Journal of Machine Engineering, 2022, Vol. 22, No. 4, 115–132
ISSN 1895-7595 (Print) ISSN 2391-8071 (Online)

Received*: 09. September 2022* / Accepted: *26. October 2022* / Published online: *04 November 2022*

machining, AI, computer vision,
image processing, chip,
chip shape, chip segmentation

Hagen KLIPPEL[1*]
Samuel PFLAUM[2]
Michal KUFFA[1]
Konrad WEGENER[1]

AUTOMATED EVALUATION OF CONTINUOUS AND SEGMENTED CHIP GEOMETRIES BASED ON IMAGE PROCESSING METHODS AND A CONVOLUTIONAL NEURAL NETWORK

The aim of this work is to present a new methodology for the automated analysis of the cross-sections of experimental chip shapes. It enables, based on image processing methods, the determination of average chip thicknesses, chip curling radii and for segmented chips the extraction of chip segmentation lengths, as well as minimum and maximum chip thicknesses. To automatically decide whether a chip at hand should be evaluated using the proposed methods for continuous or segmented chips, a convolutional neural network is proposed, which is trained using supervised learning with available images from embedded chip cross-sections. Data from manual measurements are used for comparison and validation purposes.

1. INTRODUCTION

The numerical prediction of machining processes requires constitutive material model parameters, which are difficult to determine for the prevailing conditions where high plastic strain rates are overlaid with high plastic strains and temperatures. Various material models for the simulation of machining processes exist, of which the Johnson-Cook flow stress model [1] is most often used. It requires five material dependent parameters, which are difficult to obtain at the aforementioned harsh conditions. A method to determine the constitutive material model coefficients use inverse methods where the orthogonal cutting experiment itself serves as material test. If only the process forces are used as input for the inverse identification, it can lead to unambiguous constitutive model coefficients, as discussed in [2] [3]. To reduce this unambiguity more information out of the cutting experiments has to be considered in the inverse material parameter identification, e.g. the shape of the resulting chips as demonstrated in [4].

[1] IWF, ETH Zürich, Switzerland
[2] DTDS, Bühler AG, Switzerland
[*] E-mail: hklippel@ethz.ch
https://doi.org/10.36897/jme/156091

During metal cutting processes different chip types (discontinuous, elemental, continuous, segmented) and chip forms occur, depending on the material, the tool geometry and the process parameters [5, 6]. The chip types and forms can be classified according to ISO 3685-1977, see for example the overview provided in [7]. In an attempt to get deeper process insights, the extraction of geometrical features of such chips is required. Basic geometrical definitions of the chip exist in literature [8–11], but their evaluation requires manual analyses, which is a tedious task since it is time-consuming and the reproducibility is not ensured. An approach for the automated chip thickness extraction is proposed in [4], with a limitation to numerical simulation results. For this reason, in this publication an automated solution is presented for cross-sections of chips, which enables fast and accurate chip geometry evaluations. The presented algorithm automatically evaluates average chip thicknesses and chip curling radii. In the case of segmented chips, it can be used to extract segment lengths as well as minimum and maximum segment thicknesses. The algorithm is applied to experimental test data, which has been evaluated with regards to average chip thicknesses by a manual method in [12] and to manually measured segmented chips in [13]. Because the selection of the algorithm for the evaluation of continuous or segmented chip types is manual, an AI algorithm using a convolutional neural network (CNN) is proposed to automatically identify whether the chip type is segmented or continuous. It is trained on experimental data of Ck45 and Ti6Al4V chips from [12].

2. THEORETICAL BACKGROUND

2.1. IMAGE PROCESSING

Image processing deals with the problem of processing and analysing information out of images. Usually, in a first step filter operations are used to smooth images [14], where a filtered image $I'(x,y)$ is computed by a convolution of an image $I(x,y)$ with a kernel $w(x,y)$ [15]:

$$I'(x,y) = w(x,y) * I(x,y) \tag{1}$$

Similar to the other filters, a morphological filter requires a kernel or structuring element B to perform the filtering on the input image X. The two basic morphological operations are dilation and erosion, where dilation grows an image and erosion shrinks it. Based on these two operations, other morphological operations such as opening or closing can be formed, where an example of each operation is shown in Fig. 1.

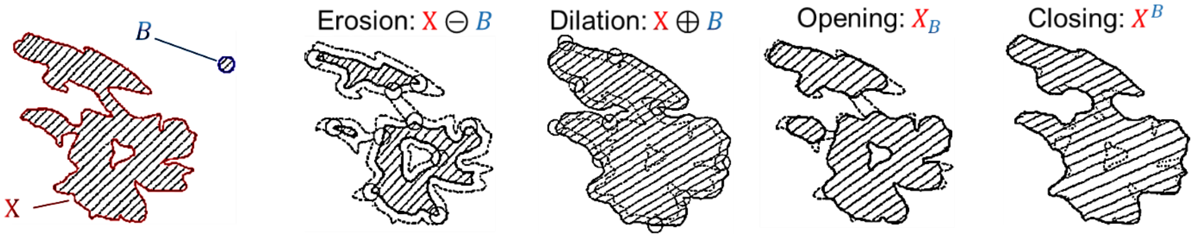

Fig. 1. Morphological operations with an input image ***X*** and a structuring element ***B***, adopted from [16]

After image preparation by filters, the image has to be separated into foreground and background. An image segmentation approach with low computational effort is the method of Otsu [17] where a greyscale image is converted into a binary image by division of the pixel intensity distributions into two classes with the help of a threshold value. The threshold value is determined such that the intra-class variance is minimized, which is similar to maximizing the inter-class variance. An example for that approach is given in Fig. 2.

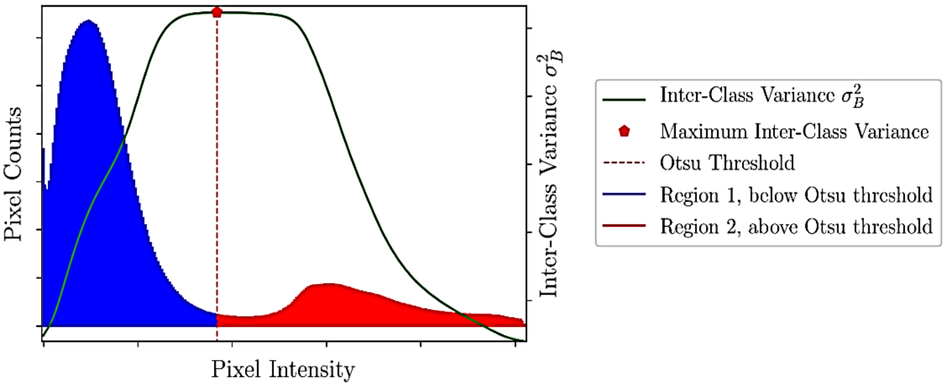

Fig. 2. Otsu’s thresholding method: the maximum inter-class variance divides the image into two regions

With the thresholding method, objects can be extracted from images, but its edges are not characterized. Various approaches for edge detection algorithms exist [18]. In this work the approach from Suzuki [19] is used, which is a border following algorithm and can detect outer and inner borders as well as several objects in an image.

Corners of an object can be used to analyse the shape or orientation of an object. The method of Chetverikov and Szabo [20] uses a variable triangle inside of the curve and considers its opening angle α.

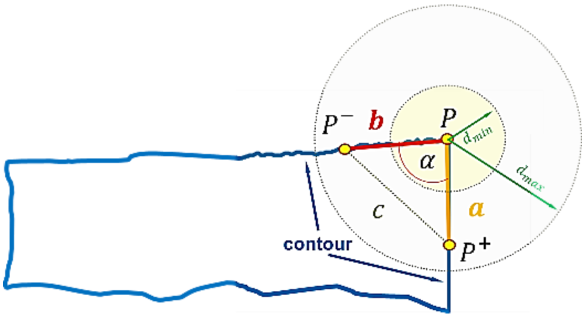

Fig. 3. Corner detection with the Chetverikov and Szabo algorithm [20]. A variable triangle is spanned by the points $\boldsymbol{P^-}, \boldsymbol{P}, \boldsymbol{P^+}$ on three points on the contour and contain the opening angle $\boldsymbol{\alpha}$. The points $\boldsymbol{P^-}$ and $\boldsymbol{P^+}$ are admissible on the contour within the distances $\boldsymbol{d_{min}}$ and $\boldsymbol{d_{max}}$

The algorithm therefore of two passes, where in a first pass the candidates for a corner point are determined and in a second pass the useless candidates are eliminated. This means that in the first pass the detector tries to create a variable triangle (P^-, P, P^+) in each point P of the curve under consideration of:

$$\begin{gathered} d_{min}^2 \le |P - P^+|^2 \le d_{max}^2 \\ d_{min}^2 \le |P - P^-|^2 \le d_{max}^2 \\ \alpha \le \alpha_{max} \end{gathered} \tag{2}$$

where P^- is the kth anticlockwise neighbour of P, P^+ is the kth clockwise neighbour of P, d_{min} is the minimum and d_{max} the maximum admissible distance of P^- and $P^+$to P, see also Fig. 3. The opening angle α is:

$$\alpha = \arccos \frac{a^2 + b^2 - c^2}{2ab} \tag{3}$$

with

$$a = |P - P^+|, b = |P - P^-|, c = |P^+ - P^-| \tag{4}$$

2.2. OBJECT DETECTION – CONVOLUTIONAL NEURAL NETWORKS

The detection of objects is a fundamental problem in computer vision where the aim is the detection and localization of objects in images. Different approaches exist for the object detection, where according to [21] large progresses in computational performance allow the application of deep learning methods to various fields. CNN is a computational processing system inspired by nature like the biological nervous system. The CNN consists of many computational nodes which are connected to each other and collaborate and weight the information of the input in such a way that a learning process is established leading to an improved output. There are many different approaches to the architecture of a CNN, see also [22]. Different algorithms have been developed for object detection and classification of which the single shot detector (SSD) is introduced here. The SSD [23] enables to perform the object detection during one stage. Therefore, the algorithm consists of two parts that are the backbone model and the SSD-head. The backbone model consists of a standard network for image classification where in the original version the VGG- 16 networks [24] is used. The usage of the backbone model is to only extract features from the input image and therefore the classification layers of the base network are not considered. The network architecture of SSD with a VGG-16 backbone model is shown in Fig. 4.

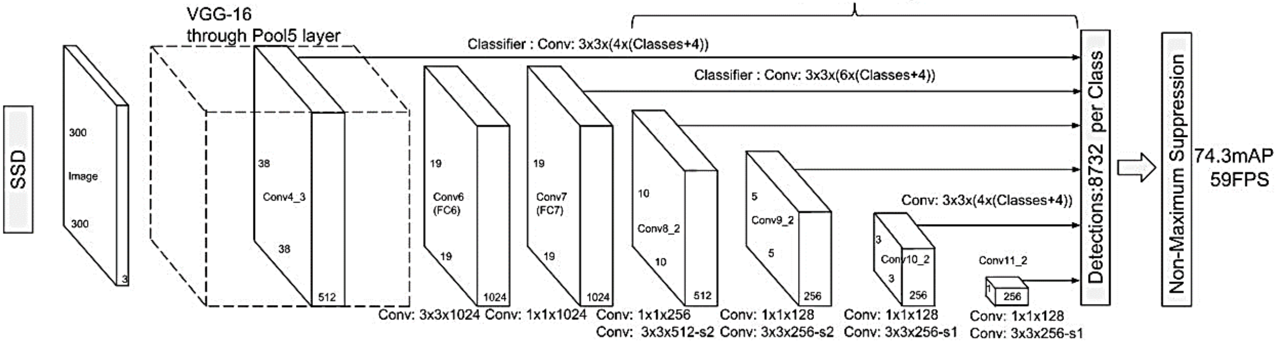

Fig. 4. SSD architecture with VGG-16 backbone model, from [23]

3. EXPERIMENTAL BASIS

In this section the experimental basis, which has been used for the development of the evaluation methods, is introduced together with the manual measurement of chip geometry features.

3.1. CUTTING EXPERIMENTS

Quasi-orthogonal cutting experiments on Ck45 and Ti6Al4V from [12, 13] serve as database for the development of an automated methodology of the chip geometry evaluation. Chips from Ti6Al4V exhibited chip segmentation behaviour, while Ck45 showed chip segmentation only at higher feed rates and cuttings speeds, otherwise continuous chips. The cutting experiments were performed for various combinations of feed rates ($f = 0.01 \ldots 0.4\ \mathrm{mm/rev}$) and cutting speeds ($v_c = 10 \ldots 500\ \mathrm{m/min}$). Images of the cross-sections of embedded chips as well as measured average chip thicknesses are documented in [12] and a small selection of segmented chips is documented in [13]. The images of embedded chips from these experiments are used as data basis for this work. An example of available chip images is shown in Fig. 5, showing the chip in raw condition, before embedding, after embedding and grinding, and after etching.

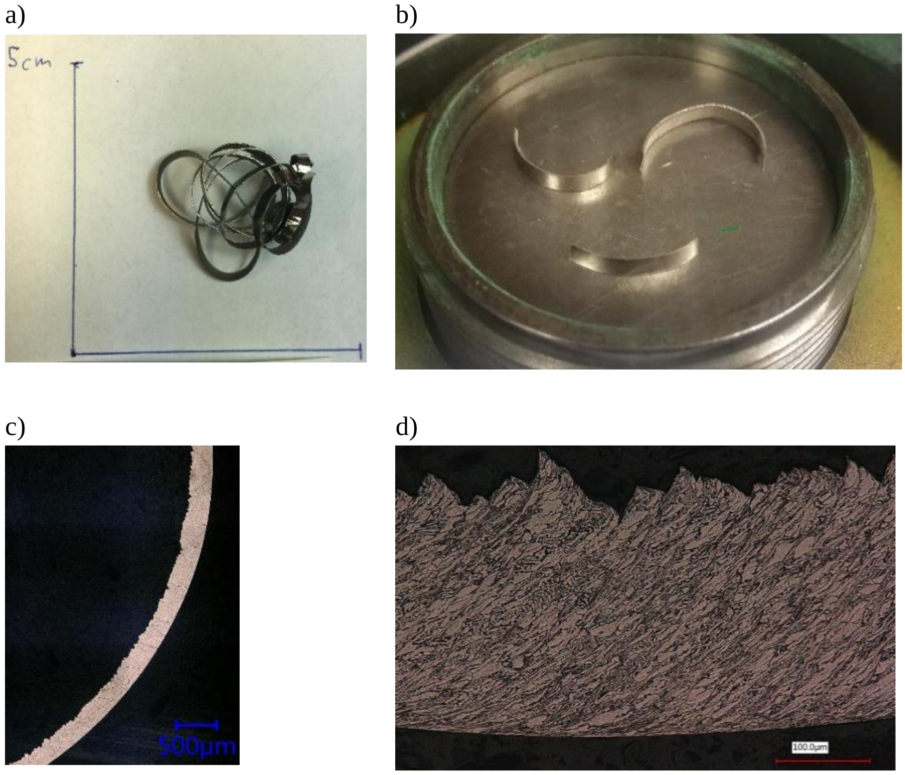

Fig. 5. A Ti6Al4V chip from experiment V0010 from [12] in raw condition (a), before embedding (b), after embedding into Bakelite and grinding (c) and a close-up revealing the microstructure after etching with Kroll (d)

3.2. MANUAL MEASUREMENTS OF CHIP GEOMETRY FEATURES

Chip geometry features are measured from embedded chip cross-sections using a Keyence VHX-5000 microscope. In a manual procedure average chip thicknesses and chip curling radii are extracted for continuous and segmented chips. For segmented chips, minimum and maximum chip thicknesses as well chip segment distances are measured at a few locations of the chip. These geometric features are shown exemplarily in Fig. 6.

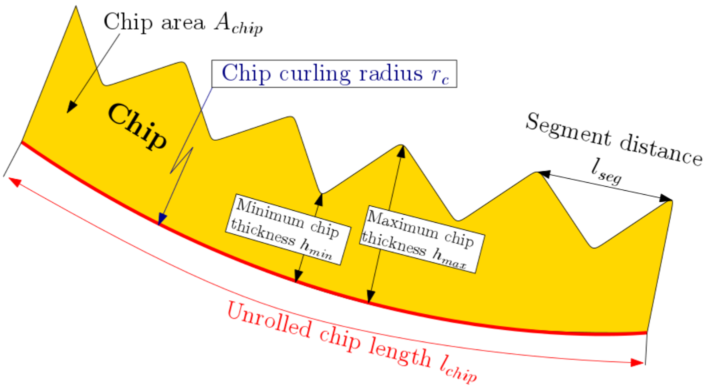
Chip area A_{chip}
Chip curling radius r_c
Chip
Segment distance l_{seg}
Minimum chip thickness h_{min}
Maximum chip thickness h_{max}
Unrolled chip length l_{chip}

Fig. 6. Basic geometry features of a chip cross-section, from [12]

The average chip thickness h_{avg} is determined from the cross-sectional area of the chip A_{Chip} and the unrolled chip length l_{Chip} by an approximation with a polygonal chain:

$$h_{avg} = \frac{A_{Chip}}{l_{Chip}} \tag{5}$$

Example measurements for a continuous and a segmented chip are shown in Fig. 7.

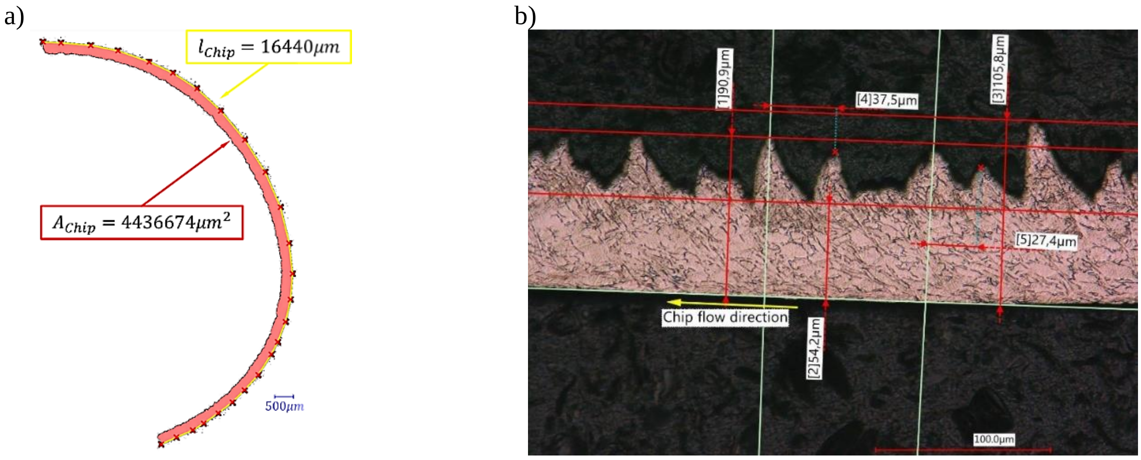

Fig. 7. (a) Measurement of the average chip thickness using the cross-sectional area and unrolled chip length of a Ti6Al4V chip (V0010) from [12], b) measurement of minimum and maximum chip thickness and chip segment distance of a Ti6Al4V chip (V0350) from [13]

4. RESULTS

In this section the image preparation process is described followed by the results of the average chip thickness measurements, the chip curling radii determination and the segmented chip evaluation. Whether the algorithms for continuous or segmented chips are applied is up to here a user decision. To automate this decision, an automated detection of continuous and segmented chips, based on a CNN, is proposed.

4.1. IMAGE PREPARATION

The automated image processing is developed in Python (v3.9) using OpenCV (v4.5.3), SciPy and NumPy. In the following the main steps of the image preparation and segmentation, the chip separation and corner detection is outlined.

Before chip features can be automatically extracted, a preparation of the chip images is required. The colour space of the images is converted from RGB into greyscale and a Gaussian filter reduces noise. In the next step Otsu's [17] thresholding method is used to binarize the image and separate the chip from the background, an example is shown in Fig. 8.

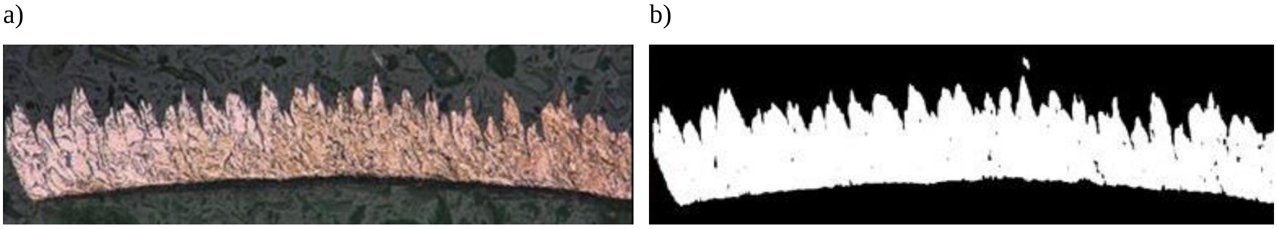

Fig. 8. Raw (a) and binarized chip (b), where the morphological operator is not yet applied

After binarization, morphological operators are applied to separate entities, which are falsely connected to the chip. An example where two chip fragments are interconnected after binarization is displayed in Fig. 9. There, the application of morphological filters lead to the separation of both chip fragments.

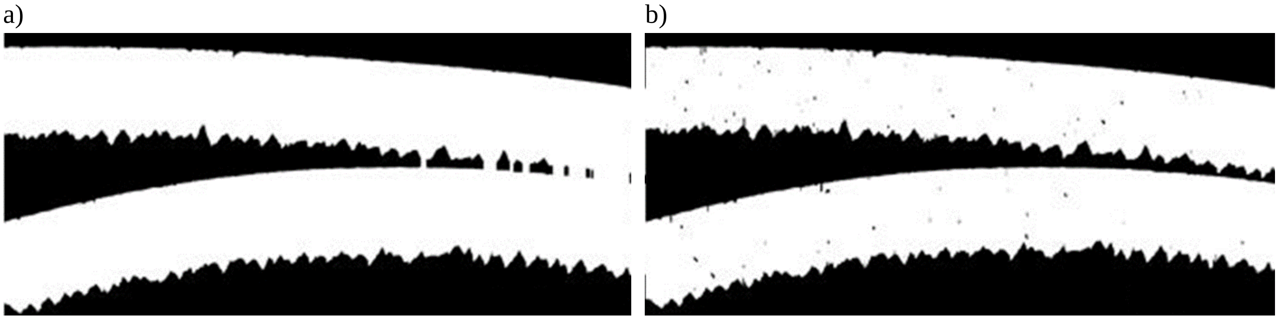

Fig. 9. Chip after binarization (a) and after application of morphological filters (b)

Suzuki's [19] border following method is applied to obtain the borders of the objects contained in the image. The border following methods returns a list with all objects, of which the largest object (largest chip fragment) in the image is considered in the following evaluation steps, see Fig. 10.

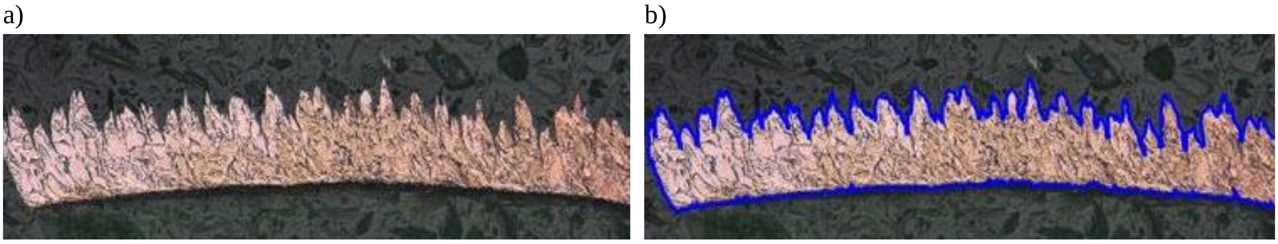

Fig. 10. Raw chip (a) and selected contour (b) with Suzuki's [19] border following method

The tool contact side of the chip is determined iteratively. In a first step, the centre of gravity (COG) of the chip is computed. From this COG the largest distance to the chip contour is determined in vertical and horizontal direction. The larger distance of both determines the alignment of the chip in the image. Next, the image is split along the shorter distance and the procedure is repeated with iteratively halving the contour through the respective COG until left and right ends of the chip are found. The procedure is depicted in Fig. 11, where both end points are connected by the black and green contour. The shorter contour of both determines the tool contact side, which is the green contour in the figure. Since this method returns only approximate start and end points of the chip, the corner detection with the method from Chetverikov and Szabo [20] is applied to find the exact positions.

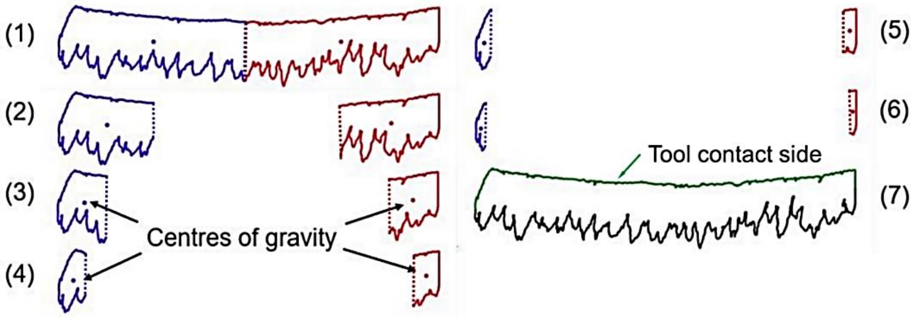

Fig. 11. Determination of the tool contact side of the chip by iteratively halving the contour (steps 1–6) through the respective centres of gravities until left and right ends of the chip are found. Both end points are connected by the black and green contour, where the shorter of both determines the tool contact side, which is here the green contour (step 7)

So far, the image processing is based on a pixel level of the images. For the evaluation of the chip geometry, the ratio of pixel-to-micrometre ratio is required. Unfortunately, this value is not accessible from the raw data of the images. Instead, the length scale of the picture is automatically detected and the length value is identified using optical character recognition (OCR) implemented in Tesseract [25].

4.2. AVERAGE CHIP THICKNESSES AND COMPARISON TO MANUAL ANALYSIS

The chip thicknesses are determined for each contour point of the tool contact side of the chip. For this purpose, local edge normal are computed on the tool contact side of the chip and their respective intersections with the free surface of the chip yield the local chip thickness. Since the tool contact side of the chip is usually not smooth, a Savitzky–Golay filter [26] is first applied for smoothing the contour on the tool contact side of the chip. The approach for the local chip thickness determination is shown in Fig. 12.

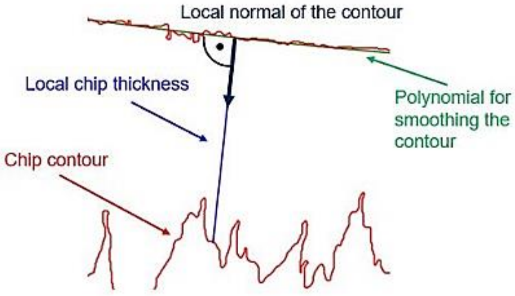
Local normal of the contour
Local chip thickness
Polynomial for smoothing the contour
Chip contour

Fig. 12. Local chip thickness determination with a surface normal on the smoothed contour of the chip contact side

The local chip thicknesses along a single chip are shown in Fig. 13 (a & b). The average of all local thicknesses determines the average chip thickness in the automatic method. It is determined as $h_{avg\_auto} = 35.7$ µm with a standard deviation of $\sigma_{h_{avg\_auto}} = 7.4$ µm and compares well with a manually measured chip thickness of $h_{avg\_manual} = 34.3$ µm in [12]. The rather high standard deviation is because the displayed chip is not continuous but segmented, which hints that a more elaborated method is required to extract further chip features like minimum and maximum chip thickness as well as the chip segment length and is introduced later.

The automated chip geometry evaluation algorithm is used to compare average chip thicknesses versus manual measurements conducted in [12]. The results are displayed in Fig. 13c and show a very good agreement with only a few outliers. Potential reasons are due to inaccuracies in the manual measurement as well as different sections of the chips are used for the manual and automatic evaluation.

Alternatively, the chip compression ratio (CCR) can be computed from the manual and automated average chip thickness with:

$$CCR = \frac{h_{avg}}{f} \tag{6}$$

The CCRs computed from manual and automated chip thickness measurements are displayed in Fig. 14. Similar to the comparison in Fig. 13 the agreement between manual and automated method is very good with only a few outliers. It has to be noted that towards very low feed rates the CCR tends to show very high values which indicates that for thin chips the measurement uncertainty of the microscope image negatively impacts the accuracy for both, the manual and automated measurement. Since images of both continuous and segmented chips are evaluated, the CCR may become meaningless for segmented chips, since CCR<1 may occur in some cases.

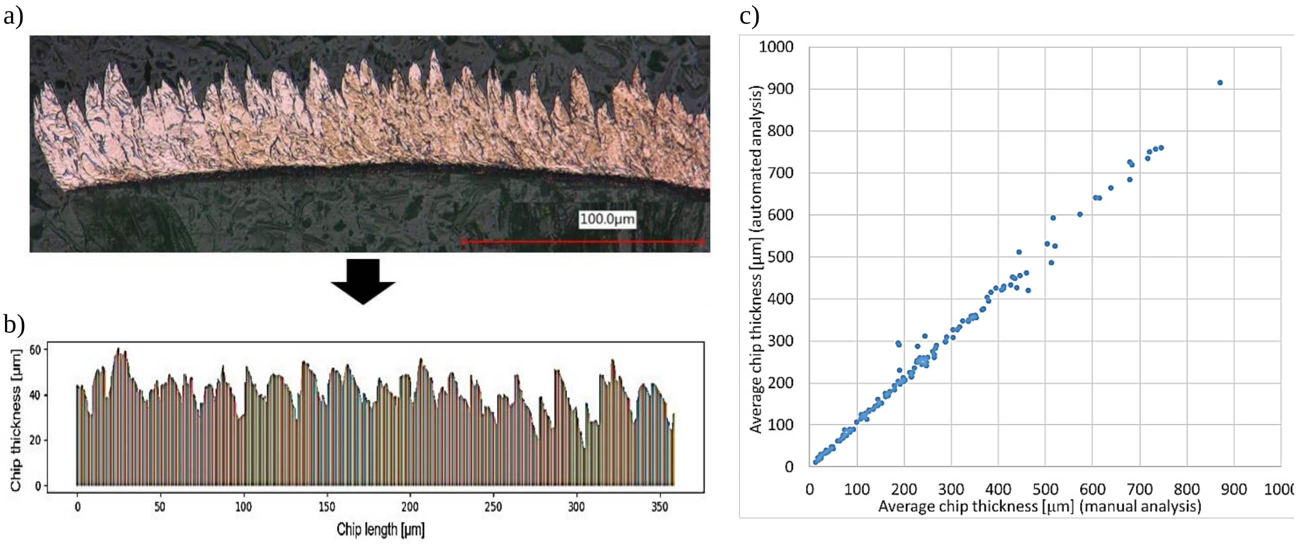

Fig. 13. Chip image (a) and its chip thickness variation along the chip length (b) and comparison of manual and automated analysis of the average chip thickness (c) – for 140 different chips showing a good agreement

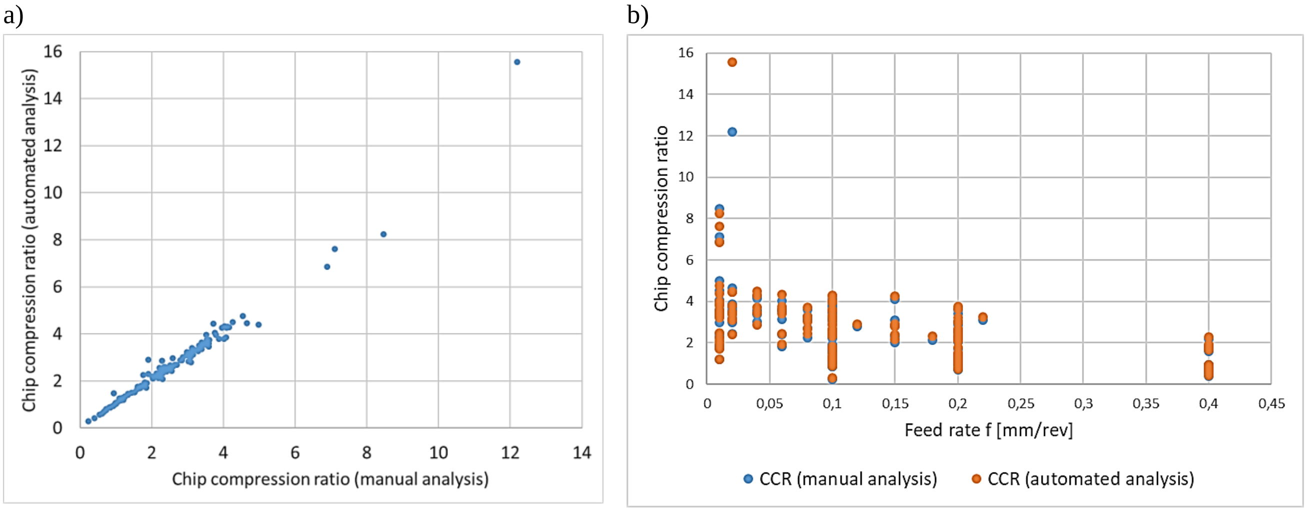

Fig. 14. Comparison of CCR determined from manual and automatic chip thickness measurement (a) and display of both CCR versus feed rate

4.3. CHIP CURLING RADIUS DETERMINATION

The chip curling radii r_c are determined from the coordinates of the tool contact side contour points $P(x, y)$ of the chip by using [27]:

$$r_c = \frac{(\dot{x}^2 + \dot{y}^2)^{3/2}}{\dot{x}\ddot{y} - \ddot{x}\dot{y}} \tag{7}$$

In Fig. 15 an embedded chip is shown together with a histogram of the corresponding chip curling radii.

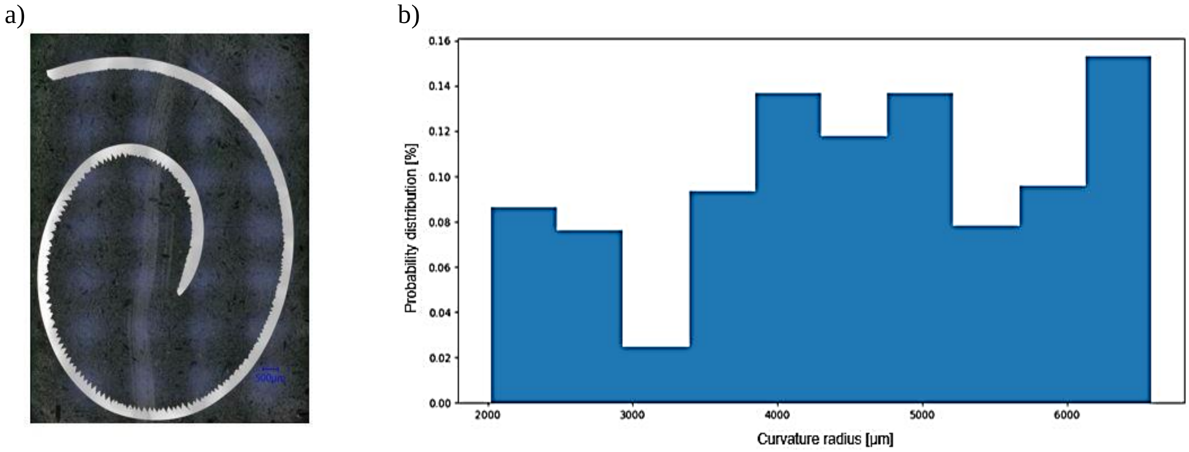

Fig. 15. Example evaluation of the curvature radius from test V0012. Chip in embedded condition (a) and histogram of the corresponding chip curling radii (b)

The average chip curling radii $\mathrm{r_{c\_mean}}$ are given for a selection of chips in Table 1 together with minimum and maximum values as well as the standard deviations σ_{r_c}. It can be seen that the standard deviation is very high. The reason for this is that the local chip curvature for a chip can change very strongly, and since the chip curvature radius is determined for the individual contour points, the radius can vary greatly. The results of the curvature radii are only of limited significance, as the curvature pictured does not necessarily correspond to the true curvature of the chip. The reason for this is that the chips, especially thinner ones, have a low stiffness and can bend easily, which leads to deformations in the embedding process under high pressure. On the other hand, embedding required sometimes fixations with clamps or rolling of thin chips into plastic foils for stabilization, which in turn could lead to a change in the curvature.

Table 1. Chip curling radii determination with the automated method on a selection from [12]

Test	v_c [m/min]	f [mm/rev]	r_{c_mean}[mm]	r_{c_min}[mm]	r_{c_max}[mm]	σ_{r_c}[mm]
V0002	12.6	0.01	9.9	0.6	3213.3	82.5
V0012	10.5	0.2	5.5	2.3	9	1.9
V0024	74.3	0.01	7	0.01	5415.2	105.3
V0035	190.5	0.1	248.2	21	721731.3	7456.2
V0042	190.5	0.4	927.7	36.1	1311014.6	17712.5
V0068	254.1	0.2	40	5.5	5037.2	108.1
V0198	199.9	0.04	97.1	4.7	137590	1242.3
V0261	250	0.01	26.8	0.1	64580.7	624.9
V0302	10	0.02	14.5	1.9	7081.8	144.8

4.4. SEGMENTED CHIP GEOMETRY EVALUATION

While the extraction of features from continuous chips is rather straightforward, the analysis of segmented chips proves more difficult. An average thickness can be determined as well, but additionally each segment has a minimum thickness h_{min}, a maximum thickness h_{max} and a distance between two segments l_{Seg}, see also Fig. 16.

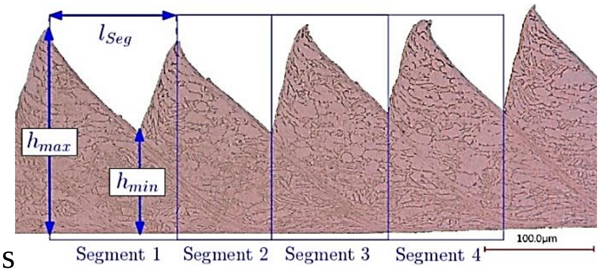

Fig. 16. Geometric features of segmented chips, here an etched Ti6Al4V chip from test V0062 in [12]

Since these segments repeat periodically, an Ansatz for automatic feature extraction is to first perform a fast Fourier transform (FFT) of the chip thickness profile and determine the frequency with the largest amplitude see Fig. 17. The frequency at the largest amplitude corresponds to the segment distance l_{Seg}. A chip contour and its FFT spectrum are displayed in Fig. 17.

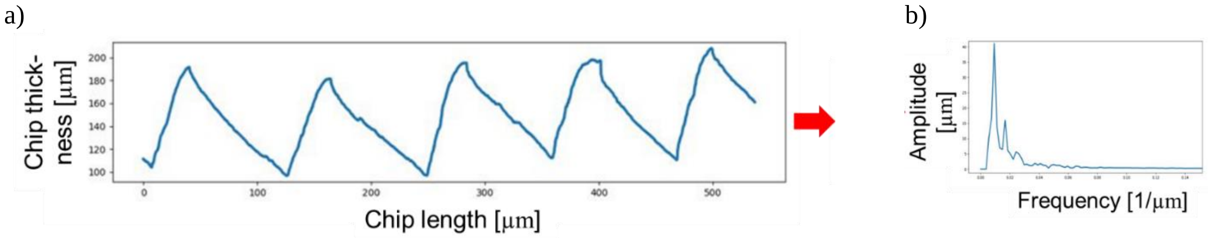

Fig. 17. Chip thickness profile (a) and FFT of the chip thickness profile (b) from test V0062 in [7]

Using the frequency at the largest amplitude the chip contour can be reconstructed with an inverse FFT, see Fig. 18.

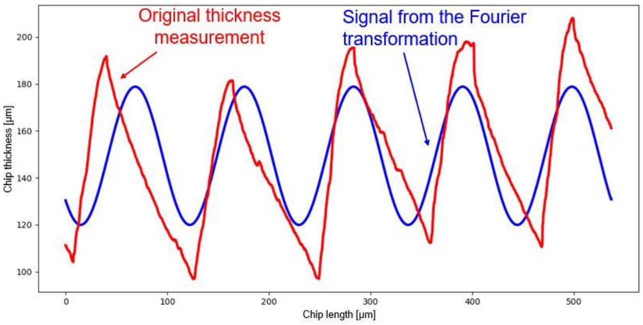

Fig. 18. Original chip shape (red) and reconstructed shape (blue) with the maximum peak from the FFT

The segment distance l_{Seg} is used in the next step to extract the local minima (h_{min}) and maxima (h_{max}) of each segment. For this purpose, the evaluation starts on the left side of the chip contour and within a distance l_{Seg} the next extreme point is searched. This extreme point serves then as the new starting point for the identification of the next extreme point within a distance l_{Seg}. This search procedure is shown in Fig. 19, where red line segments correspond to the current search range for extreme points and green dots denote identified extreme points.

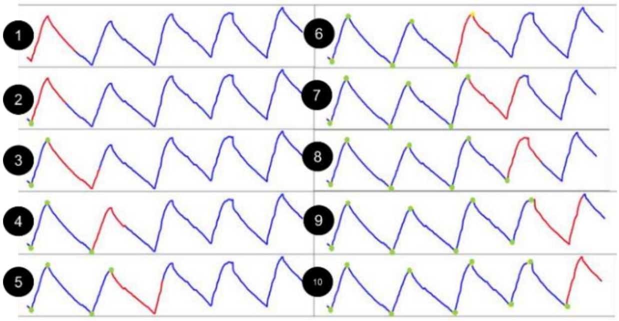

Fig. 19. Segment-wise identification of local extreme points (green points) in 10 steps. The red marked area equals the average segment length determined from the FFT and is used to find piece-wise the next extreme point of the contour

The final extreme points are overlaid with the original chip image in Fig. 20 and demonstrate that the algorithm properly selects the minima and maxima of the chip contour.

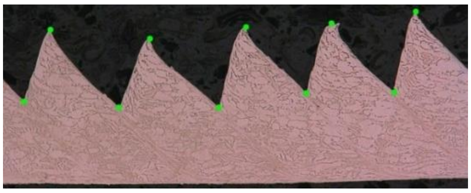

Fig. 20. Detected extreme points marked on the contour with green dots

A comparison of manually measured minimum and maximum chip thicknesses is provided with Table 2 and shows a good agreement of the automated and the manual measurement.

Table 2. Comparison of manual and automated measurement of minimum and maximum chip thicknesses from a selection of segmented chips

Test	Cutting speed	Feed rate	Min. chip thickness h_{min} [µm]			Max. chip thickness h_{max}[µm]		
	v_c [m/min]	f [mm/rev]	manual [13]	$h_{\min\_auto}$	$\sigma_{h_{\min\_auto}}$	manual [13]	$h_{\max\_auto}$	$\sigma_{h_{\max\_auto}}$
V0060	381.3	0.1	80	82	11	150	138	12
V0320	19.9	0.01	24	14	2	35	25	3
V0325	19.9	0.04	57	67	5	69	81	4
V0348	125	0.01	12	12	3	23	21	3
V0350	125	0.04	54	51	3	106	74	7
V0461	400.1	0.01	7–12	9	2	23-25	19	2
V0471	400.1	0.04	33-36	31	4	63–71	61	8

Similarly, Table 3 shows a comparison of manually measured average chip segment lengths l_{Seg} versus automatically determined values from average distances between subsequent minimum chip thicknesses h_{min}, maximum chip thicknesses h_{max} and from the FFT. The different chip segment lengths from the automated method are given with their respective standard deviations, except for the one from the FFT, since only the largest peak from the FFT-spectrum has been used. The three automatically determined segment lengths l_{Seg} are similar at the higher feed rates, but towards lower feed rates ($f < 0.04$ mm/rev) the deviations to the manual measurements increase.

Table 3. Comparison of manual and automated measurement of chip segment length

Test	Cutting speed	Feed rate	Chip segment length l_{Seg} [µm]						
	v_c [m/min]	f [mm/rev]	manual [13]	from h_{min}	$\sigma_{l_{Seg\_min}}$	from h_{max}	$\sigma_{l_{Seg\_max}}$	from FFT	$\sigma_{l_{Seg\_FFT}}$
V0060	381.3	0.1	51–56	55	9	54	8	66	-
V0320	19.9	0.01	13–23	30	18	30	13	16	-
V0325	19.9	0.04	11–16	26	9	27	10	25	-
V0348	125	0.01	6–15	24	7	24	7	44	-
V0350	125	0.04	27–38	37	10	38	6	43	-
V0461	400.1	0.01	15–24	20	8	20	6	45	-
V0471	400.1	0.04	34–44	42	11	42	8	48	-

The reason for this is that the segmentation frequency of the chips is not regular, see for example the chip in Fig. 21. The distance between subsequent chip segments varies strongly, which makes it difficult for the algorithm to identify all peaks.

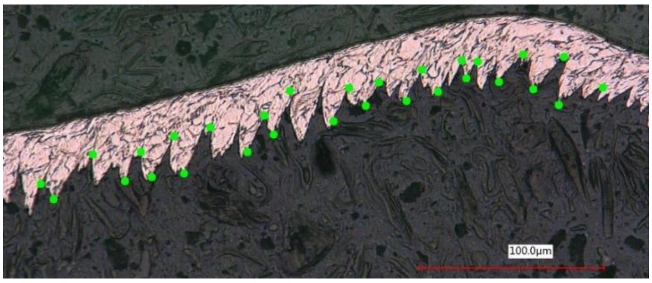

Fig. 21. Very irregular chip segmentations can cause problems in the automated extreme point determination, here on a chip generated at a very low feed rate of *f* = 0.01 mm/rev (experiment V0001 from [12])

4.5. CLASSIFICATION OF CHIP TYPES USING CNN

In order to fully automate the evaluation, an automated distinction between continuous and segmented chips is required to select whether average chip thicknesses (continuous chips), or minimum and maximum chip thicknesses and chip segment lengths (segmented chips) are exacted. For this purpose a CNN is trained using a SSD-network [23] with a subset of images from embedded chips from [12]. An SSD-network with a MobileNet V2 backbone and a FPNLite feature extractor is used. This SSD-network is provided with the Model Zoo, was pre-trained with the COCO2017 dataset and is implemented in TensorFlow [28] (v2.8). Here, it is trained on an individual dataset created from a selection of 303 chip images from [12]. In each chip image, bounding boxes are manually created and the chips are classified into continuous and segmented chips. These chip images are then rotated 3 times (90°, 180°, 270°) to increase the total number of images to 1’212 of which 1’032 are used for training and the remaining 180 images are used for testing. Parameters according to Table 4 are used for the training and testing of the SSD-model.

Table 4. Parameters for training and testing of the SSD-model

Batch size	Training steps	Training set	Test set
8	50’000	1’032 images	180 images

The model training required 12 hours on a GPU NVidia GTX 970 and with the test images a mean average precision (mAP) of 0.84 is observed. This means that 84% of all detections correctly identified continuous or segmented chips, which is a high value compared to results obtained in [23]. Figure 22 shows the learning rate during the training steps, the losses on the training and test set of the images as well as the mAP for different threshold value for the intersection over union (IoU).

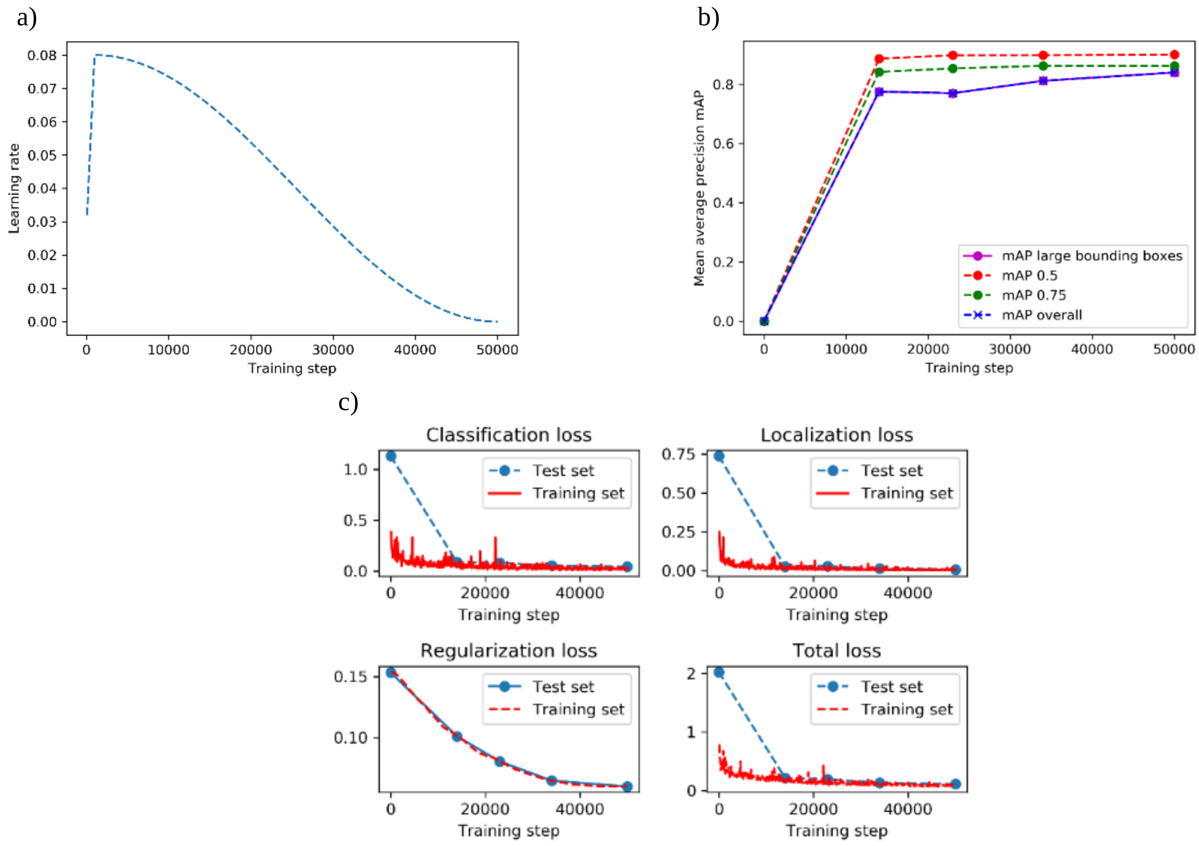

Fig. 22. Trends of the learning rate (a), mean average precisions mAP (b) and losses (c) during the training

One potential reason for the good mAP-value is that the considered images are similar. For training and testing, the images of etched specimens of Ti6Al4V-chips and Ck45-chips were used where the background for all images is quite similar and the chip looks similar, since there is not a great change in colour. Furthermore, all Ti6Al4V-chips are segmented and only the Ck45-chips contain segmented and continuous chips, which means that it is possible that the classification is mainly based on other features then the actual chip segmentation. This means that the usage of the trained network is limited to images, which were created from embedded chips. In Fig. 23, examples for true and false detections are shown.

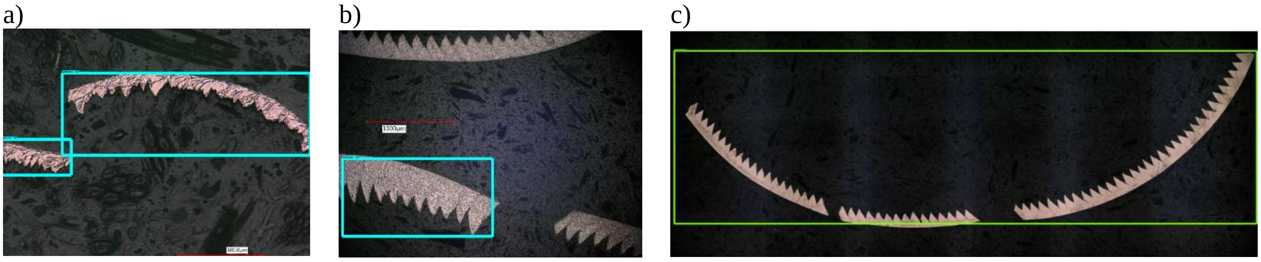

Fig. 23. Example detections with the trained SSD-network, the box colour indicates the detection of segmented (cyan) and continuous (green) chips. (a): successful detection of chip segmentations on two chip fragments, (b): questionable detection as only one of three segmented chips is found and (c): a segmented chip is incorrectly detected as a continuous chip, which is indicated by the green frame

5. SUMMARY

A method for the automated analysis of chip geometries was developed utilizing supervised learning with a CNN, image processing methods and other algorithms. The advantages of the method are the low evaluation times, the reproducibility and the high quality of the results. The method was tested on embedded chips from quasi-orthogonal cutting experiments of Ck45 and Ti6Al4V.

Average chip thicknesses computed with the proposed method showed a good agreement to manual measurements. The curvature radii of the chip contours were successfully determined. The determined radii correspond to the radii of the imaged chips, although it is difficult to judge whether these radii correspond to the actual radii of curvature, as these can change significantly as a result of the embedding process of the chips.

For segmented chips, an automated method for the evaluation of geometric features was proposed based on the FFT of the chip thickness profile. The method was validated against manual measurements of segmented chips and showed very good agreements. However, towards very low feed rates the chip segmentation becomes irregular in the experimental chips, which made it difficult to obtain meaningful results for both, manual and automated analysis.

A CNN was trained to detect objects in the chip images and classify them into continuous and segmented chips. The network is based on the SSD-model and was trained with images from available experimental chips. The classification into continuous and segmented chips achieved a comparably high mean average precision of 0.84 on the test set.

The detection of several chip fragments in one image requires improvements, since not always all of the fragments are found.

In future work the classification from the object detection could be used as a switch to automatically select if a continuous or segmented chip is to be evaluated with the image processing methods. It is further recommended to increase the image database for learning and testing of the CNN. Moreover, it would be desirable to investigate other properties of the chip, such as built-up edges or the thickness and distance of the individual shear zones. The methods developed in this work form a good basis that can be expanded in future work based on the points mentioned above.

REFERENCES

[1] JOHNSON G.R., COOK W.H., 1983, *A Constitutive Model and Data for Materials Subjected to Large Strains, High Strain Rates, and High Temperatures*, Proc. 7th Inf. Sympo. Ballist., 541–547.

[2] SHROT A. BÄKER M., 2010, *Is it Possible to Identify Johnson-Cook Law Parameters from Machining Simulations?*, Int. J. Mater. Form., 3, 443–446, doi: 10.1007/s12289-010-0802-4.

[3] SHROT A., BÄKER M., 2012, *A Study of Non-uniqueness During the Inverse Identification of Material Parameters*, Procedia CIRP, 1, 72–77, doi: 10.1016/j.procir.2012.04.011.

[4] HARDT M. BERGS T., 2021, *Considering Multiple Process Observables to Determine Material Model Parameters for FE-Cutting Simulations*, Int. J. Adv. Manuf. Technol., 113/11–12, 3419–3431, doi: 10.1007/s00170-021-06845-6.

[5] JEMIELNIAK K., 2021, *Review of New Developments in Machining of Aerospace Materials*, J. Mach. Eng., 21/1, 22–55, doi: 10.36897/jme/132905.

[6] TANABE I., YAMAGAMI Y., HOSHINO H., 2020, *Development of a New High-Pressure Cooling System for Machining of Difficulut-to-Machine Materials*, J. Mach. Eng., 20/1, 82–97, doi: 10.36897/jme/117776.

[7] GRZESIK W., 2017, *Advanced Machining Processes of Metallic Materials: Theory, Modelling and Applications*, Second edition, Amsterdam, Boston, Elsevier.

[8] KHARKEVICH A. VENUVINOD P.K., 1999, *Basic Geometric Analysis of 3-D Chip Forms in Metal Cutting*, Int. J. Mach. Tools Manuf., 39/5, 751–769, doi: 10.1016/S0890-6955(98)00065-0.

[9] KHARKEVICH A.G., VENUVINOD P.K., 2002, *Extension of Basic Geometric Analysis of 3-D Chip Forms in Metal Cutting to Chips with Obstacle-Induced Deformation*, Int. J. Mach. Tools Manuf., 42/2, 201–213, doi: 10.1016/S0890-6955(01)00115-8.

[10] KOUADRI S., NECIB K., ATLATI S., HADDAG B., NOUARI M., 2013, *Quantification of the Chip Segmentation in Metal Machining: Application to Machining the Aeronautical Aluminium Alloy AA2024-T351 with Cemented Carbide Tools* WC-Co, Int. J. Mach. Tools Manuf., 64, 102–113, doi: 10.1016/j.ijmachtools.2012.08.006.

[11] DEVOTTA A., BENO T., LÖF R., 2017, *Finite Element Modelling and Characterisation of Chip Curl in Nose Turning Process*, Int. J. Mach. Mach. Mater., 19/3, 277–295, doi: 10.1504/IJMMM.2017.084009.

[12] KLIPPEL H., SÜSSMAIER S., KUFFA M., WEGENER K., 2022, *Dry Cutting Experiments Database Ti6Al4V and Ck45*, arXiv:2209.04197, doi: 10.48550/ARXIV.2209.04197.

[13] KLIPPEL H., 2021, *Constitutive Equations for Simulation of Metal Cutting with Meshless Methods on GPU*, Doctoral Thesis, ETH Zurich, doi: 10.3929/ETHZ-B-000527668.

[14] BURGER W., 2013, *Principles of digital image processing*, New York, Springer.

[15] SIEGWART R., NOURBAKHSH I.R., SCARAMUZZA D., 2011, *Introduction to autonomous mobile robots*. Cambridge, Mass, MIT Press.

[16] MARAGOS P., SCHAFER R., 1987, Morphological filters-Part I: *Their_Set-Theoretic Analysis and Relations to Linear Shift-Invariant Filters*, IEEE Trans. Acoust. Speech Signal Process., 35/8, 1153–1169, doi: 10.1109/TASSP.1987.1165259.

[17] OTSU N., 1979, *A Threshold Selection Method from Gray-Level Histograms*, IEEE Transactions on System Man Cybernetics, 9, 62–66, doi:10.1109/TSMC.1979.4310076.

[18] BUDZYN G. RZEPKA J., 2020, *Review of Edge Detection Algorithms for Application in Miniature Dimension Measurement Modules*, J. Mach. Eng., 20/4, 74–85, doi: 10.36897/jme/130876.

[19] SUZUKI S. ABE K., 1985, *Topological Structural Analysis of Digitized Binary Images by Border Following*, Comput. Vis. Graph. Image Process., 30/1, 32–46, doi: 10.1016/0734-189X(85)90016-7.

[20] Chetverikov D., 2003, *A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves*, Computer Analysis of Images and Patterns, 2756, N. Petkov and M.A. Westenberg, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 746–753, doi: 10.1007/978-3-540-45179-2_91.

[21] ZOU Z., SHI Z., GUO Y., YE J., 2019, *Object Detection in 20 Years: A Survey*, arXiv:1905.05055, doi: 10.48550/ARXIV.1905.05055.

[22] COSCUN M., YILDIRIM O., UCAR A., DEMIR Y., 2017, *An Overview of Popular Deep Learning Methods*, Eur. J. Tech., 7/2, 165–176, 2017, doi: 10.23884/ejt.2017.7.2.11.

[23] LIU W., ANGUELOV D., Dumitru ERHAN D., et al., 2016, *SSD: Single Shot MultiBox Detector*, Computer Vision – ECCV, 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 21–37. doi: 10.1007/978-3-319-46448-0_2.

[24] SIMONYAN K. ZISSERMAN A., 2014, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv:1409.1556, doi: 10.48550/ARXIV.1409.1556.

[25] SMITH R., 2007, *An Overview of the Tesseract OCR Engine*', Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) 2, Curitiba, Parana, Brazil, 629–633. doi: 10.1109/ICDAR.2007.4376991.

[26] SAVITZKY A., GOLAY M.J.E., 1964, *Smoothing and Differentiation of Data by Simplified Least Squares Procedures*, Anal. Chem., 36/8, 1627–1639, doi: 10.1021/ac60214a047.

[27] BRONŠTEJN I.N., Ed., 2006, *Taschenbuch der Mathematik*, 6., Vollst. überarb. und erg. Aufl., Nachdr. Frankfurt am Main, Deutsch, ISBN 3817120168, 9783817120161.

[28] ABADI M., et al., 2016, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*, arXiv:1603.04467, doi: 10.48550/ARXIV.1603.04467.