DEEP LEARNING IN HISTORICAL GEOGRAPHY

Jakub Vynikal*[,1,2], Jan Pacina[1,2]

*jakub.vynikal@fsv.cvut.cz
[1]Department of Geomatics, Faculty of Civil Engineering, Czech Technical University in Prague, Prague, Czech Republic.
[2]Department of Geoinformatics, Faculty of Environment, J. E. Purkyně University in Ústí nad Labem, Ústí nad Labem, Czech Republic

Abstract
In relation to the rapid development of artificial intelligence, the possibilities of automatic processing of spatial data are increasing. Scanned topographical maps are a valued source of historical information. Neural networks allow us to extract information quickly and efficiently from such data, eliminating the difficult and repetitive work that would otherwise have to be done by a human. The article presents two case studies exploring the possibilities of using deep learning in historical geography. The first one is concerned with detecting and extracting swamps from topographic maps, while the second one attempts to automatically vectorize contours from the State Map 1 : 5 000.

Keywords
Deep learning, historical geography, segmentation, vectorization, scanned maps

1 INTRODUCTION

Scanned topographical maps are an important source of information about various spatial phenomena, including land use and land cover, altitude, historic human settlements, and many more. These maps were created before the onset of GIS, and they are mostly available as physical printed assets. These can be scanned to a raster form. It is generally desirable to generate vector features for further processing. Features are generally manually traced in point, line and polygon form, and associated with corresponding legend description based on their attributes.

Although previous research has explored utilization of deep learning in remote sensing field extensively, a comprehensive review of the available literature indicates a noticeable scarcity of studies dedicated to the extraction of information from topographic maps and other cartographic representations. Indeed, the application of machine learning (ML) or deep learning (DL) techniques for the extraction of features from topographic maps remains relatively underrepresented in current research.

In their research, Li et al. [1] delved into the realm of text recognition on topographic maps using deep learning (DL). Simultaneously, Uhl et al. [2] investigated the utilization of convolutional neural networks (CNNs) for the extraction of human settlements. Lui et al. [3] explored the application of CNNs for the broad segmentation of topographic maps. Maxwell et al. [4] used a modified UNet neural network to delineate surface mine extents from topographic maps. Vectorization attempts of old cadastral maps with various edge detectors were conducted by Chen et al. [5]. An extensive thesis was written by Petitpierre [6] about semantic segmentation of historical city maps. Symbol extraction of fixed size along with coarse segmentation of regions indicated by them was explored by Smith and Pillat [7] using a modified YOLO neural network. Dolejš et al. [8] exploited precise digital terrain model (DMR5G) to detect aerial bombing craters with CNN method.

In this paper, we present two case studies based on and furthering the listed research. The first case study describes the detection and delineation of swamps on historic TM10 maps by using unorthodox object detection solution, instead of conventional semantic segmentation, due to the peculiar nature of swamps signs on these maps. Swamps are illustrated as horizontal blue lines on various backgrounds. Areas of land use and land cover (LULC) are usually extracted by semantic segmentation, even in historical maps, if the classes have distinct spectral characteristics [9]. In this case however, this approach was replaced by object detection, namely SSD (Single-Shot Detector) neural network.

The second case study concerns automatic vectorization of elevation contours from SMO-5 maps using deep edge filtering. Aside from manual vectorization, attempts at semi-automatic vectorization were made by Kratochvílová et al. [10] on the same maps. The SMO-5 maps were converted the SMO-5 raster map to a binary space, only distinguishing contours and background. Small groups of connected pixels were then erased and contours were vectorized in ESRI ArcScan. This approach certainly increased automation a lot, but much of manual cleaning and correcting of problematic places was still required. Moreover, annotations of contours created many gaps in the final vector layer, which had to be dealt with (using ETGeoWizards software), and the

DOI 10.13164/juniorstav.2024.24097

height of the contour had to be added manually to the attribute. This reconstructed elevation model was used in the pre-dam valley reconstruction of the Vltava river [11], where accuracy of the approach and the existence of other applicable maps was discussed. Methods based on deep learning should further increase the automation. Based on the research by Chen et al. [5], numerous deep edge filters were trained to semantically segment the contours. The three neural networks trained were HED (Holistically Edge Detector), BDCN (Bi-Directional Cascade Network for Perceptual Edge Detection), and UNet, named after its characteristic U shape.

2 METHODOLOGY

ArcGIS Pro, which was used throughout the entirety of the process, supports various deep learning libraries like TensorFlow, PyTorch, cuDNN and others. These libraries are integrated into ArcGIS Pro interface and Python API. The general workflow for deep learning detection in ArcGIS Pro is as follows : collecting training samples, exporting them in a suitable format, training a deep learning model, and applying the model to new data for detection.

Swamp detection

TM10 topographic military maps in 1 : 10,000 scale were initially created between 1957 and 1971, spanning the entire territory of the Czech Republic. Their content includes land cover, elevation contours, watercourses and water surfaces, human settlements, important landmarks, and others. As depicted in Fig. 1, swamps are illustrated with blue horizontal lines. These are to be automatically detected and delineated in polygons for an environmental analysis.

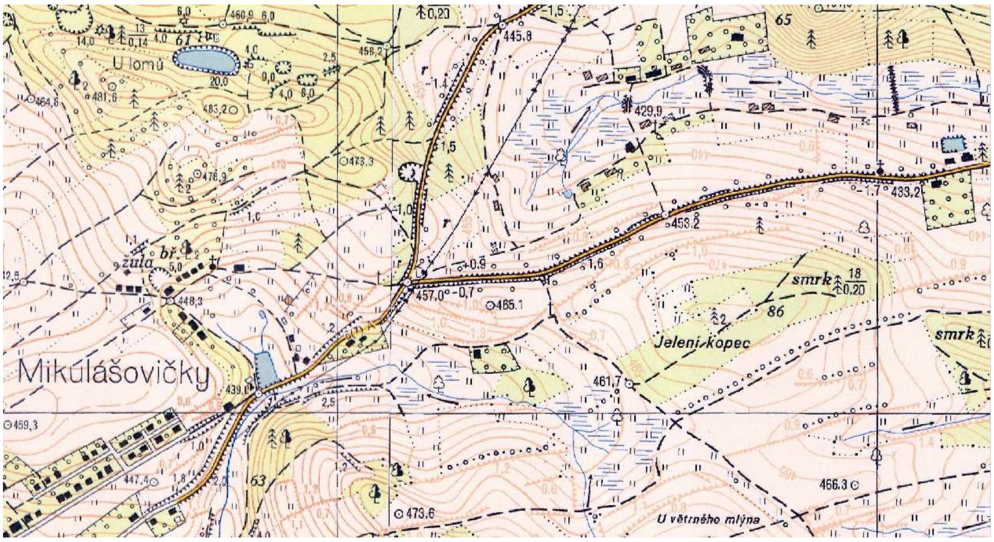

Fig. 1 An excerpt of TM10 maps with abundant swamp signs.

To prepare the maps for further work, it is necessary to place the individual rasters in a coordinate system. Aside from České Švýcarsko, which is the area of interest, two other similarly large areas were selected to serve as a source of training data. These are described in Tab. 1. The two training areas were chosen specifically because of highly occurrent swamps.

Tab. 1 TM10 areas used for training and evaluation of the model.

	N. of map sheets	**Swamp area [km^2]**	**Area [km^2]**	**Pct. of swamps [%]**
Třeboňsko	*32*	*16.9*	*677.2*	2.5
Podyjí	*15*	*6.7*	*318.9*	2.1
České Švýcarsko	*33*	*6.8*	671.7	1.0

The rasters were referenced using Projective Transformation Method in ArcGIS Pro, with corners of the map sheets serving as identical points. The rasters were then mosaicked to eliminate parts of maps beyond the

DOI 10.13164/juniorstav.2024.24097

frame. The raster mosaic dataset could be used for the vectorization, but ArcGIS Pro usually crashes with such a big mosaic. Therefore the three mosaics were exported as three big seamless rasters.

The swamps were manually delineated on the training rasters. This polygon layer then served to export image chips to train the model. The size of the image chips was determined based on the average length of a swamp line. The chip size was chosen 30 × 30 pixels, representing an area of about 32.5 × 32.5 meters. Because we want to detect only one type of symbol, no other class was defined. A similar process would work with any other map symbol, most likely with better success in the case of scale and rotation invariant symbols. In our case, there are differences in spacing, length, and shade of blue lines between various samples. Because of the variance, relatively big amount of samples is required. The swamps are always horizontal, which makes rotating the chips for data augmentation irrelevant. Augmentation was done with striding – chips were overlapped every 10 pixels, roughly tripling the the amount of chips in every dimension. Overall, around half a million chips were exported.

The model was then trained, with 10% samples set aside for validation. ResNet-50 was used as the backbone. The training (process) took approximately 27 hours.

Contour vectorization

SMO-5 maps (Fig. 2) was published gradually from 1950 in various editions. The maps used here are the oldest available edition from the 1950s, without dams built in the following years. Their contents are less detailed than in the TM10 maps, consisting only of cadaster maps, altimetry contours and some miscellaneous content. The color scale is also poorer, with only three colors being used, black for planimetry, brown for altimetry, and beige for the background. The contours are spectrally distinct, regularly interrupted with height annotations.

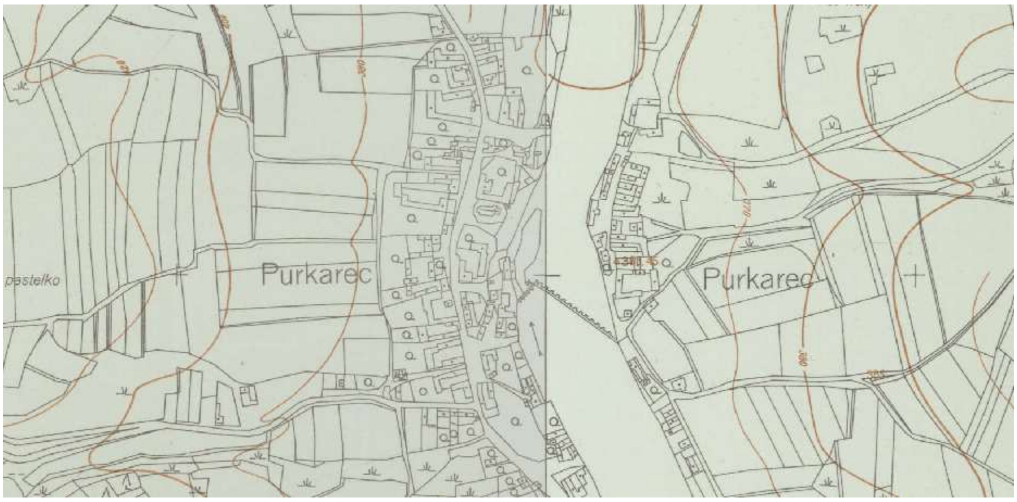

Fig. 2 A snippet of a SMO-5 first edition map.

The contours from the previous Vltava project were used to generate the training data. From their line layer, a buffer zone of one meter wide on each side was created. These polygons were then rasterized and exported as 500 × 500 pixel images with 50% stride and variable 180° rotation to augment the training set. For segmentation purposes, the training set always consists of a pair of images with the same reference: one original raster and the other a segmented binary bitmap (is contour/is not contour). This data was generated from 39 map sheets, resulting in 33,954 image chips in JPG format.

Three neural networks were trained, namely HED (Holistically Edge Detector), BDCN (Bi-Directional Cascade Network for Perceptual Edge Detection), and UNET. Ten per cent of the training set was used for validation, VGG16 was used as the backbone model. The training took six hours in the case of HED, and 44 hours in the case of BDCN.

3 RESULTS

After successful training, the inference took place. The models were put to use on new and unseen data; swamps were detected in the national park and contours were segmented on 22 new map sheets.

DOI 10.13164/juniorstav.2024.24097

The inference on national park swamps took about 90 minutes. Every instance of a swamp was delineated with a bounding box. These boxes overlap, so it is possible to merge them into a polygon layer later, encompassing all the swamp area.

Fig. 3 Succesfully detected swamps.

In the case of contours, this process takes only about 10 minutes. At this point it is basically a semantic segmentation of the map. The BDCN model came out as the best one as it generalized very well on the new data and was able to cope even with contour annotations (Fig. 3 left). One of the problems was contours densely stacked side by side (Kratochvílová et al. [10] mentioned it too). The HED model performed significantly worse with large gaps. The UNET model performed well, but with gaps in unexpected places (Fig. 4 right).

Fig. 4 Results of BDCN (left), HED (middle) and UNET (right) networks.

4 DISCUSSION

While both use-cases demonstrate a level of success, there are many things to be improved.

Firstly, the swamp detection model indicates many false positives, mainly along watercourses spanning in east-west directions, mimicking the swamp appearance. The resulting polygons were limited by their probability to at least 90% to filter out most of them, but some remained. One of the possible causes of suboptimal performance is the percentage of false positive training chips, which were created from the edges of training polygons. Only later it was found that many of them do not contain any swamp signs at all. To optimize the algorithm further, it is suitable to explore other deep learning solutions.

The result of contour edge detection is a segmented binary raster, similar to preceding research [10]. The binary raster can be converted to vector lines just like before, with some notable improvements. Above all else, the lines from the BDCN model are rarely interrupted, even in places of inserted annotations. This makes the vectorization procedure much easier and automatic. The connections of contours between map sheets are also smoother, even though not perfect. The binary raster can be thinned and converted to polylines. There are some resulting topological errors to be fixed.

5 CONCLUSION

This article showcased the possibilities of deep learning uses in historical geography. The demonstrated ones include symbol detection and contour vectorization. The models were trained and applied with moderate success, encouraging future development.

To automate the contour processing further, contours annotations are to be optically recognized and placed to the attribute of the newly created vector elements. This is a difficult task, as varying rotation introduces uncertainties such as ambiguity between 6 and 9. Along with variable contour interval (5-10-20 meters in different map sheets), this poses a challenge for a complete solution of automatic elevation model creation.

Acknowledgements

This research was funded by the Student Grant Competition of Czech Technical University in Prague, grant No. SGS23/050/OHK1/1T/11, and TAČR project (no. SS05010090) “Water in the region of Czech Switzerland” and a Ministry of Culture of the Czech Republic NAKI programme (o. DH23P03OVV055) “Vltava 2 – transformation of historical landscape, river as a connection and barrier”.

References

[1] LI, Huali, LIU, Jun and ZHOU, Xiran. Intelligent map reader: A framework for topographic map understanding with Deep Learning and Gazetteer. *IEEE Access* [online]. 2018. Vol. 6, pp. 25363–25376. DOI 10.1109/access.2018.2823501

[2] UHL, Johannes. H., LEYK, Stefan., YAO-YI Chiang, WEIWEI Duan and KNOBLOCK, Craig A. Extracting human settlement footprint from Historical Topographic map series using context-based machine learning. *8th International Conference of Pattern Recognition Systems (ICPRS 2017)* [online]. 2017. DOI 10.1049/cp.2017.0144

[3] LIU, Tiange, MIAO, Qiguang, XU, Pengfei and ZHANG, Shihui. Superpixel-based shallow convolutional neural network (SSCNN) for scanned topographic map segmentation. *Remote Sensing* [online]. 2020. Vol. 12, no. 20, p. 3421. DOI 10.3390/rs12203421

[4] MAXWELL, Aaron, BESTER, Michelle, GUILLEN, Luis, RAMEZAN, Christopher, CARPINELLO, Dennis, et al. Semantic segmentation deep learning for extracting surface mine extents from historic topographic maps. *Remote Sensing* [online]. 2020. Vol. 12, no. 24, p. 4145. DOI 10.3390/rs12244145.

[5] CHEN, Yizi, CARLINET, Edwin, CHAZALON, Joseph, MALLET, Clément, DUMENIEU, Bertrand, et al. Vectorization of historical maps using deep edge filtering and closed shape extraction. In: *International conference on document analysis and recognition* [online]. Cham: Springer International Publishing, 2021. pp. 510–525. DOI 10.1007/978-3-030-86337-1_34.

[6] PETITPIERRE, Rémi. Neural networks for semantic segmentation of historical city maps: Cross-cultural performance and the impact of figurative diversity. *arXiv preprint arXiv:2101.12478* [online]. 2021. DOI 10.13140/RG.2.2.10973.64484.

[7] SMITH, William A. and PILLATT, Toby. You only look for a symbol once: An object detector for symbols and regions in documents. *Lecture Notes in Computer Science* [online]. 2023. pp. 227–243. DOI 10.1007/978-3-031-41734-4_14

[8] DOLEJŠ, Martin, PACINA, Jan, VESELÝ, Martin and BRÉTT, Dominik. Aerial bombing crater identification: Exploitation of precise digital terrain models. *ISPRS International Journal of Geo-Information* [online]. 2020. Vol. 9, no. 12, p. 713. DOI 10.3390/ijgi9120713

[9] PROCHÁZKA, Matěj. Utilization of advanced classification methods for automatic element extraction from maps. thesis. Department of Geoinformatics, Faculty of Environment, J. E. Purkyně University in Ústí nad Labem (UJEP), 2022 [online] https://portal.ujep.cz/StagPortletsJSR168/PagesDispatcherServlet?pp_destElement=%23ssSouboryStudentuDivId_6230&pp_locale=cs&pp_reqType=render&pp_portlet=souboryStudentuPagesPortlet&pp_page=souboryStudentuDownloadPage&pp_nameSpace=G232335&soubidno=127386.

[10] KRATOCHVÍLOVÁ, Darina and CAJTHAML, Jiří, 2020, Using the automatic vectorisation method in generating the vector altimetry of the historical Vltava River Valley. *Acta Polytechnica* [online]. 2020. Vol. 60, no. 4, pp. 303–312. DOI 10.14311/ap.2020.60.0303

[11] PACINA, Jan, CAJTHAML, Jiří, KRATOCHVÍLOVÁ, Darina, POPELKA, Jan, DVOŘÁK, Vojtěch et al. Pre-Dam Valley Reconstruction based on archival spatial data sources: Methods, accuracy, and 3D printing possibilities. *Transactions in GIS* [online]. 2021. Vol. 26, no. 1, pp. 385–420. DOI 10.1111/tgis.12854