CIL Course Project Report 2021: Road Segmentation

Team name: Corpus Inscriptionum Latinarum (CIL)
Akanksha Baranwal, Jona Braun, Andreas Kaufmann, Frederike Lübeck
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—Recent research in the field of semantic image segmentation predominantly directs attention to the development of slight architecture variations with increasing complexity. In this work, we analyze the impact of architecture modifications of a U-Net, namely the GC-DCNN and other self-developed variations. Our experiments with fine tuning and model architecture alterations lead us to a novel better variant of GC-DCNN. We also propose two novel post-processing techniques to remove artefacts in predictions. Although these methods improve the visual quality, they do not improve prediction accuracy significantly due to patch abstraction. With the help of numerous experiments, we conclude that the greatest improvement is observed by making the training dataset diverse.**

I. Introduction

Road Segmentation is a category in semantic image segmentation tasks where the goal is to detect roads in aerial images. More specifically, given a set of labelled images, our task is to label each pixel as either road, or non-road, respectively. Challenges arise as parts of roads can be covered by trees or shadows of large buildings, resulting in fragmented predictions. Furthermore, roads can look considerably different in images taken at different locations or varying light conditions.

Since the success of Convolutional Neural Networks (CNNs) in the 2012 ImageNet challenge [1], CNNs and its variants are widely applied. For such segmentation tasks, a particular encoder-decoder CNN architecture, the U-Net [2], has proven successful as it enables the precise localization of objects in an image. A remarkable research area focuses on extending the plain U-Net by proposing a slightly modified network structure or by adding new components. The Res-U-Net [3], for example, adapts the concept of residual learning [4] to introduce so-called residual blocks, and the Global Context based Dilated CNN (GC-DCNN) [5] in turn, makes use of dilated convolutions to further enlarge the receptive field.

However, not only are these variants increasingly complex to understand, they also require more sophisticated hyper parameter tuning and significantly longer training times. Therefore, we are posing the following research question: *What is the contribution of these complicated network architectures in terms of prediction accuracy compared to alternative methods such as image augmentations, post-processing and using additional training data?*

(a) Input Image (b) True Label (c) Prediction

Figure 1: Example image with prediction obtained from a U-Net trained on ETH-data, *GMaps-public* and *GMaps-custom*.

In our analysis, we show that by appropriate tuning, we can reach similar or even higher prediction accuracy, while requiring significantly less computing resources with the U-Net compared to the GC-DCNN. An example prediction can be seen in Figure 1.

While fine tuning the baseline models, we extensively experimented with alterations to the model architecture itself especially for GC-DCNN. This lead us to a novel variant GC-DCNN-Plus which improved the baseline model. Additionally, we applied post-processing methods to remove recurring artefacts such as disjoint line segments, noisy images etc in the model predictions. Motivated by the idea that a human could heuristically connect line fragments just by looking at the predicted mask, we came up with two novel post-processing techniques. One approach retrains the network on the binary predictions using partial convolution layers. Another approach makes use of Hough Transformations [6] to connect road fragments by completing lines and is motivated by the human behavior of annotating roads. Retraining on the binary predictions has improved prediction accuracy.

The remainder of this report is structured as follows: In Section II, we describe the data we used and image augmentations we applied. Section III presents our baseline models, describes the experiments we conducted to improve on these baselines and extensively elaborates on the contributions of model architecture. Our proposed post-processing techniques are explained and demonstrated in Section IV. Finally, results are compared in Section V and discussed in Section VI.

II. Data

For this project, ETH provided 100 labelled aerial RGB images of size 400 by 400 pixel, acquired from Google Maps. Since deep learning approaches often require a large

amount of training data, we pursued two strategies to enlarge the data set for training our models.

A. Augmentations

First, we applied several transformations to each image using the python *Albumentations* library [7]. We applied rotations and flips to make the model more robust against variations in road orientation. In in order to keep the absolute widths of roads, cars, trees etc. fixed, we didn't consider cropping and resizing transformations. These augmentations increased the data set by a factor of eight.

Additionally, we applied random transformations during training, which are described in Section III.

B. Additional Data Sets

Because the provided data set is not only small, but also lacks diversity, we included a publicly available data set which we found on Github [8], which was retrieved from Google Maps (2411 images). In the following, we refer to this data set as *GMaps-public*.

We additionally scraped our own images from Google Maps using the tool of [8] where we focused on including a more diverse set of roads: 1) Roads with a red, light blue, light yellow, white, black and grey color tone, 2) Roads of different quality, e.g. including some with a lot of cracks, some with unusual texture and some without surface markings. The contribution of these different data sets are evaluated in Section III. In the following, we refer to this data set as *GMaps-custom*. This set contains 2366 images.

C. Labels

Even though we have pixel-level labels for our training images, our task is to predict one label per 16x16-pixel patch. We experimented with approaches to directly predict the label per patch. However, we came to the conclusion that both visually and in terms of prediction accuracy, a pixel-level prediction with subsequent voting per patch with a threshold of 25% produces better results. That is, if at least 25% of the pixels in a 16x16 px patch are classified as road, the entire patch is classified as a road.

III. Models and Methods

A. Baseline Models

We have focused on two different model architectures as our baselines: U-Net [2] and GC-DCNN [5].

1) Baseline U-Net: A U-Net is based on a fully connected CNN and consists of a contracting and an expanding path, arranged symmetrically in a U-shape. The contracting path has the architecture of a typical CNN, it repeatedly applies convolutions and pooling operations to extract meaningful features. This results in reduced spatial information. By combining these features with high-resolution information from the contracting path, the expanding path is able to precisely locate detected objects on a high-resolution output mask. This propagation of context information between different levels in the network is enabled by skip connections.

2) Baseline GC-DCNN: The Global Context based Dilated CNN (GC-DCNN) builds upon the structure of a U-Net. Encoder blocks in the contracting path are replaced by Residual Dilated Blocks (RDBs) to further enlarge the receptive field, by using the dilation technique [9]. A dilated convolution leaves a specific number of pixels spare between the rows and columns of a filter. Residual blocks have been introduced in [4], and are used to facilitate information propagation by adding shortcut connections to jump over layers. Additionally, a Pyramid Pooling Module (PPM) is used in the bottleneck layer, which applies multilevel global average pooling to obtain a global representation. See Figure 4 in the appendix for an illustration.

In order to improve the two baseline models and to tailor them to our specific problem of segmenting roads on aerial images, we conducted the following experiments. For each experiment, we provide an intuition on why it might improve our model and draw conclusions. This results in our final versions of the U-Net and the GC-DCNN compared in Section V.

Experimental setup: We train each network for 200 epochs and save the model weights of the best epoch in terms of validation accuracy. We use the optimizer Adam and the dice-loss as our loss function. As initial learning rates, we use 10^{-3} for the U-Net and 10^{-4} for the GC-DCNN, as we have figured out that these are good values to prevent overfitting as well as divergence. Additionally, we use a learning rate scheduler, which drops the learning rate by a factor of 10 at epochs 50 and 100. We train our models on the ETH data set including the augmented images and on the data set *GMaps-public*. We used a fixed validation set in order to compare validation accuracy among different experiments. Images which contain only non-road pixels are removed from the training set. During training, we apply the transformation *ShiftScaleRotate* with probability 50% to each image, which is explained in more detail later.

We evaluate our models in terms of validation accuracy on our validation set and the public score on Kaggle.

B. Experiment: Training Data

In this experiment we evaluate the contribution of using additional data described in Section II B. We always use a validation set of 20% and report in Table I the public score reached on Kaggle as well as the training epoch with the highest validation accuracy. A comparison of the validation accuracy itself is not helpful, since in this experiment the validation set is different for every experiment.

The results clearly show that using more data as well as a diverse set of images as in the union of all three sets can lead to a great improvement in prediction accuracy.

Data Set	Public Score	Best Epoch
ETH	89.162	77
GMaps-public	90.584	81
GMaps-custom	91.716	120
ETH + GMaps-public + GMaps-custom	**93.077**	72

Table I: Results of training on different data sets (U-Net)

C. Experiment: Augmentations During Training

In order to achieve better generalization and to make our models robust against small variations in the training images, we apply different random augmentations during training. Here we compare the augmentations ShiftScaleRotate (SSR), RandomContrast (RC) and GaussNoise (GN) from the albumentations library [7] using default parameters.

We apply each augmentation with probability 50% if we use one or two augmentations simultaneously. When using three augmentations, we apply each with probability 40%. The results are shown in Table II.

Model	Method	Validation Accuracy	Public Score	Best Epoch
U-Net	no augmentation	96.894	91.935	51
	SSR	97.185	92.352	124
	SSR+RC	**97.202**	**92.687**	83
	SSR+RC+GN	97.145	92.510	177
GC-DCNN	no augmentation	97.004	91.256	56
	SSR	97.270	92.242	71
	SSR+RC	**97.309**	92.317	155
	SSR+RC+GN	97.273	**92.422**	95

Table II: Results of Augmentation Experiments for U-Net and GC-DCNN

D. Experiment: Architecture Alterations U-Net

In order to fine tune the U-Net architecture to our specific task, we experimented with varying dilation, convolutional filter sizes and max pooling kernel sizes. Table IV in the Appendix shows the variations and results in detail. The best architecture, henceforth referred to as U-Net-Plus, was obtained using a pooling kernel of size 4 which increased the public score (in %) by 0.457.

E. Experiment: Architecture Alterations GC-DCNN

We experimented with the following alterations to the GC-DCNN architecture.

Deep: The original GC-DCNN was designed for images of size 256x256. However, as we use images of size 400x400, the bottleneck layer has a higher resolution. Therefore, we study the effect of a deeper network. Instead of 3 RDBs with filters [128, 256, 512] we use 4 RDBs with an additional filter of size 1024. This reduces the image size in the bottleneck layer by a factor of 16.

Atrous Spatial Pyramid Pooling (ASPP): We replace the bridge (PPM) with the ASPP [10] module.

(a) (b) (c)

Figure 2: Retrain on binary predictions: a) Original prediction, b) Retrain simple U-Net, c) Retrain U-Net with partial convolution and dilation 3.

Attention: We use an attention gate, as proposed by Oktay et al. [11], in the expanding path of the GC-DCNN. For this, we apply the attention gate to the decoder tensors before concatenating them with the skip connections.

The best result is obtained by combining all three above-mentioned alterations which creates our novel altered model GC-DCNN-plus. The results in Table V in the Appendix show that our architecture variations can increase the public score by up to 0.307 percentage points.

IV. Post-processing

We observed recurring visual artefacts like disconnected road fragments, see Fig. 1 c), in our predicted results. Initially we attempted to resolve this using classical methods such as morphological operations (e.g. erosion, dilation) and median filtering. While these structure based approaches helped by connecting disjoint road segments, they introduced new artefacts in some predictions by merging adjacent roads. Due to the diversity of the images, the exact filter type, shape and size, which worked best for a prediction was sensitive to the road structure. Instead of a manual grid search for the best possible parameters per image, we developed two novel learning based post processing methods.

A. Retraining on binary predicted images

We used the best predictions of the U-Net & GC-DCNN as a training set and retrained the network to learn how to connect roads by joining lines and to remove noisy predictions.

1) Partial convolution: Partial convolution layers are commonly used for image inpainting of irregular holes [12]. Instead of learning different morphological filters, we decided to use these layers to inpaint pixels to recreate road images. When retraining on the binary images, we replaced the normal convolution layers with partial convolution layers as in [12]. As seen in Figure 2 c) this leads to cleaner boundaries between roads and denoised predictions.

2) Increasing the receptive field: As seen in Fig. 2 b), some of the roads still remained disconnected. Increasing the dilation helped to increase the receptive field of the network to learn whether disjoint segments separated by a large black margin indeed constitute a road. This improved the quality of predicted images both for the U-Net with partial convolution

and the normal U-Net as shown in Tables VI and VII in the Appendix. Retraining helped to produce cleaner images, but improved the public score by only 0.199 percentage points due to the 16x16 patch abstraction.

B. Learning with Hough Transformations

Another idea we came up with was to nudge the network towards predicting connected roads by explicitly presenting possible connected line fragments. A classical computer vision technique to detect shapes such as lines in images is the so-called Hough transformation [6]. By looking at the predicted mask and counting the number of pixels that lie on straight lines in specific angles, this method returns all possible lines that exceed a certain threshold. We draw these lines with the width of an average street on the predicted masks, concatenate this image with the prediction and the original RGB image and train a post-processing network (U-Net) on it. Thus, our model is trained on images with 5 channels (RGB, prediction, lines). The network should decide which of these roads (if any) it should keep.

This can be interpreted in several different ways. First, one could see it as letting the network learn how to remove predicted roads, thereby concentrating exactly on the parts between already identified road fragments. Secondly, one could also interpret this as a kind of topological prior. The prior probability of another road pixel on the connecting line between two fragments is higher than in any other part of the image. Thirdly, by providing these lines, we give the network an additional set of features, which can be used to improve the current predictions – or can be ignored.

(a) (b) (c) (d)

Figure 3: Hough Transformations: (a), (c) show the original predictions. (b), (d) show the completed lines.

For an illustration of the Hough Transformation, see Fig. 3. In the first image (a, b), we see that the lines are very simple and therefore this approach works fine. However, when the road network is more complicated as in (c), there are many wrong lines (d). This leads our model to use the original prediction and mostly ignore the lines. Therefore, this post-processing technique does not help in improving prediction accuracy.

C. Ensemble Methods

Averaging the predictions from different models is another form of post-processing, as it helps to remove artefacts, such as separated road predictions. If at least 50% of the models predicted a pixel as road, then the ensemble prediction considers it to be road. Afterwards, the predictions are patched as usual. This procedure has shown to significantly improve prediction accuracy, as shown in Section V.

V. Results

Table III summarizes the results of our final models by comparing the Public Scores on Kaggle. The trend is that our proposed post-processing technique (Retrain on binary, IV-A) improves the accuracy. The best results are obtained with an ensemble of the best predictions from each model specification. For the ensemble, post-processing did not improve the accuracy. A reason might be the already removed artefacts by the ensemble.

Model	Specification	Data Sets	PP	Public Score
U-Net	Baseline	①, ②	-	92.352
U-Net	Aug.: SSR + RC	①, ②, ③	-	92.689
U-Net-plus	Aug.: SSR + RC	①, ②, ③	-	92.835
			✓	93.034
GC-DCNN	Baseline	①, ②	-	92.242
GC-DCNN	Aug.: SSR + RC + GN	①, ②, ③	-	92.457
GC-DCNN-plus	Aug.: SSR + RC + GN	①, ②, ③	-	92.999
			✓	93.065
Ensemble	All models from above, not post-processed		-	**93.497**
			✓	93.303

Table III: Results of Final Models: U-Net-plus Sec: III-D, GC-DCNN-plus Sec: III-E, PP: Post-processing Sec: IV-A). Data sets: ① ETH, ② GMaps-public, ③ GMaps-custom.

VI. Discussion

We observe that an accurately tuned U-Net performs similar to the more complex GC-DCNN and other variants with slight architecture alterations, suggesting that the exact model architecture plays a minor role. The largest contribution to prediction accuracy was reached by using additional diverse training images. Averaging multiple predictions in an ensemble significantly improves the predictions by removing artefacts and by combining the strengths of several models. Despite not improving Kaggle score significantly, our proposed post-processing method: retraining on binary images with a large receptive field was able to fill small gaps between roads. Since we are evaluated on the patch-level accuracy, these slight improvements did not reflect in the patched predictions. A general drawback of post-processing techniques is their required extra work to train another network.

VII. Summary

In this project we investigated the contribution of complex model architectures in comparison to other factors and conclude that the exact model architecture plays only a minor role. Furthermore, we proposed and evaluated two post-processing techniques, which improved our Kaggle public score only slightly, but enhanced the visual quality of the predictions significantly.

APPENDIX

A. Pyramid Pooling Module

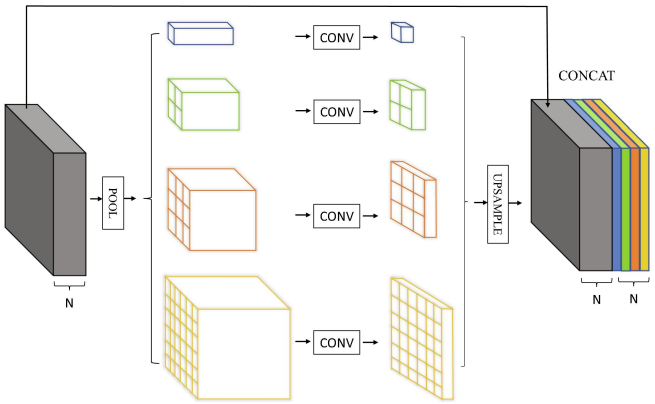


Figure 4: Pyramid Pooling Module used in GC-DCNN [5]

B. Augmentation parameters

In total we experimented with over twenty different augmentations. A key result of these experiments is, that if ShiftScaleRotate (SSR) is not applied the model overfits and achieves a training accuracy of over 99%. For brevity, we here only list the explanation of the top three augmentations.

- ShiftScaleRotate (SSR): Randomly shift, scale and rotate the image. We uniformly draw a shift value in [-0.0625, 0.0625], a scale value in [-0.1, 0.1] and a rotation value in [-45, 45] degrees.
- RandomContrast (RC): Change the contrast with a random factor in the range [-0.2, 0.2]
- GaussNoise (GN): We apply Gaussian noise with a mean of 0 and a variance limit of [10.0, 50.0].

C. Results of architecture alterations

1) U-Net: In the following we list the explanations of the different U-Net architecture tunings of which the results can be found in table IV.

- Dilation: Dilations with factors 3 and 18 were applied to all filters
- Filtersize: Filtersize was increased to 5 instead of 3
- PoolKernel: The Max Pooling kernel was increased to size 4 instead of 2
- Dilation and LR-Change: Dilation with factor 3 with a initial learning rate of 10^{-4} instead of 10^{-3}

method	Validation Accuracy	Public Score	Best Epoch
Baseline U-Net	**97.185**	92.352	124
Dilation (3)	96.817	91.942	34
Dilation (18)	96.280	90.941	86
Filtersize (5)	97.123	92.793	105
PoolKernel (4)	97.169	**92.809**	149
Dilation (3) lr=10^{-4}	97.110	92.203	70

Table IV: Results of U-Net Architecture Alterations

2) GC-DCNN: Results of architecture variations of the GC-DCNN can be seen in Table V. As our final model we choose the model ASPP + Attention + Deep, since it achieved the best validation score as well as a high public score.

Method	Validation Accuracy	Public Score	Best Epoch
-	97.270	92.242	71
Attention	97.279	92.246	60
ASPP	97.3	**92.549**	97
ASPP + Attention	97.145	92.201	51
Deep	97.272	92.195	178
ASPP + Deep	97.301	**92.468**	105
Attention + Deep	97.235	92.049	102
ASPP + Attention + Deep	**97.319**	**92.430**	132

Table V: Results of GC-DCNN Architecture Alterations

D. Postprocessing results

Model	Validation Accuracy	Public Score	Dilation
U-Net-plus	96.150	92.835	-
Retrain U-Net	**97.433**	92.964	1
Retrain U-Net	97.449	**93.034**	5
Retrain U-Net-PCONV	97.439	92.987	1
Retrain U-Net-PCONV	97.450	93.001	3
Retrain U-Net-PCONV	97.422	93.028	5

Table VI: Results of postprocessing binary retraining using U-Net on U-Net-plus predictions

Model	Validation Accuracy	Public Score	Dilation
GCDCNN-plus	96.28	92.99	-
Retrain U-Net	97.484	93.032	1
Retrain U-Net	97.480	**93.065**	5
Retrain U-Net-PCONV	97.490	93.041	1
Retrain U-Net-PCONV	97.547	93.038	3
Retrain U-Net-PCONV	**97.485**	93.029	5

Table VII: Results of postprocessing binary retraining using U-Net on GC-DCNN-plus predictions

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[3] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "Resunet-a: a deep learning framework for semantic segmentation of remotely sensed data," *CoRR*, vol. abs/1904.00592, 2019. [Online]. Available: http://arxiv.org/abs/1904.00592

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[5] M. Lan, Y. Zhang, L. Zhang, and B. Du, "Global context based automatic road segmentation via dilated convolutional neural network," *Information Sciences*, vol. 535, pp. 156–171, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025520304862

[6] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, p. 11–15, Jan. 1972. [Online]. Available: https://doi.org/10.1145/361237.361242

[7] A. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *ArXiv e-prints*, 2018.

[8] J. Friedman, A. L. John, R. Menta, and D. Weibel, "CIL-FS20-ETHZ," https://github.com/jkfrie/CIL-FS20-ETHZ, 2020.

[9] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *ArXiv e-prints*, mar 2016.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[11] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention u-net: Learning where to look for the pancreas," 2018.

[12] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 89–105.

GLOSSARY

ASPP
: Atrous Spatial Pyramid Pooling. 3

CNN
: Convolutional Neural Network. 1, 2

ETH
: Eidgenössische Technische Hochschule. 3

GC-DCNN
: Global Context based Dilated CNN. 1–5

GN
: GaussNoise. 3, 5

PPM
: Pyramid Pooling Module. 2, 3, 5

RC
: RandomContrast. 3, 5

RDB
: Residual Dilated Block. 2, 3

RGB
: Red Green Blue. 1

SSR
: ShiftScaleRotate. 3, 5

U-Net
: U-Network. 2–5

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

CIL Course Project Report 2021: Road Segmentation

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Baranwal	Akanksha
Braun	Jona
Kaufmann	Andreas
Lübeck	Frederike

With my signature I confirm that

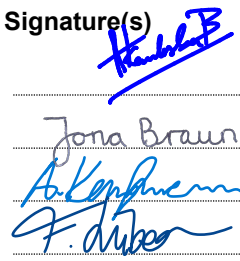
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

31.07.2021

Signature(s)



For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.