SEMANTIC URBAN MESH SEGMENTATION BASED ON AERIAL OBLIQUE IMAGES AND POINT CLOUDS USING DEEP LEARNING

Ł. Wilk [1, 2], D. Mielczarek [2], W. Ostrowski [1] *, W. Dominik [2], J. Krawczyk [2]

[1] Department of Photogrammetry, Remote Sensing and Spatial Information Systems, Faculty of Geodesy and Cartography, Warsaw University of Technology, Warsaw, Poland - (lukasz.wilk3.stud, wojciech.ostrowski)@pw.edu.pl
[2] OPEGIEKA, Elbląg, Poland – (dominik.mielczarek, wojciech.dominik, jakub.krawczyk)@opegieka.pl

Commission II, WG II/4

KEY WORDS: 3D Mesh, oblique images, LiDAR, semantic segmentation, neural networks

ABSTRACT:

The use of deep machine learning methods for semantic classification of city mesh models is one of the current trends in geoscience development. Thanks to the thriving development of Convolutional Neural Networks (CNNs) it is now achievable to conduct fully automated process of building aforementioned 3D model by means of photogrammetric techniques and supplement it with additional semantic information obtained by Artificial Intelligence (AI) algorithms. In order to guarantee the comprehensiveness of said information it is essential to use an extensive range of 3D data including oblique aerial imagery and aerial laser scanning (ALS). Such comprehensive 3D mesh models may be later implemented in many Digital Twin class solutions additionally supported with modern GIS systems and its algorithms. To proof the validity of this thesis, the article showcases results of research conducted using deep learning based solutions tested on two datasets - real-world data in the form of oblique aerial images and ALS point clouds acquired in Bordeaux, France using novel Leica CityMapper-1 multisensoral system and large-scale dataset from SUM: A Benchmark Dataset of Semantic Urban Meshes. Both subalgorithms make use of CNNs as its core-feature. To perform accurate classification of oblique aerial scenes PSP-Net architecture accelerated by techniques of transfer learning has been used. Second algorithm destined for ALS point clouds utilizes CNN as well, but in this case implementation is based on proprietary architecture. The results of the experiments demonstrate that the utilizing these two mutually complementary solutions to extract new semantic information for city mesh models in proposed manner compared with the state-of-the-art methods achieves competitive classification performance.

1. INTRODUCTION

Urban 3D mesh models, created mostly with Dense Image Matching of oblique aerial images (Haala, 2015), have become more and more popular as a method for representation of complex cities environments in recent years. The superiority of 3D meshes over LOD CityGML is obvious in the case of models created solely for the purpose of visualization. However, they are much harder to use for Digital Twin applications (City Information Modelling) as they are complex and geometrically unstructured which makes them less favourable to store semantic information required for complex spatial analysis (Lehner & Dorffner, 2020). The recently published Cesium 3D Tiles Next (Cozzi, 2020) format could be a game-changer in the field of semantic 3D models, as it promises an efficient way of streaming 3D meshes together with Semantic Metadata which could be stored on the various level. Storing semantic metadata not only for the purpose of geometric features but also in terms of texture pixels also warrants interest.

Several approaches to semantic segmentation based on photogrammetric data had been already introduced: Rouhani et al. (2017) proposed a supervised approach for classifying textured meshes, Blaha et al. (2017) presented a method of mesh surface refinement involving semantic information from images, Rong et al. (2021) used multiview oblique images to acquire semantic information from many images for each facet of the 3D mesh model. Methods of 3D mesh segmentation or labelling based solely on triangle faces have their limitations – small objects (like windows), could be merged with surrounding objects (like walls) on the geometry level. Frommholz et al. (2016) have shown that such objects could be classified directly on images and then projected onto the model. Therefore semantic segmentation of oblique images, which are often used for creating a 3D mesh, and then a projection of received classes from images to 3D meshes (textures) should provide reasonable results. Classifying oblique aerial images with Convolution Neural Networks in order to distinguish elements of the urban environment or building parts is a relatively new research topic (Huang, 2019; Liu, 2019).

The second typical source of data for urban modelling are point clouds from Airborne Laser Scanning, the classification of which is a common challenge among existing 3D urban benchmarks (Gao et. al 2021). Recently point clouds from ALS also become more and more popular supporting data for the creation of 3D meshes. This relatively new trend in mapping city areas is connected with hybrid mapping systems - combining LiDAR sensors with a nadir and oblique images (Toschi, et al., 2018; Bacher, 2021).

The method proposed in this paper (as opposed to segmenting 3D mesh) directly focuses on classifying data (oblique images, and LiDAR point cloud) that could be used for both mesh creation and adding semantic information. Proposed methodology was developed with the real-word dataset for Bordeaux collected with Leica CityMapper and then further evaluated with data from SUM: a Benchmark Dataset of Semantic Urban Meshes (Gao et. al 2021).

* Corresponding author

The Bordeaux (real-world dataset), consists of simultaneous collected: LiDAR point cloud (with the density of 10 pts/m2), nadir images, and oblique images in the four cardinal directions. Data were acquired from an altitude of 850 m above ground with a hybrid sensor – Leica CityMapper-1. This resulted in the 5 cm Ground Sample Distance (GSD) of nadir images with 80% overlap along flight lines and 60% overlap across flight lines. From Bordeaux dataset 11 oblique images (Fig. 1) are fully manually labelled into four semantic classes: facades, windows, roofs, and ground accompanied by an additional "other" class. For training of a neural network 8 images were used other three were used as a test dataset for evaluation of the achieved result.

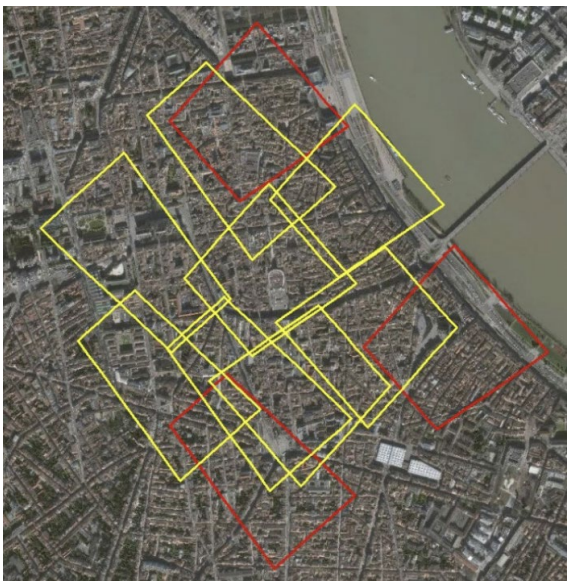

Figure 1. Bordeaux data set. The footprints of oblique images used for training (outline in yellow) and test (outline in red) of the neural network.

Dataset from SUM: A Benchmark Dataset of Semantic Urban Meshes consists of 64 tiles of Helsinki 3D textured meshes, each of which covers about 250x250 m. Tiles were divided into three groups: 40 were used as training data, 12 tiles as test data, and 12 tiles as validation during training. Meshes were created from oblique aerial images with 7.5 cm GSD, the data were labelled per facet of the meshes with 6 semantic classes: terrain, high vegetation, building, water, car, boat accompanied by an additional "unclassified" class. Within benchmark data classified point clouds generated from meshes are also provided, during our experiment we use one of three with a sampling density of 30 pts/m^2.

With SUM meshes 120 images (30 per direction with overlaps ca. 60/60%) were rendered with pyrender library. The intrinsic camera parameters have been selected in such a way as to simulate the Leica CityMapper-2 flying 2250 meters above the ground to match the resolution of SUM 3D meshes (GSD of 8 cm). The outputs of pyrender library are images with RGB and label values from mesh and depth maps.

2. METHODOLOGY

The main idea of the presented hybrid method consists of three parts: (1) first to classify input data used for photogrammetric modelling (oblique images and point clouds) instead of a classification of a final product – a 3D meshes; (2) then results of images classification are projected on the mesh and stored as texture files (due to overlap between images each pixel of texture (texel) should be visible and classified on at least a few images – differences between classification on different images can be solved by voting in texture space); (3) finally results from images classification are overwritten on a mesh level with result of LiDAR point cloud classification for that classes which can be classified with higher accuracy form lidar data than from images.

For point cloud classification, a convolutional neural network using only point cloud geometry has been used. This deep learning method was developed in-house by the Polish Geoinformatics R&D Centre OPEGIEKA (OPEGIEKA, 2020). Segmentation of aerial oblique images has been performed using the convolutional neural network architecture of the Pyramid Scene Parsing Network (PSP-Net). So both presented methods utilize completely different information for distinguishing the same objects with deep learning, image segmentation is based only on RGB values, and point cloud classification use only geometric information.

OPEGIEKA's solution for ALS point clouds classification is based on a fully convolutional neural network (Dominik et al. 2021). The solution core is an algorithm of point cloud transformation to a regular array accompanied by internally designed convolutional neural network architecture. In the first step, the point cloud is divided in the horizontal plane into grid cells of 1x1 meter. In the next step, raster images are generated layer by layer starting from the lowest points in each grid cell. The coordinates of the consecutive points in ascending elevation order are written as cell attributes of the consecutive images which are then stacked together.

The result of those steps is an array (4-dimensional). The first two dimensions of the array are the spatial width and height of the point cloud divided into 1 meter grid. The third dimension is the order of the points in the vertical direction in a given grid cell. The fourth dimension stores point coordinates. The size of the array is 64 by 64 by 64 by 3 in most cases. If in a certain grid cell the number of points exceeds the size of the third dimension points are randomly selected. The 4-dimensional array is then fed to a fully convolutional neural network constructed from 3D convolutional layers. During training, the data is generated by a generator that randomly selects patches of point clouds and transforms them into the 4-dimensional array.

After training the neural network, the classification of the point cloud is carried out by prediction of the neural network. For practical reasons, the LiDAR data is divided into 500 by 500 meters tiles. For experiments with data from SUM Benchmark original division into 250 by 250 meters tiles is used. The full process of classification of such tile takes about 3-4 minutes on a single machine equipped with a graphic card (GPU).

Segmentation of aerial oblique images has been performed using the convolutional neural network architecture of the Pyramid Scene Parsing Network (PSP-Net) proposed by (Zhao et al., 2017). CNN implementation was based on source code published in the segmentation-models library (Yakubovskiy, 2019) using python and Keras deep learning API. PSP-Net architecture can be divided into two main parts. The aim of the first module is to extract deep image feature maps used as a source for later image segmentation. In our case, the extraction was performed with ResNet-18 (He, 2015) CNN, which presented the highest level of performance among other architectures (eg. VGG) both in independent tests and in the literature (Garcia-Garcia et al., 2018; H. Zhao et al., 2017) . In principle, the core feature of PSP-Net architecture is to intercept more contextual information during

the final image segmentation, which allows the algorithm to extract multi-scale topological dependencies among the distinguished semantic classes. That's what the second part of CNN is designed for. Pyramid Pooling Module performs a fusion of feature maps obtained on four levels of image pyramids (extracted by four average-pooling layers with different kernel sizes). Finally, the result tensor is used as a data source for prediction/classification.

Original oblique images (approx. 80 Mpx for Leica CityMapper-1 data and 150 Mpx for rendered mesh scenes simulating CityMapper-2) are too big to be directly fed into the network to overcome this problem input images were divided into smaller tiles. Because of CNN architecture (combination of convolutional and max-pooling layers), each image dimension must be 48-divisible, that's why each input image has been cropped to size 624 x 624 px (approx. 50 by 50 m on the ground). While images from the training dataset have been tiled without any overlap, test images were generated with 50% of overlap (vertical and horizontal) between two consecutive tiles. The reason why the images were generated with such coverage was that there were worse prediction results at the edges of tiles.

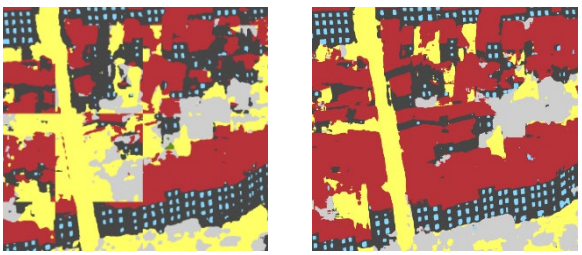

Figure 2. Bordeaux dataset – oblique aerial images CNN prediction results without (left) and with mosaic mechanism (right).

To avoid that problem, the final assumption for merging results of the prediction was that the pixels closer to the centre of the tile are more reliable. Based on this assumption final predictions were built using only middle parts of tiles (Fig. 2, 3).

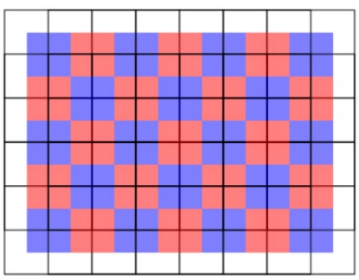

Figure 3. Final prediction building scheme. Red and blue parts of pictures (middles of tiles) are connected side by side to reduce effect of false classification on edges.

Another problem that was addressed during the first experiment with Bordeaux dataset was the relatively small size of the training set for real-world images. In comparison to 120 images rendered from SUM dataset, 11 full CityMapper-1 images are not enough to perform reliable prediction (Tab. 1) Because of that, the data augmentation process was applied during CNN training. Our augmentation process, inspired by operations presented in similar approaches (Liu, 2019), consisted of 3 possible operations:

1. Horizontal flip (probability=0.5)
2. Contrast and brightness manipulation (p=0.8)
3. Affine transformation (p=0.5).

For contrast and brightness manipulation specified values were changed in the range of ± 20% in comparison to original value. During affine transformation every image has been scaled by ± 15% and then could be either sheared by an random angle in range ± 10° or rotated by an random angle in range ± 20°. Using the following augmentation process implemented with albumentations, the python library training dataset was doubled.

Dataset	Training dataset size	Effective training dataset size
Real-world (CityMapper-1)	885 Mpx	1 834 tiles
SUM (render)	18240 Mpx	14 706 tiles

Table 1. Sizes of the training datasets (used for semantic segmentation of oblique images) in Mpx and number of tiles (624 x 624 px each).

To further accelerate and regularise the learning process in every case the transfer-learning technique was used. The first part of PSP-Net used for feature extraction was pretrained using ImageNet dataset (Deng et al., 2009). Whole learning process was conducted using categorical cross-entropy (CCE) loss function and Adam optimizer (Kingma, Ba, 2014) with an initial learning rate of 0.001 further reduced when plateau was detected. Then results of semantic segmentation from images were projected into the mesh and stored as a texture.

In order to project every pixel from classified oblique images into a texture map, a simple texturization algorithm was applied. First for every pixel of the texture map (texel) and for every triangle of the model a position on the model is determined. Using multilinear interpolation, which can be viewed as an affine transformation (Tymchyshyn, 2019), transformation from UV space of texture coordinate to model coordinate we are able to perform texel mapping. The affine transformation matrix is obtained with the Laplace elimination method. The derived transformation matrix with a translation vector transforms points from the mesh model texture projection to the model projection.

The height of the individual pixels in the new 3D model projection is determined using the plane equation. The values of plane parameters are determined as normal planes using the cross-product of two vectors constructed from the points of a single plane of the mesh model. As the result, all raster cells (texels) of the UV texture map are projected on the mesh model. The density of points is directly related to the resolution of the UV texture image. Finally, for each texel 3D coordinates in model space are projected on the corresponding image (with known interior and exterior orientation parameters, and depth map used for visibility check) and then classification results are acquired using nearest neighbour resampling.

With this method for each 3D mesh tile, for every corresponding oblique image, a separate texture is created maintaining UV coordinates of the original texture map. Then all textures for a single mesh tile are stacked together, for SUM Benchmark single mesh tile was visible on 6-9 rendered images which result in more than 30 textures with classification results per every mesh tile. The majority voting in UV coordinates of texture space was used for the creation of the final texture with classification results.

3. RESULTS

Results of the semantic segmentation of oblique images, from the real-world (Bordeaux) dataset, were presented in Table 2. Per-pixel evaluation metrics were calculated based on three fully manually labelled test images (Fig. 1). The lowest accuracies, as well as Intersection over Union (IoU) values, were achieved for pixels labelled (classified) as windows and ground. The most probable reason for the worse accuracies of these two classes is the unbalanced training dataset for the test field (less than 12% of pixels were labelled as a ground on oblique aerial images).

Class	Accuracy	IoU
Facades	71.82%	61.81%
Windows	53.90%	38.11%
Roofs	86.28%	78.59%
Ground	71.33%	54.11%
Mean	70.83%	58.16%

Table 2. Real-word dataset. Results of image segmentation with PSP-Net.

To improve achieved results hybrid approach (Fig. 1) of ground classification was proposed – the ground was classified on the LiDAR point cloud and then corresponding faces of 3D mesh were labelled as ground and then projected to the oblique images overriding the classification achieved from image segmentation. This process improved achieved accuracy and IoU for ground class pixels up to 89.3% and 77.3% respectively (Fig. 4).

For further investigation of differences in performance of semantic classification of point clouds and oblique images data from SUM: a Benchmark Dataset of Semantic Urban Meshes were used. Results of classification of the point cloud (with a density of 30 pts/m2) with OPEGIEKA's CNN evaluated on 12 test tiles achieved Overall Accuracy (OA) of 91.4%. These results were achieved with classification including six semantic classes as well as class unclassified. Per-class metrics calculated for points for six semantic classes (Tab. 3) show results comparable with the top results achieved in SUM Benchmark experiments (Gao et. al 2021).

Class	Precision	Recall	F1	IoU
Ground	89.6%	92.9%	91.2%	83.8%
Vegetation	92.4%	95.1%	93.7%	88.1%
Building	93.7%	97.4%	95.5%	91.5%
Water	94.4%	89.5%	91.9%	85.0%
Car	75.6%	59.6%	66.6%	49.9%
Boat	79.4%	16.3%	27.1%	15.7%
Mean	87.5%	75.1%	77.7%	69.0%

Table 3. SUM benchmark dataset. Per-class results of point cloud classification.

Class	Precision	Recall	F1	IoU
Ground	85.5%	89.5%	87.4%	77.7%
Vegetation	91.8%	86.3%	89.0%	80.2%
Building	91.7%	94.9%	93.3%	87.4%
Water	92.6%	59.9%	72.7%	57.2%
Car	45.5%	79.4%	57.9%	40.7%
Boat	19.7%	78.5%	31.5%	18.7%
Mean	71.2%	81.4%	72.0%	60.3%

Table 4. SUM benchmark dataset. Per-class results of oblique image segmentation.

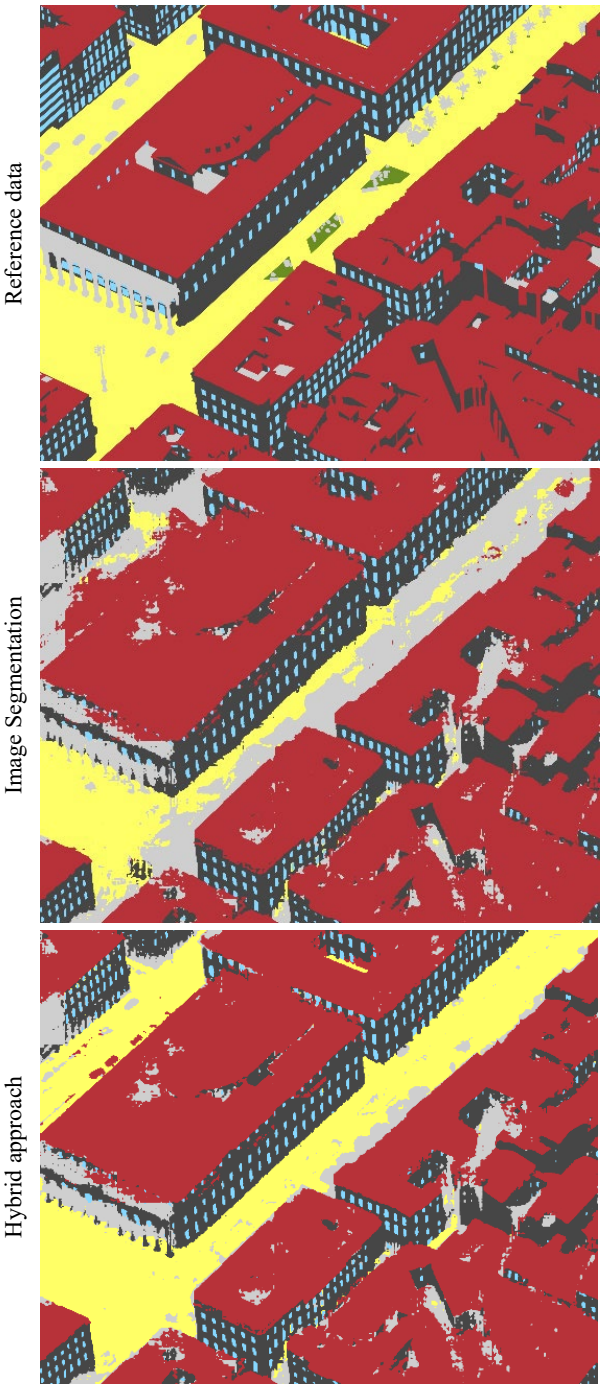

Figure 4. From top: Reference data from manually annotated oblique image; results of image segmentation with CNN, results from hybrid approach (ground classification from lidar data).

Results of the semantic segmentation of oblique images rendered from the SUM Benchmark dataset firstly were evaluated in the image space of oblique images. In order to calculate evaluation metrics the prediction with the PSP-Net was run on fully rendered RGB images and then test and validation tiles were masked out – comparison between reference classes rendered from mesh and prediction results were performed only on sections of images corresponding with test 3D mesh tiles. Achieved results (Tab. 4) with OA of 84.1% are slightly worse than in the case of point cloud classification with OPEGIEKA's CNN, but still.

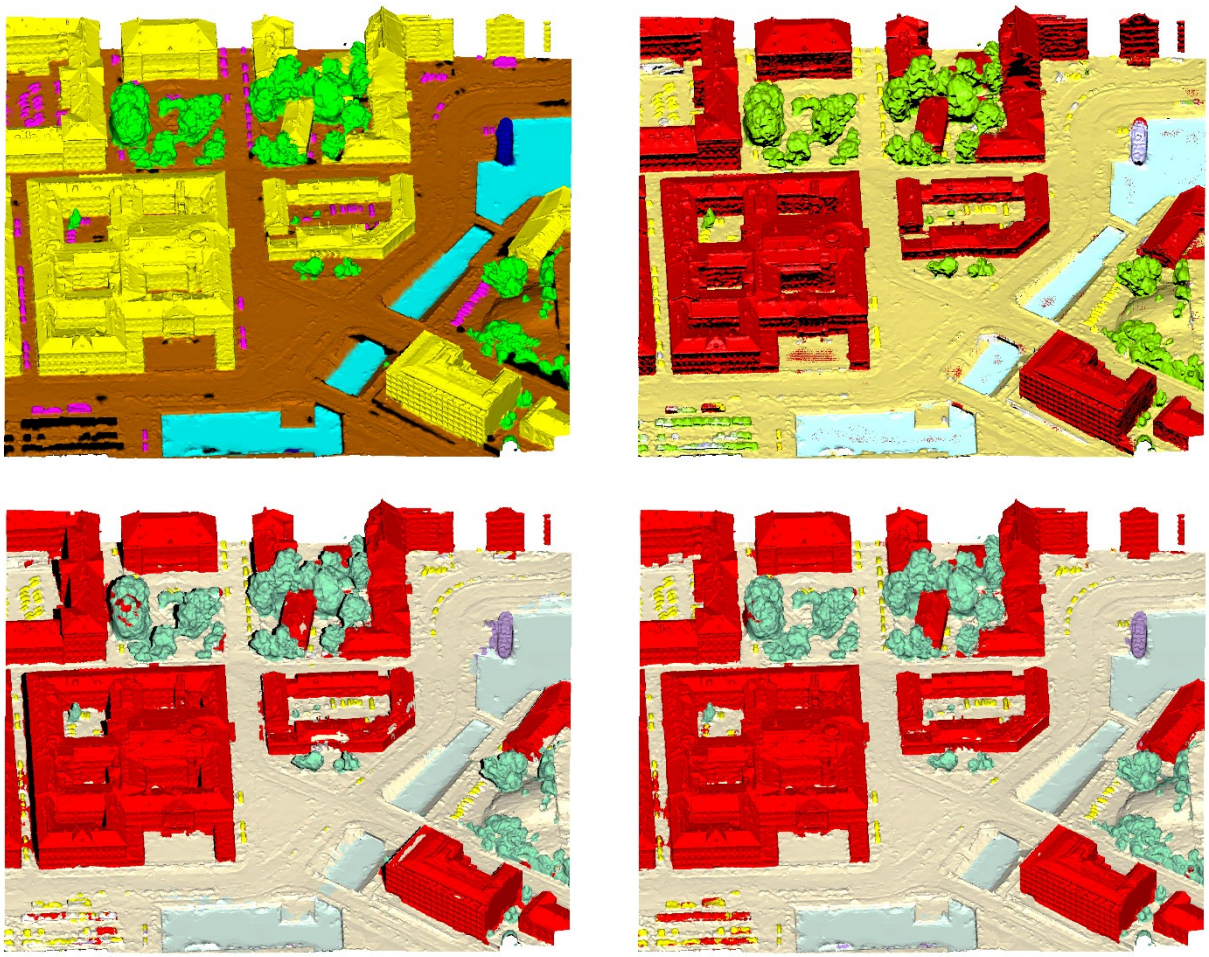

Figure 5. One of the SUM Benchmark's test tiles, with the texture generated from reference data (top-left), point cloud classified with OPEGIEKA'S CNN (top-right), results from semantic segmentation of single image - occluded areas are black (bottom-left), results of merging classification from many oblique images (bottom-right).

comparable with the top results achieved in SUM Benchmark experiments (Gao et. al 2021).

Merging classification from many images and voting in UV coordinates of texture space improved achieved results (Tab. 5), OA increase up to 89,9%. For the calculation of evaluation metrics, texture maps generated from 3D mesh tiles were used. Texture maps were created for each of 12 test 3D mesh tiles using classification stored in the labelled mesh tiles, textures were generated with a resolution of the original RGB texture. Per-class comparison of results achieved before and after voting show improvement of F1 and IoU values for all classes with exception of the boat class. The highest improvement is visible for water class.

Class	Precision	Recall	F1	IoU
Ground	86.7%	90.2%	88.4%	79.2%
Vegetation	89.5%	90.7%	90.1%	82.0%
Building	95.5%	94.2%	94.9%	90.2%
Water	97.7%	78.9%	87.3%	77.4%
Car	43.8%	92.2%	59.4%	42.2%
Boat	12.0%	92.3%	21.2%	11.8%
Mean	70.9%	89.8%	73.5%	63.8%

Table 5. SUM benchmark dataset. Per-class results of image segmentation after texturing and voting.

The qualitative evaluation for one of the test mesh tiles from SUM Benchmark is shown in Fig. 5. Results of classification of the point cloud with OPEGIEKA's CNN show the high completeness and quality of classification seldomly part of the object are wrongly labelled. However, some errors of classification are visible on the ground and water – portions of the points are wrongly classified as a ground – the effect is similar to the noise. Misclassification errors are mostly visible on small-scale objects.

Comparison of the results from semantic segmentation of a single image and achieved after merging classes from many oblique images (by voting in texture space) show (Fig. 5) improvement mostly with the classification of the small parts of the object which were wrongly classified on the of a single image. But still even after voting some errors of misclassification between water and ground or trees and buildings are visible.

Finally, from both – the qualitative evaluation (Fig. 5) as well as from the mean of evaluation metrics (Tab. 6), it is clearly visible that point cloud classification with OPEGIEKA's CNN provides slightly better results that semantic segmentation of oblique images with PSP-Net. That result could be easily explained by the types of classes that are proposed in SUM Benchmark. All semantic classes used in SUM Benchmark are similar to classes present in ALS data classification and are possible to distinguish only with geometry information from a point cloud. Therefore

solution that was developed especially for the classification of this type of data provided slightly better results. On the other hand, image segmentation was limited to oblique images, when a full set of hybrid data should also contain nadir images with a NIR channel, using both of them might greatly improve results of image segmentation.

	Point cloud	Oblique images	Texture voting (images)
mIoU	69.0%	60.3%	63,8%
OA	91.4%	84.1%	89,9%
mAcc	75.1%	81.4%	89,8%
mF1	77.7%	72.0%	73,5%

Table 6. SUM benchmark dataset – comparison of results from different methods.

It is also noteworthy that evaluation metrics for each of the datasets compared in Tab. 6 are calculated in different spaces. Point cloud comparison was performed in point by point manner, for oblique images pixel by pixel in image space calculation was used and for texture voting pixel by pixel (texel by texel) in texture space (UV texture map). What more reference data in SUM benchmark were created by mesh segments and then stored in another space – defined by faces of the 3D mesh.

4. CONCLUSIONS

We have compared two different approaches to semantic classification of the urban 3D mesh, which are focused on classifying the input data (oblique aerial images or point clouds), using deep learning methods instead of segmentation 3D mesh itself. Results achieved with data from SUM Benchmark show that both methods provide good results when an appropriate set of training data is available. Results from Bordeaux dataset shown that data augmentation cannot replace diverse training dataset in case of image segmentation. However, results achieved with images rendered from SUM Bencharks's 3D meshes could be biased by higher homogeneity of renders than it is possible to achieve in the case of oblique images in a real-world scenario.

Overall results for classification of different types of data show that point cloud classification provides slightly better results than semantic segmentation of oblique aerial images) at least as long as all classes can be distinguished based on geometry information from a point cloud. For future work, we plan to evaluate both methods on different real-world datasets. We will also investigate possibilities of multi-modal information transfer between imagery, point clouds, and meshes – that problem was already highlighted by Laupheimer & Haala (2021), for merging classification from different sources.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Hexagon (Leica Geosystems) for providing the Bordeaux CityMapper-1 dataset.

REFERENCES

Bacher, U., 2021. 3D content generation using hybrid aerial sensor data. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 43, 297–303. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-297-2021

Blaha, M., Rothermel, M., Oswald, M.R., Sattler, T., Richard, A., Wegner, J.D., Pollefeys, M., Schindler, K., 2017. Semantically Informed Multiview Surface Refinement. ICCV 2, 3–8.

Cozzi, P. 2021, Introducing 3D Tiles Next, Streaming Geospatial to the Metaverse. Cesium blog 10 November 2021 https://cesium.com/blog/2021/11/10/introducing-3d-tiles-next/ (15 January 2021)

Deng, J., Dong, W., Socher, R., Li, L. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 248– 255. https://doi.org/10.1109/CVPR.2009.5206848

Dominik, W., Bożyczko, M., Tułacz-Maziarz, K., 2021. Deep learning for automatic lidar point cloud processing. Archives of Photogrammetry, Cartography and Remote Sensing, 33, 1-10.

Frommholz, D., Linkiewicz, M., Poznanska, A.M., 2016. Inlining 3D reconstruction, multi-source texture mapping and semantic analysis using oblique aerial imagery. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLI, 605–612.

Gao, W., Nan, L., Boom, B., Ledoux, H., 2021. SUM: A benchmark dataset of Semantic Urban Meshes. ISPRS J. Photogramm. Remote Sens. 179, 108–120. https://doi.org/10.1016/j.isprsjprs.2021.07.008

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., Garcia-Rodriguez, J. (2018). A survey on deep learning techniques for image and video semantic segmentation. Applied Soft Computing, 70, 41–65. https://doi.org/10.1016/J.ASOC.2018.05.018

Haala, N., Rothermel, M., Cavegn, S., 2015. Extracting 3D urban models from oblique aerial images, in: 2015 Joint Urban Remote Sensing Event (JURSE). IEEE, pp. 1–4. https://doi.org/10.1109/JURSE.2015.7120479

He, K., Zhang, X., Ren, S., & Sun, J., 2015. Deep Residual Learning for Image Recognition. https://doi.org/10.48550/arxiv.1512.03385

Huang, S., 2019. Building segmentation in oblique aerial imagery. University of Twente.

Kingma, D., Ba, J. 2014. Adam: A Method for Stochastic Optimization. International Conference on Learning Representations. http://arxiv.org/abs/1412.6980

Laupheimer, D., Haala N., 2021. Juggling with representations: On the information transfer between imagery, point clouds, and meshes for multi-modal semantics. *ISPRS Journal of Photogrammetry and Remote Sensing*, 176, 55-68.

Lehner, H., Dorffner, L., 2020. Digital geoTwin Vienna: Towards a Digital Twin City as Geodata Hub. PFG – J. Photogramm. Remote Sens. Geoinf. Sci. 88, 63–75. https://doi.org/10.1007/s41064-020-00101-4

Liu, L.I., 2019. Semantic segmentation of urban airborne oblique images. University of Twente.

OPEGIEKA, 2020. Refine LiDAR classification using deep learning algorithms, 19 June 2020. https://lab.opegieka.pl/refine-lidar-classification-using-deep-learning-algorithms,30,en

Rong, M., Shen, S., Hu, Z., 2021. 3D Semantic Labeling of Photogrammetry Meshes Based on Active Learning, in: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, pp. 3550–3557. https://doi.org/10.1109/ICPR48806.2021.9412358

Rouhani, M., Lafarge, F., & Alliez, P. (2017). Semantic segmentation of 3D textured meshes for urban scene analysis. ISPRS Journal of Photogrammetry and Remote Sensing, 123, 124–139. https://doi.org/10.1016/j.isprsjprs.2016.12.001

Toschi, I., Remondino, F., Rothe, R., Klimek, K., 2018. Combining airborne oblique camera and LiDAR sensors: Investigation and new perspectives. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 42, 437–444. https://doi.org/10.5194/isprs-archives-XLII-1-437-2018

Tymchyszyn, V. B., Khlevniuk, A. V., 2019. *Beginner's guide to mapping simplexes affinely*, Bogolyubov Institute for Theoretical Physics.

Yakubovskiy, P. (2019). *Segmentation Models*. GitHub repository. https://github.com/qubvel/segmentation_models

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J. (2017). Pyramid scene parsing network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2881–2890. https://doi.org/10.1109/CVPR.2017.660