



30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021)
15-18 June 2021, Athens, Greece.

Benchmark of Automated Machine Learning with State-of-the-Art Image Segmentation Algorithms for Tool Condition Monitoring

B. Lutz^{a,b,*}, R. Reisch^{a,c}, D. Kisskalt^b, B. Avci^{a,c}, D. Regulin^a, A. Knoll^c, J. Franke^b

^aSiemens AG, Otto-Hahn-Ring 6, 81739 München, Germany

^bInstitute for Factory Automation and Production Systems, Friedrich-Alexander University Erlangen-Nürnberg, Egerlandstr. 9, 91058 Erlangen, Germany

^cChair of Robotics, Artificial Intelligence and Real-time Systems, Technical University of Munich, Boltzmannstraße 3, 85748 Garching at Munich, Germany

* Corresponding author. Tel.: +49-1522-2947688; E-mail address: lutz.benjamin@siemens.com

Abstract

In condition monitoring of cutting inserts for machine tools, vision-based solutions enable detailed wear pattern analysis. Besides the main failure modes of flank wear and tool breakage, other defects, such as chipping, grooves, and build-up-edges, can be detected and quantified. However, manual analysis of the images is time consuming and traditional machine vision systems have limited capabilities adapting to changing situations, such as different insert types. As a result, robust machine learning techniques are researched to support the process of classifying images and detecting defects through image segmentation. For the latter, a variety of highly optimized networks exists. Still, these networks require tuning by machine learning experts. In contrast, automated machine learning is a recent trend that greatly reduces these efforts through automated network selection and hyperparameter optimization. In this study, automated machine learning is compared with manually trained segmentation networks on the example of tool condition monitoring. To achieve this, a heterogeneous dataset of over 200 industrial cutting tool images is recorded and evaluated. Comparing the manually trained segmentation networks to the automated machine learning framework, it is determined that the automated machine learning solution is easier to handle, faster to train and achieves better accuracies than other approaches.

© 2020 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the scientific committee of the FAIM 2021.

Keywords: Tool Condition Monitoring, Image Segmentation, Machine Learning, Automated Machine Learning

1. Introduction

In industrial manufacturing, optimizing machining processes is important to reduce manufacturing costs and increase workpiece quality. One of the main cost drivers in subtractive manufacturing is the cutting tool as it gets worn out during machining and needs to be replaced and reworked frequently. Thus, for cost efficient machining, it is essential to use cutting tools as long as possible but exchange them before the product quality drops. For that reason, tool condition monitoring (TCM) systems are used to monitor the cutting tool condition during operation and prevent the machine from damage in case of a broken tool. [1, 2]

For signal processing in TCM, machine learning (ML) has become a popular method. ML provides the advantage of deducing knowledge from data instead of manually modeling the relations between cause and effect. Typical use cases include recommendations on e-commerce websites, real-time object detection for autonomous driving [3], process monitoring, and optimization in production [4].

Existing applications can be grouped in three categories. The first of these categories includes classification tasks where the goal is to determine one of many classes an image represents. Such an example can be found in [5] where Liu et al. use Convolutional Neural Networks (CNNs) to assign a class to a given image indicating the level of wear. The goal of the second category, regressions, is to determine one or

multiple continuous variables from the provided data, such as the exact flank wear width. As an example, [6] implements a regression model to predict surface roughness and tool wear condition. Thirdly, image segmentation approaches are investigated. These contain the most information as the algorithm's primary goal is to determine the meaning, i.e. the respective class, of each pixel in the image, thus detecting different regions of interest. This information can then consecutively be further processed to compute a variety of metrics. In [7], a window-based approach is proposed showing that image segmentation using CNNs is capable of detecting and differentiating between different type of wear defects.

Image segmentation is used in various areas, such as autonomous driving [8], medical imaging [9, 10], augmented reality [11], and satellite observation [12]. However, adapting existing solutions to a new problem or designing a completely new network requires high expertise. This is a major challenge for implementing such solutions.

Alternatively, automated machine learning (AutoML) is a recent trend in ML with the goal of focusing on the application rather than the model. Using AutoML, it is only necessary to provide the data and formulate the broad learning task, while the model selection and optimization is carried out by the respective framework. [13] investigates different AutoML tools and concludes that these tools are useful, but human input cannot be replaced.

As shown in [7], semantic image segmentation can be applied for tool condition monitoring of cutting tool inserts. However, the proposed approach requires expertise for set-up and is computational expensive due to the high amount of predictions that must be carried out. Thus, in this paper, both one-pass semantic segmentation networks and the sliding-window semantic segmentation algorithm merged with AutoML are investigated. All approaches are benchmarked on several quantitative and qualitative criteria.

First, different one-pass image segmentation networks are reviewed and the concept of using AutoML for the semantic segmentation task is explained in Section 2. In Section 3, the experimental settings and the training procedure for both one-pass networks and the AutoML integrated sliding window approach is detailed. Section 4 is dedicated to the comparison and evaluation of the experiment results of one-pass image segmentation networks and AutoML semantic segmentation. Section 5 concludes the paper and outlines future research activities.

2. Methods

ML has become a new trend for data analysis in industrial applications, learning relationships from data rather than labor-intensive manual modeling. Deep learning, as a subset of ML, has been studied and used for various vision tasks in recent years thanks to a higher amount of data available and progresses in the development of graphics processing unit (GPU) technologies. These improvements make it possible to train complex artificial neural networks (ANNs) in a faster way.

Especially CNNs, a type of ANN, are a promising alternative in many vision applications [14]. These networks

consist of two main parts. In the first convolutional part, convolutional layers detect two-dimensional features through multiple filters being applied to the input image. Detected features are aggregated through consecutive convolutional layers, with deeper convolution layers detecting more detailed and precise features. Furthermore, the filters are translation invariant, which allows for the detection of the same feature at different parts of the image. In each convolutional layer, the input image is downsampled, decreasing the image resolution. The second fully connected part consists of fully connected layers, whose last layer outputs the class probabilities. The class of the image is determined as the class with the highest probability at the output layer. [14]

2.1. Segmentation networks

Unlike image classification networks, image segmentation networks assign a class to every pixel in the input image. Thus, image segmentation networks are designed as encoder-decoder networks, where the encoder part is a convolutional network like in classification networks. Instead of aggregating the detected features after the convolutional part to a single class by a fully connected network, the decoder part is a reversed convolutional neural network increasing the image size with every layer until the original dimensions are met. Typically, the final output layer of such a network has the same width and shape as the input image, with the third dimension showing the class of each pixel. [15]

The size of the receptive field is the implication of how much contextual information is used during the segmentation task. Multi-scale semantic segmentation networks have filters with different receptive fields in convolutional layers, which allows the network to take advantage of global information during segmentation [16, 17].

In recent years, much research has been carried out to find and optimize architectures for image segmentation algorithms using deep learning. These have been used in various fields as mentioned in Section 1. It is assumed that an architecture designed for one application can also be applied to a different application. Thus, several state-of-the-art networks, FCN [18], U-Net [9], SegNet [19], LinkNet [20], and PSPNet [16], are compared for one-pass semantic segmentation in cutting tool images. The characteristics of each network are introduced briefly. More details about each architecture can be found in the respective original articles. FCN, U-Net, SegNet, and LinkNet are encoder-decoder networks, while PSPNet is a multi-scale network.

FCN is an end-to-end encoder-decoder image segmentation network. The decoder module upsamples the output image coming from the encoder module with a skip architecture, which combines semantic information from shallow layers in the encoder module with deep layers in the decoder module. Thereby, finer segmentation outputs can be obtained. [18]

U-Net was proposed for biomedical image datasets with a small amount of training data. For effective network training, U-Net applies strong data augmentation. The decoder module

has a skip architecture where entire feature maps in the encoder module are concatenated with upsampled outputs in the decoder module. [9]

SegNet is developed as a time and memory efficient encoder-decoder network. To achieve this, SegNet stores the maximum indices obtained during maximum pooling in the encoder module to be used for upsampling in the decoder module. Afterwards, the upsampled maps are inserted into the convolutional layers with trainable filters to obtain dense results in the decoder module. [19]

LinkNet aims to construct a real-time image segmentation network. For this reason, the encoder network is an image classification network [21] with a small number of parameters compared to other image segmentation networks. [20]

PSPNet is able to learn from different sizes of receptive fields. In this network, feature maps of different sizes are obtained parallelly with differently sized filters. All feature maps are upsampled to the same resolution and concatenated. Thus, the final feature map leverages from information obtained from different scales. [16]

2.2. Sliding window and AutoML

Besides analyzing an entire image at once, the image can be analyzed pixel by pixel using a sliding window approach for generating a feature map (Figure 1). A window with a certain width and height slides through the image treating each window as an input feature map. At the borders of the image, padding is applied for sliding feature map generation. These feature maps are evaluated by a classification model predicting the class of the central pixel. Once this process is carried out for every pixel, the predictions are recombined, resulting in a fully segmented image. Thus, the segmentation task can be reduced to a multitude of simpler classification tasks. Instead of handcrafting a classification model, AutoML is investigated for the classification task. As a framework, Google Cloud AutoML Vision is used in this study [22].

For implementation, all windows are created first. Consecutively, they are separated into training, validation, and test sets. All data is fed to the AutoML training pipeline. The classification task is carried out by the AutoML framework in a black box process without the need to select a model or adjust any hyperparameters. However, some limitations, such as the maximum computation time, can be set. After training, the AutoML framework returns the trained model and performance metrics from the test set.

2.3. Benchmark criteria

To compare the one-pass neural networks and the window-based algorithms for semantic segmentation, different quantitative and qualitative metrics are used.

The most common metrics for semantic segmentation are pixel-wise accuracy, intersection over union (IoU), and mean intersection over union (mIoU). Pixel-wise accuracy is calculated by dividing the number of correctly predicted pixels by the total number of pixels. IoU is defined as the ratio of the number of correctly predicted pixels to the total number of pixels in the union area of predicted segmentation and ground

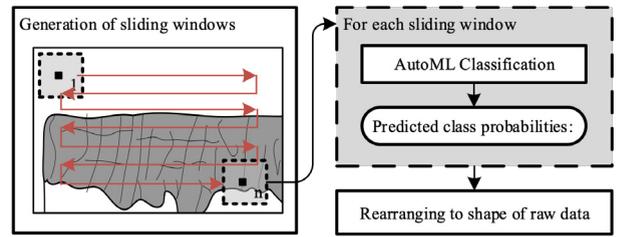


Fig. 1. Procedure of the sliding window approach

truth segmentation. mIoU is calculated by averaging IoU over different classes. [23]

Let c_{ij} be the number of pixels whose true class is i and predicted as j . The total number of classes is defined as n . $g_i = \sum_j c_{ij}$ is defined as the number of pixels belonging to class i . $p_j = \sum_i c_{ij}$ is the number of pixels predicted as class j . Based on these definitions, pixel-wise accuracy (1), IoU (2) and mIoU (3) can be computed.

$$\text{Pixel - wise accuracy} = \frac{\sum_i^n c_{ii}}{\sum_i^n g_i} \quad (1)$$

$$\text{IoU}_i = \frac{c_{ii}}{g_i + p_i - c_{ii}} \quad (2)$$

$$\text{mIoU} = \frac{1}{n} \sum_i^n \frac{c_{ii}}{g_i + p_i - c_{ii}} \quad (3)$$

For imbalanced datasets, pixel-wise accuracy might provide misleading results, as the metric might become biased towards the majority class. Since the segmentation task includes several classes, the state-of-art one-pass semantic segmentation networks are compared based on the mIoU score.

Besides the performance of the model, other factors might influence the decision of which model to implement. Therefore, the performance of the approaches in terms of needed ML expertise, flexibility of input image size, explainability of the model, and the time required for set-up and testing is investigated further.

3. Experimental procedure

3.1. Dataset

For training, 207 cutting tool images from two different cutting tool inserts are available. The images are taken with an optical microscope, showing the flank of each cutting tool insert. All images contain the regions *background* and *undamaged tool body* by default. Depending on the tool's condition, regions depicting different tool wear defects are present and visible as well. In general, the defects visible on the flank of the cutting tool can be grouped into *flank wear*, *chip notch*, *peening wear*, *build-up-edge* (BUE), and *groove* [24]. In the present dataset, 206 out of the 207 images show flank wear, 40 images depict *grooves*, and 123 images contain *BUE*. Other defects are not visible.

All images are labeled manually by process experts. The final dataset consists of raw images showing the data acquired

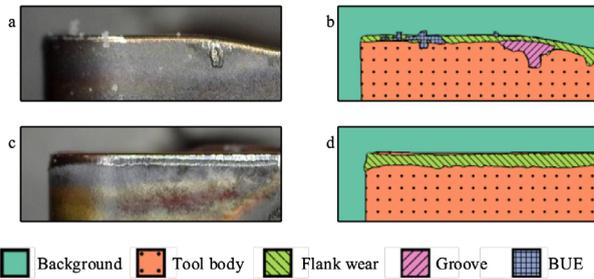


Fig. 2. Raw image of cutting tool type one (a) with its respective mask (b) and raw image of cutting tool type two (c) with respective mask (d)

by the microscope (Figure 2, a & c) and the respective image masks showing the class of each pixel (Figure 2, b & d).

3.2. Training procedure

To avoid overfitting, the dataset is split into a training set, validation set, and test set, with a ratio of approximately 8:1:1. As a result, there are 167, 20, 20 images for training, validation, and testing, respectively. For balanced learning, the splitting is carried out based on the flank wear width such that all datasets have a similar distribution of tool conditions. The training dataset is used to train and optimize each model. During parameter optimization, the validation set is used to monitor the training progress and rate different alternatives. The final metrics are computed on the test set, which is not used for training nor for parameter optimization.

The most present classes in each image are *background* and *undamaged insert body*. Thus, to focus on the classes showing wear defects, the images are cropped to the region of interest starting at the upper edge of the insert. As a result, the image size is decreased from 1280x1024 pixels to 1024x256 pixels.

Each network is trained for 50 epochs with a batch size of 8. Optimizers and parameters are set as described in the papers [9, 16, 18–20] (Table 1). Most networks are optimized with stochastic gradient descent (SGD) while LinkNet is optimized using the root mean square prop (RMSProp) optimizer.

Since the amount of training data is small and increasing the dataset through additional experiments is resource-intensive, data augmentation is applied to increase the variety within the existing data. Most of the existing variety in the data is caused by differences in brightness, blur, and contrast. Thus, augmentation is carried out by means of random brightness, clahe, random gamma, blur, motion blur, random contrast, and hue-saturation-val. The probability for each augmentation to be applied, as well as the chosen settings, are shown in Table 2.

Table 1: Optimizer, learning rate, momentum and weight decay values for each network

	Optimizer	Learning rate	Momentum	Weight Decay
FCN	SGD	10e-5	0.9	5e-4
U-Net	SGD	10e-5	0.99	-
SegNet	SGD	10e-2	0.9	-
LinkNet	RMSProp	5e-4	-	-
PSPNet	SGD	10e-3	0.9	10e-5

Table 2: Overview of the applied augmentation types and their parameters

Augmentation Type	Parameters	Probability
Clahe	clip_limit: 4	0.30
RandomBrightness	limit: (-0.2, 0.6)	0.30
RandomGamma	gamma_limit: (100, 200)	0.30
Blur	blur_limit: (3, 7)	0.45
MotionBlur	blur limit: (3, 12)	0.45
RandomContrast	limit: 0.9	0.45
HueSaturationValue	hue shift limit: 20; saturation shift limit: 30; value shift limit: 50	0.45

For implementation, the Albumentations library [25] is used. Examples of the resulting images with applied data augmentation can be seen in Figure 3.

4. Results and discussion

In this section, the different one-pass image segmentation networks explained in Section 2 are compared in terms of their mIoU score. Moreover, the sliding-window image segmentation approach, proposed in [7], is benchmarked with the AutoML classification based on pixel-wise accuracy. Finally, the best results of one-pass image segmentation networks and the sliding window approaches are investigated thoroughly.

4.1. Comparison of different segmentation networks

In the first experiment, the optimizers and parameters of every network are set according to the original papers as mentioned in Section 3. The scores are computed with and without data augmentation and can be seen in Table 3. U-Net, PSPNet, and LinkNet achieve similar high scores of 0.69, 0.68,

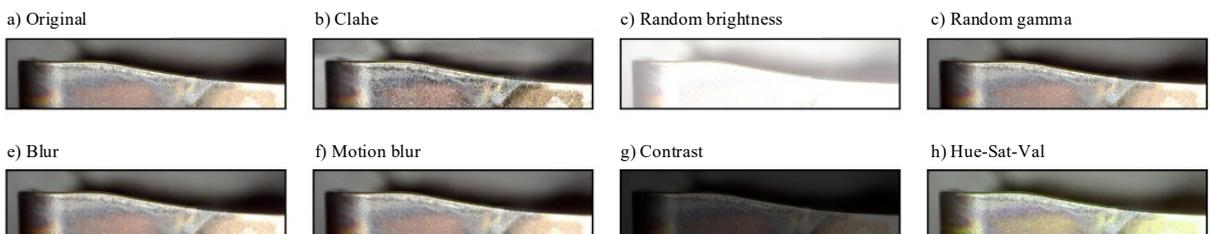


Fig. 3. Original image (a) and the resulting augmented images with the different augmentation strategies applied (b-h)

and 0.67 mIoU, respectively, for training without data augmentation. These findings support the claim of U-Nets suitability for small datasets, as mentioned in Section 2.1.

Using data augmentation, some networks show an improved learning behavior, whereas others do not benefit from the artificially created data. The biggest increase of 0.13 is observed for LinkNet. FCN and PSPNet also show slight increases, while U-Net and SegNet show reduced scores with augmentation. Overall, the highest score of 0.80 with the original configurations is achieved by using LinkNet and data augmentation. Besides, LinkNet does not only provide a high mIoU score, but also offers real time inference due to an efficient network architecture, making it a promising candidate for industrial applications.

Furthermore, the individual IoU scores of each class are a direct indicator of the network's performance in detecting specific segmentation classes (Table 3). All networks achieve the highest IoU scores for the classes *background* and *undamaged tool body*. This result indicates that the majority classes of the images are the easiest to detect for one-pass networks. As *flank wear* is the most important defect, it is necessary to segment *flank wear* as accurately as possible. LinkNet achieves the highest score of 0.55 for *flank wear* segmentation in addition to the highest mIoU score. For the class *groove*, U-Net and PSPNet reach the highest score of 0.80, whereas for the class *BUE*, LinkNet returns the best IoU score of 0.79. The FCN and SegNet networks are not able to detect the classes *flank wear*, *groove*, and *BUE* in the present configuration, limiting their prediction capabilities for the given situation.

4.2. Evaluation of AutoML approach

In the second experiment, the sliding window segmentation approach using AutoML for classification is investigated. First, windows with sizes of 48x48 pixels are created from the input images. Out of all windows created, 4000 windows are randomly selected for the training dataset and 500 each for the test and validation datasets, with an even distribution among all classes in every set. The AutoML classification pipeline is fed with the corresponding data to train a model to predict the class of the center pixel of the input window.

According to the confusion matrix (Table 4), *background* and *groove* are the classes predicted the best. 97% of true *background* and *groove* windows are predicted correctly. Since *background* pixels are darker compared to the other pixels, the easy distinction among those seems reasonable. As the second highest percentage, 94% of *BUE* windows are predicted

Table 3: The mIoU and IoU scores show the effect of data augmentation and performance for the individual classes of each network with *background* as Bg. and *undamaged tool body* as tool

	Aug. (mIoU)		Individual class scores (IoU)				
	without	with	Bg.	Tool	Wear	Groove	BUE
FCN	0.32	0.36	0.94	0.87	~0	~0	~0
U-Net	0.69	0.63	0.95	0.85	0.07	0.80	0.50
SegNet	0.47	0.31	0.78	0.76	~0	~0	~0
LinkNet	0.67	0.80	0.99	0.96	0.55	0.70	0.79
PSPNet	0.68	0.73	0.97	0.93	0.41	0.80	0.54

Table 4: Confusion matrix of image segmentation with AutoML

Ground truth	Prediction				
	Background	Tool	Flank Wear	Groove	BUE
Background	97%	1%	0%	1%	1%
Tool	2%	88%	9%	1%	0%
Flank Wear	1%	1%	87%	1%	10%
Groove	0%	2%	0%	97%	1%
BUE	0%	0%	6%	0%	94%

correctly. *BUE* pixels are very bright compared to the other classes and accumulate at the top of the tool. Thus, the *BUE* windows lie on the border of *background* and *undamaged tool body*. Thereby, they are providing characteristic data, which can be seen in the high true classification rate. The lowest percentage of correctly predicted pixels belongs to the classes *undamaged tool body* and *flank wear*. *Flank wear* is located at the upper part of the tool and is not very bright. The windows of the classes *flank wear* and *undamaged tool body* are rather similar with only small visual differences. Thus, the model is not able to predict *undamaged tool body* and *flank wear* windows as precisely as the other classes.

Comparing the AutoML solution to the manual segmentation approach based on deep learning described in [7], a slightly higher accuracy can be observed. This can be assumed to be due to the higher amount of data available or better models being used by the AutoML framework.

4.3. Comparison of one-pass segmentation with the sliding window approach

In this part, LinkNet, as the best one-pass network, and the sliding window approach merged with AutoML are compared. Based on the mIoU score, the AutoML approach achieves 0.86, outperforming LinkNet with a 0.80 score under similar conditions. In addition, the individual IoU scores for each class are investigated (Table 5). While LinkNet shows higher numbers than AutoML for the classes *background* (0.99 compared to 0.94) and *undamaged tool body* (0.96 compared to 0.85), AutoML shows higher number for the three defect classes *flank wear* (0.76 compared to 0.55), *groove* (0.94 compared to 0.70), and *BUE* (0.84 compared to 0.79). As it is more important to identify the defect classes correctly, AutoML performs better than LinkNet on a class level as well.

Besides comparing the prediction accuracies, other factors such as the model complexity and the effort needed for getting started are important factors influencing the decision of which approach to choose. Thus, the one-pass segmentation network, the sliding window approach with a manually constructed network and the sliding window approach using the model generated by AutoML are compared based on the complexity

Table 5: The IoU scores for each class show the good defect detection capabilities of the AutoML model

IoU	Background	Tool	Flank wear	Groove	BUE
LinkNet	0.99	0.96	0.55	0.70	0.79
AutoML	0.94	0.85	0.76	0.94	0.84

Table 6: The three approaches, one-pass segmentation networks, sliding window with a custom network and sliding window with AutoML are compare based on model accuracy, model complexity, input flexibility, model explainability, time needed for setup, and inference time. A high number of stars represents desirable behavior.

	One-Pass Segmentation	Sliding window	
		Custom network	AutoML
Accuracy	★★	★★	★★★
Complexity	★	★	★★★
Flexibility	★★	★★★	★★★
Explainability	★★	★★	★
Setup time	★	★	★★
Inference time	★★★	★	★

of the model, the explainability of the model, the flexibility of the model, the time needed for training, and the time needed for inference (Table 6).

Looking at the model complexity, the first two approaches, one-pass image segmentation and sliding window with custom networks, require manual network creation and, thus, a high level of ML expertise. Implementing LinkNet from scratch requires not only knowledge about ML, but also of the model's architecture and necessary programming languages and libraries. In contrast, for setting up the AutoML approach, only a few adjustments must be made by the expert since most of the process is handled by the framework. This way, process experts can obtain a semantic image segmentation model with little ML knowledge. However, some level of expertise is still needed for formulating the learning task and setting up the necessary AutoML framework.

Whereas open source solutions for AutoML might return the optimized networks, many commercial solutions only provide a prediction pipeline. Thus, further investigation and understanding of the models found with AutoML approaches might be limited. Contrarily, implementing LinkNet from scratch allows experts to understand and improve the details about the model, increasing its explainability.

For the flexibility of the model regarding the input image size, the one-pass segmentation networks do allow various image sizes. However, they are scaled or cropped to a predefined network-specific size, losing information in the process. Both sliding window approaches are more robust against different sizes and shapes of the image, if the minimum window size is met.

Regarding the training time, no significant differences were observed among the different algorithms. For inference however, the one-pass segmentation networks are faster than the sliding window approaches, as only one evaluation has to be carried out.

For time critical applications, different strategies might help reducing the total inference time. Looking at the hardware used for model execution, specialized hardware such as GPUs can calculate the prediction results more efficiently. On the software side, the prediction process for the sliding window approach, consisting of a multitude of prediction tasks, could be separated into sub processes that could be distributed to multiple machines, thus further reducing inference times.

In industrial settings, increased variety among cutting tool

inserts, especially regarding size, shape, and color, is expected. Here, the sliding window approaches should be more resistant against cutting tools of different sizes and shapes, as only small parts of the cutting tool that are independent of the overall insert size and shape are analyzed. For the consideration of different colors of the cutting tool, data augmentation is promising. Thereby, the color of the cutting tools present in the dataset can be artificially manipulated to generate data that represents cutting tools of different color. Thus, the model is trained with data from multiple colors, which improves its prediction performance when encountering such tools during operation.

5. Summary and Outlook

In this paper, different machine learning approaches for tool condition monitoring and semantic image segmentation are investigated. Analyzing the one-pass networks, FCN, U-Net, SegNet, LinkNet, and PSPNet, LinkNet shows the most promising results. Using data augmentation, a high mIoU score can be achieved, even though the data set is rather small.

As an alternative to handcrafted networks, automated machine learning is presented, which requires less machine learning expertise and reduced set-up time, decreasing the complexity of the process. It is found that using a sliding window approach in combination with the automated machine learning for the same learning tasks yields better results compared to LinkNet, the best performing one-pass network. Only the computational effort for inference is significantly higher for the sliding window approach, as many predictions have to be carried out. It can be concluded that automated machine learning is a promising and adequate choice for image segmentation in tool condition monitoring.

In future work, a more diverse dataset containing multiple different types of tools should be investigated, thus allowing the approaches to be rated based on their general usability. Besides increasing the training data and adapting the network architecture, transfer learning strategies should be investigated for an efficient knowledge transfer among different cutting tools.

References

- [1] Y. Zhou, W. Xue, Review of tool condition monitoring methods in milling processes, *The International Journal of Advanced Manufacturing Technology* 96 (2018) 2509–2523. <https://doi.org/10.1007/s00170-018-1768-5>.
- [2] A. Siddhpura, R. Paurobally, A review of flank wear prediction methods for tool condition monitoring in a turning process, *The International Journal of Advanced Manufacturing Technology* 65 (2013) 371–393. <https://doi.org/10.1007/s00170-012-4177-1>.
- [3] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, Real-Time Object Detection (2016).
- [4] A. Mayr, D. Kießkalt, M. Meiners, B. Lutz, F. Schäfer, R. Seidel, A. Selmaier, J. Fuchs, M. Metzner, A. Blank, J. Franke, Machine Learning in Production – Potentials, challenges and exemplary applications, *Procedia CIRP* 86 (2019) 49–54. <https://doi.org/10.1016/j.procir.2020.01.035>.
- [5] X. Wu, Y. Liu, X. Zhou, A. Mou, Automatic identification of tool wear based on convolutional neural network in face milling process, *Sensors (Basel, Switzerland)* 19 (2019). <https://doi.org/10.3390/s19183817>.
- [6] T. Özel, Y. Karpat, Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks, *International Journal*

- of Machine Tools and Manufacture 45 (2005) 467–479. <https://doi.org/10.1016/j.ijmachtools.2004.09.007>.
- [7] B. Lutz, D. Kisskalt, D. Regulin, R. Reisch, A. Schiffler, J. Franke, Evaluation of deep learning for semantic image segmentation in tool condition monitoring, in: *Proceedings of ICMLA 2019*, pp. 2008–2013.
- [8] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, R. Urtasun, MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving, 2016.
- [9] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, 2015.
- [10] F. Milletari, N. Navab, S.-A. Ahmadi, V-Net: Fully convolutional neural networks for volumetric medical image segmentation (2016).
- [11] P. Pigny, L. Dominjon, Using CNNs for users segmentation in video see-through augmented virtuality (2019).
- [12] A. Yoshihara, T. Hascoet, T. Takiguchi, Y. Ariki, Satellite image semantic segmentation using fully convolutional network (2018).
- [13] A. Ebadi, Y. Gauthier, S. Tremblay, P. Paul, How can automated machine learning help business data science teams? (2019).
- [14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition (1998).
- [15] X. Liu, Z. Deng, Y. Yang, Recent progress in semantic image segmentation, *Artif Intell Rev* 52 (2018) 1089–1106. <https://doi.org/10.1007/s10462-018-9641-3>.
- [16] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, 2016.
- [17] F. Yu, V. Koltun, Multi-Scale context aggregation by dilated convolutions, 2015.
- [18] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation (2015).
- [19] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, *IEEE transactions on pattern analysis and machine intelligence* 39 (2017) 2481–2495. <https://doi.org/10.1109/TPAMI.2016.2644615>.
- [20] A. Chaurasia, E. Culurciello, LinkNet: exploiting encoder representations for efficient semantic segmentation (2017) 1–4. <https://doi.org/10.1109/VCIP.2017.8305148>.
- [21] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015.
- [22] E. Bisong, Google AutoML: cloud vision, in: *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, Apress, Berkeley, CA, 2019, pp. 581–598.
- [23] G. Csurka, D. Larlus, F. Perronnin, F. Meylan, What is a good evaluation measure for semantic segmentation?, in: *BMVC, 2013*, p. 2013.
- [24] M. Lanzetta, A new flexible high-resolution vision sensor for tool condition monitoring, *J Mater Process Technol* 119 (2001) 73–82. [https://doi.org/10.1016/S0924-0136\(01\)00878-0](https://doi.org/10.1016/S0924-0136(01)00878-0).
- [25] A. Buslaev, A. Parinov, E. Khvedchenya, V.-I. Iglovikov and A.-A. Kalinin, Albumentations: fast and flexible image augmentations, 2018.