layers may lack certain inductive biases inherent to ConvNets, necessitating more data and computational resources. Consequently, recent research has focused on merging ConvNet and Transformer attributes [30], [31], [32], but often lacks a systematic understanding of their respective roles. The challenge lies in effectively combining them to strike a balance between accuracy and efficiency. CoAtNet, a model that integrates convolutional and attention layers, achieves state-of-the-art performance under comparable resource constraints across various data sizes. In low-data settings, CoAtNet benefits from ConvNet's strong generalization properties due to its favorable inductive biases. With abundant data, CoAtNet not only leverages Transformer models' superior scalability but also attains faster convergence and improved efficiency [20]. This survey by Shafiq and Gu [33] explores the methods and advantages of deep ResNets, their performance on ImageNet, and their potential applications beyond image classification tasks that makes it ideal to use for our purpose. In another paper Gupta et al. [34], have used different deep learning techniques along with ResNet18 to find the optimal model to classify defective and non-defective casting for industrial uses. Similarly, AlexNet have been used for transfer learning model in several industrial applications, such as detecting fabric defects in garments sector by Şeker [35] analyzing casting surface defect images of a metal pump impeller for performance testing in the study by Thalagala and Walgampaya [36].

III. METHODOLOGY

The experimentation is performed using different deep learning architectures including a simple CNN architecture and some advanced architectures like AlexNet, EfficientNetB0, MobileNetV2, and Resnet18. Further, in the recent developments of deep learning architectures, some sophisticated architectures integrating convolution and self-attention are also proposed. One such architecture is CoAtNet. In this study, CoAtNet-0 is evaluated for its performance to see how such architectures respond to these specific datasets. Subsequently, PyTorch and Torchvision libraries were used for AlexNet, EfficientNetB0, MobileNetV2, and Resnet18.

In this study, some common transformation techniques were used for every model. Such as, all input images were resized to (224, 224) pixels to ensure consistent size for model compatibility and normalized within a range of [−1, 1] using 'transforms.Normalize((0.5,), (0.5,))' from PyTorch.

Data augmentation techniques from PyTorch's 'torchvision.transforms'—such as random flips, rotation, resized crop, color jitters, affine transformations, and Gaussian blur—enhance the diversity of minority class samples, addressing issues like class separability, noise sensitivity, and computational complexity. Augmented examples are shown in Figure 12. For downsampling, we randomly removed 50% of the majority of 'Accepted' class, leaving 'Optimal' and 'Unaccepted' classes unchanged. These steps ensured a comprehensive preprocessing and augmentation approach.

Further background details on the architectures are provided below.

A. CONVOLUTIONAL NEURAL NETWORK (CNN)

A simple CNN architecture is used to test its performance on the machining dataset. The network consists of a feature extraction module followed by fully connected layers for classification. The feature extraction module comprises two convolutional layers, each succeeded by a Rectified Linear Unit (ReLU) activation and max-pooling operation. The first convolutional layer processes three input channels with a 3 × 3 kernel and 16 output channels, while the subsequent layer has 16 input channels and 32 output channels. The ReLU activations introduce non-linearity, and max-pooling layers reduce spatial dimensions. The flattened output from the feature extraction module is then forwarded through two fully connected layers, consisting of 128 units and ReLU activation in the first layer, and a linear layer producing the final classification output with a user-specified number of classes in the second layer. The overall architecture is designed for simplicity and efficiency, making it suitable for the machining image classification tasks. The input images are preprocessed using a set of transformations, including resizing to (224, 224) and normalization (pixel values scaled to a range of [−1, 1]). This simple CNN architecture served as a fundamental component in our experimental framework, demonstrating its efficacy in capturing discriminative features for machining image classification. Further details of the architecture are shown in Figure 1. corresponding model summery is presented model in table 1.

B. COATNET

This architecture explores the optimal integration of convolution and self-attention mechanisms in a computational block, addressing two key aspects: (i) combining these mechanisms effectively, and (ii) vertically stacking them in a network. The similarities between depthwise convolution and self-attention are noted, emphasizing their per-dimension weighted sum approach within a receptive field. The study by Dai et al [20] further highlights that self-attention excels in capturing intricate relational interactions but may be prone to overfitting with limited data, while convolution benefits from translation equivalence, making it superior for smaller datasets. Additionally, the authors stress the importance of receptive field size. They propose a method to combine convolution and self-attention by summing a global static convolution kernel with the adaptive attention matrix. They also introduce a variant of relative self-attention. In terms of network layout design, they compare strategies to manage computational complexity associated with global attention and select the C-C-T-T multi-stage layout for CoAtNet, considering factors such as generalization, model capacity, transferability, and efficiency. This comprehensive study provides valuable insights into the strengths and considerations of combining these critical mechanisms in neural networks. In this study, we have evaluated this model on our dataset to test its capabilities on data containing some real challenges. In a recent study by Basit et al. [37], the authors have used CoAtNet-0 to classify oil spillage over sea on the spaceborne synthetic aperture radar (SAR) data. In this study, a similar CoAtNet-0(without pretrained weights) architecture was applied on the dataset that contains 3 classes. The architecture of the CoatNet-0 model is presented in Figure 2.

The model summery of the CoatNet-0 is provided in Table 2.

C. RESNET18

For our work, we have utilized an 18-layer CNN deep learning model known as Residual Network(ResNet18) [38] which is specially developed for image classification tasks. This model is a pretrained model on the ImageNet dataset which includes approximately 1.3 million images. First, to adapt our input images for processing, they resized into the 224*224 resolution required for our model and normalized before feeding to the model.

ResNet is a widely used deep learning model for its innovative use of ‘skip connections’. These ‘skip connections’ address one of the most frequently occurring problems ‘vanishing gradient’ where the model ceases to learn further. Among all the versions of ResNet, the standard 18-layer version seemed to be an ideal balance for our needs; it is sufficiently deep to capture complex features without being overly extensive comparing the size of our dataset. The other versions might not offer proportional benefits for our specific dataset.

The ResNet-18 model had 18 layers for efficiently classifying images of different objects from different categories. To adapt the model to our specific needs, we added 3 fully connected(fc) layers comprising 512, 256, and 128 neurons respectively, and added a dropout of 0.4 after the fc layer to reduce any overfitting and we have used SGD optimizer for this purpose. We have applied this model on two datasets(containing identical images), one with 2 outputs and another with 3 outputs, the final output layer was different on training on each dataset. This model worked well on both upsampled and downsampled data. However, on actual data, its performance is not acceptable. The architecture of the pretrained ResNet18 is shown in Figure 3.

D. ALEXNET

Following our experiment with ResNet18, we shifted our focus on a simpler architecture, AlexNet. As demonstrated by Krizhevsky, Sutskever, and Hinton [39], AlexNet is one of the pioneering and most influential deep learning models comprising 5 convolutional layers and 3 layers of fully connected neurons. It addresses the overfitting issue in deep neural networks by using dropout techniques. In our experiment with AlexNet, we introduced two additional fully connected layers with configurations of 640 neurons (accompanied by a 50% dropout rate) and 256 neurons, respectively. The output layer was modified to fit the requirements of our two datasets, one with three classes and the other with two classes. For training, we have used it with pre-trained weights from ImageNet [40] with Stochastic Gradient Descent (SGD) as the optimizer. We transformed our image dataset into 224*224 resolution and normalized them before the training.

Remarkably, our AlexNet model achieved competitive performance, showing equally well performance as the previous model. It demonstrated higher accuracy for both the three-class and two-class datasets, outperforming all other models we considered for this research. This outcome shows AlexNet’s efficiency with medium to small datasets in deep learning tasks, even when compared to more complex models. 3 presents the architecture of the pretrained AlexNet used in this study.

E. EFFICIENTNETB0

A pre-trained baseline EfficientNet model was tested while freezing it and adding additional layers for our 3 class and 2 class outputs respectively. EfficientNetB0 is also a CNN-based architecture that has been trained on images from the ImageNet database. The EfficientNetB0 architecture employs a compound scaling method that offers a coordinated way of enlarging convolutional neural networks. Unlike conventional scaling practices that typically expand a single dimension—like width, depth, or resolution —which may result in low performance or inflated computational demands, this method enhances all three dimensions in a balanced manner that helps to optimize performance while maintaining computational efficiency [29]. The pre-trained weights were used, so instead of starting from scratch (random initialization), the model starts with patterns and features already learned from a vast and varied dataset. 3 fully connected layers were added at the end of the model comprising of 1280, 640, and 256 neurons respectively for the model to adapt to our 3 class dataset. After that, the model was fine-tuned

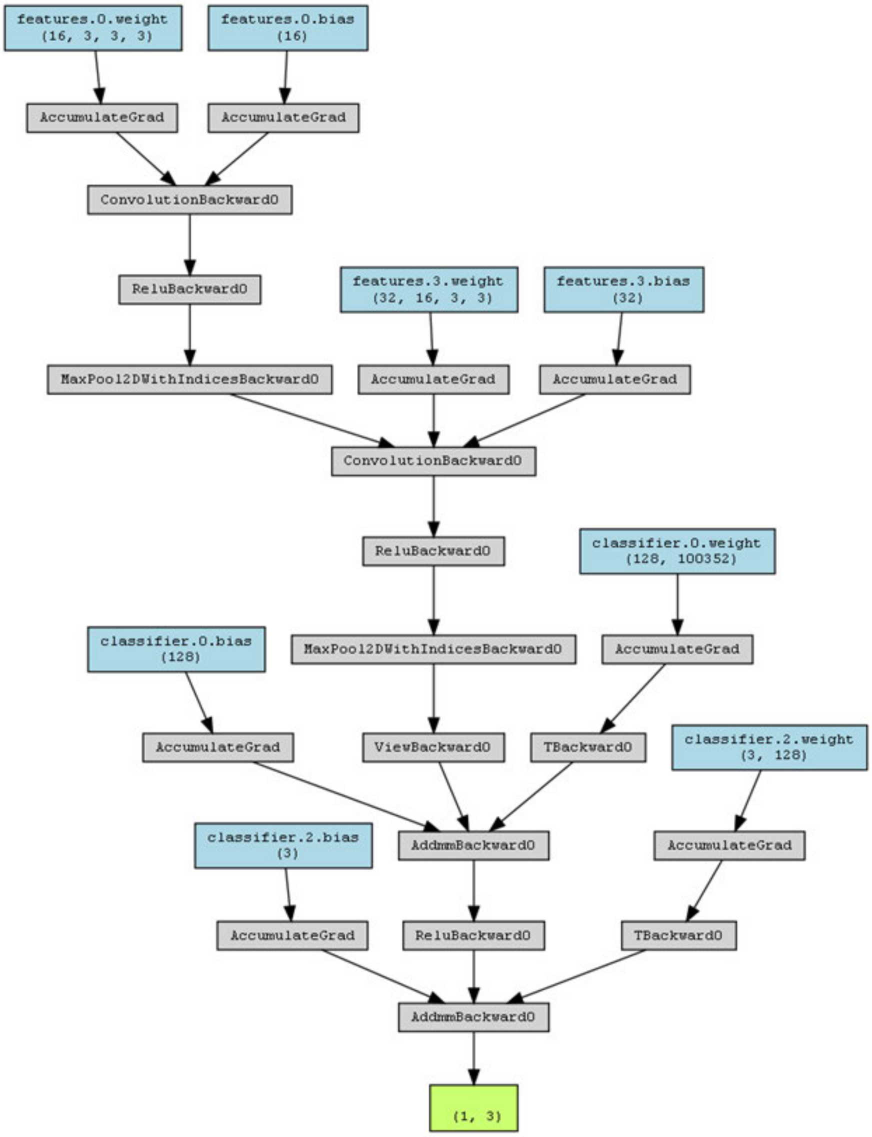


FIGURE 1. CNN architecture: Mainly two sequential 2D convolution blocks with RelU, followed by fully connected linear layers.

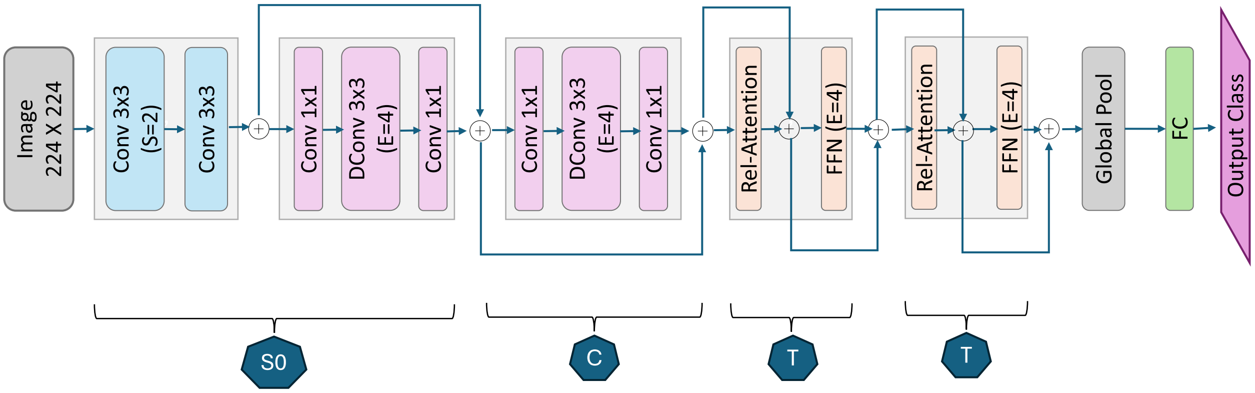


FIGURE 2. CoAtNet-0 architecture.

TABLE 1. Summary table for the CNN model.

Layer	Type	Output Shape	Details
1	Conv2d + ReLU	(16, 224, 224)	3 input channels, 16 output channels, kernel size 3, padding 1
2	MaxPool2d	(16, 112, 112)	kernel size 2, stride 2
3	Conv2d + ReLU	(32, 112, 112)	16 input channels, 32 output channels, kernel size 3, padding 1
4	MaxPool2d	(32, 56, 56)	kernel size 2, stride 2
5	Flatten	(32 * 56 * 56)	-
6	Linear + ReLU	(128)	32 * 56 * 56 input features, 128 output features
7	Linear	(3)	128 input features, num_classes output features

TABLE 2. Summary table for the CoAtNet-0 model.

Layer	Type	Output Shape	Details
1	Conv2d + BN + GELU	(64, 112, 112)	3 input channels, 64 output channels, kernel size 3, stride 2, padding 1
2	MBConv	(96, 56, 56)	64 input channels, 96 output channels, 2 blocks, downsample Yes
3	MBConv	(192, 28, 28)	96 input channels, 192 output channels, 3 blocks, downsample Yes
4	Transformer	(384, 14, 14)	192 input channels, 384 output channels, 5 blocks, downsample Yes
5	Transformer	(768, 7, 7)	384 input channels, 768 output channels, 2 blocks, downsample Yes
6	AvgPool2d	(768, 1, 1)	kernel size (7, 7)
7	Linear	(num_classes)	768 input features, num_classes output features

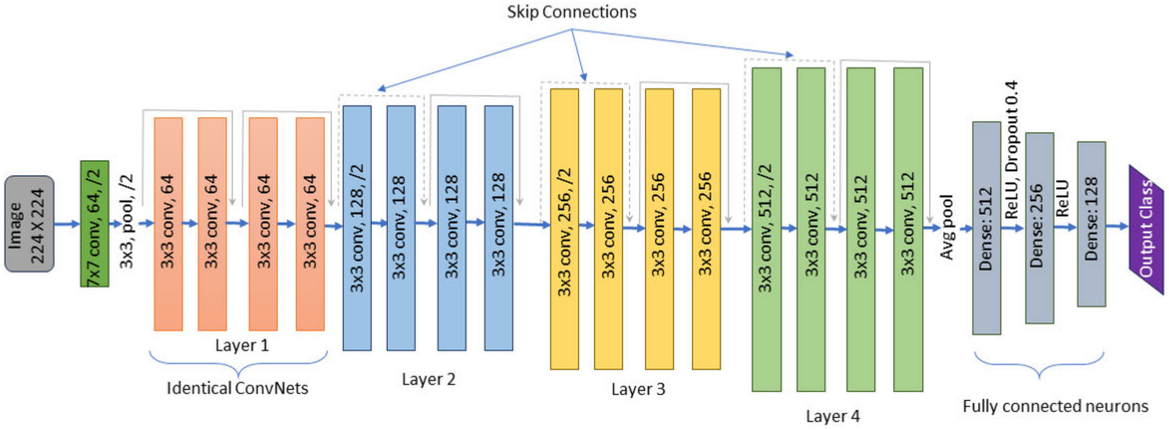


FIGURE 3. Resnet18 pretrained architecture used for transfer learning.

to get the maximized accuracy and paired it with SGD optimizer. For better understanding, the architecture of the EfficientNetB0 is given in Figure 5.

The EfficientNetB0 model demonstrated excellent performance on the upsampled data for the three-class system. However, it showed poor performance on the actual imbalanced data and downsampled data. So, we didn't further progress with the binary class dataset. As EfficientNetB0 has approximately 237 layers we assume that the actual data and downsampled data were comparatively too small for the model to be generalized, resulting in overfitting for our particular data.

F. MOBILENETV2

In our previous experiment with baseline EfficientNet, we observed impressive performance was confined to scenarios involving upsampled data only. This led us to presume that EfficientNetB0's deapth could be excessive for our dataset's specific characteristics. So, we shifted our minds to another lightweight architecture known as MobileNetV2. MobileNetV2 is distinguished by its inverted residual structure where the input and output of the residual block are thin bottleneck layers [41]. Prior research has shown that MobileNetV2 is highly efficient at extracting features from segments, and it has a good potential for use in mobile vision applications [42]. therefore we decided to use it for our research.

In our research, we have used the pretrained MobileNetV2 model, originally trained on ImageNet dataset. Our preprocessing steps mirrored those used in earlier experiments to ensure consistency across models. The model is already 53 layers deep, we added two fully connected layers with 640 and 256 neurons, respectively, incorporating a 20% dropout rate and paired it with Stochastic Gradient Descent (SGD) optimizer. Finally, we adapted the output for both our 3-class datasets and hypertuned it for optimal accuracy.

The architecture of the model is prvided in Figure 6 for better visualization.

However, like the previous model, MobileNetV2's performance was markedly better with upsampled data. Given this outcome, we decided not to proceed with testing on the two-class dataset.

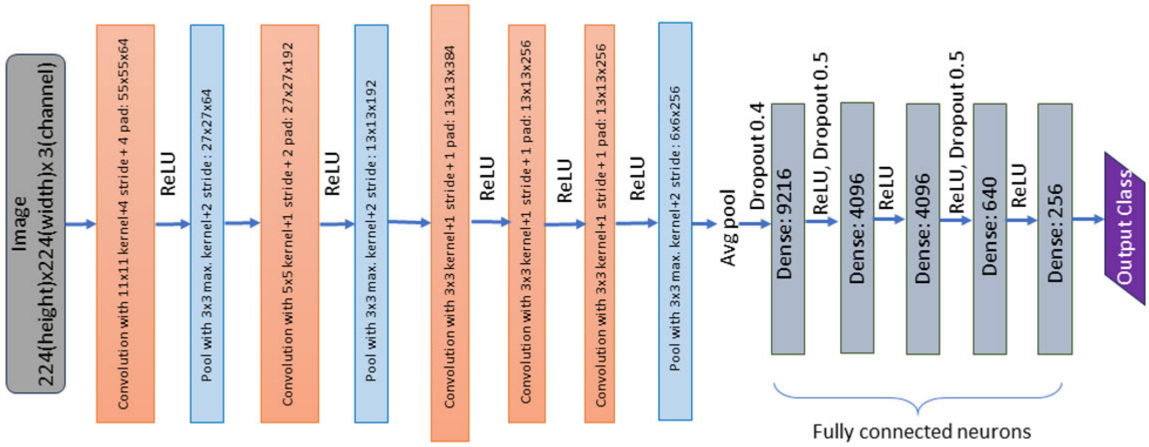


Image
224(height)x224(width)x 3(channel)
Convolution with 11x11 kernel+4 stride + 4 pad: 55x55x64
ReLU
Pool with 3x3 max. kernel+2 stride : 27x27x64
Convolution with 5x5 kernel+1 stride + 2 pad: 27x27x192
ReLU
Pool with 3x3 max. kernel+2 stride : 13x13x192
Convolution with 3x3 kernel+1 stride + 1 pad: 13x13x384
ReLU
Convolution with 3x3 kernel+1 stride + 1 pad: 13x13x256
ReLU
Convolution with 3x3 kernel+1 stride + 1 pad: 13x13x256
ReLU
Pool with 3x3 max. kernel+2 stride : 6x6x256
Avg pool
Dropout 0.4
Dense: 9216
ReLU, Dropout 0.5
Dense: 4096
ReLU
Dense: 4096
ReLU, Dropout 0.5
Dense: 640
ReLU
Dense: 256
Output Class
Fully connected neurons

FIGURE 4. AlexNet pretrained architecture used for transfer learning.

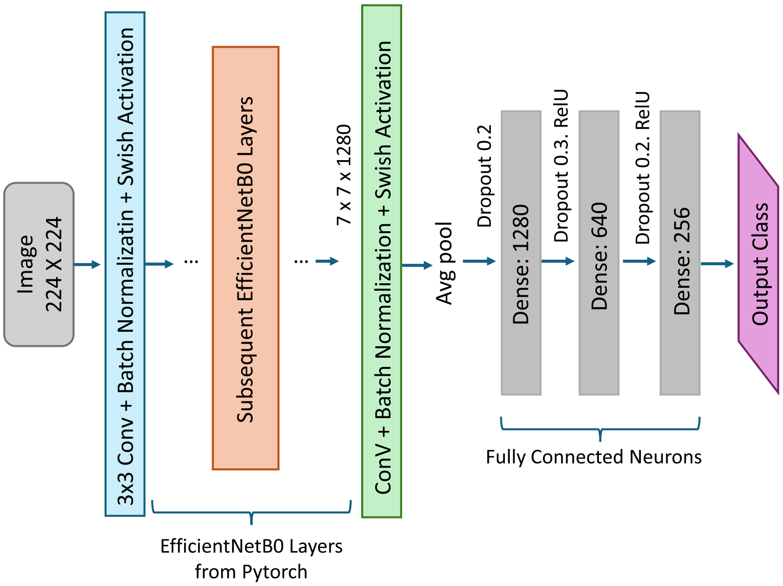


FIGURE 5. EfficientNetB0 pretrained architecture used for transfer learning.

IV. EXPERIMENTAL SETUP

A general flow of chip collection process is provided in Figure. 8. Chip is formed when a metallic workpiece is machined based on Computer Aided Manufacturing (CAM) program instructions. Here, the cutting Parameters: Cutting parameters are derived from engineering machining calculations based on the workpiece's specific metal and cutting insert combination. Cutting Parameter such as Depth of Cut (mm), Feed Rate(mm/s) and Cutting Speed (revolution / s or mm /s) are specific parameters which determines the machining or material removal rates chip is removed optimally on balance of these 3 parameters for any given material types. Depth of Cut (mm): it is amount of insert inside the workpiece while material is being removed. In the images below you can see the formation of chips. Depth of cut determines the width of chip whereas federate or feed determines the thickness of chip. The length of chip can be determined by many parameters but primarily on Chip breaker Design on Insert and combination of Depth of cut, Feed and Cutting Speed. For better visualization 2 figures Figure: 7a and Figure: 7b are added.

During the machining experiments, the ambient temperature of the workshop was maintained between 21-25°C. To manage the heat generated during the cutting process and to improve the quality of the chips produced, we employed a e.g., 'water-soluble cutting fluid' delivered through a e.g., 'flood cooling system'. The CNC turning machine used in our experiments was regularly calibrated and maintained to ensure optimal stability. The experiment was conducted on a Turning Machine with Turning Inserts on Various Cut Data Parameters to identify the right machining parameter for a given workpiece. These cut data parameters are Depth of Cut, Federate and cutting speed. However, the scope of study was limited to Identify the optimal cutting parameters (DoC, Feed, Cutting Speed) which was prime part of study as they play crucial role in determining the quality of chips and

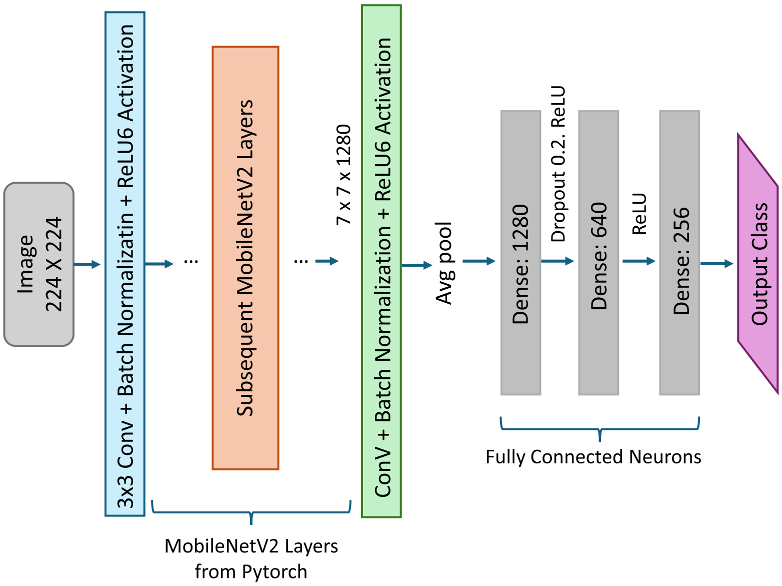


FIGURE 6. MobileNetV2 pretrained architecture used for transfer learning.

Cooling/ Lubrication conditions, temperature and machine stability are known to determine the tool life.

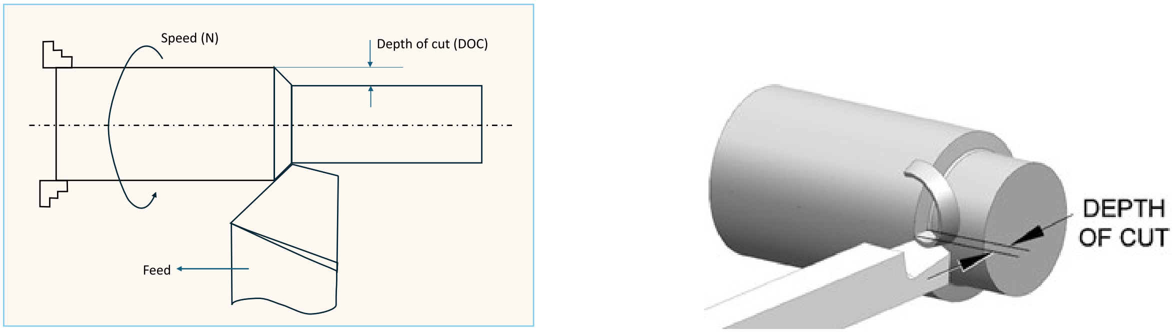
After machining, the chips are carefully collected from the predefined cutting zones. The process involves halting the machine at regular intervals to gather the chips, ensuring that they represent the specific machining parameters used at each stage (e.g., depth of cut, cutting speed, feed rate). The collected chips are then arranged on a display table known as a Chip Chart. This table is designed with a grid layout where each cell corresponds to a unique combination of machining parameters. The grid helps in systematically organizing the chips, making it easier to record and analyze the conditions under which each type of chip was produced. Each grid cell on the Chip Chart is labeled with the specific machining parameters used to produce the chips placed in that cell. This includes details such as the depth of cut, cutting speed, and feed rate. By doing so, the Chip Chart serves as a visual and documented record of the experimental conditions. A Chip chart is the output of experiment to determine within which machining range of machining parameters we get which kind of chips. This is experimental. In the above image experiment was first set up in following order:

1) Cutting Speed V_c is determined.
2) Choices of Feed, f, are determined.
3) Choices of Depth of Cut are determined.
4) Experiments were conducted and machining was performed.
5) Chip formed after running each choice was then collected manually from inside the machine from the chip tray/belt.
6) The chip was then put on display with the empty chart.
7) Once the experiment was completed with all the choices and all chips were collected one by one on the display table, then a photograph was taken for further studies.

These choices of cutting parameter are step values and are determined by the design of Insert geometrical attribute and other engineering calculations. An example of various quality chips produced and arranged in the chip chart is shown in Figure. 9. After identifying proper cutting speed (Vc) based on metal cutting calculations or recommendations based on prior history, the chip was collected for various depths of cut at regular intervals and feed combinations. These combinations produce different kinds of chips, which were later classified as optimal, acceptable, and unaccepted.

Chips are visually inspected and compared against standard samples to determine their length and size, which helps in categorizing them into the optimal, acceptable, and unaccepted categories. Experienced machinists or engineers play a crucial role in the classification process, using their expertise to identify the optimal balance between chip size, surface finish, and material removal rates. Chips that are excessively long can lead to entanglement with machining parts and inserts, potentially damaging equipment. Such chips also pose handling issues for disposal after machining, even if high material removal rates or surface finish are achieved. Therefore, excessively long chips are classified as unaccepted. Chips that are too small may indicate low material removal rates. While they may result in good surface finish and are easy to dispose of later, they are considered suboptimal for machining processes due to their impact on material removal rates. Optimal chips are those that strike a balance between the surface finish, ease of disposal, and material removal rates. They are of the right size to ensure efficient machining while maintaining good surface quality and easy handling for disposal.

Our objective is to determine the 'optimal' chips. The definition of 'Optimal' Chip is the Chip which should not interfere while machining while high material removal rate or high surface finish. Following characteristic can be obtained



(a)

(b) (b)

FIGURE 7. (a)Chip breaker Design on Insert and combination of Depth of cut,Feed and Cutting Speed (b)Depth of Cut(DOC) (Angular view).

by studying chips. a) Chips that are very long: It may lead to entanglement of chips with machining parts and inserts and lead to damaging of equipment. This also leads to handling issues for disposal of chips later after machining, even if we achieve high material removal rates or high surface finish it is unacceptable. b) Chips that are too small: Low material rates, may be good surface finish easy to dispose off later. c) Optimal Chips: Right chip size where we have balance of optimal surface finish, easy to dispose off chips and good material rates. This classification of chip based on grading is detailed in Table. 3 and is shown in Figure. 10.

A. DATASET

The data collected is from a real industry environment, therefore, it poses some real challenges, and the most critical is the class imbalance problem. There are very few manual preparations needed for the practical application, for example, cleaning and maintenance of chip adhesion including manual inspection, however, it doesn't affect the outcome of the study. As there are only a few unaccepted samples available compared to the accepted samples, therefore it causes a huge imbalance in the classes.

From the several chip charts available, the individual chips are extracted to create the dataset. Some samples of extracted chips and their corresponding labels are shown in Figure. 11. In total, 9,795 individual chips are extracted, out of which 7,523 are 'Accepted', 850 are 'Optimal', and 1,422 are 'Unaccepted'. The data we got were clearly imbalanced. In addressing the challenge of imbalanced data, we explored two distinct strategies: upsampling and downsampling. Therefore, the two minority class samples are up-sampled through data augmentation to decrease the imbalance ratio between classes. Rather than using more sophisticated oversampling approaches like KNNOR [43] or SMOTE [44], the data augmentation is done using PyTorch's torchvision's 'transforms' module. This is done to avoid issues like intrinsic class separability, sensitivity towards noise, and computational complexity. To add more data diversity, the augmentation is done through random horizontal/vertical Flips, random rotation, random resized crop, color jitters, random Affine, and Gaussian blur. Examples as a result of augmentation on two minority class samples are provided in Figure. 12. Following the data balancing process, we achieved a more even distribution: 9,404 instances in the 'Accepted' class, 4,776 in the 'Optimal' class, and 8,882 in the 'Unaccepted' class. To enhance robustness, we upsampled the 'Accepted' class, which primarily contained regular images, by introducing flipped, rotated, and cropped variants as these modifications were also introduced in the other two classes. This strategy was employed to align with the modifications made in the other two classes.

TABLE 3. Classification of chip based on grading.

Sr. No	Classification	Grade
1	Optimal	6,7,8,9
2	Accepted	3,4,5
3	Unaccepted	1,2

For the downsampling approach, we randomly removed 50% of only the majority 'Accepted' class, with no modifications done on the other two ('unaccepted/optimal') classes. The summary of data samples is provided in Table 4.

In an industrial context where promptly identifying any 'Unaccepted' items is crucial, the study focused on decreasing the false positive rate for the 'Unaccepted' class. The best performing model was optimized for heightened sensitivity to this category. Additionally, binary classification was experimented with to simplify the task and focus on distinguishing between acceptable and unacceptable tool wear states. A binary version of the dataset was created with only two classes: 'Accepted' and 'Unaccepted'. The 'Optimal' class, representing the highest quality of chips, was merged with the 'Accepted' class, while the 'Unaccepted' class remained unchanged. This restructuring aimed to simplify the classification task. For the binary classification's upsampled dataset, the upsampled instances of the 'Accepted' and 'Optimal' classes were combined with the upsampled instances of the 'Unaccepted' class, ensuring a balanced representation for the binary classification model.

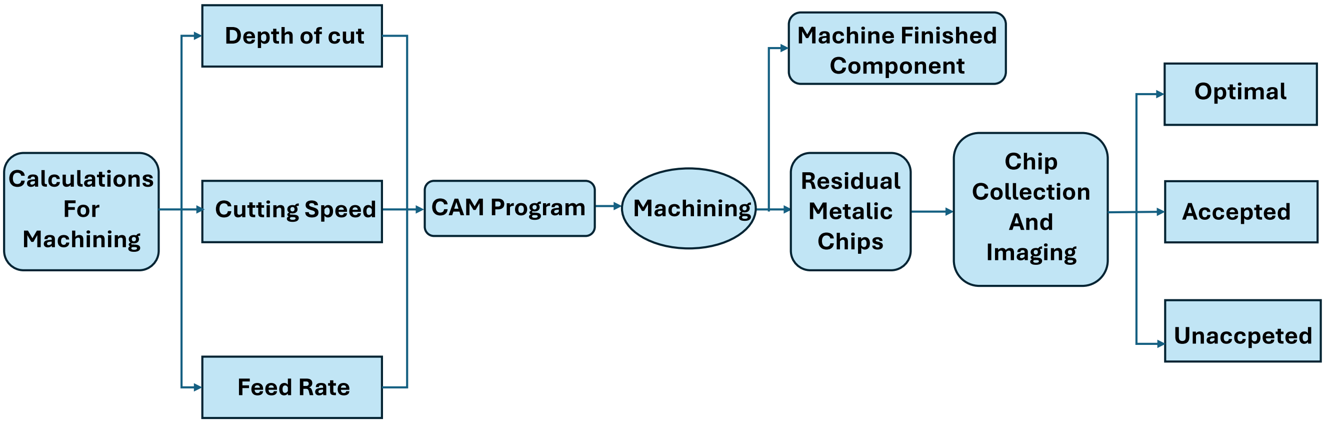


FIGURE 8. General Flow of Experimental Setup for chip collection.

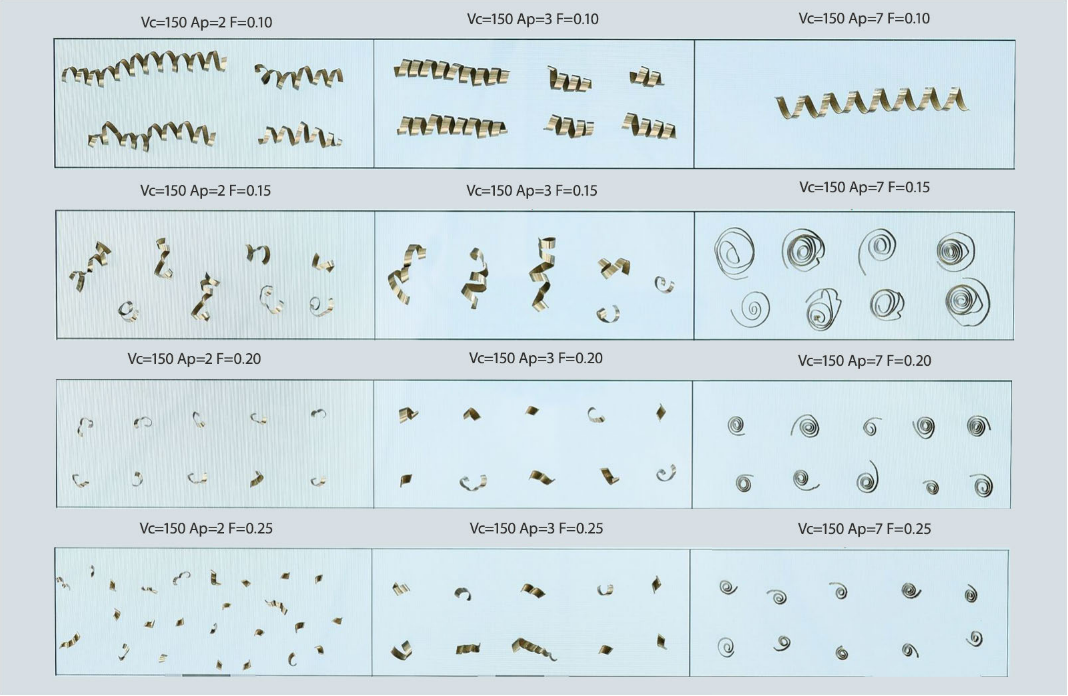


FIGURE 9. Classic way of various quality chips produced and arranged in a chip chart.

V. PERFORMANCE EVALUATION

All the models are evaluated with different experimental settings using their train and validation loss curves to ensure each model has learned properly and is reported using 5-fold cross-validation. The experimental settings that produced optimal results for different models are reported in Table 5.

A. EVALUATION METRICS

The models are evaluated based on precision, recall, F1-score, accuracy, macro average precision, and weighted average precision. Moreover, all the models are statistically tested and validated against their performance. For statistical testing, the Friedman test followed by Nemenyi Post Hoc test is performed to see the performance difference between different models. The next section details the results of the evaluation of different models.

B. MULTI-CLASS RESULTS

The evaluation results of the CNN model, as presented in Table 6, 7, 8, demonstrate its varying performance across different datasets. When tested on actual data, despite its simple architecture, the model exhibited a commendable accuracy of 86%. However, its performance was notably reduced on down-sampled data, where accuracy dropped to 75%. The down-sampled scenario presented challenges, particularly in achieving a balance between precision and recall for each class. Notably, the precision for the ‘Accepted’ class suffered a significant drop, indicating difficulties in

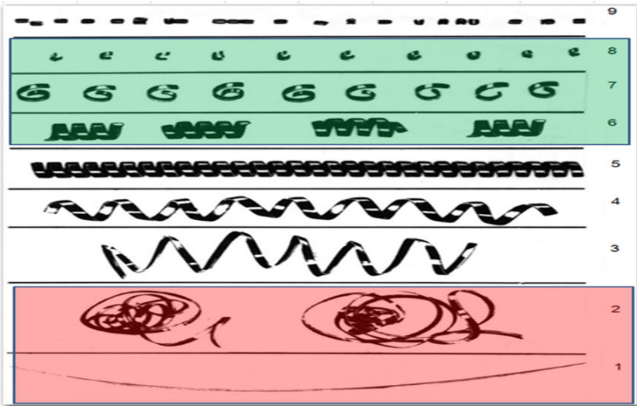


FIGURE 10. A visual example of chip classification based on grading displayed in Table. 3.

correctly identifying positive cases under down-sampled conditions. In contrast, the up-sampled data showcased the model's capability to achieve perfect precision, recall, and F1-score for all classes, emphasizing the importance of data balance in enhancing model performance.

In the case of CoAtNet-0, the evaluation results (Table 6, 7, 8) reveal a similar trend. The model achieved an accuracy of 75% on actual data, which dropped to 69% on down-sampled data. This decrease is consistent with the challenges posed by class imbalance, affecting both precision and recall. The 'Unaccepted' class, in particular, faced a decline in recall. However, on up-sampled data, CoAtNet-0 showcased a remarkable accuracy of 94%, underlining its adaptability to a more balanced dataset. Despite a lower macro average, the model's weighted average remains high, emphasizing its ability to handle imbalanced data effectively. The results for EfficientNetB0, AlexNet, ResNet18, and MobileNetV2 pretrained models on the up-sampled data, the down-sampled data, and the actual data are provided in Table 6, Table 7 and, Table 8, respectively. The EfficientNetB0 model demonstrates high precision, recall, and F1-score across all three classes on the upsampled data(Table 6), resulting in an impressive overall accuracy of 95%. This indicates the effectiveness of EfficientNetB0 in handling somewhat balanced datasets, making it a good candidate for tool wear classification. However, its effectiveness diminishes with down-sampled(Table 7) and actual(Table 8) datasets, particularly struggling with the 'Optimal' and 'Unaccepted' classes due to the inherent class imbalance in real-world data. This indicates that EfficientNetB0 excels when there's plenty of data, thus, it tends to favor the 'Accepted' class, which is most prevalent in the dataset. Increasing the training iterations would lead to overfitting this class without effectively improving the model's ability to recognize the less common classes.

TABLE 4. Summary of number of data samples in each class.

Data	Multiclass Samples		
	Accepted	Unaccepted	Optimal
Actual	7525	1422	850
Up-sampled	9404(1.25x)	8882(6.2x)	4776(5.62x)
Down-sampled	3763(0.5x)	1422(1x)	850(1x)
Data	Binary Samples		
	Accepted	Unaccepted	
Actual	8375	1422	
Up-sampled	14180(1.7x)	8882(6.2x)	
Down-sampled	5185(0.62x)	1422(1x)	

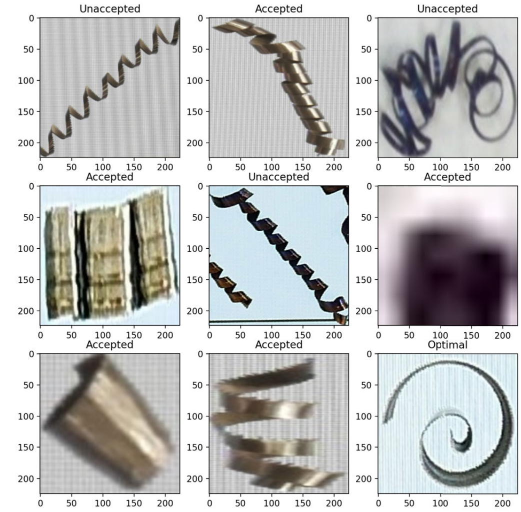
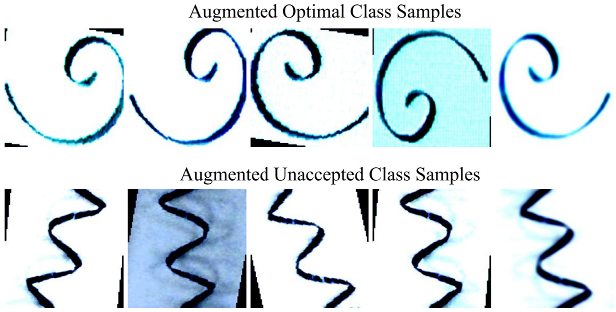


FIGURE 11. Example of extracted samples from the chip chart.


Augmented Optimal Class Samples

Augmented Unaccepted Class Samples

FIGURE 12. Examples from augmented data samples from two minority classes.

The evaluation results for AlexNet on up-sampled data (Table 6) also show robust performance with an accuracy of 96%. The model achieves perfect precision, recall, and F1-score for the 'Accepted' class. However, it shows a relatively lower F1-score for the 'Optimal' class, which may be attributed to challenges in correctly identifying instances of this class. It achieves an accuracy of 86% on down-sampled data(Table 7). It demonstrates balanced precision and recall for all classes, indicating its ability to handle class imbalances to some extent. Furthermore, it performs well on actual data with an accuracy of 90%. It exhibits high precision, recall, and F1-score for the 'Accepted' and 'Unaccepted' classes. However, it struggles with the 'Optimal' class, where precision is compromised. As its result was quite satisfactory on the multiclass data, we further explored its efficiency on Binary class data which is demonstrated in the later part.

ResNet18, as depicted in Table 6, exhibits consistent performance across all metrics, with an accuracy of 96%. The model achieves perfect precision, recall, and F1-score for the 'Accepted' class. It also maintains high scores for the 'Optimal' and 'Unaccepted' classes, showcasing its reliability in tool wear classification tasks. It performs consistently on down-sampled data(Table 7), achieving an accuracy of 85%. It maintains high precision, recall, and F1-score for the 'Accepted' and 'Unaccepted' classes, but faces challenges in correctly classifying the 'Optimal' class. Furthermore, it demonstrates consistent performance on actual data, achieving an accuracy of 86%. It maintains high precision, recall, and F1-score for the 'Accepted' and 'Unaccepted' classes. Challenges persist in correctly classifying the 'Optimal' class. ResNet18 was further assessed on a binary class dataset due to its strong performance on the multiclass dataset.

Table 6 also presents the evaluation results for MobileNetV2 on up-sampled data. The model achieves a commendable accuracy of 95%, with balanced precision, recall, and F1-score for all three classes. This highlights MobileNetV2 as a reliable choice for tool wear monitoring in machining processes if ample data is present. However, in the case of the imbalanced datasets, such as the downsampled (Table 7) and actual data (Table 8), ResNet18 did not perform as well compared to the previous models. This outcome highlights its limitations in adjusting to handle imbalanced data effectively.

The comparative analysis of models, as evaluated by the weighted average precision, is presented in Table 9. Complementing this tabular representation, a visual illustration of the comparative performance is depicted in Figure. 13. The results of this performance assessment reveal that AlexNet exhibits superior efficacy compared to other models under consideration. On the original dataset, without any data augmentation, AlexNet attains the highest performance, with a noteworthy second place achieved by a simplistic Convolutional Neural Network (CNN) architecture. Although subtle, the discernable differences in overall performance among the models become more pronounced when augmenting data samples from minority classes. Strikingly, augmenting the dataset enhances the performance of all models. In contrast, downsampling the majority class samples leads to a decline in performance across all models. The results of the statistical significance test done through Friedman Test followed by Nemenyi Post Hoc Test are depicted by the heat maps as shown in Figure. 14. From these results, it can be seen that the performance of AlexNet and ResNet is significantly better throughout the datasets, mainly compared to CNN, CoAtNet, and MobileNet.

C. BINARY CLASS RESULTS

We later evaluated the two best-performing models, AlexNet and ResNet18, on up-sampled data for Binary classes (Accepted and Unaccepted), and the result is summarized in Table 10, 11, 12. Both models demonstrate strong performance on the up-sampled data, with high precision, recall, and F1-scores for both classes. ResNet18, in particular, shows slightly better precision for the Accepted class, contributing to its higher overall performance in terms of macro and weighted averages. The accuracy for both models is consistent at 96%. Overall, these models exhibit robust performance in handling imbalanced data through upsampling. The results on actual data using the same two models are detailed in Table 11. Both models show solid performance on actual data, with high precision, recall, and F1-scores for the Accepted class. However, the Unaccepted class poses a challenge, especially for AlexNet, where the recall is relatively low. ResNet18, on the other hand, maintains a higher recall for the Unaccepted class, contributing to its better overall performance in terms of macro and weighted averages. The models achieve accuracy levels of 93% and 94%, respectively. These results highlight the models' capability to handle real-world data, with ResNet18 demonstrating a slight advantage in performance over AlexNet. Similarly, the performance for both models on down-sampled data for two classes is shown in Table 12. Both models exhibit robust performance on downsampled data. AlexNet maintains high precision and recall for the Accepted class, contributing to a strong F1-Score. However, it faces a challenge in correctly classifying Unaccepted samples, resulting in a lower recall for this class. ResNet18, while slightly lower in precision for both classes, compensates with better recall for the Unaccepted class, achieving comparable F1-Scores. The models achieve an accuracy of 91%, and their macro and weighted averages are consistent, indicating stable performance across both architectures on downsampled data.

D. DISCUSSION

The dataset was collected from real machining operations, reflecting the natural variability and diversity present in industrial settings. The dataset already contained classes for chips, as classified by experts or based on specific industrial standards. These classes were not controlled or standardized but were inherent to the dataset. There is currently no one rule to fit all machining strategies as some machines determine high material removal rate without worrying about surface finish, whereas some determine high surface finish with low material rates. Chip Size and quality are mostly determined by playing within the range of machining parameters for given machining strategies. Therefore, the parameters or conditions during chip collection have not been manipulated to control the variability or standardize the classification.

The design of image or feature enhancement methods specific to chips, considering their symmetry and geometric features was performed using traditional methods like random flipping, rotating, and scaling due to lack of specific domain knowledge that includes geometric calculation and geometrical attributes and other engineering calculations.

TABLE 5. Experimental Settings for different models.

Multiclass dataset						
Dataset	Model	Learning Rate	Optimizer	Loss Function	Epochs	Batch Size
Upsampled	CNN	0.001	Adam	Cross Entropy Loss	64	10
	CoatNet	0.001	Adam	Cross Entropy Loss	64	10
	EfficientNetB0	0.01	SGD	Cross Entropy Loss	30	128
	AlexNet	0.001	SGD	Cross Entropy Loss	100	128
	ResNet18	0.001	SGD	Cross Entropy Loss	20	64
	MobilenetV2	0.01	SGD	Cross Entropy Loss	17	64
Downsampled	CNN	0.001	Adam	Cross Entropy Loss	64	10
	CoatNet	0.001	Adam	Cross Entropy Loss	64	10
	EfficientNetB0	0.001	SGD	Cross Entropy Loss	64	30
	AlexNet	0.001	SGD	Cross Entropy Loss	64	30
	ResNet18	0.001	SGD	Cross Entropy Loss	64	30
	MobilenetV2	0.001	SGD	Cross Entropy Loss	32	30
Actual	CNN	0.001	Adam	Cross Entropy Loss	64	10
	CoatNet	0.001	Adam	Cross Entropy Loss	64	10
	EfficientNetB0	0.001	SGD	Cross Entropy Loss	64	50
	AlexNet	0.001	SGD	Cross Entropy Loss	64	50
	ResNet18	0.001	SGD	Cross Entropy Loss	20	16
	MobilenetV2	0.001	SGD	Cross Entropy Loss	20	16
Binary class dataset						
Dataset	Model	Learning Rate	Optimizer	Loss Function	Epochs	Batch Size
Upsampled	AlexNet	0.001	SGD	Cross Entropy Loss	30	64
	ResNet18	0.001	SGD	Cross Entropy Loss	15	64

TABLE 6. Evaluation matrix for different models on Upsampled data for Multiclass classification.

	EfficientNetB0			AlexNet			ResNet18			MobileNetV2			CoAtNet			CNN		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Accepted	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Optimal	0.87	0.89	0.88	0.90	0.93	0.91	0.93	0.90	0.91	0.90	0.86	0.88	0.91	0.78	0.84	0.89	0.82	0.84
Unaccepted	0.94	0.93	0.93	0.96	0.94	0.95	0.95	0.96	0.95	0.92	0.95	0.94	0.89	0.96	0.92	0.90	0.92	0.91
accuracy	0.95			0.96			0.96			0.95			0.94			0.93		
macro avg	0.94			0.95			0.96			0.94			0.93			0.92		
weighted avg	0.95			0.96			0.96			0.95			0.94			0.93		

TABLE 7. Evaluation Matrix for different models on Downsampled data for Multiclass classification.

	EfficientNetB0			AlexNet			ResNet18			MobileNetV2			CoAtNet			CNN		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Accepted	0.83	0.94	0.88	0.88	0.93	0.90	0.87	0.94	0.90	0.81	0.94	0.87	0.78	0.85	0.81	0.76	0.92	0.83
Optimal	0.97	0.04	0.08	0.75	0.77	0.76	0.79	0.65	0.71	1.00	0.03	0.06	0.49	0.57	0.53	0.67	0.54	0.60
Unaccepted	0.57	0.70	0.63	0.87	0.71	0.78	0.82	0.74	0.78	0.61	0.70	0.65	0.55	0.36	0.43	0.71	0.40	0.51
accuracy	0.75			0.86			0.85			0.76			0.69			0.75		
macro avg	0.79			0.83			0.83			0.81			0.60			0.72		
weighted avg	0.79			0.86			0.85			0.79			0.68			0.74		

TABLE 8. Evaluation Matrix for different models on Actual data for Multiclass classification.

	EfficientNetB0			AlexNet			ResNet18			MobileNetV2			CoAtNet			CNN		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Accepted	0.86	0.98	0.91	0.93	0.95	0.94	0.89	0.97	0.93	0.89	0.96	0.92	0.88	0.84	0.86	0.91	0.93	0.92
Optimal	1.00	0.02	0.03	0.7	0.73	0.72	0.82	0.28	0.32	0.80	0.17	0.28	0.38	0.45	0.41	0.65	0.61	0.63
Unaccepted	0.55	0.46	0.50	0.84	0.70	0.76	0.65	0.64	0.64	0.62	0.65	0.64	0.41	0.46	0.43	0.71	0.67	0.69
accuracy	0.82			0.90			0.86			0.85			0.75			0.86		
macro avg	0.80			0.83			0.79			0.61			0.56			0.76		
weighted avg	0.82			0.89			0.85			0.83			0.77			0.86		

Each chip of the dataset reflects the state of the cutting tool and as soon as some 'unaccepted' classified chips are collected, it indicated that the tool is needed to be replaced to ensure better quality chips. The dataset doesn't provide continuous data on tool wear progression and it wasn't specified if all those chips were cut using the same tool reflecting its gradual degradation on the chips with uses. Therefore, this study proposed for a classification model instead of a regression one.

While choosing models to experiment with, this study conducted research on different models from classical, lightweight, to deep models with pretrained weights. The reason behind this was the nature of data used in this study. The data was imbalanced, but our goal was to

TABLE 9. Performance comparison of models based on the weighted average precision using the up-sampled data.

Data	Weighted Average Precision					
	CNN	CoAtNet	EfficientNetB0	MobileNetV2	ResNet18	AlexNet
Actual	0.86	0.77	0.82	0.83	0.85	0.89
Down sampled	0.74	0.68	0.79	0.79	0.85	0.86
Up sampled	0.93	0.94	0.95	0.95	0.96	0.96

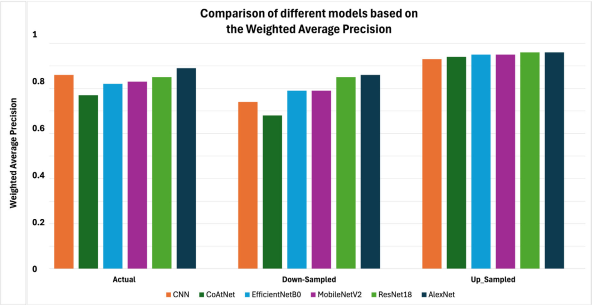


FIGURE 13. Comparison of different models based on the Weighted Average Precision. Three groups show results on three data cases.

make model that is sensitive to the 'Unaccepted' class which was more than 5 times less than the 'Accepted' class. To achieve this, we experimented with upsampling and downsampling techniques. We also assumed that the classical and lightweight models would do better when we experimented with the Actual and the down-sampled data. However, when we work with the upsampled data, the deep models would work better. The overall goal was to find a technique that provides good accuracy and precise response, especially with the unaccepted class.

Initially, this study used a simple 2-layers CNN structure and the bare architecture of CoAtNet0. For AlexNet, ResNet18, MobileNetV2 and EfficientNetB0, their pre-trained weights from ImageNet dataset were used. First, all the layers in the models were frozen to utilize their pretrained feature extractors. By doing this, it was ensured that only the newly added classifier layers are updated during training. This allows the models to maintain the robust features it learned from the ImageNet dataset, while the new classifier layers adjust to perform well on our specific task. Every model was fine-tuned by monitoring the loss curve, adjusting training parameters to minimize the loss function, ensuring optimal convergence, and preventing overfitting.

Both CNN and CoAtNet-0 models faced challenges in scenarios with imbalanced data, especially under down-sampled conditions. The results highlight the importance of considering data distribution during model training, with up-sampling proving effective in improving performance. Notably, CoAtNet-0 demonstrated consistent and competitive performance with CNN, showcasing its potential as an alternative model for tool wear classification tasks. The findings underscore the significance of selecting appropriate models and data preprocessing strategies to address the intricacies of real-world machining datasets.

EfficientNetB0, AlexNet, ResNet18, and MobileNetV2, alongside CNN and CoAtNet-0, demonstrate strong performance on up-sampled data. They showcase their ability to handle class imbalances and effectively classify tool wear states. The choice of the most suitable model may depend on specific requirements, computational resources, and the complexity of the machining dataset.

The models show varying performances on down-sampled and actual data. While some models maintain robustness in handling imbalanced datasets, challenges persist in correctly classifying less represented classes. The choice of a model should consider the specific characteristics of the machining dataset and the importance of accurate classification for each tool wear state. In summary, all six models exhibit promising results, emphasizing the feasibility of employing various deep learning architectures for real-world tool wear classification tasks in machining processes.

To ensure a fair comparison between the different models, this paper standardized the evaluation process across all models. This included training and testing the models on the same datasets, using consistent evaluation metrics (e.g., precision, recall, F1-score, accuracy, macro average precision, and weighted average precision), and applying uniform training procedures (e.g., number of epochs, learning rates, and optimization techniques) to all models. Additionally, all models were evaluated on the same hardware to ensure consistent computational comparisons.

1) OVERALL COMPARISON OF PERFORMANCE

Determining the best-performing model depends on several factors. The following are the observations based on different factors in this case:

1) **Overall Accuracy:**
   - **Up-sampled Data:** All models perform exceptionally well with high accuracy (above 94%). In this scenario, all models are competitive, with little variation in accuracy.
   - **Actual Data:** AlexNet and ResNet18 outperform other models with accuracies of 90% and 86%, respectively. These models demonstrate better generalization to real-world machining scenarios.
   - **Down-sampled Data:** ResNet18 and AlexNet show the highest accuracies (86% and 85%) on down-sampled data, indicating their ability to handle reduced data scenarios.
2) **Handling Class Imbalance:**
   - **Up-sampled Data:** All models show excellent performance on the imbalanced up-sampled data,

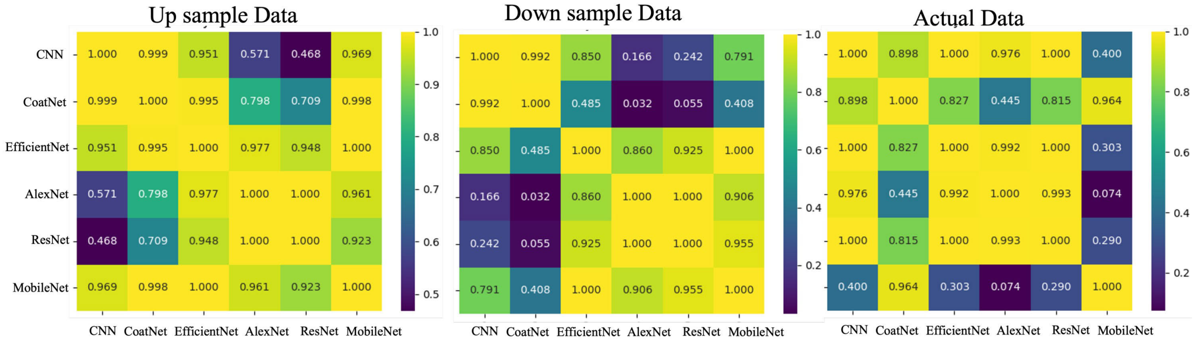


FIGURE 14. Heat map of the statistical performance of different Models on three different data sets.

TABLE 10. Evaluation matrix for different models on Upsampled data for binary class classification.

	AlexNet			Resnet18		
	Precision	Recall	f1-score	Precision	Recall	f1-score
Accepted	0.96	0.97	0.96	0.98	0.97	0.97
Unaccepted	0.95	0.94	0.94	0.95	0.96	0.95
accuracy	0.96			0.96		
macro avg	0.95			0.96		
weighted avg	0.96			0.96		

TABLE 11. Evaluation matrix for different models on actual data for binary class classification.

	AlexNet			Resnet18		
	Precision	Recall	f1-score	Precision	Recall	f1-score
Accepted	0.94	0.98	0.96	0.95	0.98	0.96
Unaccepted	0.85	0.64	0.73	0.85	0.70	0.77
accuracy	0.93			0.94		
macro avg	0.89			0.90		
weighted avg	0.93			0.94		

TABLE 12. Evaluation matrix for different models on downsampled data for binary class classification.

	AlexNet			Resnet18		
	Precision	Recall	f1-score	Precision	Recall	f1-score
Accepted	0.92	0.97	0.94	0.93	0.96	0.94
Unaccepted	0.87	0.72	0.79	0.85	0.75	0.80
accuracy	0.91			0.91		
macro avg	0.89			0.89		
weighted avg	0.91			0.91		

with high precision, recall, and F1-scores across all classes.

- **Actual Data:** AlexNet and ResNet18 exhibit balanced performance on the ‘Accepted’ and ‘Unaccepted’ classes, indicating their ability to handle class imbalances in real-world machining data.
- **Down-sampled Data:** AlexNet and ResNet18 also demonstrate somewhat balanced precision and recall on down-sampled data, showcasing their robustness in scenarios with limited samples.

3) **Robustness Across Scenarios:**

- **AlexNet:** Performs consistently well across different datasets and scenarios, making it a strong candidate for tool wear classification.
- **AlexNet and ResNet18:** Demonstrate versatility, showing competitive performance across up-sampled, down-sampled, and actual datasets.

Though upsampling is considered superior to downsampling, there could be scenarios where there is abundance of a particular class of data, but the model needs to perform well in the minor class. This can be attributed to the nature of data in real machining industry settings, where there are typically ample majority-class samples (representing good tool conditions) and few minority-class samples (representing various stages of tool wear). Downsampling the majority class preserves much of the information as these samples are often similar and redundant, whereas upampling the minority class can introduce noise and synthetic samples, potentially leading to overfitting. In this kind of situation, downsampling can reduce the redundancy of the majority class while also mitigating the overfitting issues. However, in this study upsampling performed comparably better to downsampling in tool wear monitoring despite having chances to get contradictory results. The findings of the study suggest that downsampling can be a viable option, especially when computational efficiency and simplicity are crucial. However, before applying the random downsampling technique, it should be ensured that the downsampled instances represent the actual data without losing important features which was limited in this study. This underscores the importance of considering the dataset's characteristics and practical constraints when choosing between upsampling and downsampling techniques in tool wear monitoring. The broader implication of our findings is that the choice between upsampling and downsampling should be guided by the specific characteristics of the dataset and goal, and the practical constraints of the application. For tool wear monitoring, where real-time processing and model simplicity can be critical, downsampling presents a compelling alternative. We plan to reapply the downsampling approach in future, this time ensuring that the instances remain representative of the actual data introducing an evaluation method.

It is true that the modern architecture like CoAtNet-0 was far behind than classic AlexNet on our dataset, the reason behind that is: while using upsampled data, the accuracy of AlexNet and CoatNet-0 both were noticeably well. The average accuracy of AlexNet was 96% and the average accuracy of CoatNet-0 was 94%. However, when we used the actual data and downsampled data, though AlexNet performed satisfactorily, CoAtNet-0 performed poorly. The drastic difference between these results could be described by their training system. the study have used pretrained AlexNet while the CoatNet-0 had only the bare architecture. So, as the training instances became lower, the complicated architecture without previous training failed to capture enough details from the data. CoAtNet-0 represents a more advanced architecture, its performance is highly dependent on the availability of sufficient data and appropriate training. However, as AlexNet had previously trained weights from ImageNet, it could still perform better than more modern CoAtNet-0. Thus, AlexNet can leverage prior knowledge to perform better on smaller or less diverse datasets, making them valuable even in the presence of more modern architectures

In conclusion, the choice of the best-performing model depends on the specific characteristics and requirements of the machining dataset. ResNet18 and AlexNet stand out as strong contenders, with AlexNet being particularly notable for its consistent performance across scenarios. Furthermore, it's recommended to consider factors such as computational efficiency, and interpretability.

E. FUTURE WORK

In order to consider the symmetrical and the geometric feature, future research directions could include the development of custom augmentation algorithms that leverage the specific geometric and symmetrical properties of chips. Such algorithms could include symmetry-based augmentation techniques, such as mirroring along specific axes, and geometric feature augmentation methods that preserve or emphasize chip-specific shapes and patterns. Additionally, we plan to explore feature extraction and enhancement techniques tailored to chips, such as edge detection and pattern recognition, to further improve the robustness and accuracy of our models. These enhancements aim to not only improve the diversity of our training dataset but also enhance the generalization of our approach to new and unseen chip images.

Again, determining the tool wear condition through chip analysis future study is needed. Based on the data considered in the study, there is no step-by-step link indicating tool wear conditions. This determination requires high-frequency data collection, detailed physical examinations, and laboratory analyses to provide immediate feedback on tool wear. Consequently, assessing any potential lag in determining tool wear conditions is beyond the scope of our research.

VI. CONCLUSION

In this study, we delved into the critical realm of tool wear monitoring in machining processes, recognizing its profound impact on final output quality and production efficiency. Our exploration of direct and indirect tool monitoring methods revealed the importance of a dependable system for early tool wear detection. Leveraging recent advancements in artificial intelligence (AI) and aligning with the principles of Industry 4.0, we focused on employing AI models for monitoring and classifying tool wear using authentic image data from machining processes.

The evaluation of multiple convolutional neural network (CNN) architectures, including AlexNet, MobileNetV2, ResNet18, and the recently introduced CoAtNet-0, across various datasets provided nuanced insights. While each model exhibited strengths under different scenarios, the overall robustness of AlexNet stood out, demonstrating consistent performance across up-sampled, down-sampled, and actual datasets. ResNet18 showcased adaptability to class imbalances, generalization to real-world machining scenarios, and competitive accuracy.

Our findings underscore the significance of considering the specific characteristics of machining datasets when selecting an appropriate tool wear monitoring model. ResNet18 and AlexNet emerge as strong candidates, with AlexNet being particularly noteworthy for its versatility and resilience across different scenarios. As we move towards the era of highly automated manufacturing, the integration of AI models in tool wear monitoring presents itself as a promising avenue for achieving adaptable and learning-enabled machining processes.

In conclusion, this study contributes valuable insights to the machining industry, addressing challenges related to tool wear monitoring and emphasizing the role of AI in enhancing the efficiency and reliability of manufacturing processes. Future research could explore additional AI architectures, refine model interpretability, and investigate real-time implementation aspects to further advance the field of intelligent tool wear monitoring.

REFERENCES

[1] M. Günay, M. E. Korkmaz, and N. Yaşar, ''Performance analysis of coated carbide tool in turning of nimonic 80A superalloy under different cutting environments,'' *J. Manuf. Processes*, vol. 56, pp. 678–687, Aug. 2020.

[2] R. W. Maruda, G. M. Krolczyk, S. Wojciechowski, B. Powalka, S. Klos, N. Szczotkarz, M. Matuszak, and N. Khanna, ''Evaluation of turning with different cooling-lubricating techniques in terms of surface integrity and tribologic properties,'' *Tribol. Int.*, vol. 148, Aug. 2020, Art. no. 106334.

[3] M. Cheng, L. Jiao, P. Yan, H. Jiang, R. Wang, T. Qiu, and X. Wang, ''Intelligent tool wear monitoring and multi-step prediction based on deep learning model,'' *J. Manuf. Syst.*, vol. 62, pp. 286–300, Jan. 2022.

[4] M. A. Erden, N. Yaşar, M. E. Korkmaz, B. Ayvacı, K. N. S. Ross, and M. Mia, ''Investigation of microstructure, mechanical and machinability properties of Mo-added steel produced by powder metallurgy method,'' *Int. J. Adv. Manuf. Technol.*, vol. 114, nos. 9–10, pp. 2811–2827, Jun. 2021.

[5] M. E. Korkmaz and M. Günay, ''Finite element modelling of cutting forces and power consumption in turning of AISI 420 martensitic stainless steel,'' *Arabian J. Sci. Eng.*, vol. 43, no. 9, pp. 4863–4870, Sep. 2018.

[6] S. Wojciechowski, R. W. Maruda, P. Nieslony, and G. M. Krolczyk, ''Investigation on the edge forces in ball end milling of inclined surfaces,'' *Int. J. Mech. Sci.*, vol. 119, pp. 360–369, Dec. 2016.

[7] N. Szczotkarz, R. Mrugalski, R. W. Maruda, G. M. Królczyk, S. Legutko, K. Leksycki, D. Dębowski, and C. I. Pruncu, ''Cutting tool wear in turning 316L stainless steel in the conditions of minimized lubrication,'' *Tribol. Int.*, vol. 156, Apr. 2021, Art. no. 106813.

[8] R. Teti, K. Jemielniak, G. O'Donnell, and D. Dornfeld, ''Advanced monitoring of machining operations,'' *CIRP Ann.*, vol. 59, no. 2, pp. 717–739, 2010.

[9] R. W. Maruda, G. M. Krolczyk, P. Nieslony, S. Wojciechowski, M. Michalski, and S. Legutko, ''The influence of the cooling conditions on the cutting tool wear and the chip formation mechanism,'' *J. Manuf. Processes*, vol. 24, pp. 107–115, Oct. 2016.

[10] T. Mohanraj, S. Shankar, R. Rajasekar, N. Sakthivel, and A. Pramanik, ''Tool condition monitoring techniques in milling process—A review,'' *J. Mater. Res. Technol.*, vol. 9, no. 1, pp. 1032–1042, 2020.

[11] N. Ambhore, D. Kamble, S. Chinchanikar, and V. Wayal, ''Tool condition monitoring system: A review,'' *Mater. Today, Proc.*, vol. 2, nos. 4–5, pp. 3419–3428, 2015.

[12] F. Akkoyun, A. Ercetin, K. Aslantas, D. Y. Pimenov, K. Giasin, A. Lakshmikanthan, and M. Aamir, ''Measurement of micro burr and slot widths through image processing: Comparison of manual and automated measurements in micro-milling,'' *Sensors*, vol. 21, no. 13, p. 4432, Jun. 2021.

[13] M. E. Korkmaz, N. Yaşar, and M. Günay, ''Numerical and experimental investigation of cutting forces in turning of nimonic 80A superalloy,'' *Eng. Sci. Technol., Int. J.*, vol. 23, no. 3, pp. 664–673, Jun. 2020.

[14] H. Yurtkuran, M. E. Korkmaz, and M. Günay, ''Modelling and optimization of the surface roughness in high speed hard turning with coated and uncoated CBN insert,'' *Gazi Univ. J. Sci.*, vol. 29, no. 4, pp. 987–995, 2016.

[15] A. U. Rehman, M. U. Ahmed, and S. Begum, ''Cognitive digital twin in manufacturing: A heuristic optimization approach,'' in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov.* Cham, Switzerland: Springer, 2023, pp. 441–453.

[16] D.-H. Kim, T. J. Y. Kim, X. Wang, M. Kim, Y.-J. Quan, J. W. Oh, S.-H. Min, H. Kim, B. Bhandari, I. Yang, and S.-H. Ahn, ''Smart machining process using machine learning: A review and perspective on machining industry,'' *Int. J. Precis. Eng. Manuf.-Green Technol.*, vol. 5, no. 4, pp. 555–568, Aug. 2018.

[17] K.-D. Thoben, S. Wiesner, and T. Wuest, '''Industrie 4.0' and smart manufacturing—A review of research issues and application examples,'' *Int. J. Autom. Technol.*, vol. 11, no. 1, pp. 4–16, 2017.

[18] V. I. Guzeev and D. Y. Pimenov, ''Cutting force in face milling with tool wear,'' *Russian Eng. Res.*, vol. 31, no. 10, pp. 989–993, Oct. 2011.

[19] A. U. Rehman, M. A. Kabir, M. Ijaz, H. M. Al-Mohsin, and A. Bermak, ''Salp swarm algorithm for drift compensation in E-nose,'' in *Proc. 15th Int. Conf. Adv. Comput. Intell. (ICACI)*, May 2023, pp. 1–6.

[20] Z. Dai, H. Liu, Q. V. Le, and M. Tan, ''CoAtNet: Marrying convolution and attention for all data sizes,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 3965–3977.

[21] G. Wang and Y. Cui, ''On line tool wear monitoring based on auto associative neural network,'' *J. Intell. Manuf.*, vol. 24, no. 6, pp. 1085–1094, Dec. 2013.

[22] B. H. Freyer, P. S. Heyns, and N. J. Theron, ''Comparing orthogonal force and unidirectional strain component processing for tool condition monitoring,'' *J. Intell. Manuf.*, vol. 25, no. 3, pp. 473–487, Jun. 2014.

[23] E. Kuram and B. Ozcelik, ''Micro-milling performance of AISI 304 stainless steel using Taguchi method and fuzzy logic modelling,'' *J. Intell. Manuf.*, vol. 27, no. 4, pp. 817–830, Aug. 2016.

[24] G. Wang, Y. Yang, and Z. Li, ''Force sensor based tool condition monitoring using a heterogeneous ensemble learning model,'' *Sensors*, vol. 14, no. 11, pp. 21588–21602, Nov. 2014.

[25] L. Xu, C. Huang, C. Li, J. Wang, H. Liu, and X. Wang, ''Estimation of tool wear and optimization of cutting parameters based on novel ANFIS-PSO method toward intelligent machining,'' *J. Intell. Manuf.*, vol. 32, no. 1, pp. 77–90, Jan. 2021.

[26] D. McParland, S. Baron, S. O'Rourke, D. Dowling, E. Ahearne, and A. Parnell, ''Prediction of tool-wear in turning of medical grade cobalt chromium molybdenum alloy (ASTM F75) using non-parametric Bayesian models,'' *J. Intell. Manuf.*, vol. 30, no. 3, pp. 1259–1270, Mar. 2019.

[27] A. U. Rehman, S. B. Belhaouari, M. A. Kabir, and A. Khan, ''On the use of deep learning for video classification,'' *Appl. Sci.*, vol. 13, no. 3, p. 2007, Feb. 2023.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, ''An image is worth 16 × 16 words: Transformers for image recognition at scale,'' 2020, *arXiv:2010.11929*.

[29] M. Tan and Q. Le, ''EfficientNet: Rethinking model scaling for convolutional neural networks,'' in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[30] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, ''Scaling local self-attention for parameter efficient visual backbones,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 12894–12904.

[31] Y. Fang, X. Wang, R. Wu, and W. Liu, ''What makes for hierarchical vision transformer?'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12714–12720, Oct. 2023.

[32] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, ''CvT: Introducing convolutions to vision transformers,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.

[33] M. Shafiq and Z. Gu, ''Deep residual learning for image recognition: A survey,'' *Appl. Sci.*, vol. 12, no. 18, p. 8972, Sep. 2022.

[34] R. Gupta, V. Anand, S. Gupta, and D. Koundal, ''Deep learning model for defect analysis in industry using casting images,'' *Expert Syst. Appl.*, vol. 232, Dec. 2023, Art. no. 120758.

[35] A. Seker, ''Evaluation of fabric defect detection based on transfer learning with pre-trained AlexNet,'' in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 1–4.

[36] S. Thalagala and C. Walgampaya, ''Application of AlexNet convolutional neural network architecture-based transfer learning for automated recognition of casting surface defects,'' in *Proc. Int. Res. Conf. Smart Comput. Syst. Eng. (SCSE)*, vol. 4, Sep. 2021, pp. 129–136.

[37] A. Basit, M. A. Siddique, M. K. Bhatti, and M. S. Sarfraz, ''Comparison of CNNs and vision transformers-based hybrid models using gradient profile loss for classification of oil spills in SAR images,'' *Remote Sens.*, vol. 14, no. 9, p. 2085, Apr. 2022.

[38] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[39] A. Krizhevsky, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. 25th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 1, 2012, p. 1097.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ''ImageNet: A large-scale hierarchical image database,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[42] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[43] A. Islam, S. B. Belhaouari, A. U. Rehman, and H. Bensmail, "KNNOR: An oversampling technique for imbalanced datasets," *Appl. Soft Comput.*, vol. 115, Jan. 2022, Art. no. 108288.

[44] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

ATIQ UR REHMAN received the bachelor's degree (Hons.) in computer engineering from COMSATS University Islamabad, Pakistan, in 2010, the master's degree in computer engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2013, and the Ph.D. degree in computer science and engineering from Hamad Bin Khalifa University, Doha, Qatar, in 2019. Following his academic achievements, he embarked on a journey as a Postdoctoral Researcher with the College of Science and Engineering, Hamad Bin Khalifa University, making significant research contributions, from 2019 to 2022. Subsequently, he assumed the role of an Assistant Professor with the Department of Electrical and Computer Engineering, Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur, Pakistan. Currently, he is a Postdoctoral Researcher with the Artificial Intelligence and Intelligent Systems Research Group, School of Innovation, Design, and Engineering, Mälardalen University, Västerås, Sweden. His extensive body of work, comprising nearly 50 published research articles, revolves around the development of evolutionary computation, pattern recognition, and machine learning algorithms. He continues to make valuable contributions to the ever-evolving landscape of research in these domains.

TAHIRA SALWA RABBI NISHAT is currently pursuing the Ph.D. degree with Mälardalen University. Her research interests include lifelong machine learning was implemented with the random forest algorithm and propagated the current acquired knowledge to the next learning phase using the genetic algorithm. Her greatest interests include Python and MySql, also computer vision, e.g., OpenCV toolbox in MATLAB. Machine learning algorithms (e.g., linear regression, logistic regression, SVM, kNN, K-Means, Naive, Bayes, PCA, and LDA) and some Python libraries like Numpy, Pandas, and Matplotlib. Soft computing, hands-on experience on several neural networks and deep learning algorithms (e.g., LSTM, RNN, CNN, NLP, shallow, and deep neural networks) used different machine learning libraries like PyTorch, Keras, and Scikit-learn.

MOBYEN UDDIN AHMED received the M.Sc. degree in computer engineering (specialization in intelligent systems) from Dalarna University, Sweden, and the Ph.D. degree in artificial intelligence/computer science from Mälardalen University, in 2011. From 2012 to 2014, he completed his postdoctoral study in computer science and engineering with the Center for Applied Autonomous Sensor Systems, School of Science and Technology, Orebro University, Sweden. He is currently a Professor in artificial intelligence/computer science with the Artificial Intelligence and Intelligent Systems Group, School of Innovation, Design and Engineering, Mälardalen University, and a member of the ESS-H Embedded Sensor Systems for Health Research Profile. He has more than 150 scientific publications and more than 3529 citations. He research work have been selected twice in IVA-100 list, e.g., 2023: Trustworthy AI (https://www.iva.se/det-iva-gor/utmarkelser/ivas-100-lista/trustworthy-ai/). He is involved in teaching and is responsible for 19 online and campus-based courses in artificial intelligence.

SHAHINA BEGUM received the Ph.D. degree in artificial intelligence from Mälardalen University, in 2011.

She is currently a Professor and the Deputy Leader of the Artificial Intelligence and Intelligent Systems Group, MDU. Her research interests include developing intelligent systems in medical and industrial applications, artificial intelligence, multimodal machine learning and reasoning, explainable AI (XAI), data analytics, decision support systems, knowledge-based systems, and intelligent monitoring and prediction systems. She has been the Principal Applicant and the Project Manager for a number of research projects with MDU. She received a Swedish Knowledge Foundation's Prospect individual grant for prominent young researchers, in 2011, and is today leading several research projects in the area of intelligent-monitoring and prediction systems in collaboration with industrial partners. She has been listed amongst the 100 most relevant researchers in sustainable AI algorithm development by the Royal Swedish Academy of Engineering Sciences, in 2020.

ABHISHEK RANJAN received the bachelor's degree (Hons.) in mechanical engineering from Vellore Institute of Technology, India, and the master's degree (Hons.) in artificial intelligence and machine learning from Liverpool John Moores University, U.K. His research on electron microscope image analysis for tungsten carbide particles. He is currently an accomplished Data Scientist based in Fagersta, Sweden, boasting a remarkable 11 year career in manufacturing and five years dedicated to data science. In his current role as the Data Scientist with SECO Tools AB, Sweden, he leads the charge in implementing cutting edge AI and ML solutions, particularly in digital twins and cognitive manufacturing methodologies. His contributions include establishing MLOps practices, creating frameworks for AutoML on time-series data pipelines using Azure, and collaborating with industry leaders and universities for cutting-edge research. His professional journey encompasses impactful roles, such as implementing real-time cutting tool wear analytics and managing design automation teams. His skills span system and solution design, machine learning methodologies, computer vision, and expertise in tools like Python, Azure ML Studio, and TensorFlow. He stands as a distinguished professional with a proven track record of academic excellence and a significant impact on the forefront of data science and manufacturing innovation.