NordDesign 2022
August 16-19, 2022
Kgs. Lyngby, Denmark

ResNet networks for plausibility detection in Finite Element simulations

Sebastian Bickel[1*], Benjamin Schleich[1], Sandro Wartzack[1]

[1]Engineering Design,
Friedrich-Alexander-Universität Erlangen-Nürnberg,
Martensstraße 9, 91058 Erlangen,

[*]mail: bickel@mfk.fau.de

Abstract

Nowadays, the efficient execution of product design and process planning activities without the extensive use of Finite Element (FE) simulation is hardly conceivable. The ability to virtually test components at an early stage can reduce costs while significantly increas-ing product quality. In addition, the current business environment promotes ever shorter development times coupled with a greater variety of products. This combination poses the risk that more and more inexperienced users from the field of product design have to perform simulation and validation tasks. An idea to support the less experienced users is to automatically check their FE models and simulations for plausibility. This helps to con-duct simulations independently by alerting the designers when major errors occur. One approach to the automatic plausibility check of FE simulations utilizes simulations that have already been calculated as a database and applies them to train a Deep Learning model that classifies the new simulations into plausible and non-plausible. This requires converting the existing data into a uniform format by the projection method, so it is read-able for a Deep Learning network.

However, for this method to be applied in an industrial environment, a high recognition accuracy is required for unknown simulations. Therefore, the goal of this paper is to investigate the ability of a new Deep Learning architecture to check the plausibility of FE simulations. The objective is to achieve the highest possible recognition accuracy. So far, mainly serial network types have been used for this procedure, which will now be extended by the application of ResNet networks. These have different paths in their structure through the addition of skip-connections, allowing for a theoretically better-trained model. This will be analysed together with a new dataset of simulations. These components for the structural mechanical simulation are geometrically varied and served to evaluate the new network types and their achieved accuracy.

Keywords: data mining, structural analysis, data driven design, Deep Learning, Plausibility checks

1 Introduction

The need for shorter development times, caused by a growing variety of product variants and a decreasing time to market (Kadam and Apte, 2016), forces many CAD designers to take over product simulation and validation tasks. Accordingly, Finite Element (FE) simulations in virtual product development are increasingly performed by users without in-depth FE know-how for the simulative evaluation of product behaviour (Kestel et al. 2019). At the same time, the amount of simulation data is expanding due to more powerful computational capabilities and available data storage. This existing data is often stored for legal reasons only and is not further utilized as it could be.

An automatic plausibility check of FE simulations to detect gravely wrong results is consequently a promising digital support tool for the virtual product development. It allows previously calculated and validated simulations to be used to provide feedback to the simulation user, enabling the less-experienced employees to create and iterate FE simulations independently. After the iteration, calculation experts take over the final validation of the product without unnecessary iteration loops between simulation and design departments.

When classifying FE simulations into plausible and non-plausible, numerous influencing variables must be taken into account since the relationships between the input variables and the target variables of a numerical calculation are often complex and non-linear. Especially the inhomogeneous data structure from FE simulations poses a problem for the automatic analysis of simulations. For this reason, the paper aims to improve an existing method for plausibility checking through a higher classification accuracy of the prediction, thereby increasing the applicability of the technique.

2 State of the art

There are several methods to improve the quality of FE simulations for specific domains. Examples are the SACON system of Bennett et al. (1978), which improves the development of wings for the Boeing 747, or the method presented by Johansson (2008) for the rotary draw bending of aluminium tubes. In contrast to a domain specific method, developed Spruegel et al. (2015) an approach for classifying FE simulations according to their plausibility which was further improved through Spruegel et al. (2021). The general idea is to provide a system that classifies a calculated simulation into plausible and non-plausible. In the context of finite element simulation, the term plausibility has been described by Spruegel et al. (2015) as a FE simulation that does not contain any obvious errors, which an experienced calculation engineer would recognize immediately. These include wrongly attributed units (e. g. bar and MPa), missing bearings or too coarse meshing or too high target values of the investigated component. The term plausibility is very well described by the English term "likely valid".

The first step in the method is the preparation of the calculated FE simulations. These must be converted to a standardized structure so that a neural network can process them. In this context, the method presented in Spruegel et al. (2016) projects the nodes from the center point onto a detector sphere, which is divided into several regions, similar to the longitudes and latitudes on a globe. The resulting areas are called pixels and all projected results are evaluated for each pixel. After the projection onto the sphere, it is unfolded like a world map, providing a matrix as a result. The projection is performed for the different elements of a simulation; this includes the mesh, the boundary conditions, the stress and deformation results. Each parameter is transferred to its own matrix, which is then arranged channel-wise, comparable to an RGB image. Figure 1 illustrates the entire process of the plausibility check. After an experienced FE simulation engineer has labelled the results, the matrices form the database for training the Deep Learning model.

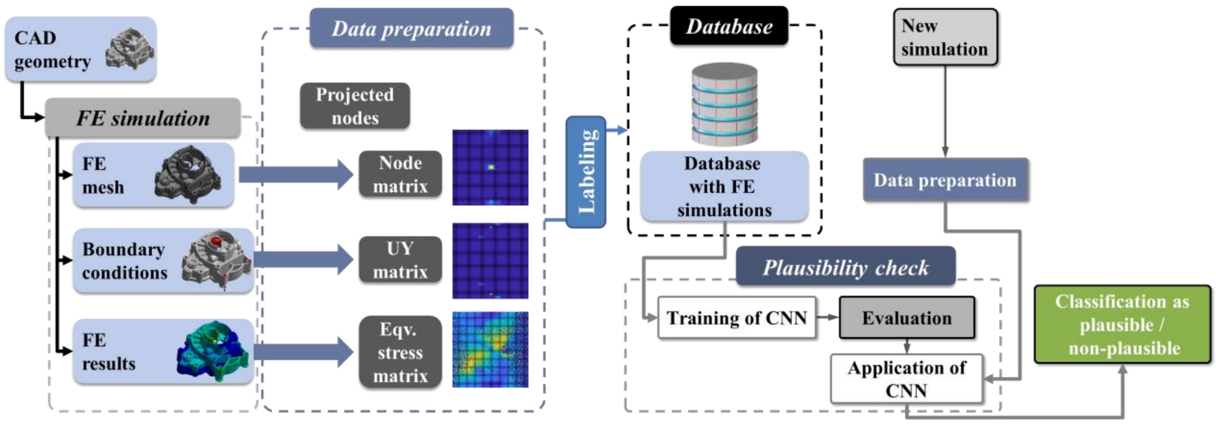

Figure 1: Overview of the plausibility-check method according to Spruegel et al. (2021)

The accuracies achieved in the publication of Spruegel et al. (2021) were high in some instances but should be further improved. The best result in Spruegel et al. (2021) could be generated by the own normalization of the data, which depends on the orientation of the components. Consequently, problems can occur, especially when comparing data across components. However, for a more general applicability of the method, new components should be investigated first. Currently, three geometric types have been investigated in different FE simulation parameter studies so far: bulky, planar and round/long. Consequently, this analysis focuses on components that are round but short and secondly less planar.

Besides providing a more extensive part database, the current paper explores the potential for higher classification accuracies by investigating a more general data normalization technique and by applying a new network type, ResNet models, to the task of detecting the plausibility of FE simulations. Increasing the recognition accuracy and with it the method's reliability for detecting the plausibility of simulations, leads to the method being able to be transferred to industrial applications in the future.

3 Deep Learning architecture

The presented task, the classification of FE simulations, is well suited for machine learning methods. As a result of the uniform data processing, Convolutional Neural Networks (CNN) are particularly appropriate since they were developed to recognise images. CNNs use the visual cortex as a template and model its functioning with different neurons, processing, and pooling layers. Convolutional Neural Networks are a particular form of artificial neural networks and their name addition refers to a convolution that takes place before the actual neural network to filter more information from the data.

CNNs can be categorized as Deep Learning, a subcategory of Machine Learning (ML). Deep Learning differs from Machine Learning through the way of generating the relevant features for the model training. In a classic ML model, the developer specifies these features, whereas a Deep Learning model creates them independently.

3.1 Methodical background

One of the first successful applications of CNNs was the recognition of handwritten numbers by the LeNet of Lecun et al. (1998). With the increasing computer power from 2010 onwards, it was possible to use CNNs practically and beneficially. The AlexNet from Krizhevsky et al. (2012) subsequently attracted particular attention with its results at an image detection competition in 2012. Compared to the LeNet, the AlexNet has significantly more layers and is suitable for higher resolution images. Two years later, an even deeper network, vgg16, was presented by Simonyan et al. (2014), which successfully participated in an image recognition competition.

These three CNN types are all serial, which forces the network to pass through all layers in sequence.
In theory, deeper networks should adapt better to a given task because the parameter space for adaption increases, but the achieved accuracy tends to saturate and then degrades, according to Srivastava et al. (2015). A possible approach to solving this problem was to incorporate multiple paths in a CNN. One of the first implementations was the ResNet of He et al. (2015). This kind of CNN aimed to solve the problem of the vanishing gradient by using a so-called residual block. An example of this network extension is provided in Figure 2.

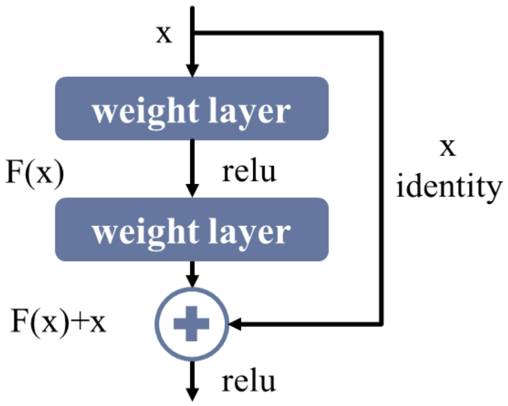

Figure 2: Example of a residual building block according to He et al. (2015)

The idea is to add residual mapping to the CNN model instead of an unreferenced mapping. For the execution in the CNN model, shortcut connections were built into the model, which execute identity mapping. Following this, the results of the skip connection and the stacked layers are added. The application of this type of network has demonstrated that image recognition accuracy increases significantly in many competitions and benchmarks. Therefore, this type of network will be applied to classify FE simulations and compared with a serial network.

3.2 Adaption for plausibility checks

A structure based on the vgg16 was chosen as a comparison model to the ResNet CNN. In the publication of Spruegel et al. (2021), this network type is also applied. In this study, however, a less deep variant is adopted. Figure 3 shows the linear structure of this network architecture. The vgg16-derivative uses five convolution layers, each followed by a ReLU (Rectified Linear Unit) layer. A max pooling layer is utilized three times and a drop out is inserted for each of the fully connected layers at the end.
The comparison with the ResNet highlights the difference between the two types of networks. The new CNN layout with skip connections consists of six simple blocks, called *convUnit* in the figure, each containing a convolution, batch normalization, ReLU activation, convolution, and batch normalization layer. The last ReLU layer connects the two paths again. The filter size is 3x3 and the values in the square brackets describe the number of filters and the stride for the convolution layer. Two blocks need an extra skip convolution layer so that the results have the same format when merging. Figure 3 again summarizes the structure of the two networks and shows the feature vectors generated per layer in each case.
Both networks are adapted to handle the 100 x 100 x 20 input and the high number of additional channels. In the first layer, called the input layer, a normalization between zero and one is automatically performed in both networks. The available hardware was also taken into account when setting up the network and therefore rather shallower networks were chosen, especially with regards to the vgg16-derivative. After the description of the CNN setup, the FE simulations for analysing the different networks are then explained.

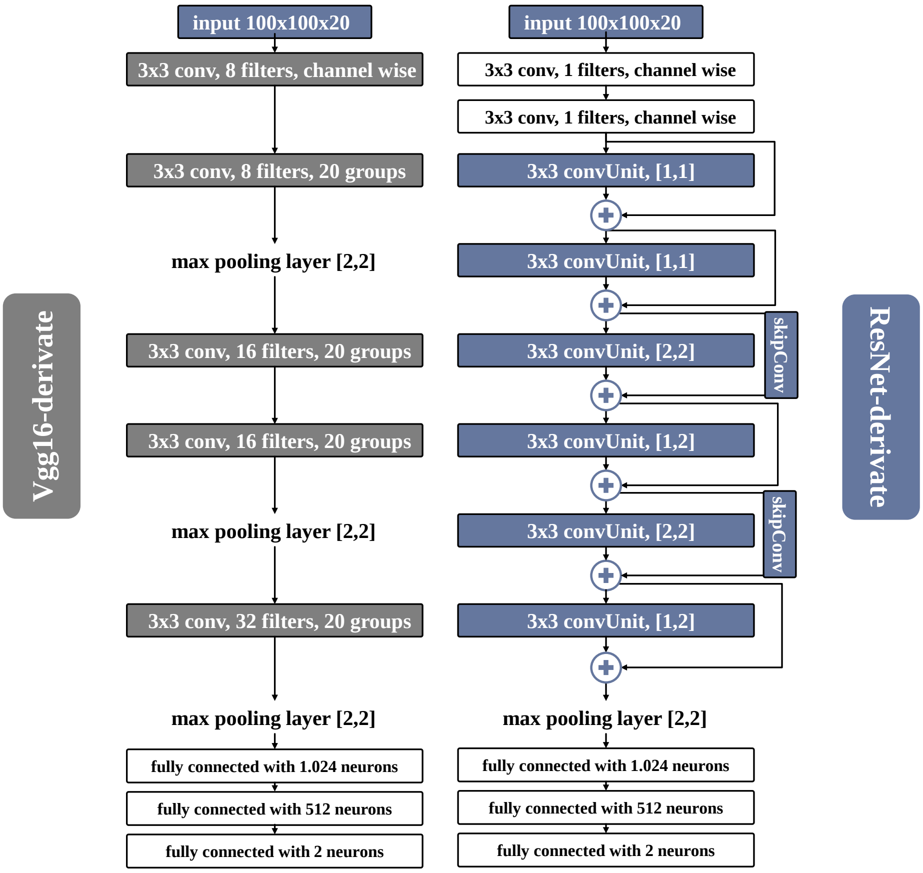

Figure 3: Comparison of both network architectures adapted for channel-wise input

4 Demonstrators

In order to test the application of the two CNN architectures, a demonstrator dataset consisting of two components has been created. The two parts form the basis for the structural mechanics simulations. To verify the method as thoroughly as possible, one component is kept round and short, whereas the other is flatter and thinner. This intends to represent different basic geometric shapes, which can also vary in simulations in an industrial application. The selected components are a vehicle rim and a brake lever for a bicycle. These parts are chosen because they have several unique characteristics for a parametric study, such as complexity of geometry, good derivation of the load case, available material data and a sustainable number of parameterizable quantities. The following two subchapters present the components and the generation of the entire dataset in detail.

4.1 Car rim

The first new component for a parameter study is an automobile rim. A car rim acts as a link between the tire and the rest of the vehicle. It must absorb a correspondingly high force due to the vehicle weight and driving motions. Rims are often made of steel or light metals such as aluminum or magnesium. They can be screwed, welded or forged and their dimensions are usually standardized (e.g., diameter or width). The design of the rims can have purely visual reasons, however, the goal is often to minimize the rim weight in order to keep the unsprung masses as low as possible.
In order to build the database, the first step is designing the component in CAD using PTC Creo with great attention to the parameterization of the geometry. A total of eight geometry param-

eters are defined, which control the following dimensions: offset, bolting thickness, overall diameter, rim width, spoke thickness and width / height for the narrow and wide cutouts respectively. The drawings in Figure 4 display the different geometrical parameters.

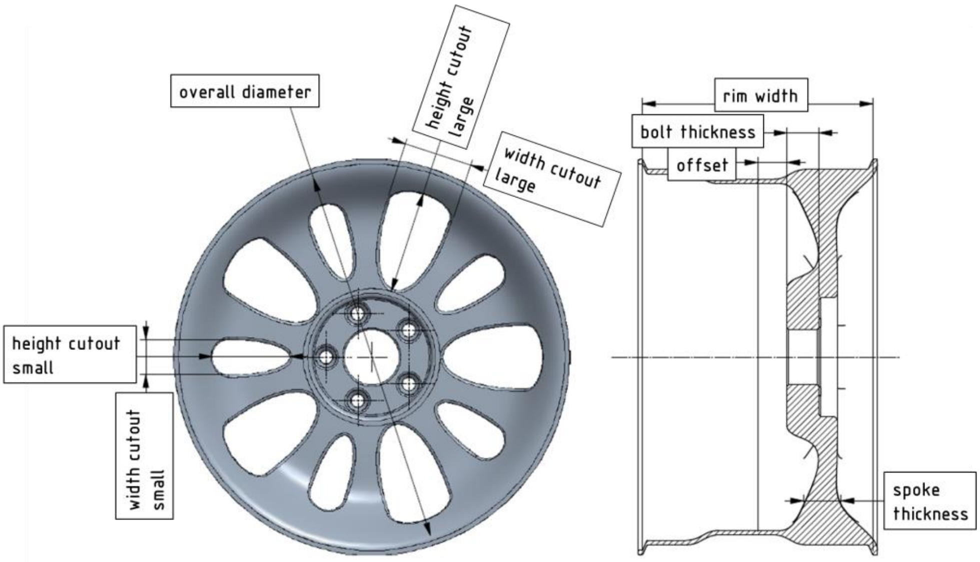

Figure 4: Geometrical parameters for the car rim

The simulation model is based on the rotary fatigue test according to Wang et al. (2011) and Jape et al. (2016). In this test, the rim is rotated and subjected to a moment to represent the loads during cornering. According to the SAE standard, the test setup comprises a rotating table and an axle connected to the rim. A constant force is applied to this axis, which can be calculated using the following equation.

$$M = \big((\mu * R) + d\big) * F * S$$

with M = Bending Moment, μ = Friction Coefficient, R = Radius Tire, d = Offset Rim, F = Maximum Load, S = Coefficient specified by Standards

To simplify the parameter study, a constant force has been defined for all rim sizes. A fixed constraint is defined on the rim flanks and the boreholes have been selected as the force application point. Furthermore, the rotation is represented over eight different load steps, with the applied force rotating around the center of the rim in 45 ° steps in each case. Figure 5 displays the setup of the simulation with all boundary conditions.

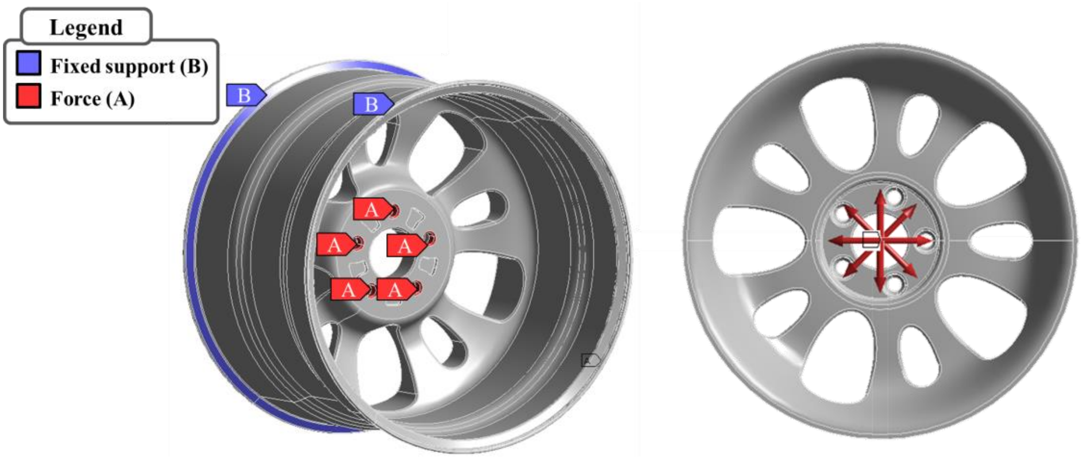

Figure 5: Setup of the FE simulation for the car rim with all eight load cases displayed

4.2 Bike Brake lever

The second part is a brake lever for a bicycle. The component is generally more planar and slimmer than the car rim. The lever transmits the force applied by the rider's hand to the brake shoes. For this purpose, a steel Bowden cable or hydraulic lines are often used. The demonstrator is a brake lever with a steel cable. The lever is pulled towards the rider to apply the brake, which tensions the steel cable and presses the brake shoes against the bicycle hub. For this purpose, the lever is pivoted so when the brake force is applied, the lever moves in a circular path.

Often brake levers are made of light metals, and their shape is adapted to the application. For a standard city bike, the length of the levers usually corresponds to a hand width, whereas for mountain bikes, the width of the lever is much shorter, usually half a hand width or less. Also, the thickness, curvature and possible cutouts depend on the intended use and targeted product cost.

Before setting up the simulation, the part is designed in CAD with multiple geometrical parameters. A total of eight parameters are applied to the model: the handle length and angle, the diameter of the bearing and the cable application point, the distance between the bores, the thickness of the neck, the total thickness and the wall thickness of the cutout. An example of different geometric variations and the representation of the geometry parameters are shown in Figure 6.

Figure 6: Example of the brake lever with different geometrical parameters

The FE simulation of component is modelled according to the German Institute for Standardization (DIN, 2014). This norm specifies both the force location and value that a brake lever must encounter. A value of 450 N is specified for a brake lever of this type. The gripping surface is defined as the force application area. A fixed support is applied at the bore for the brake cable. The pivot point of the lever is determined as a cylindrical bearing. Figure 7 illustrates the entire structure of the simulation, including the boundary conditions.

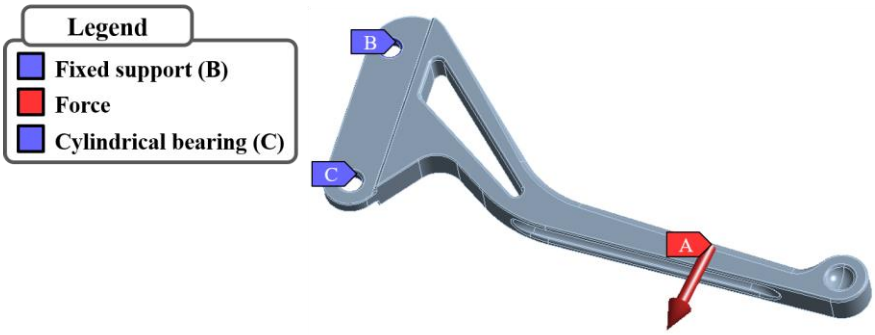

Figure 7: Setup of the brake lever simulation

4.3 Parameter study

To further build the database for the plausibility checks, parameter studies have subsequently been performed for both simulations. For the implementation, a d-optimal experimental plan was first created, containing all parameters of the respective simulation model. Besides the geometry parameters, the meshing and load values can also be controlled by parameters. Each experimental plan comprises 10.000 simulations, whereby for the brake lever, this also corresponds to 10.000 design points (dp); for the vehicle rim it takes 1.250 design points since eight load cases are calculated per dp. An APDL script saves the results for each simulation as a text file, including the stresses, deformations, boundary conditions and the general mesh. Table 1 lists all other relevant information about the dataset.

Table 1: General information about the demonstrator study

Dataset	Number of simulations in DOE	Successful simulations	Plausible	Non-plausible	Storage space
Vehicle rim	10.000	9.954	5.894	3.964	676 GB
Brake lever	10.000	9.858	3.734	6.230	574 GB
Whole dataset	20.000	19.812	9.628	10.194	1.250 GB

The entire dataset ends up with 19.812 simulations, of which 9.628 are labelled as plausible and 10.194 as non-plausible. The labelling was conducted through a combination of automated (e.g. rule-based for too high force values) and manually labelling. Subsequently, the dataset is randomly split into training and test data for each demonstrator, in a ratio of 80/20. This data is then used to compare how well the different network architectures predict the plausibility for a simulation. According to the method described in section 2, all simulations are projected onto a 100 x 100 matrix, with 20 matrices per simulation are used. This adds up to an input for the CNN model of 100 x 100 x 20.

5 Comparison between two network types

The evaluation metric for comparing the two network architectures is the classification accuracy. This metric is calculated for the three datasets: the training, validation, and test dataset. The equation for measuring the accuracy according to Fawcett (2006) and Powers (2011) is:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

with True Positive (TP), True Negatives (TN), False Positive (FP) and False Negatives (FN).

The test accuracy is particularly relevant since it is entirely unknown to the trained model. All training and evaluation steps were performed on a workstation PC with an Intel Xeon W-2125, 32 GB RAM and an Nvidia Titan V.

The different network models have been defined according to the setup from section 3.2 and trained with slight adjustments to the hyper-parameters, which are listed in Table 2. It was especially noticeable that the learning rate has a high influence on the achieved accuracy of the trained networks.

Table 2: Overview of the applied hyper-parameters for both network-architectures

Model	Mini-Batch Size	Initial Learn Rate	Solver	Shuffle Dataset	Learn Rate Drop Factor
Vgg16-derivate	64	0.0001	adam	every-epoch	-
ResNet-derivate	128	0.001	sgdm	every-epoch	0.1

The results of the evaluation for both networks are presented in Table 3. For the evaluation, five models per network type were trained with the same hyper-parameters and then evaluated. In Table 3 the arithmetic means of the accuracy for training, validation and test dataset are presented.

Table 3: Average accuracy over five trained models for each CNN type

Model	Accuracy		
	Training	Validation	Test
Vgg16-derivate	98,73 %	98,64 %	98,72 %
ResNet-derivate	99,98 %	99,99 %	99,98 %

The results consistently show a very high classification accuracy for both network architectures, through all datasets. This is a good sign if different network types work with one dataset, supporting the plausibility check method. The accuracy variation among the three datasets is relatively small, which indicates that there is no overfitting of the models. The vgg16-based model achieves an accuracy of 98.72 %, while to the ResNet network reaches an even higher accuracy of 99.98 % on average. This demonstrates that the ResNet model, in combination with the normalization of the data, was able to achieve a better classification result than the vgg16 model. This demonstrates that the new network architecture works for an input with multiple channels, derived from FE simulations.
Nevertheless, the next step should be to investigate further the validity of this finding, primarily through several more simulation parameter studies. Especially the generalizability to different components in one training dataset, is particularly relevant for the application of the method.

6 Conclusion and outlook

In summary, this paper examined two Deep Learning models for classifying two different FE simulations according to their plausibility. A model based on vgg16 was compared with a ResNet model. Two parameter studies were generated to investigate better the different architectures, each containing about 10.000 simulations.
The achieved accuracies show that the plausibility could be classified very precisely, even for a dataset with simulations from different domains. However, only the general plausibility of the simulations is determined and detailed feedback could not be provided as application-specific systems can. Nevertheless, the achieved result provides the basis for subsequent investigations, where simulations are classified according to a more specific reason for the non-plausibility, for example, a too large element size or wrong force values. Furthermore, it would be feasible to define a score for plausibility, similar to a percentage value or transform it into a traffic light system. For this purpose, the result before the softmax layers of the fully connected layer could be used as the initial result. In addition, the general idea of plausibility checks could be expanded to assemblies or other simulation types like CFD or modal analysis.

Acknowledgement

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 Titan GPU.

Citations and References

Bennett, J., Creary, L., Englemore, R., & Melosh, R. (1978). SACON: A Knowledge-Based Consultant for Structural Analysis. Stanford Univ Calif Dept. of Computer Science

DIN. German institute for standardization e. V. (2014). Cycles – Safety requirements for bicycles – Part 4: Braking test methods (ISO 4210-4:2014)

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

He, K., Zhang, X., Ren, S., Sun, J. (2015), Deep Residual Learning for Image Recognition, CoRR, abs/1512.03385.

Jape, R., Jadhav, S.G. (2016). CAD Modeling and FEA Analysis of Wheel Rim for Weight Reduction. International Journal of Engineering Science and Computing, June 2016

Johansson, J. (2008). Manufacturability analysis using integrated KBE, CAD and FEM. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 43291, pp. 191-200).

Kadam, S., Apte, D. (2015). A survey on short life cycle time series forecasting, International Journal of Application or Innovation in Engineering & Management (IJAIEM) , Volume 4, Issue 5, May 2015 , pp. 445-449 , ISSN 2319 - 4847.

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. https://dx.doi.org/10.1016/j.aei.2019.02.001

Krizhevsky, A., Sutskever, I., Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In NIPS

Lecun, Y., Bottou, L., Bengio, Y., Haffner P. (1998). Gradient-based learning applied to document recognition, in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791

Powers, D. M. W. (2011). Evaluation: From precision, recall, and f-measure to ROC, informedness, markedness & correlation. The Journal of Machine Learning Technology, 2(1), 37–63.

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

Spruegel, T., Bickel, S., Schleich, B., 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, Issue 1, February 2021, Pages 298–315, https://doi.org/10.1093/jcde/qwaa079

Spruegel, T., Hallmann, M., Wartzack, S. (2015). A concept for FE plausibility checks in structural mechanics. In: Summary of Proceedings NAFEMS World Congress 2015, 21.-24. June 2015, San Diego, USA.

Spruegel, T., Kestel, P., Wartzack, S. (2016). FEA-Assistenzsystem – Plausibilitätsprüfung für Finite-Elemente-Simulationen mittels sphärischer Detektorflächen. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.), Design for X (DFX) 2016, Hamburg: TuTech

Srivastava, R. K., Greff, K., Schmidhuber, J. (2015). Highway networks. arXiv:1505.00387.

Wang, L., Chen, Y., Wang, C., Wang, Q. (2011). Fatigue Life Analysis of Aluminum Wheels by Simulation of Rotary Fatigue Test. Strojniski Vestnik. 57. 31-39. 10.5545/sv-jme.2009.046.