

EUCLIDIAN DISTANCE TRANSFORMATION, MAIN AXIS ROTATION AND NOISY DILITATION SUPPORTED CROSS-SECTION CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS

Denk, Martin (1);
Rother, Klemens (2);
Höfer, Tobias (3);
Mehlstäubl, Jan (1);
Paetzold, Kristin (1)

1: Bundeswehr University Munich;
2: Munich University of Applied Sciences, Institute for Material and Building Research;
3: Munich University of Applied Sciences, Competence Center Image Processing (CCBV)

ABSTRACT

Polygon meshes and particularly triangulated meshes can be used to describe the shape of different types of geometry such as bicycles, bridges, or runways. In engineering, such polygon meshes can be supplied as finite element meshes, resulting from topology optimization or from laser scanning. Especially from topology optimization, frame-like polygon meshes with slender parts are typical and often have to be converted into a CAD (Computer-Aided Design) format, e.g., for further geometrical detailing or performing additional shape optimization. Especially for such frame-like geometries, CAD designs are constructed as beams with cross-sections and beam-lines, whereby the cross-section is extruded along the beam-lines or beam skeleton. One major task in the recognition of beams is the classification of the cross-section type such as I, U, or T, which is addressed in this article. Therefore, a dataset consisting of different cross-sections represented as binary images is created. Noisy dilatation, the distance transformation, and main axis rotation are applied to these images to increase the robustness and reduce the necessary amount of samples. The resulting images are applied to a convolutional neuronal network.

Keywords: Artificial intelligence, Machine learning, Computer Aided Design (CAD), Geometric Deep Learning, Cross-Sections

Contact:

Denk, Martin
Bundeswehr University Munich
Insitute for Technical Product Development
Germany
martin.denk@unibw.de

Cite this article: Denk, M., Rother, K., Höfer, T., Mehlstäubl, J., Paetzold, K. (2021) 'Euclidian Distance Transformation, Main Axis Rotation and Noisy Dilatation supported Cross-Section Classification with Convolutional Neural Networks', in *Proceedings of the International Conference on Engineering Design (ICED21)*, Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.140

1 INTRODUCTION

Various applications such as animation design, topology optimization, or 3D laser scanning apply discrete triangulated meshes to represent the surface of a 3D geometry. Triangulated surfaces describe the shape of the object by separating the boundary surface into discrete triangles. Engineering applications for manufacturing, optimization, or design modification require a parametric representation such as constructive solid geometry (CSG), free form surfaces of the geometry (Bénière et al., 2013; Vidal et al., 2014), or skeleton-based representations (Denk et al., 2020a). The shape of an object can be reconstructed with primitive surface types (Bénière et al., 2013; Vidal et al., 2014) or free form surfaces (Louhichi et al., 2015; Yoely et al., 2018), which can be a representation as a connected boundary representation (Brep) or the Boolean of different bodies as CSG-Objects. Primitive bodies such as cylinders are typically used for (CSG) (Bénière et al., 2013; Gauthier et al., 2017; Vidal et al., 2014), whereas free form surfaces are used for the reconstruction of organic geometries as boundary representations (Louhichi et al., 2015; Yoely et al., 2018). The following figure shows a skeleton-based, boundary-based, and CSG-based shape description.

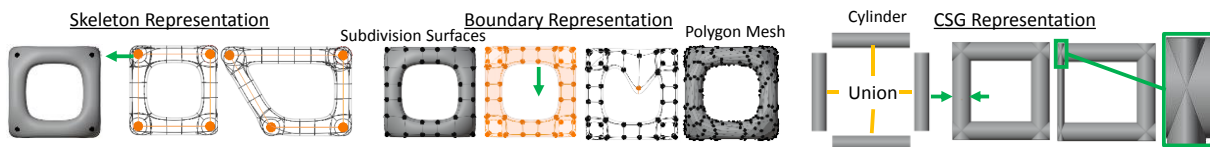


Figure 1: Different shape descriptors

The manual parametrization of these polygon meshes can be time-consuming and can lead to inaccurate approximations or altering the topology. Geometric reverse engineering attempts to redesign a parametric shape and topology of representations such as polygon meshes (Bénière et al., 2013; Gauthier et al., 2017; Vidal et al., 2014), or volumetric geometries (Bremicker et al., 1991). Most of the approaches perform a so-known mesh segmentation (Agathos et al., 2007), where the boundary of the geometry is split into several patches (Bénière et al., 2013; Gauthier et al., 2017; Vidal et al., 2014). Afterward, the surface of the geometry can be parametrized in surface types like primitive patches (Bénière et al., 2013; Gauthier et al., 2017; Vidal et al., 2014), or nonuniform rational B-spline (NURBS) surfaces (Ben Makhlouf et al., 2019; Louhichi et al., 2015). This segmentation only results in surface parametrization (Bénière et al., 2013; Gauthier et al., 2017; Vidal et al., 2014) so that the topology itself is not covered quite well. Therefore so-called curve (middle line) or surface skeleton (middle surface) can be used for part segmentation (Agathos et al., 2007; Feng et al., 2015; Reniers and Telea, 2008), which can lead to a more beamline representation (Bremicker et al., 1991; Nana et al., 2017; Stangl and Wartzack, 2015). These curve skeletons serve as a reasonable shape descriptor for tube-like organic geometries (Tagliasacchi et al., 2016). Such geometries can be derived in topology optimization for linear elastic static, heat transfer, or considering both (Denk et al., 2020b). These skeletonization methods in reverse engineering are often applied to organic and tube-like shapes leading to circular cross-sections (Denk et al., 2020a; Kresslein et al., 2018; Nana et al., 2017). Such skeletonization method can also be applied for non-circular shapes, which can result in various of different types of cross-sections. The following figure shows the concept of generating skeletons using (Lee et al., 1994), which can be used to cut along a polygon model or a 3D binary image. The cross-section of the binary image shows a sample of a cap-profile, while the polygon cross-sections can be expressed as a circular section. Such skeleton-based reverse engineering can be applied for deformed FE-meshes represented in (Louhichi et al., 2015), point cloud reconstruction (Kresslein et al., 2018), the determination of center lines in CT-Data (Computer Tomography) to reconstruct blood vessels (Hua Li and Yezzi, 2006) or vascular skeletons (Lidayová et al., 2016) or topology optimization results (Nana et al., 2017; Yin et al., 2020).

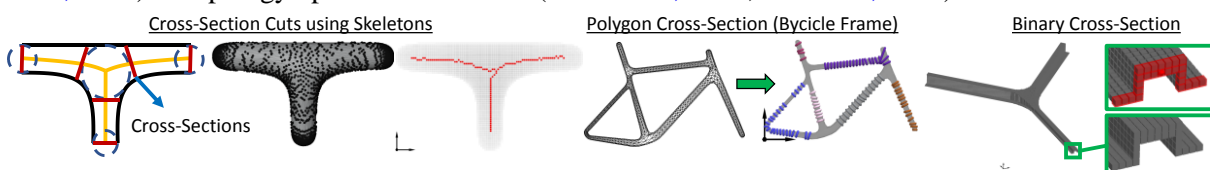


Figure 2: Cross-section in reverse engineering of polygon meshes and binary images

Additional to the reverse engineering task, such classification can also be used for object retrieval tasks (Yang et al., 2014) to find for example, specific parts in 2D technical drawings of various amount of data sets. For reverse engineering, the parametrization of that cross-section, such as the radius or the size parameter of the cap and the type of cross-section particular for junctions, is required. Based on that parametrization, the cross-section can be swept along the curve skeleton, and a smooth transition of different cross-section types at the junction can be ensured. Additionally, the skeleton and the cross-section can also be applied to perform a topology optimization using a finite element model based on beams (Changizi and Warn, 2020) or applying flexible shape adjustment (Denk et al., 2020a). To find an automatically cross-section parametrization, a classification task can be first applied so that afterward a regression of the shape parameters can be applied. This article addresses the automated classification of such cross-sections using geometric deep learning, which will be extended in further work applying a regression task. We apply that classification task on “Z”, “T”, “U”, “L”, “Cap”, “T”, “Rectangle” and “B-Spline” cross-sections. These classes are a subset of common cross-section types on which, in our case, organic or circular shapes are referred to as the class “B-Spline”, which can be extended by generating different classes and data samples.

Geometry deep learning is one of the recent challenges, especially for 3D objects (Ahmed et al., 2019; Bronstein et al., 2017; Cao et al., 2020). Due to the rapid development in deep learning, these models can perform classification and shape recognition tasks for different kinds of 3D geometries (Ahmed et al., 2019; Bronstein et al., 2017; Cao et al., 2020; Denk et al., 2019). Especially the use of convolutional neuronal networks (CNN) allows a high degree of parallelization. CNN layers are typically used for Euclidean data structures such as a 2D pixel or 3D voxel. While deep learning methods applied on Euclidean data structures such as 3D images (voxels) or 2D images (pixels) are largely established, current research fields summarized in (Ahmed et al., 2019; Bronstein et al., 2017; Cao et al., 2020) use CNN for non-Euclidean data such as graphs, polygonized surfaces, and point clouds. Properties such as shift-invariance and a global parametrization in the image can miss in non-Euclidian shapes (Bronstein et al., 2017), so that processing on non-Euclidian data is quite challenging (Bronstein et al., 2017; Denk et al., 2019). In our contribution, we address the topic of geometric learning for polygon cross-section. To avoid non-Euclidian shapes, the cross-sections are rasterized to enable global parametrization and shift-invariance. Instead of using common data argumentation to increase the number of data samples by rotating the images or scaling the image (Mikołajczyk and Grochowski, 2018), feature engineering is applied. The Euclidian distance transformation is used as preprocessing feature engineering due to its connection to skeletonization (Arcelli et al., 2011), part segmentation (Reniers and Telea, 2008), and object classification (Shen et al., 2014). Furthermore, the cross-sections are rotated into the main axis and scaled to its bounding box. These steps should reduce the necessary amount of data.

2 STATE OF THE ART

State of the art addresses mostly the rule-based estimation of cross-sections, the classification of objects in 2D binary images, and the data set for binary images. In most of the skeletonization methods, a skeleton is computed, which serves as a guideline for the cross-section cuts or segmentation. If a beamline representation is established, the corresponding cross-section can be determined. The authors of (Stangl and Wartzack, 2015) manually define elliptical and spherical cross-sections in topology optimization. In contrast to the manual selection, the authors of (Cuillère et al., 2018; Mayer and Wartzack, 2020; Nana et al., 2017; Yin et al., 2020) automatically use shape properties such as the radius of circular cross-sections (Cuillère et al., 2018; Nana et al., 2017) or the local radius (Mayer and Wartzack, 2020). The authors of (Lidayová et al., 2016) determine the cross-section of vascular bloodlines to remove nodes of a skeleton that does not suit the anatomy. In a recent publication, the authors of (Lidayová et al., 2017) use CNN based on the cross-sections on greyscale images to remove unnecessary nodes. In contrast to the primitive cross-section of (Lidayová et al., 2017; Nana et al., 2017; Stangl and Wartzack, 2015; Yin et al., 2020), the authors of (Tang and Chang, 2001) manually determine surface skinning with B-spline cross-sections to cover the shape of the topology optimization. Therefore, the whole smooth geometry is directly defined by the control grid. In our approach, we use similar to (Lidayová et al., 2017) a convolutional neuronal network to classify the type of cross-sections instead of the classification of node points. Additionally, we focus on the classification of several different kinds of cross-sections such as B-Spline and elliptical cross-sections, thin-walled cross-sections, and rectangle cross-sections as 2D binary images.

Several datasets for 2D binary images such as Kimia-216, Kimia-99, and MPEG-7 classification are available (Shekar and Pilar, 2014). These datasets address a huge variety of different objects such as birds, elephants, or bones. In our classification task, we build up a new dataset, especially for common cross-section shapes in mechanical engineering. For the binary image classification, the authors of (Shekar and Pilar, 2014) use local morphological and binary patterns for the classification of objects. The authors of (Shen et al., 2014) use the contours associated with the skeleton of a binary image and the authors of (Yang et al., 2014) uses a contour segment matching to measure the similarity of shapes for object retrieval, which are segmented by a skeletonization method. Due to the rapid development in machine learning, object retrieval tasks for 2D images can be solved using for example, neuronal networks, support vector machines, or random forest (Bansal et al., 2020).

The classification of images with deep learning is well known for 3D images (Gomez-Donoso et al., 2017; Maturana and Scherer, 2015; Qi et al., 2016; Sedaghat et al., 2017; Su et al., 2015) and 2D images (Gomez-Donoso et al., 2017; Lidayová et al., 2017). The authors of (Maturana and Scherer, 2015) determine a greyscale 3D image by the underlying point cloud, whereas the authors of (Lidayová et al., 2017) use the greyscale of computer tomography (CT) for 2D images. The authors of (Su et al., 2015) rotate the 3D geometry into multiple views to classify the 3D object with a 2D convolutional neuronal network. The authors of (Agh Atabay, 2016) use CNNs directly on binary images as a classification task of the MPEG-7 dataset. In contrast, we use the distance transformation of the binary image for the foreground and background as greyscale values, which also can be used for the skeletonization and segmentation of 3D images (Arcelli et al., 2011). Additionally, these binary cross-sections are rotated into the main axis based on the second moment of area to limit the number of rotations to four. For further object retrieval tasks and techniques for 2D images, we refer to the recent state of the art of (Bansal et al., 2020).

Deep Learning for image classification tasks uses a huge amount of 2D images of different resolutions and deep architectures with a huge number of learning parameters. Processing on 3D images is quite more challenging due the additional dimension, so that the amount of necessary data increases, and the resolution of the image itself is often restricted to the GPU memory size (Cao et al., 2020). Alternative representations provided with feature engineering presented in this work can reduce data variance and increase robustness without increasing the data samples.

3 SYNTHETIC DATA SET GENERATION

A synthetic data set is created for the classification with a CNN. There are several use-cases where polygon cross-section or cross-section inform of binary images needs to be classified. Therefore, we present a workflow covering both cases by rasterizing all polygon cross-sections. To reduce the number of configurations of the training and test samples, we a) reduce the number of image rotations, b) apply the distance transformation, and c) use a Euclidian data structure. The following figure covers the different steps in data generation.

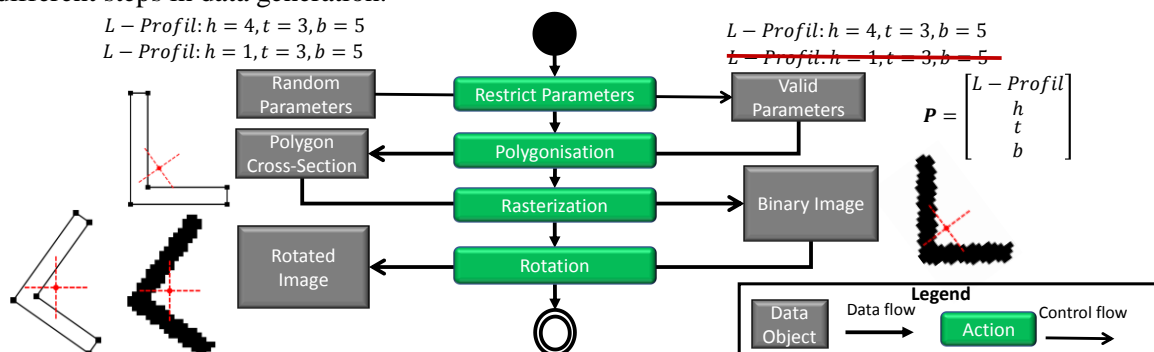


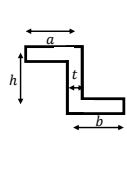
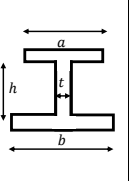
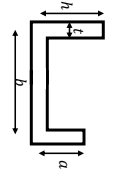
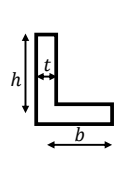
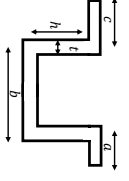
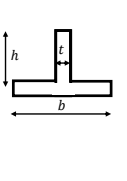
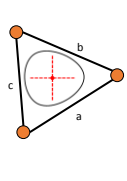
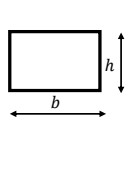
Figure 3: Generation of the data set

First, a set of random parameters are calculated. Second, the parameter combination has to be restricted so that only valid profiles are possible (See Table 1). Third, the parameters are normalized to reduce the design space. Forth, this parametric cross-section is converted to a polygon cross-section. Fifth, the cross-section is rasterized. Sixth, the image is rotated into the main axis. As our result, binary images and the corresponding label are stored for training.

3.1 Restriction of the cross-section parameter

For each class (8 classes), 500 samples are generated. The following table covers the restriction and the parameters of common cross-sections. The limits are defined, so that the main shape is preserved.

Table 1: Restrictions for the selected cross-sections

Section	Z	I	U	L	Cap	T	B-Spline	Rec
Limits	$\frac{t}{2} < a$ $\frac{t}{2} < b$ $t < h$	$t < a$ $t < b$ $t < h$	$\frac{t}{2} < a$ $\frac{t}{2} < h$ $t < b$	$\frac{t}{2} < b$ $\frac{t}{2} < h$	$\frac{t}{2} < a$ $\frac{t}{2} < c$ $t < h$ $t < b$	$t < b$ $\frac{t}{2} < h$	$c + b > a$ $a + c > b$ $a + b > c$	-
Shape								

3.2 Image rotation using the moment of areas

The rotation of the image into the main axis can be applied by calculating the moment of area of the binary image or the polygon cross-section. So the rotation can be applied to the polygon cross-section before rasterization by calculating the moment of areas using the pixel representation. The following figure covers the main shape parameter and coordinate systems calculating the moment of areas.

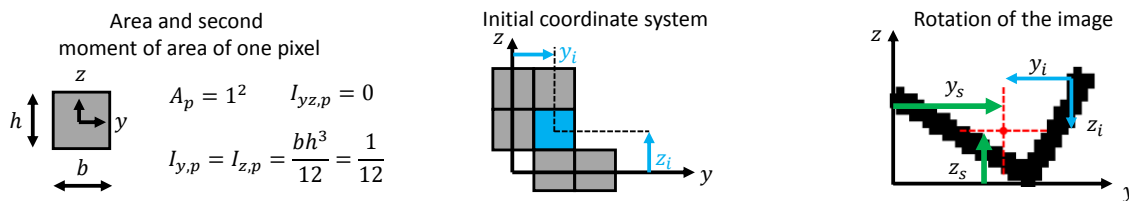


Figure 4: Moment of area calculation for binary images

The area A can be calculated by summing up all black pixels $i \in B$. The first moment of area S_y and S_z can be calculated using the distance of the coordinate system y and z to each pixel in B with

$$S_y = \int_A z dA = \sum_{i \in B} z_i 1^2 \text{ and } z_s = \frac{S_y}{A}; \quad S_z = \int_A y dA = \sum_{i \in B} y_i 1^2 \text{ and } y_s = \frac{S_z}{A} \quad (1)$$

where y_s and z_s represents the center of the area according to the coordinate system y, z . The second moment of area can be calculated in the shifted reference coordinate system z_s and y_s with

$$I_{\bar{y}} = \sum_{i \in B} I_{y,p} + \bar{z}_{s,i}^2 A_p = \sum_{i \in B} \left(\frac{1}{12} + (z_s - z_i)^2 \right) \quad (2)$$

$$I_{\bar{z}} = \sum_{i \in B} I_{z,p} + \bar{y}_{s,i}^2 A_p = \sum_{i \in B} \frac{1}{12} + (y_s - y_i)^2 \quad (3)$$

$$I_{\bar{yz}} = \sum_{i \in B} I_{yz,p} - \bar{z}_{s,i} \bar{y}_{s,i} A_p = - \sum_{i \in B} (y_s - y_i)(z_s - z_i) \quad (4)$$

where $A_p, I_{y,p}, I_{z,p}$ represents the moment of area for a local pixel and $\bar{z}_{s,i}, \bar{y}_{s,i}$ the distance of the center of mass to the corresponding pixel i . Based on these moments of areas, the image can be rotated into the main axis by calculating the angle ϕ with

$$\tan(2\phi) = \frac{I_{yz}}{I_{\bar{y}} - I_{\bar{z}}}. \quad (6)$$

3.3 Feature-Engineering: Distance Transformation CNN

Feature-Engineering can be used to reduce the number of necessary training samples. Therefore, based on the concept of distance transformation supported skeletonization, we perform a distance transformation on the background and the foreground. First, the binary image is padded so that no black pixel touches the image boundary. Based on that, a distance transformation is performed on the cross-section and the background. The distance field serves as the input for the convolutional neuronal network. The distance values of the background are multiplied with minus one so that the foreground is represented with positive distance values and the background with negative distance values.

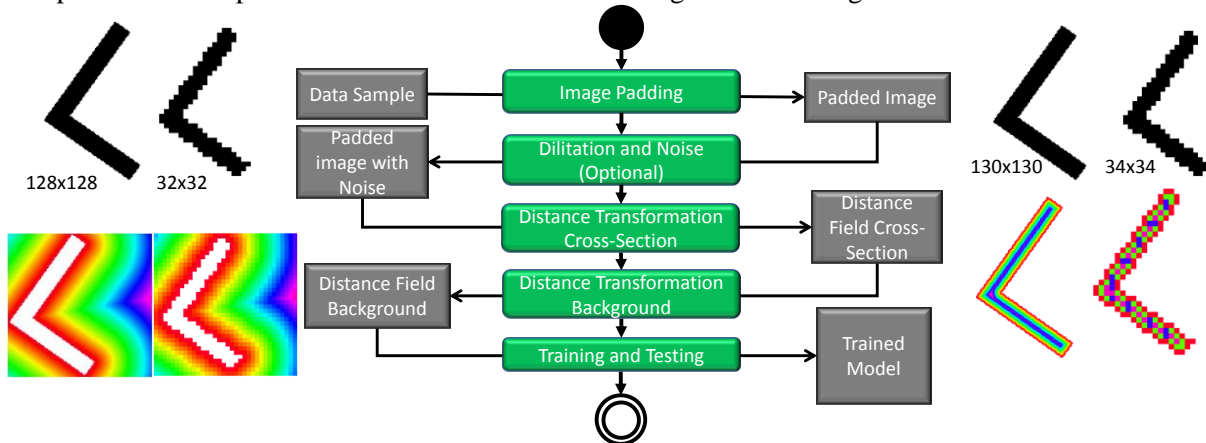


Figure 5: Distance transformation and noise

Additionally, as an optional criterion, noise on the boundary of the cross-section can be added using randomly applied dilatation. The following figure shows two different results of the rasterization.

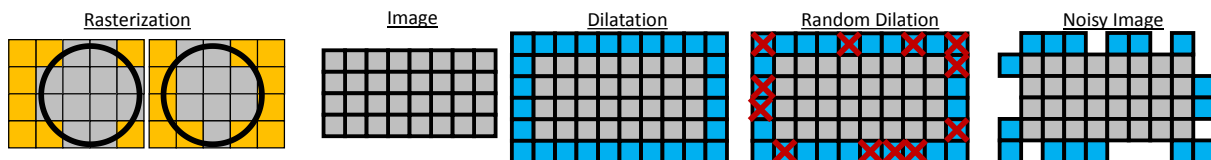


Figure 6: Rasterization using different grid overlays and noisy dilatation.

To increase the robustness of different rasterization or cuts through a volumetric model presented in Figure 2, noisy dilatation is applied. The image is randomly dilatated using a 1x1 structure element.

4 CROSS-SECTION CLASSIFICATION AND REGRESSION

Two common (from scratch), not specialized convolutional neuronal networks are selected as the architectures. The following figure shows the architecture for the classification and the corresponding activation functions.

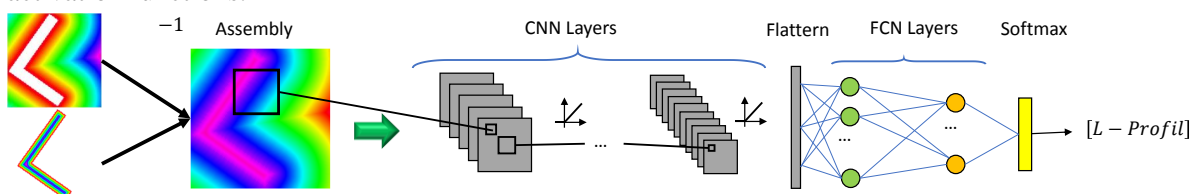


Figure 7: Architecture for the classification

To increase the robustness, we use bias, activity and kernel regularization, dropout, and batch normalization. For the classification, the categorical cross-entropy for the M cross-section categories for n samples are selected with the loss-function

$$L = - \sum_{i=1}^n \sum_{j=1}^M \hat{c}_{i,j} \log(c_{i,j}) \quad (8)$$

where $c_{i,j}$ is the predicted probability and $\hat{c}_{i,j}$ the true probability for class j for the sample i .

5 EXPERIMENT

The proposed method is applied to several different use cases with different cross-section shapes. For classification accuracy, we use the median value of ten runs. The following figure shows the accuracy and the loss function during the training for a test and train case. As first architecture, a 3x3 kernel size and the three convolutional layers with 32, 64, 64 filters followed by two fully connected layers with the size of 64 and 8 is selected. A mean accuracy of 98% for the original and 97% for the noise images by applying ten pieces of training was achieved.

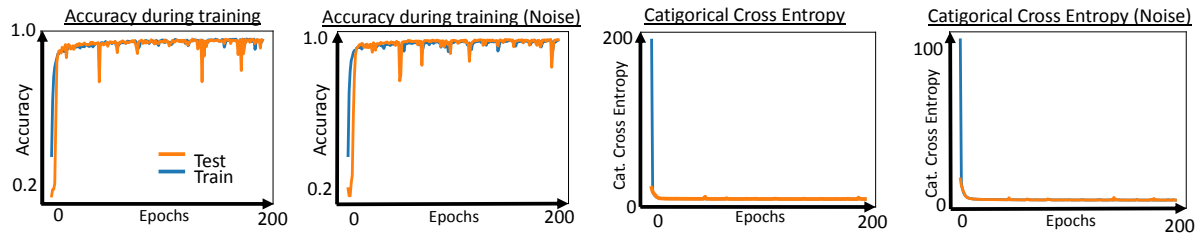


Figure 8: Training of the classification of cross-sections of one sample run

The following table covers different samples (noise and no noise), on which the first three cases are falsely predicted, and the last three are correctly predicted.

Table 2: Classification samples

Noise Sample												
Sample												
y	L, L	U, U	T, T	I, I	Z, I	I, I	T	T	B	B	Cap	Cap
\hat{y}	L, L	U, U	T, T	I, I	Z, I	I, I	T	T	B	B	Cap	Cap

The first two examples visually do not match with the cross-section, but the predicted class matches visually (false positive). Such cases need to be corrected in the dataset. The third sample shows an example where the noise image matches the target I, and the clean image misclassifies the image as a Z. All misclassifications are covered in the following confusing matrix (No noise).

Table 3: Confusion matrix using distance transformation and rotation into the main axis

$\hat{y} \setminus y$	Rec	L	CAP	I	U	T	Z	B-Spline
Rec	122	0	0	0	0	0	0	0
L	0	119	0	0	0	1	0	2
CAP	0	0	116	0	0	0	2	0
I	1	0	0	128	0	1	1	0
U	0	7**	0	0	111	0	0	0
T	3*	0	0	0	0	114	0	0
Z	0	0	0	0	1	0	126	0
B-Spline	0	0	0	0	0	0	0	147
Precision	97	94	100	100	99	98	98	99
Recall	100	98	98	98	94	97	99	100

To compare different feature engineering strategies, we lower the number of learnable parameters by using an architecture with a 3x3 kernel size, three convolutional layers with 2, 4, 4 filters two fully connected layers with the size of 4 and 8. This also should lead visually to higher impacts changing the feature engineering strategy. For each strategy, ten samples are generated, which are represented in

the following figure as boxplots. Each strategy is marked with a color, and the best strategy (median) for the different resolution are marked with **.

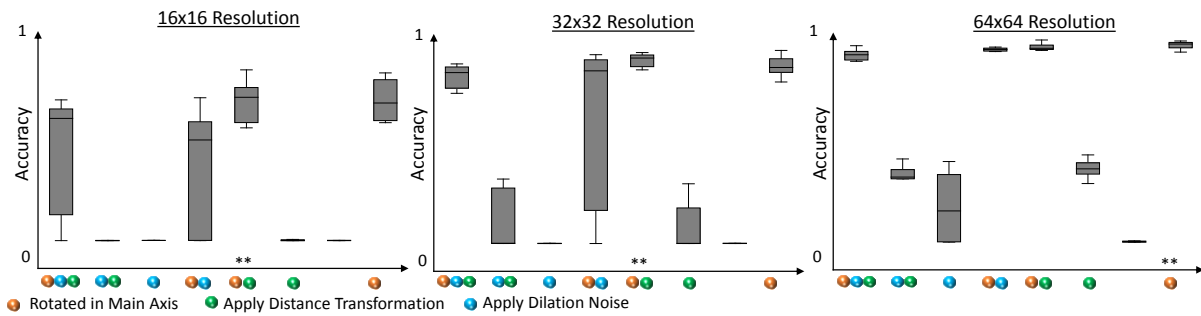


Figure 9: Comparison of feature engineering strategies

6 DISCUSSION

The provided classification leads to reasonable results. The accuracy of 98% shows a robust classification of the chosen categories so that directly applying the regression after classification can be achieved. By adding dilation noise, the accuracy is still 97%, so that unclear cross-section can also be covered. The misclassification is often bounded to a too-small resolution. The following figure shows misleading samples, whereas the polygon section itself hardly represents a U or an I. Due to the rasterization (32x32), too small parts of the U and the I vanish, which can be corrected for these cases using a higher resolution (128x128).

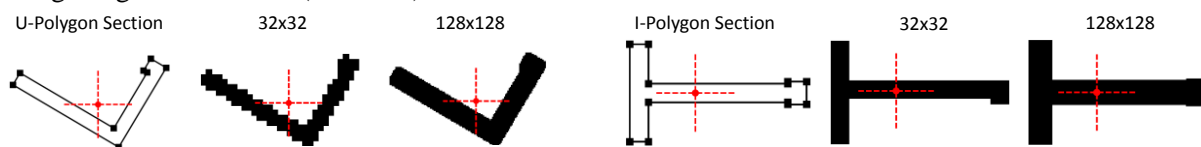


Figure 10: Sample of misleading cross-sections

For further research purposes, the noise and the rasterization resolution should be considered in Table 1 so that the main shape of the cross-section is preserved. Therefore, the selected restriction of reasonable parameters for cross-sections should also contain the noise. Additionally to the classification itself, the impact of misclassification using the confusion matrix shows that certain types of cross-sections can be recognized quite robust while cross-sections such as L or U profiles are sometimes classed as each other. This can occur if the parameter h or a of U are too small, so that U is shaped similar to L. Wrong classification of cross-sections can lead to inaccurate reverse engineering. So, in this case, at least L-Profiles can be checked manually to the local precision of 94% in Table 3. Furthermore, the mechanical cross-section properties such as the stiffness can be altered, which impacts the magnitude of stresses and strains. In further research, miss-classification should be evaluated by using rule-based assumptions such as moments of areas comparison of the image and the regressed cross-section.

Applying feature engineering strategies can increase the accuracy compared in Figure 9. The most impact in increasing the accuracy can be achieved by choosing the rotation in the main axis of the images. Smaller impacts occur by choosing the distance transformation, particularly if noise is applied. The distance transformation leads to a more robust representation if noise is applied.

7 SUMMARY

The classification of cross-section types is required in reverse engineering of the beam-like shapes. In this article, a convolutional neuronal network for the classification of different cross-sections is shown, which can be downloaded at (Denk, 2021). A data set using polygon and image cross-section is created for eight common cross-section types such as I, U, or T sections. The robustness is increased by applying randomized dilatation and feature engineering such as the rotation into the main axis and applying the Euclidian distance transformation to the image cross-sections. The Euclidian distance transformation is applied at the background and the foreground of the image so that the decision of the cross-section type is determined by a local feature size and not by a sharp black-white pattern on the cross-section boundary. The categorical cross-entropy is chosen as the loss-function. The neuronal

network architecture consists of common layers such as convolutional layers or dense layers. The proposed method shows a median accuracy of 98% of ten training steps for the classification.

ACKNOWLEDGMENTS

This work is part of the project ANGORA, which is supported by the “Federal Ministry for Economic Affairs and Energy “with the funding indicator ZF4428401BZ7 in the context of the research network IraSME and ZIM. Responsibility for the content of this publication lies with the authors.

REFERENCES

- Agathos, A., Pratikakis, I., Perantonis, S., Sapidis, N., Azariadis, P., 2007. 3D Mesh Segmentation Methodologies for CAD applications. *Comput.-Aided Des. Appl.* 4, 827–841. <https://doi.org/10.1080/16864360.2007.10738515>
- Agh Atabay, H., 2016. Binary shape classification using Convolutional Neural Networks. *IIOAB J.* 7, 332–336.
- Ahmed, E., Saint, A., Shabayek, A.E.R., Cherenkova, K., Das, R., Gusev, G., Aouada, D., Ottersten, B., 2019. A survey on Deep Learning Advances on Different 3D Data Representations. *ArXiv180801462 Cs*.
- Arcelli, C., Baja, G.S. di, Serino, L., 2011. Distance-Driven Skeletonization in Voxel Images. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 709–720. <https://doi.org/10.1109/TPAMI.2010.140>
- Bansal, M., Kumar, Munish, Kumar, Manish, 2020. 2D Object Recognition Techniques: State-of-the-Art Work. *Arch. Comput. Methods Eng.* <https://doi.org/10.1007/s11831-020-09409-1>
- Ben Makhlof, A., Louhichi, B., Mahjoub, M.A., Deneux, D., 2019. Reconstruction of a CAD model from the deformed mesh using B-spline surfaces. *Int. J. Comput. Integr. Manuf.* 32, 669–681. <https://doi.org/10.1080/0951192X.2019.1599442>
- Bénière, R., Subsol, G., Gesquière, G., Le Breton, F., Puech, W., 2013. A comprehensive process of reverse engineering from 3D meshes to CAD models. *Comput.-Aided Des.* 45, 1382–1393. <https://doi.org/10.1016/j.cad.2013.06.004>
- Bremicker, M., Chirehdast, M., Kikuchi, N., Papalambros, P.Y., 1991. Integrated Topology and Shape Optimization in Structural Design*. *Mech. Struct. Mach.* 19, 551–587. <https://doi.org/10.1080/08905459108905156>
- Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P., 2017. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Process. Mag.* 34, 18–42. <https://doi.org/10.1109/MSP.2017.2693418>
- Cao, W., Yan, Z., He, Zhiquan, He, Zhihai, 2020. A Comprehensive Survey on Geometric Deep Learning. *IEEE Access* 8, 35929–35949. <https://doi.org/10.1109/ACCESS.2020.2975067>
- Changizi, N., Warn, G.P., 2020. Topology optimization of structural systems based on a nonlinear beam finite element model. *Struct. Multidiscip. Optim.* <https://doi.org/10.1007/s00158-020-02636-x>
- Cuillière, J.-C., François, V., Nana, A., 2018. Automatic construction of structural CAD models from 3D topology optimization. *Comput.-Aided Des. Appl.* 15, 107–121. <https://doi.org/10.1080/16864360.2017.1353726>
- Denk, M., 2021. Cross-Section Dataset. URL <https://github.com/Foxelmanian/CrossSectionClassification>
- Denk, M., Paetzold, K., Rother, K., 2019. Feature line detection of noisy triangulated CSGbased objects using deep learning, in: *Proceedings of the 30th Symposium Design for X (DFX 2019)*, DfX. Presented at the DfX Symposium 2019, The Design Society, Jesteburg, Germany, pp. 239–250. <https://doi.org/10.35199/dfx2019.21>
- Denk, M., Rother, K., Paetzold, K., 2020a. Fully Automated Subdivision Surface Parametrization for Topology Optimized Structures and Frame Structures using Euclidean Distance Transformation and Homotopic Thinning, in: *Proceedings of the Munich Symposium on Lightweight Design 2020*. Springer Nature, Munich, Germany. <https://doi.org/10.1007/978-3-662-63143-0>
- Denk, M., Rother, K., Paetzold, K., 2020b. Multi-Objective Topology Optimization of Heat Conduction and Linear Elastostatic using Weighted Global Criteria Method, in: *Proceedings of the 31st Symposium Design for X (DFX2020)*, DfX. Presented at the DfX Symposium 2020, The Design Society, Bamberg, pp. 91–100. <https://doi.org/10.35199/dfx2020.10>
- Feng, C., Jalba, A.C., Telea, A.C., 2015. Part-Based Segmentation by Skeleton Cut Space Analysis, in: Benediktsson, J.A., Chanussot, J., Najman, L., Talbot, H. (Eds.), *Mathematical Morphology and Its Applications to Signal and Image Processing*, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 607–618. https://doi.org/10.1007/978-3-319-18720-4_51
- Gauthier, S., Puech, W., Bénière, R., Subsol, G., 2017. Analysis of digitized 3D mesh curvature histograms for reverse engineering. *Comput. Ind.* 92–93, 67–83. <https://doi.org/10.1016/j.compind.2017.06.008>
- Gomez-Donoso, F., Garcia-Garcia, A., Garcia-Rodriguez, J., Orts-Escolano, S., Cazorla, M., 2017. LonchaNet: A sliced-based CNN architecture for real-time 3D object recognition, in: *2017 International Joint Conference on Neural Networks (IJCNN)*. Presented at the 2017 International Joint Conference on Neural Networks (IJCNN), pp. 412–418. <https://doi.org/10.1109/IJCNN.2017.7965883>

- Hua Li, Yezzi, A., 2006. Vessels as 4D Curves: Global Minimal 4D Paths to Extract 3D Tubular Surfaces, in: 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06). Presented at the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), pp. 82–82. <https://doi.org/10.1109/CVPRW.2006.210>
- Kresslein, J., Haghighi, P., Park, J., Ramnath, S., Sutradhar, A., Shah, J.J., 2018. Automated cross-sectional shape recovery of 3D branching structures from point cloud. *J. Comput. Des. Eng.* 5, 368–378. <https://doi.org/10.1016/j.jcde.2017.11.010>
- Lee, T.C., Kashyap, R.L., Chu, C.N., 1994. Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms. *CVGIP Graph. Models Image Process.* 56, 462–478. <https://doi.org/10.1006/cgip.1994.1042>
- Lidayová, K., Frimmel, H., Wang, C., Bengtsson, E., Smedby, Ö., 2016. Fast vascular skeleton extraction algorithm. *Pattern Recognit. Lett., Special Issue on Skeletonization and its Application* 76, 67–75. <https://doi.org/10.1016/j.patrec.2015.06.024>
- Lidayová, K., Gupta, A., Frimmel, H., Sintorn, I.-M., Bengtsson, E., Smedby, Ö., 2017. Classification of Cross-sections for Vascular Skeleton Extraction Using Convolutional Neural Networks, in: Valdés Hernández, M., González-Castro, V. (Eds.), *Medical Image Understanding and Analysis*. Springer International Publishing, Cham, pp. 182–194.
- Louhichi, B., Abenhaim, G.N., Tahan, A.S., 2015. CAD/CAE integration: updating the CAD model after a FEM analysis. *Int. J. Adv. Manuf. Technol.* 76, 391–400. <https://doi.org/10.1007/s00170-014-6248-y>
- Maturana, D., Scherer, S., 2015. VoxNet: A 3D Convolutional Neural Network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922–928. <https://doi.org/10.1109/IROS.2015.7353481>
- Mayer, J., Wartzack, S., 2020. Ermittlung eines Skelettierungsverfahrens zur Konvertierung von Topologieoptimierungsergebnissen, in: *Proceedings of the 31st Symposium Design for X (DFX2020)*. Presented at the Symposium Design for X 2020, Bamberg, pp. 111–120. <https://doi.org/10.35199/dfx2020.12>
- Mikołajczyk, A., Grochowski, M., 2018. Data augmentation for improving deep learning in image classification problem, in: 2018 International Interdisciplinary PhD Workshop (IIPHDW). Presented at the 2018 International Interdisciplinary PhD Workshop (IIPHDW), pp. 117–122. <https://doi.org/10.1109/IIPHDW.2018.8388338>
- Nana, A., Cuillière, J.-C., Francois, V., 2017. Automatic reconstruction of beam structures from 3D topology optimization results. *Comput. Struct.* 189, 62–82. <https://doi.org/10.1016/j.compstruc.2017.04.018>
- Qi, C.R., Su, H., Niessner, M., Dai, A., Yan, M., Guibas, L.J., 2016. Volumetric and Multi-View CNNs for Object Classification on 3D Data. *ArXiv160403265 Cs*.
- Reniers, D., Telea, A., 2008. Part-type Segmentation of Articulated Voxel-Shapes using the Junction Rule. *Comput. Graph. Forum* 27, 1845–1852. <https://doi.org/10.1111/j.1467-8659.2008.01331.x>
- Sedaghat, N., Zolfaghari, M., Amiri, E., Brox, T., 2017. Orientation-boosted Voxel Nets for 3D Object Recognition. *ArXiv160403351 Cs*.
- Shekar, B.H., Pilar, B., 2014. Shape Representation and Classification through Pattern Spectrum and Local Binary Pattern – A Decision Level Fusion Approach, in: 2014 Fifth International Conference on Signal and Image Processing. Presented at the 2014 Fifth International Conference on Signal and Image Processing, pp. 218–224. <https://doi.org/10.1109/ICSIP.2014.41>
- Shen, W., Jiang, Y., Gao, W., Zeng, D., Wang, X., 2014. Shape Recognition by Bag of Skeleton-associated Contour Parts. *ArXiv160506417 Cs* 483, 391–400. https://doi.org/10.1007/978-3-662-45646-0_40
- Stangl, T., Wartzack, S., 2015. Feature based interpretation and reconstruction of structural topology optimization results, in: Weber, M., C.; Husung, S.; Cascini, G.; Cantamessa, M.; Marjanovic, D.; Bordegoni (Ed.), *Proceedings of the 20th International Conference on Engineering Design (ICED15)*. Design Society, p. Vol. 6, 235–245.
- Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G., 2015. Multi-view convolutional neural networks for 3d shape recognition, in: *Proc. ICCV*.
- Tagliasacchi, A., Delame, T., Spagnuolo, M., Amenta, N., Telea, A., 2016. 3D Skeletons: A State-of-the-Art Report. *Comput. Graph. Forum* 35, 573–597. <https://doi.org/10.1111/cgf.12865>
- Tang, P.-S., Chang, K.-H., 2001. Integration of topology and shape optimization for design of structural components. *Struct. Multidiscip. Optim.* 22, 65–82. <https://doi.org/10.1007/PL00013282>
- Vidal, V., Wolf, C., Dupont, F., 2014. Mechanical Mesh Segmentation and Global 3D Shape Extraction.
- Yang, C., Tiebe, O., Pietsch, P., Feinen, C., Kelter, U., Grzegorzek, M., 2014. Shape-based object retrieval by contour segment matching, in: 2014 IEEE International Conference on Image Processing (ICIP). Presented at the 2014 IEEE International Conference on Image Processing (ICIP), pp. 2202–2206. <https://doi.org/10.1109/ICIP.2014.7025446>
- Yin, G., Xiao, X., Cirak, F., 2020. Topologically robust CAD model generation for structural optimisation. *Comput. Methods Appl. Mech. Eng.* 369, 113102. <https://doi.org/10.1016/j.cma.2020.113102>
- Yoely, Y.M., Amir, O., Hanniel, I., 2018. Topology and shape optimization with explicit geometric constraints using a spline-based representation and a fixed grid. *Procedia Manuf.*, 15th Global Conference on Sustainable Manufacturing 21, 189–196. <https://doi.org/10.1016/j.promfg.2018.02.110>