INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.152

A New Projection Based Method for the Classification of Mechanical Components Using Convolutional Neural Networks

S. Bickel ✉, B. Schleich and S. Wartzack

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
✉ bickel@mfk.fau.de

Abstract

Digital engineering is increasingly established in the industrial routine. Especially the application of machine learning on geometry data is a growing research issue. Driven by this, the paper presents a new method for the classification of mechanical components, which utilizes the projection of points onto a spherical detector surfaces to transfer the geometries into matrices. These matrices are then classified using deep learning networks. Different types of projection are examined, as are several deep learning models. Finally, a benchmark dataset is used to demonstrate the competitiveness.

Keywords: data-driven design, artificial intelligence (AI), data mining, deep learning, part classification

1. Introduction

Driven by the continuous improvement of computation performance as well as huge advances in the field of machine learning methods, digital engineering and data-driven design tools reform established product development processes (Bickel et al., 2019). In this context, particularly the automatic classification of mechanical engineering parts throughout the design process receives increasing attention. Various application options are conceivable, starting from a simple component search in created assemblies over the employment in FE-simulation. Moreover, with the aim to simplify the pre-processing of finite element simulations, Kestel et al. (2019) use a combination of text mining and ontologies to independently extract FE-specific knowledge from databases and FE simulation reports, which can be merged with part detection for a better automation of the preprocessing stage. A first concept proposes Spruegel and Wartzack (2015) to this objective, with an automatic part recognition in FE simulations. Even in the reverse engineering process, the automatic detection of part classes enhances the quality of the reconstructed part model (Qie et al., 2021).
In this paper, a new method is presented which can classify parts from the field of mechanical engineering by applying deep learning in combination with the projection method. The performance of the new method is first compared with initial approach, followed by a comparison with established algorithms from computer vision via a benchmark dataset.

2. State of the art in the part classification

The idea of classification is to assign a specific predefined class label to a given input. Consequently, it is a supervised machine learning method, which has a huge application range, from classifying images, time-sequences or text data. This working principle is also adapted for 3D geometry parts.
Many methods and new algorithms have been developed to achieve this objective. In particular, the discovery of Convolutional Neural Networks (CNNs) had an enormous impact on the development of

geometry classification algorithms. CNNs machine-imitate human vision by arranging multiple filters and layers one after the other to transform the input image into a large feature map. Afterwards fully connected neural networks or other classifiers, such as support vector machines or decision trees, can be applied for the classification. Almost all current methods for part classification have CNNs as the foundation for detection. The different methods can be distinguished by their geometry input format. A short overview is presented in Figure 1.
The shown formats are all derived from 3D surface meshes. Compared to other geometry formats (e. g. STEP or IGES) they require less memory and can be easily manipulated while representing the geometry accurately enough without providing unnecessary information.

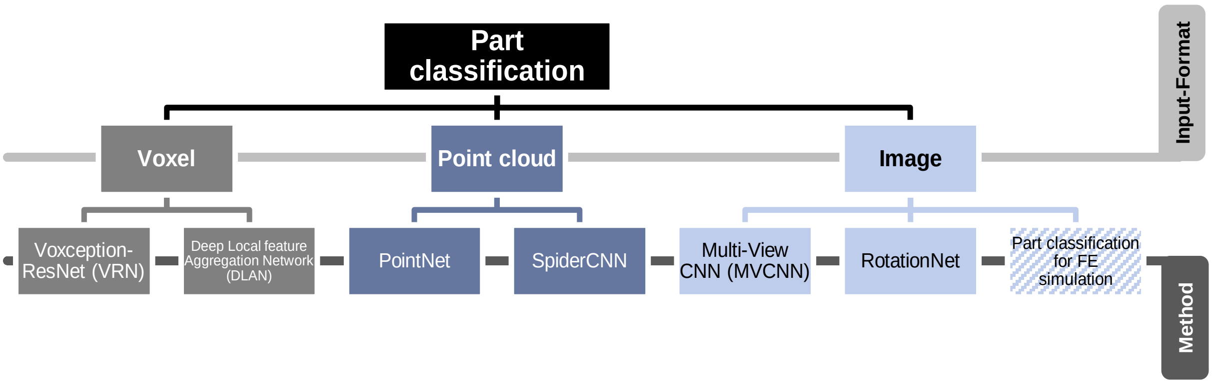

Figure 1. Overview of different part classification methods sorted by their input format

The first geometry representation is called voxel and its name is composed of the acronym "vox" for volume and "el" for element. It is a spatial 3D grid that represents geometry in a defined area. A point of this grid is called a voxel, comparable to a pixel within an image. This format offers the advantage of providing geometry uniformly in the grid representation. Several methods have taken this into consideration and are therefore built on the voxel representation. Brock et al. (2016) developed a new approach based on the VoxNet architecture, which was extended by inception-style modules and a relatively shallow network was chosen. This new procedure was called Voxception-ResNet (VRN), due to the integration of the inception modules. The approach of Furuya and Ohbuchi (2016) combines the description of 3D objects via local features with rotation normalized grids into a new method called Deep Local feature Aggregation Network (DLAN).
In contrast to voxels, point clouds are irregularly and disorderly distributed in space, which makes the application of typical CNNs impossible. This problem has been solved by PointNet, which is a deep learning architecture invented by Qi et al (2016) that extracts features from point clouds. The described method is not only able to recognise components, but also to segment them. The part detection is based on two steps, the input transformation and feature generation, as well as the classification of global features. For input transformation, a combination of transformation networks and matrix multipliers are utilised, which are transferred to a global feature vector via max pooling. This vector is then classified by a multi-layer perceptron (MLP) and the corresponding label is determined. The described procedure was improved further by Qi et al (2017) to PointNet++, with the aim of recognising more fine-grained patterns.
An alternative approach for point clouds has been developed by Xu et al. (2018), the SpiderCNN. This architecture consists of so-called SpiderConv units, which extend the convolutional operations to irregular point sets by parametrizing a family of convolutional filters. These filters are defined as the product of a simple step function for detecting local features and a Taylor polynomial that approximates the weights.
Besides using the two previously described geometry formats as input for classification, images can also be selected. The existing methods generally render several images from different views of the part, which are then classified using CNNs. A well-known method from this category is the Multi-View CNN (MVCNN), which was stated by Su et al (2015). In this method, rendered views from the

https://doi.org/10.1017/pds.2022.152 Published online by Cambridge University Press

3D object are passed through the first CNN layer to extract view-based features. In the subsequent view pooling layer, the previously extracted features from all views are combined and directed through the second CNN, which is responsible for part classification.

Another image-based approach relies not only on component but also on orientation prediction, the RotationNet developed by Kanezaki et al. (2018). Unsupervised learning helps finding the three best object poses out of twelve fixed camera perspectives, to optimise the classification accuracy. In contrast to the previous techniques, methods based on images are not able to segment 3D geometries and have therefore fewer general functions. Especially the ModelNet (Wu et al., 2015) benchmark results of the RotationNet show that the usage of images enables a high accuracy in part detection.

A special case in this collection is the method of Spruegel and Wartzack (2015), which is limited to the identification of FE simulation parts. This method has been highlighted with a hatching in Figure 1, because in this method an image is not transferred into a CNN, but into a normal neural network. For this purpose, the geometry was projected into a matrix which was then transformed into a vector. This serves as input for a neural network.

The collection of algorithms shows that image-based methods can achieve high accuracy and are partly already used in the product development environment. For this reason, the method presented below also utilises images as an input for the classification algorithm.

3. Projection method for classifying mechanical engineering components

The idea of this new approach is to project a point cloud onto a sphere and then transform it into a matrix, which is the input for the CNN. The origin of this projection method goes back to Spruegel and Wartzack (2015). The stated method has the problem that it was only applied to very specific classes, e. g. screw ISO 4762 – M6 x 20. Therefore, no transferability to new components could be verified or tested. Furthermore, the approach used a volume mesh as input, which severely limits the applicability of the method, since the reconstruction of surface meshes to solids can be costly and complicated.

Consequently, the goal of the new approach is to enable a broader class classification, like screw or alignment pin, by learning the features of the respective class. In addition, surface meshes should be taken as input to ensure better applicability and comparability. A comparison of the new method with the old one on a given data set is presented in Section 4.1.

While the procedure was significantly revised, changed and optimized in almost all partial aspects, the basic idea has been preserved by converting the geometry into a point cloud and then in a matrix.

Therefore, a 3D mesh is necessary for the execution, which serves as a starting point for the projection. In principle, point clouds can also be used as input but with the restriction of a fixed point cloud number for all parts to achieve a meaningful projection.

Unlike the established image-based methods, it does not use multiple views of the geometry, but an already processed form. The detailed structure of the new classification procedure is explained in detail in the following chapter.

3.1. General overview

In Figure 2 the complete overview of this method is shown. The process can be subdivided in two main steps, the point cloud generation and projection method, followed by the building and training of the classifier. The procedure starts with the input of a 3D geometry, which is first processed and converted and then handed over to the trained model. This classifies the matrix and provides the appropriate class for the geometry in question. Every main step consists of several sub-steps, each of which is explained in the next chapters.

https://doi.org/10.1017/pds.2022.152 Published online by Cambridge University Press

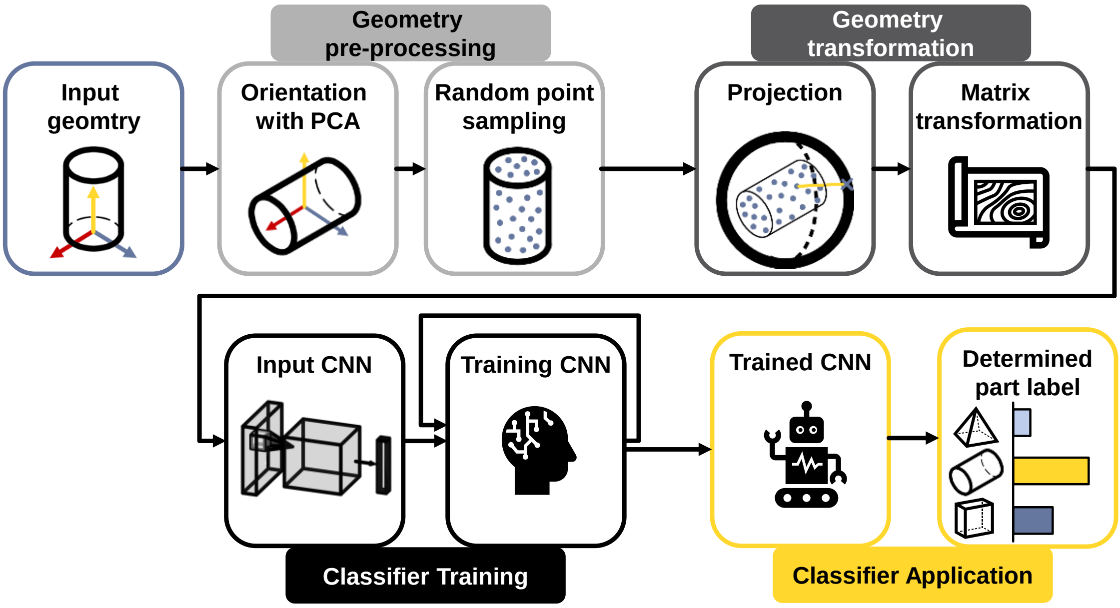

Figure 2. Overview of the general procedure of the classification

3.2. Geometry pre-processing and projection method

The first step is to read the geometry file and extract the faces and vertices. This information is then used to align the component. For this, a bounding box is generated around the part. The calculated corner points serve as input for a principal component analysis (PCA), which defines the alignment of the individual geometries.

Next, the aligned component is converted into a point cloud by randomly generating a defined number of points on the surface of the respective component. For this series of experiments, a fixed point cloud number of 25,000 per each component was selected. The number is defined by the combination of projection resolution and a feasible computing time. The random distribution on the mesh could be improved by further optimization, but due to high component numbers, the focus was set on a faster method to generate the point cloud. The projection method utilizes the information of the individual points and projects them onto a detector sphere. This globe is divided into several areas, which are called pixels. The exact structure of the sphere can be examined in the publication of Spruegel et al. (2021).

Finally, the projected points per pixel are counted. Similar to a map, the generated sphere is unfolded, and the result are considered as an input matrix for the machine learning process. Hence, the projection has two main primary parameters: the type of projection and the resolution of the matrix.

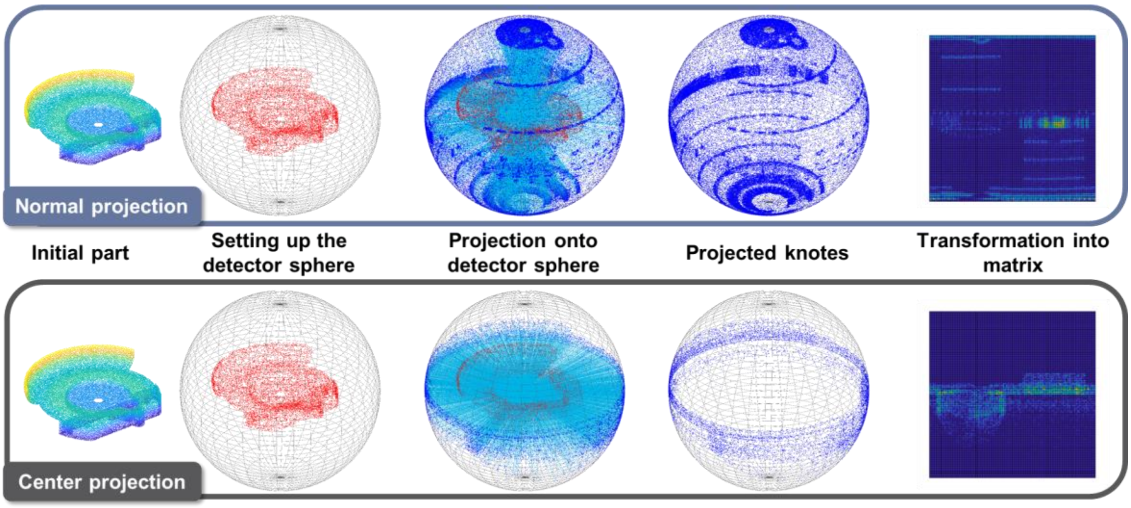

Figure 3. Examples of the two projection methods according to Bickel et al. (2020)

https://doi.org/10.1017/pds.2022.152 Published online by Cambridge University Press

The resolution of the projection matrix describes the number of pixels on the detector sphere. A matrix with the resolution 100x100 means that the detector sphere has 10,000 pixels. For this study, a fixed resolution of 256x256 is chosen, due to two reasons: On the one hand, this resolution value can still be calculated relatively time-efficiently. On the other hand, the network architectures have a standard input size between 230-260 pixels and can provide reasonable results with this input size.

The projection method can also be varied. The original version uses a center projection, as stated by Spruegel and Wartzack (2015). This general idea was enhanced with the normal projection from Bickel et al. (2020). The new aspect is, unlike the central projection, to use the normal direction of each point for the projection. The normal direction can either be determined via the mesh faces or calculated using the neighbouring points. The general procedure of these two ways of projecting points can be seen in Figure 3, which shows the difference in the two types of geometric transformation.

Besides counting the projected points per pixel, other values can also be used. A newly developed variant is called "distance projection". In this case, the projection follows the example of the center method. Rather than counting the projections per pixel, the distance between the center point and the point cloud point is projected into the pixel. The mean value of the distances per pixel is then determined and transferred to the matrix.

This newly developed type of geometry transformation solves a significant problem of the two previous projection types. Through the additional information about the mean value of the distance, it is possible to create an approximation to the geometry, starting from a created transformation matrix. The process is not fully reversible, as the mean calculation per pixel contributes to the inaccuracy of the reconstruction. It also offers the advantage for the classification process of more collected information in the transformation matrix. The distance had not been considered before and could help to improve the accuracy of the classification.

The last step of the geometry processing is the conversion of the generated matrices into RGB images. For this purpose, an existing function in the image datastore of Matlab is employed, converting the 1-channel images into 3-channel images. These images are then passed on to the machine learning model in the next step. The necessary structure and training of the models are explained in depth in the subsequent section.

3.3. Building and training of the classifier

The procedure for the building and training of the classifier model is based on the standard process of creating machine learning models with transfer learning. This means that pre-trained networks are taken as a starting point and specific layers are exchanged for the particular task. The network is only fine-tuned, which usually leads to faster and more stable trained models. First, the available data is split into training- and test-datasets. The training dataset is used for fitting and optimizing the pretrained model, while the test-set is applied afterwards for evaluating the accuracy of the trained model. As mentioned earlier, CNNs are broadly deployed for classifying images and geometries and therefore are chosen for finding the right class for the generated matrices.

Over the recent years, a various number of CNN architectures have been developed. A huge step in the classification accuracy of image detection was achieved with the alexnet in 2012 by Krizhevsky et al (2012). The next significant improvement was the introduction of the VGG 16 and VGG 19 architecture through Simonyan et al. (2014). These very deep neural networks could increase the accuracy of the classification, but due to their high number of layers, require a lot of computing power and time. This was followed by the development of the resnet structures through He et al. (2015), which are also available in different numbers of layers (e. g. resnet18, 50, 101). The benefit of this network structure is the skipping of layer and as a result, very deep networks can be realised that are still less computationally intensive.

Older architectures such as the alexnet, as well as more recent models with new layer versions were selected for this study. Overall, five different CNN models have been selected for the component detection application, which were all pretrained on the ImageNet (Deng et al. (2009)) dataset: resnet50, inceptionresnetv2 (Szegedy et al. (2016)), densenet201 (Huang et al. (2016)), alexnet and inceptionv3 (Szegedy et al. (2015)).

https://doi.org/10.1017/pds.2022.152 Published online by Cambridge University Press

There are two main reasons for choosing multiple networks. Firstly, it should be ensured that the presented projection method can transform geometry into a matrix that offers enough specific features for successful classification. If different CNNs can correctly classify the converted data, it indicates that the geometry conversion was successful. Furthermore, the different CNNs will be tested to determine which is the best architecture for the problem, compared by the achieved test accuracy of each CNN-structure.

4. Evaluation of the new approach

In this section, the previously presented method is examined and the influence of the different CNNs is shown. This is followed by a comparison with its native approach and established algorithms in the field of computer vision. For all comparisons, the stated test and training datasets from a benchmark are used. The training dataset is applied to create and optimise the machine learning model. The finished model is then checked against the unknown test data to verify that the generated model is applicable. The dataset chosen is a benchmark dataset for mechanical components; this new geometry collection has been developed specifically for the product development domain and therefore differs significantly from other established datasets. In the following chapter, this dataset is briefly illustrated and explained to present the results of the classification in more detail.

4.1. Presentation of the dataset

The applied dataset for this new classification method was developed by Kim et al. (2020) and is named "Mechanical Component Benchmark" (MCB). It consists of 58,696 different geometries, sorted into 68 classes. The dataset is divided into two different sets, whereby dataset A was used for this study, which contains the same statistics as the original dataset. A short sample of components from the collection is shown in Figure 4.

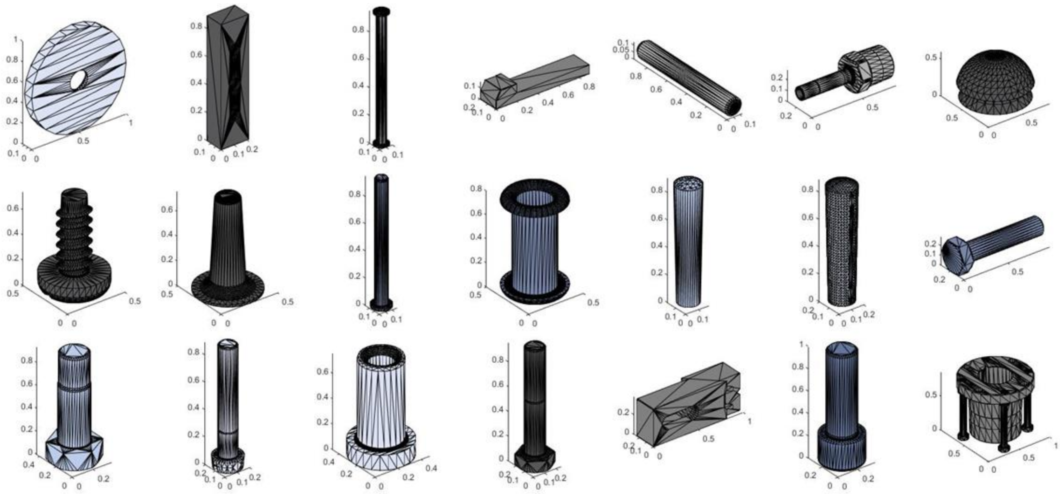

Figure 4. Overview of random parts in the MCB-dataset

For the evaluation of the results, the metrics have been adopted from the MCB dataset. The applied formula for the accuracy is listed below.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

The used variables in the equation are true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), according to Fawcett (2006).

4.2. Comparison to the old approach

To demonstrate that the new method provides higher accuracy values than the old approach, initial investigations have been carried out. A workstation PC with 32 GB RAM, an Intel XEON W-2125 and a Nvidia Titan V was used for the training of the all stated models in this paper.

As previously explained, the original method is not able to use surface meshes as an input format. For this reason, the procedure was adapted by the process step "geometry preprocessing" from Figure 2. It was ensured that the quotient of "point cloud number" / " total number of pixels" was in the same range as the new approach. The number of pixels is specified as 36x36 which corresponds to a point cloud number of 1000. As described in the publication of Spruegel and Wartzack (2015), the matrix is subsequently transformed into a row vector and fed into a neural network. In addition to the original specification of two hidden layers with 550 neurons each, 10 other layouts were tested.
The original layout achieved a test accuracy of 62.82 %, whereas the overall maximum of 65.06 % was achieved with a different layout (280 125 45).
In contrast to these results stands the first test for the new approach with a resnet50 as architecture and also the center projection. With this combination, an accuracy of 89.03 % was achieved. This comparison demonstrates that the new method gives a significantly better result, improving the accuracy by about 25% points. The results are shown in Figure 5.

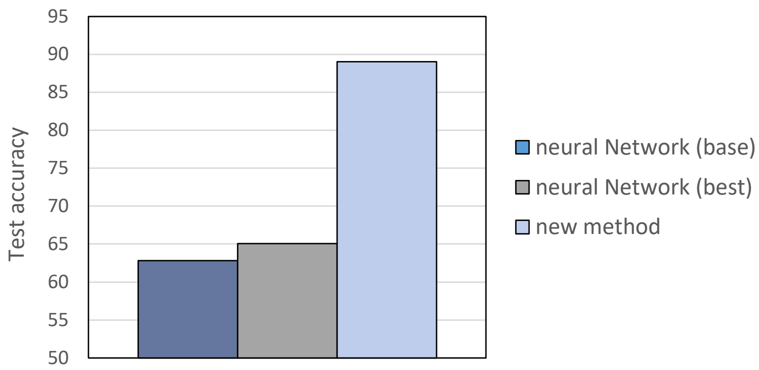

Figure 5. Comparison of the new method with the older approach

Therefore, not only in the general characteristics of the method could be improved, but also the classification of components. Based on these results, the new method will be examined in detail in the following chapters.

4.3. Evaluation of the projection methods and network architectures

To test the assumption, whether the geometry transformation provides enough information and features to classify components, the test-accuracy of all five networks is analysed. All data was prepared in the same way and the training parameters for the different CNNs also remained the same for all five models. The training duration for one model varied between 8 h and 2 days, depending on the architecture. First, the influence of the data preparation with different projection methods on the classification result has been investigated. For this purpose, the results of each projection variant for all five networks were summarised in a box plot and are presented in Figure 6.

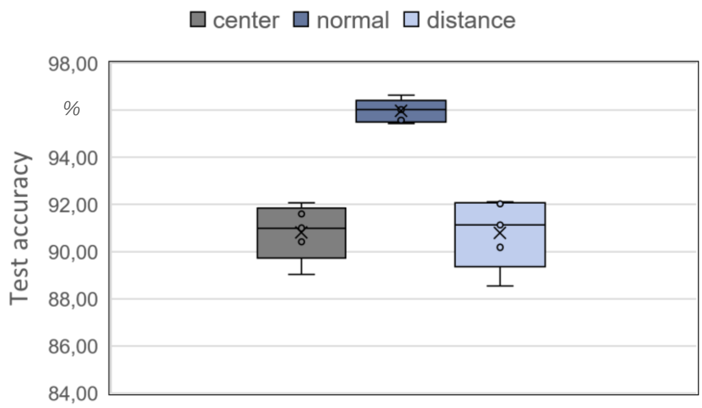

Figure 6. Comparison of the projection methods with all network types

https://doi.org/10.1017/pds.2022.152 Published online by Cambridge University Press

The diagram shows the distribution for the projection methods: center, normal and distance. It should be noted that all three projection variants show satisfactory results. The lowest classification value is 89.03 %, whereas the highest value reaches almost 96.65 %. These high values in combination with the relatively low dispersion of the results demonstrate that the geometry conversion works and provides sufficient features for classification.
Furthermore, the comparison among the projection methods clearly shows that the normal projection provides better results than the other two variants, independent of the CNN architecture. The test accuracy is on average more than 5 % higher compared to the previous two methods. The difference between the center and the distance projection reveals that the distance variant delivers slightly better results with a further scattering, but both are not comparable with the normal projection.
Proceeding from this study, the normal projection was selected for further investigation to determine the best combination with the stated CNN architectures. The results are displayed in Figure 7. The various networks and their associated classification accuracies are plotted in the bar chart. The diagram compares the five CNN models with each other and highlights that the combination of normal projection and the densenet201 delivers the best results for the MCB dataset.

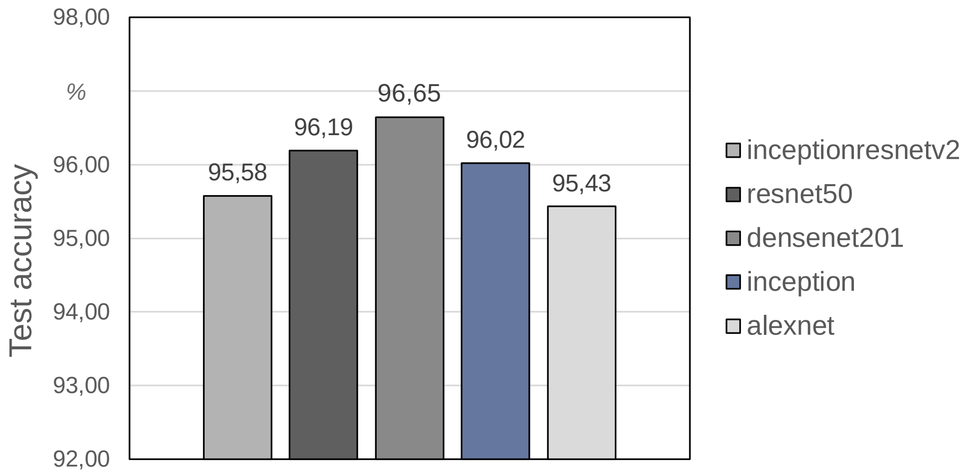

Figure 7. Comparison of different CNNs with transformed geometry by the normal projection

In summary, the results indicate that geometry conversion is effective and can provide relatively good classification accuracies. In the upcoming chapter, a comparison is conducted with established object detection methods.

4.4. Comparison to established methods

The achieved results are now compared with the accuracies produced in the MCB publication. In this paper, a total of seven state-of-the-art methods from the part classification field were compared with each other. The algorithms are explained in detail in section 2 and are called: PointCNN, PointNet++, SpiderCNN, MVCNN, RotationNet, DLAN and VRN. In addition, the results of the approach to classify parts in FE simulations were added. The procedure developed in this publication is benchmarked against these eight procedures by the test accuracy. The diagram in Figure 8 displays the comparison result.
The bar chart reveals that the developed method can compete effectively with the other approaches. Only one algorithm scores better results for the MCB-A dataset, the RotationNet. All other methods have a lower classification accuracy compared to the projection method, proving that the new procedure can compete with the available methods.
This combination of data set and capable methods offers new possibilities for the application in product development. A common problem is the unavailability of a dataset with labelled components, which is necessary for training. A possible solution to this, similar to transfer learning, would be to train a model with the large MCB data set and then optimize it with less data from the specific use case.
In addition, the data set can be used directly for an application, since it has a high number of classes, which cover many standard components from product development. One idea would be to use the data set to detect components

in assemblies and then transfer the geometrical model into a graph, similar to principle sketches. This requires a large number of specific classes of machine elements (e. g. washer bolt or countersunk screw) in conjunction with general components (e. g. turbine or valve), both of which are covered by the MCB data set. The generated result can then be combined with the results of another approach. Bickel et al. (2021) developed a method for recognising and classifying components of principle sketches and transferring them also into graphs resulting in the possibility of quickly displaying similar assemblies to generate a large variation for a problem solution in the early stages of the product development process.

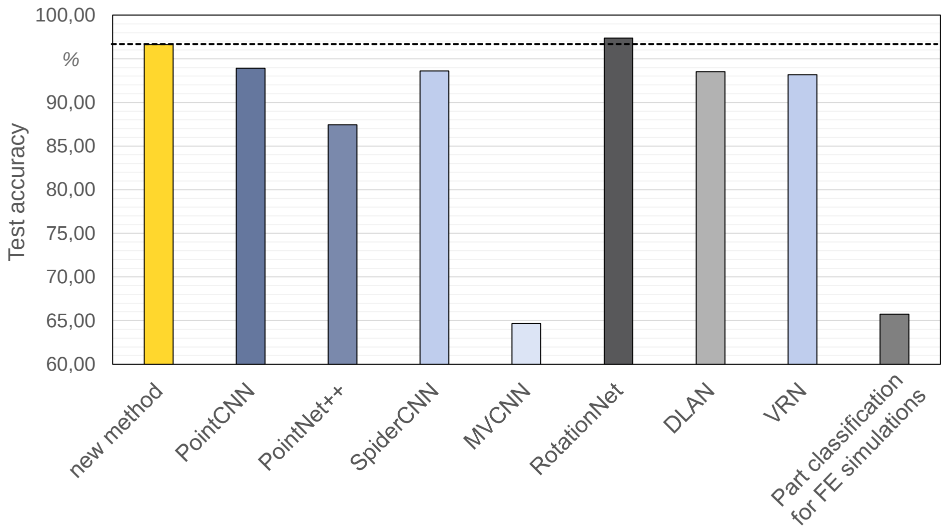

Figure 8. Comparison of the MCB results against the new method according to Kim et al. (2020)

Another idea would be to use the ability to detect specific parts within the FE simulation, but instead of applying this method in the preprocessing, this could also be used to improve the postprocessing of simulations. For example, it would be conceivable that, depending on the detected component class, neuralgic areas are automatically evaluated or frequent sources of error are pointed out.

5. Conclusion and Outlook

In summary, this paper presents a new method for the classification of 3D geometry in the field of mechanical engineering. The procedure was compared to an older approach and optimised based on various parameters and network architectures. Then, the new method was benchmarked against established methods from the field of object detection, whereby the new method performed very well in comparison. Only one method was able to achieve better accuracy values, the RotationNet.
The presented part classification method also needs continuous improvement; relevant parameters for new examinations are the hyper-parameters of the individual networks, as well as the component alignment. Especially with the orientation of the components, an optimised version should make it possible to further increase the classification result.

Acknowledgements

The authors thank the German Research Foundation for funding this research under grant number WA 2913/47-1.
The authors would like to thank the NVIDIA Corporation and the academic GPU Grant Program for the donation of a Tesla GPU.

References

Bickel, S., Sauer, C., Schleich, B. and Wartzack, S. (2020). Comparing CAD part models for geometrical similarity: A concept using machine learning algorithms. In Procedia CIRP (Eds.), Proceedings of the CIRPe 2020 - 8th CIRP Global Web Conference - (pp. 133-138).

Bickel, S., Schleich, B. and Wartzack, S. (2021). Detection and Classification of Symbols in Principle Sketches using Deep Learning. In Cambridge University Press (Eds.), Proceedings of the International Conference on

Engineering Design (ICED21) (pp. 1183 - 1192). Gothenburg, SE: Cambridge University Press. doi: 10.1017/pds.2021.118

Bickel, S., Spruegel, T., Schleich, B. and Wartzack, S. (2019). How Do Digital Engineering and Included AI Based Assistance Tools Change the Product Development Process and the Involved Engineers. Proceedings of the Design Society: International Conference on Engineering Design, 1(1), 2567-2576. doi:10.1017/dsi.2019.263

Brock, A., Lim, T., Ritchie, J. and Weston N. (2016). Generative and Discriminative Voxel Modeling with Convolutional Neural Networks.

Deng, J. et al., 2009. ImageNet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. doi: 10.1109/CVPR.2009.5206848

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. doi:10.1016/j.patrec.2005.10.010.

Furuya T. and Ohbuchi R. (2016). Deep Aggregation of Local 3D Geometric Features for 3D Model Retrieval. In Richard C. Wilson, Edwin R. Hancock and William A. P. Smith, editors, Proceedings of the British Machine Vision Conference (BMVC), pages 121.1-121.12. BMVA Press, doi: 10.5244/C.30.121

He K., Zhang X., Ren S. and Sun J. (2015), Deep Residual Learning for Image Recognition, CoRR, abs/1512.03385. doi: 10.1109/CVPR.2016.90

Huang G., Liu Z. and Weinberger K. (2016), Densely Connected Convolutional Networks, CoRR, abs/1608.06993. doi: 10.1109/CVPR.2017.243

Kanezaki A., Matsushita Y. and Nishida Y. (2018). RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints. *CVPR*, pp.5010-5019, doi: 10.1109/CVPR.2018.00526

Kestel, P., Kügler, P., Zirngibl, C., Schleich, B. and Wartzack, S. (2019). Ontology-based approach for the provision of simulation knowledge acquired by Data and Text Mining processes. Advanced Engineering Informatics, 39, 292-305.doi: 10.1016/j.aei.2019.02.001

Kim, Sangpil; Chi, Hyung-gun; Hu, Xiao; Huang, Qixing; Ramani, Karthik (2020): A Large-scale Annotated Mechanical Components Benchmark for Classification and Retrieval Tasks with Deep Neural Networks. In: Proceedings of 16th European Conference on Computer Vision (ECCV).

Krizhevsky A., Sutskever I. and Hinton G. (2012). Imagenet classification with deep convolutional neural networks. In NIPS. doi: 10.1145/3065386

Simonyan, K., Zissermann, A. (2014). Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556, 2014.

Spruegel, T., Bickel, S., Schleich, B. and Wartzack, S. (2021). Approach and application to transfer heterogeneous simulation data from finite element analysis to neural networks. Journal of Computational Design and Engineering, Volume 8(1), 298–315. doi: 10.1093/jcde/qwaa079

Spruegel, T. and Wartzack, S. (2015). Concept and application of automatic part-recognition with artificial neural networks for FE simulations. In Weber, C.; Husung, S.; Cantamessa, M.; Cascini, G.; Marjanovic, D.; Graziosi, S. (Eds.), Proceedings of the 20th International Conference on Engineering Design (ICED15) (pp. 183-193). Mailand, IT.

Su H., Maji S., Kalogerakis E. and Learned-Miller E. (2015). Multi-view convolutional neural networks for 3d shape recognition. doi: 10.1109/ICCV.2015.114

Szegedy C., Ioffe S. and Vanhoucke V. (2016), Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, CoRR, abs/1602.07261.

Szegedy C., Vanhoucke V., Ioffe S., Shlens J. and Wojna Z. (2015), Rethinking the Inception Architecture for Computer Vision, CoRR, abs/1512.00567. doi: 10.1109/CVPR.2016.308

Qi C. R., Su H., Mo K., and Guibas L. J. (2016) Pointnet: Deep learning on point sets for 3d classification and segmentation. arXiv preprint arXiv:1612.00593. doi: 10.1109/CVPR.2017.16

Qi C. R., Yi L., Su H., and Guibas L. J. (2017) Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413

Qie, Y., Bickel, S., Wartzack, S., Schleich, B. and Anwer, N. (2021). A function-oriented surface reconstruction framework for reverse engineering. CIRP Annals - Manufacturing Technology, 70(1), 135-138. doi: 10.1016/j.cirp.2021.04.016

Wu Z., Song S., Khosla A., Yu F., Zhang L., Tang X. and J. Xiao (2015). 3D ShapeNets: A Deep Representation for Volumetric Shapes, Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015), doi: 10.1109/CVPR.2015.7298801

Xu, Y., Fan, T., Xu, M., Zeng, L. and Qiao, Y. (2018). SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters.