Improved Machine Learning Based Flank Wear Identification by Incorporating 3D-Height Maps

Sebastian Schibsdat(✉) and Jan H. Dege

Institute of Production Management and Technology, Hamburg University of Technology, Hamburg, Germany
sebastian.schibsdat@tuhh.de

Abstract. Coated cutting tools are very important in the field of machining, but predicting the temporal evolution of tool wear is still a major challenge due to the high complexity of the wear process. Although machine learning (ML) shows promise in this respect, traditional supervised learning models rely on labour-intensive and error-prone manual measurement of tool wear to generate training data, which is a process that is time-consuming and inefficient. This paper uses a machine vision approach that automates the identification of flank wear on coated indexable inserts for turning applications and extends these techniques by integrating 3D tool surface data via height maps into the input. For this purpose, images of indexable inserts are taken during external longitudinal turning of C45 + N and X5CrNi18–10 using a focus variation microscope. This enables a height map of the wear area to be generated. Image annotation is performed on the 2D images to maintain comparability with traditional methods. Furthermore, the images and height maps are automatically cropped to focus on the flank wear region. This reduces the class imbalance and allows the model to maintain higher image resolution, despite the limited input. Three different U-Net models are trained on this dataset: two with height maps and one without. A comparison of the three models focusing on prediction accuracy and learning efficiency shows that including height maps improves prediction performance and reduces the required dataset size for comparable accuracy.

Keywords: Machine vision · Convolutional neural networks · Tool Wear detection · Turning

1 Introduction

Machining is a key manufacturing process. It enables the production of complex parts with great precision. Despite major advances in technology, tool wear continues to be a major problem. Predicting the wear of coated tools is challenging due to the multi-layered, process-dependent effects during cutting. These are difficult to model analytically. A new approach to these problems is the use of machine learning (ML) techniques, which shows promising results [1]. Many approaches used in the literature are based on supervised ML models that use regression or classification, for which labelled data is required [1]. Wear prediction is based on the tool's measured wear, which is usually

L. Overmeyer and B.-A. Behrens (eds.), *Production at the Leading Edge of Technology*,
Lecture Notes in Production Engineering, https://doi.org/10.1007/978-3-032-19524-1_58

measured manually by experts and can be influenced by human error. To reduce the human factor in this laborious process, machine vision models have been suggested. Miao et al. and Holst et al. use a U-Net to create a segmentation of flank wear on a turning tool and a ball end mill [2, 3]. Bergs et al. expands this by incorporating tool type identification and comparing general trained U-Net segmentations with specialised versions [4]. Yoo et al. use a similar approach, but with different CNN models [5]. Chen et al. address the problems of imbalanced datasets and blurry boundaries for wear areas with an augmentation process involving diffusion and a two-step segmentation process using a fuzzy clustering technique [6]. In Zhu et al. the U-Net tool segmentation is used to predict surface roughness [7]. This study also compares the performance of different U-Net types. All of the aforementioned approaches use a 2D image for prediction and modelling. Adhesions of the workpiece material on the tool surfaces can make identification with just colour information difficult. Integrating additional information, such as surface geometry, could improve the model's accuracy. Additionally, increasing the input dimension by one more channel representing the topology increases the available training data per instance and could reduce the amount of training data needed for the same level of prediction accuracy.

This paper presents a machine vision model that has been trained using images of coated indexable inserts for turning applications. The model takes height maps of the tools as an additional input alongside the image. A separate model that uses only the image as input is compared against it. In order to ascertain the impact of incorporating height maps on the size of the training data, new models will be trained and evaluated using subsets of the original data. To focus on improvements through height maps, the classical U-Net [8] was selected as an ML model, since it is a proven CNN network used in various segmentation tasks, characterised by efficient and stable learning. This is a similar approach to that taken by Miao et al. and Holst et al., as their research has already shown that the base U-Net is capable of achieving an acceptable level of prediction accuracy [2, 3].

2 Materials and Methods

2.1 Experimental Setup

Images of the various stages of tool wear were gathered during external longitudinal turning experiments on a CNC lathe Max Mueller Gildemeister MD5S using custom-coated, indexable inserts. The setup had an orthogonal clearance of $\alpha_0 = 16°$, a tool orthogonal rake angle of $\gamma_0 = 6°$, and a cutting edge inclination of $\lambda_s = -6°$. Indexable inserts of the CNMG160608-MM5 type, manufactured by Walter AG, were used. They were made of fine-grained tungsten carbide (WC grain sizes $\leq$1 μm) and contained a binder of 10 wt% cobalt. A custom TiAlN coating was applied. This coating had a thickness of $t_{Coating} = 3.5 \pm 0.3$ μm. Some tools also had an additional decor layer containing TiN with a thickness of around $t_{Decor} = 50$ nm. The workpiece materials used were C45 + N (1.0503) and X5CrNi18–10 (1.4301). Each workpiece was cylindrical, with a usable length of $l_W = 500$ mm and a diameter ranging from $d_{Wmax} = 150$ mm to $d_{Wmin} = 100$ mm. Additionally, one side was flattened to enable temperature measurements of the rake face for use in other research. The process parameters used are listed in

Table 1. These process parameters are taken from a larger, fractional factorial design of experiments.

Table 1 Cutting process parameter used

Workpiece material	Cutting depth a_p in mm	Cutting speed v_c in m/min	Cutting feed f in mm	Cutting edges used	Wear measurements
X5CrNi18–10	1.1–1.6	170–290	0.15–0.35	11	52
C45 + N	1.2; 1.6	130; 180	0.15; 0.25	7	134

The cutting process was interrupted at intervals of $t_c = 30$ s in order to take a microscope image of the cutting tool. To facilitate this, the microscope was mounted on a gantry above the lathe, allowing the camera to be lowered into the machine chamber. With this setup, the insert could be measured in situ. Furthermore, the images obtained were highly homogeneous due to the precisely reproducible position of the camera, which simplified the preparation of the dataset. Each cutting edge was used until it reached a maximum flank wear width of $VB_{max} = 300$ µm, tool breakage, or a significant change in chip formation. For the dataset used in this paper, 18 cutting edges were used. This resulted in n = 186 (C45: 134, X5CrNi18–10: 52) measurements of tool wear.

Images and height maps were captured using a microscope VHX-7020, Keyence Corporation, including an adjustable magnification lens. Used in conjunction with the microscope controller VHX-X1F, Keyence Corporation, this allows the usage of focus variation for greater depth of field and 3D surface mapping. For these images, the microscope was oriented orthogonally to the flank face at an angle of $\psi = 35°$ to the primary cutting edge, Fig. 1. This angle was selected to also capture the secondary cutting edge. Magnification of 150x was selected, which leads to a field of view of approximately 2.34 × 1.75 mm. If the tool wear was too large to be captured completely at this magnification, an additional image was taken at 100x magnification (3.54 × 2.65 mm).

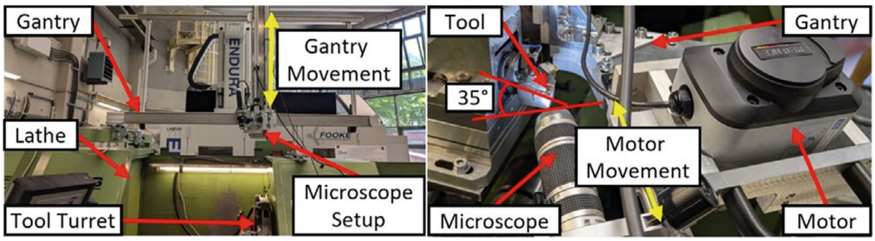

Fig. 1 Microscope setup

2.2 Hardware and Software

Image annotation was carried out manually using the MATLAB R2024b, The MathWorks Inc., 'Image Labeller' tool. Dataset preparation, hyperparameter study and final model training were conducted in Python. The main libraries used are listed in Table 2. Dataset creation, augmentation and model training were carried out on a computer with two Nvidia RTX 3090 GPUs, 128 GB of 2133 MHz RAM, an AMD Ryzen Threadripper 3970X processor and an SSD with 2 TB of storage.

Table 2 Main python libraries used for the study

Basic	Augmentation	ML training and model
Python 3.12.10 Pandas 2.2.3 Pillow 11.2.1 Numpy 2.2.5	Albumentations 2.0.7 Opencv-python 4.11.0.86	Torch 2.7.0 + cu128 Torchvision 0.22.0 + cu128 Scikit-learn 1.6.1 Optuna 4.3.0

3 ML Methodology

3.1 Database

As a result of the experiments, 404 images (.jpg format) with a resolution of 2048x1536 pixels were obtained. These included various stages of tool wear, ranging from new state to tool breakage, Fig. 2a, b. Accompanying height maps were extracted and saved as a .csv file for all images. The images were sorted into two categories according to the material used. To make creating a consistent mask easier, the tool images were first sorted by time used, and then annotated. First, the entire visible tool was segmented into the 'Tool' category, including the worn areas. Next, all types of flank wear and adhesion on the cutting edge were segmented as a single 'Wear' class, which overrode the previous 'Tool' declaration for these pixels. The mask was saved as a .png file.

Fig. 2 Example of flank wear and image pre-processing

The extracted height maps revealed an artificial surface in areas where parts of the tool were not visible. This appears to be an artefact resulting from processing through the microscope controller. To correct this, the background area of the image was identified and the corresponding areas in the height maps were set to zero.

To increase the resolution of the restricted input in the ML model, the images should be cropped to the region of interest. The first step is to rotate the images, accompanying height maps and masks so that the cutting edge aligns with the horizontal edge of the image. Additionally, the areas above and below the cutting edge are cropped. This reduces the image size to 1898 × 936 pixels, a total reduction of 43.5%, without any loss of information about wear. Figure 2c, d shows this process applied to one example, with the same changes applied to the accompanying mask created through the annotation process.

To allow for an accurate comparison of the different models, data augmentation was performed offline. First, the dataset was split into training and test data in a 70/30 ratio. The test data was saved directly without any additional modification and includes 121 images. Unless stated otherwise, all performance evaluations in this paper are based on these 121 images. To increase the available training data, multiple augmented versions of a single training image were created using the Albumentations Python library. The following augmentations were applied: Rotate, Horizontal Flip, Vertical Flip, Grid Distortion, Elastic Transform and Colour Jitter (only on image). Geometric and pixel-level augmentation were performed identically on the accompanying height map and mask. Colour space augmentation, however, could only be performed on the image.

Each augmentation had a 50% probability of applying a randomised value within a predetermined range. Multiple augmentations could be applied simultaneously. This increased the size of the training set of 282 images tenfold to a total of 2820 images.

3.2 Training Setup

The ML training was implemented with the PyTorch library. To incorporate both available graphics cards, the DataParallel wrapper of PyTorch was used. The basic U-Net [8] was selected as the model. As in the original proposed U-Net, two convolution layers with a 3 × 3 kernel and stride 1, with a rectified linear unit (ReLU) activation function, are implemented at encoder and decoder levels, with feature sizes of 64, 128, 256, 512 and 1024. Unlike the original proposed U-Net, a batch normalisation layer is implemented behind every convolution layer, as well as padding. Batch normalization was introduced to improve training speed and stability [9]. The padding ensures that the image size remains unchanged at each level. This means that the image size in the decoder is the same as in the encoder. Not only does this improve the output resolution, it also makes it easier to implement the forwarding of the encoder stage to the decoder stage and the comparison of the ground truth with the model's output.

The height map is implemented as an additional input in two ways: as an extra channel to the input image (RGB-H) and as another input (RGB + H). Including the additional input in the U-Net requires a change to the first encoder stage. Here, both inputs go through a convolution layer, a batch normalisation layer, and a ReLU layer. The two feature maps of depth 64 are concatenated to create a single feature map with depth of 128, which is then reduced to 64 using a convolution layer. A batch normalisation and ReLu layer follow. The rest of the model remains unchanged. An overview of the model structure is given in Fig. 3.

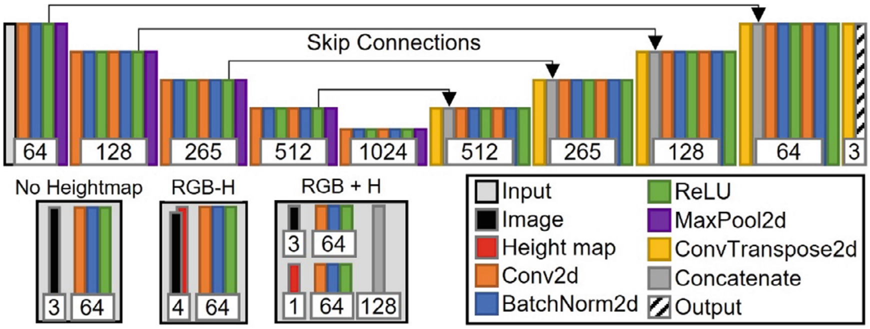

Fig. 3 Overview of the modified U-Net and input scenarios

For the model, the input size was chosen as 50% of the original image's resolution, which was 949 × 468 pixels. Normalisation of images and height maps was between 0 and 1. Cross Entropy Loss was used as the loss function and Adam as the optimiser. The model's predictions were evaluated using the Intersection over Union (IoU) score, which was calculated as the mean on the models predictions of the test set.

An initial hyperparameter study with the Optuna library was performed to determine the learning rate and batch size. The input scenario with no height map was trained 50 times with different learning rates in the range [1 × 10–5, 1 × 10–2] and batch sizes in the range [4, 24]. Validation was performed using the IoU score on a hold-out set with a 70/30 split from the training data. A hold-out sample was randomly selected and used consistently. The Tree-structured Parzen Estimator (TPE) was used as a sampler, and the Hyperbandpruner for pruning underperforming hyperparameters. Models were trained for in range [2, 10]. As the top ten models used a learning rate between 2.3×10^{-5} and 9.3×10^{-5}, so 5×10^{-5} was selected for use in this study. The hyperparameter study showed a small improvement with smaller batch sizes. However, the impact of a larger batch size was deemed acceptable in order to reduce training time. The batch size was set to 24, as this was the maximum using the available hardware.

Each of the three input scenarios (no height map, RGB-H and RGB + H) was trained for 30 epochs using the entire training dataset. A validation on the test set was performed for each epoch. This process was repeated using 80%, 60%, 40% and 20% of the training set. The reductions were achieved by excluding two to eight augmentations from all images, respectively. As there are no non-augmented images in the training set, the random distribution of test set images remains unchanged in these cases.

4 Results

For the three different models, the top plot of Fig. 4 shows the mean and standard deviation of the mean IoU score of the prediction on the test set over 30 epochs, based on five training runs. Over all training runs the model with no height map achieves its best prediction at epoch 30, with an IoU score of 0.705; the RGB-H model achieves its best prediction at epoch 30, with an IoU score of 0.7501; and the RGB + H model achieves its best prediction at epoch 28, with an IoU score of 0.7504.

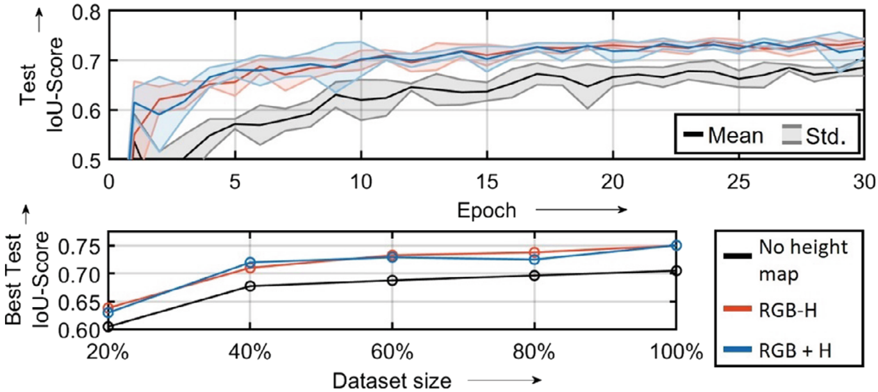

Fig. 4 Mean IoU score of the test set prediction of the three different models over the course of training on the complete dataset and smaller subsets

As the validation loss increased or stagnated from around epoch 20 in all three models, further improvement with a higher epoch count is unlikely. As the figure shows, at all epochs, the models with height maps outperformed the model that used only the image as input. Furthermore, at epochs 6 (RGB-H) and 5 (RGB + H), the models with height maps achieved a mean IoU scores within 0.05 of the final mean value, whereas the model without a height map did so at epoch 12. This demonstrates that the height map improves both the quality and speed of the model's convergence to the optimum.

At the bottom of Fig. 4, the plot shows the mean IoU score for all three models across all epochs in subsets of different sizes of the dataset. The dataset corresponding to 100% refers to the best runs of the models shown at top of Fig. 4. For datasets smaller when 100% only one training run was performed. Across different dataset sizes, the models with hight map as input both performed similar good, while the model with no height map performed worst overall. Decreasing the dataset size decreases the IoU score of the model with no height map and the model with combined RGB-H input equally. The model with split RGB + H input shows stable IoU scores between 40% and 60%. All models demonstrate a significant drop in performance when using only 20% of the dataset. As the IoU score does not plateau at or before 100% of the dataset, it can be concluded that the dataset is still too small to reach the theoretical maximum of model performance, as more training data would increase the performance further.

Fig. 5 shows an example prediction output for all models at epoch 30 for a single test image. In the model without the height map, some areas of discolouration caused by the adhesion of workpiece material were incorrectly identified as wear areas. The model with the combined RGB-H input also misidentified some discolouration, but to a lesser extent. Finally, the RGB + H split input model had an even smaller problem with discolouration, but ignored the adhesion of workpiece material to the cutting edge. As can be seen, all models produce predictions that resemble the manually created mask, albeit with an overestimation that is reduced by adding height maps. Predictions based on additional height map inputs are very similar for both models. Both models can accurately estimate wear progression along the cutting edge.

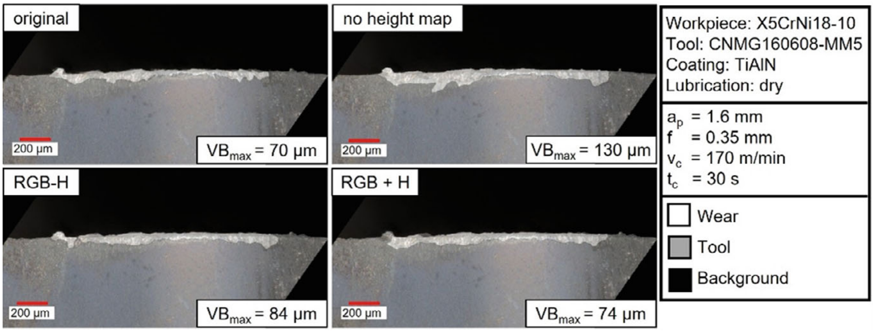

Fig. 5 Example prediction of a test image shown as overlay over the input image

5 Conclusion

This study presents a method for automatically evaluating flank wear width on indexable inserts. A dataset of 404 microscope images with height maps was created, showing varying levels of tool wear, with manual annotations for segmentation. A modified U-Net was trained to predict flank wear using three input methods: images only; images with height maps as a fourth dimension (RGB-H); and images with height maps as parallel inputs (RGB + H). Results showed incorporating height maps improved the IoU score by up to 0.045 (6.4%) and required up to 7 (58%) epochs less to achieve an IoU score close to the final value. This improvement was retained in smaller datasets.

As next steps, the implementation of wear evaluation after ISO 3685 is required so that the predicted mask can be transferred to comparable characteristic wear values. To further improve the model, different loss algorithms and models, such as the Segment Anything Model (SAM) [10] or newer variations of the U-Net [11], will be tested.

Acknowledgements. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—521385417.

Competing Interests The author(s) has no competing interests to declare that are relevant to the content of this manuscript.

References

1. Colantonio, L., Equeter, L., Dehombreux, P., Ducobu, F.: A systematic literature review of cutting tool wear monitoring in turning by using artificial intelligence techniques. Machines. **9**, 351 (2021). https://doi.org/10.3390/machines9120351
2. Miao, H., Zhao, Z., Sun, C., Li, B., Yan, R.: A U-Net-based approach for tool wear area detection and identification. IEEE Trans. Instrum. Meas. **70**, 1–10 (2021). https://doi.org/10.1109/TIM.2020.3033457
3. Holst, C., Yavuz, T.B., Gupta, P., Ganser, P., Bergs, T.: Deep learning and rule-based image processing pipeline for automated metal cutting tool wear detection and measurement. IFAC PapersOnLine. **55**, 534–539 (2022). https://doi.org/10.1016/j.ifacol.2022.04.249

4. Bergs, T., Holst, C., Gupta, P., Augspurger, T.: Digital image processing with deep learning for automated cutting tool wear detection. Procedia Manuf. **48**, 947–958 (2020). https://doi.org/10.1016/j.promfg.2020.05.134
5. Yoo, Y., Yang, G., Park, K., Hyun, Y., Jeong, S.: Extendable machine tool wear monitoring process using image segmentation based deep learning model and automatic detection of depth of cut line. Eng. Appl. Artif. Intell. **135**, 108570 (2024). https://doi.org/10.1016/j.engappai.2024.108570
6. Chen, H., Cheng, C., Hong, J., Huang, M., Kong, Y., Zheng, X.: An on-machine tool wear area identification method based on image augmentation and advanced segmentation. J. Manuf. Process. **132**, 558–569 (2024). https://doi.org/10.1016/j.jmapro.2024.10.085
7. Zhu, X., Chen, G., Ni, C., Lu, X., Guo, J.: Hybrid CNN-LSTM model driven image segmentation and roughness prediction for tool condition assessment with heterogeneous data. Robot. Comput. Integr. Manuf. **90**, 102796 (2024). https://doi.org/10.1016/j.rcim.2024.102796
8. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. (2015).
9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. (2015).
10. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R.: Segm. Anything. (2023). https://doi.org/10.48550/arXiv.2304.02643
11. AL Qurri, A., Almekkawy, M.: Improved UNet with Attention for Medical Image Segmentation. Sensors. **23**, 8589 (2023). https://doi.org/10.3390/s23208589