Leveraging uncertainty estimation and spatial pyramid pooling for extracting hydrological features from scanned historical topographic maps

Sidi Wu, Magnus Heitzler, and Lorenz Hurni

Department of Civil, Environmental and Geomatic Engineering, Institute of Cartography and Geoinformation, ETH Zurich, Zurich, Switzerland

ABSTRACT
Historical maps are almost the exclusive source to trace back the characteristics of earth before modern earth observation techniques came into being. Processing historical maps is challenging due to the factors such as diverse designs and scales, or inherent noise from painting, aging, and scanning. Our paper is the first to leverage uncertainty estimation under the framework of Bayesian deep learning to model noise inherent in maps for semantic segmentation of hydrological features from scanned topographic historical maps. To distinguish different features with similar symbolization, we integrate atrous spatial pyramid pooling (ASPP) to incorporate multi-scale contextual information. In total, our algorithm yields predictions with an average dice coefficient of 0.827, improving the performance of a simple U-Net by 26%. Our algorithm outputs intuitively interpretable pixel-wise uncertainty maps that capture uncertainty in object boundaries, noise from drawing, aging, and scanning, as well as out-of-distribution designs. We can use the predicted uncertainty potentially to refine segmentation results, locate rare designs, and select reliable features for future GIS analyses.

INTRODUCTION
Historical maps serve as useful resources to depict natural and human-induced objects and their changes on earth (Chiang, Leyk, and Knoblock 2014), especially before modern earth observation techniques came into being. The spatio-temporal dynamics of objects on earth can be unlocked from historical maps at a fine resolution over large spatial and temporal extents. Manually retrieving information from historical maps can be extremely time-consuming depending on the size and complexity of the maps. For example, manual digitization takes around 1 hour and 20 minutes for roads from a USGS Topographic Map of 2283 × 2608 pixels in size at the scale of 1:62,500 (Chiang, Leyk, and Knoblock 2013) and approximately 4 hours for forest areas from a Siegfried map of 7000 × 4800 pixels in size at the scale of 1:25,000 (Leyk and Boesch 2009). The increased availability of tremendous scanned historical maps in digital archives makes time-consuming manual digitization nearly impractical. Recognizing and extracting geographical features from historical maps automatically, nevertheless, is challenging because:

First, maps have diverse designs and scales and contain different types of symbolic representations (singular symbols, line symbols, area symbols, texts), which causes dispersion of specially tailored approaches and poses challenges on finding generic and robust solutions.

Second, scanned historical maps often suffer from graphical flaws inherent in the original paper maps (e.g. defects in manual drawing and fading papers) and the scanning process (e.g. low scanning resolution and scanning artifacts). This might impede the detection process and introduce subjectivity into further interpretation.

Third, the symbolic representations of different features can be similar in terms of colors, shapes, or textures, which can confuse the recognition process, especially with a small detection window size. has illustrated the graphic similarity between depth contour lines in (a) and streams in (b), as well as between lakes in (c) and rivers in (d). Small detection windows of 100 m in size, within which distinguishing rivers from lakes is nearly impossible, are marked by red squares. All the map sheets illustrated in our paper are oriented North-up.

CONTACT Sidi Wu sidiwu@ethz.ch

© 2022 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Fourth, typical geographical features like streams are sparse as they usually occupy only a small proportion of an entire map sheet. This leads to much redundancy when all the image pixels are searched for, and a high sensitivity of spatial accuracy when the spatial resolution alters in the extraction process (Chiang et al. 2020).

Fifth, the continuity of a geographical feature is interrupted when overlaid by other features (e.g. streams can be interrupted by roads), which requires proper approaches to trace the whole feature in the object-oriented extraction process. Figure 1

Digital map processing techniques have been studied at the intersection of cartography, geographical information science, computer vision, and pattern recognition. A digital map processing workflow (Chiang, Leyk, and Knoblock 2014) usually includes segmentation of different feature layers by identifying color-homogeneous regions based on different color prototypes, extracting features in different symbolic representations using methods like template matching, shape descriptors, morphological operators, and georeferencing maps. Despite plenty of studies on extracting roads (Chiang and Knoblock 2009; Chiang, Leyk, and Knoblock 2011; Chiang and Knoblock 2013; Liu 2002), vegetation (Leyk and Boesch 2009; Dhar and Chanda 2006; Leyk, Boesch, and Weibel 2006), texts (Dhar and Chanda 2006; Pezeshk and Tutwiler 2010, 2011), buildings (Liu 2002) and contours (Pezeshk and Tutwiler 2008; Samet and Hancer 2012; Miao et al. 2012), highly automatic approaches are relatively rare or generate only coarse results. In fact, those studies were often tediously tailored to certain map series or features with carefully crafted algorithms and manually fine-tuned parameters (Dhar and Chanda 2006; Bin and Cheong 1998; Leyk and Boesch 2009), or involved user interventions like manual annotations or corrections (Khotanzad and Zink 2003; Samet and Hancer 2012; Chiang, Leyk, and Knoblock 2011; Chiang and Knoblock 2013). Some of those tediously tailored or semi-automatic methods have achieved remarkable accuracy for the given maps, for example, the average completeness and correctness reported by Chiang, Leyk, and Knoblock (2013) are both above 95% for road extraction; the average accuracy reported by Leyk and Boesch (2009) is 98% for forest extraction tested on three map pages. However, those metrics are often hardly comparable due to multiple factors, such as specifics of the operated maps and features. In addition, the costs of those algorithms with respect to manual efforts, time consumption, and the generalizability to other features or maps cannot be neglected. Notable efforts of automatic waterbody segmentation can be seen from the works of Leyk (2009), Leyk and Boesch (2010), and Iosifescu, Tsurini, and Hurni (2016) which focus on refining color-based segmentation. However, these color-based methods are not robust enough against color variations, mixed/false colors, and map noise (Leyk and Boesch 2010). In addition, none of those studies has stepped further into classifying different hydrological features from the segmented layer.

Recent advances in the field of computer vision have shifted the digital map processing research toward deep-learning-based methods, aiming at a highly automated process to reduce manual labor significantly, a generalizable process to deal with different symbolic representations and maps of diverse designs and scales, as well as a robust process to

**Figure 1.** Examples of similar symbolic representations. The first pair is between depth contour lines in (a) and streams in (b), both from old national maps at the scale of 1:25,000 in 1976. The second pair is between lakes in (c) and rivers in (d), both from Siegfried maps at the scale of 1:25,000 in 1885 and 1876, respectively. The red squares are examples of small detection windows of 100 m in size, within which distinguishing rivers and lakes is nearly impossible. We re-scale maps for illustration. The scale bar of 100 m is shown at the corner for each cropped map.
handle maps of low graphic quality. Uhl et al. (2017) and Uhl et al. (2018, 2020) used LeNet (LeCun et al. 1989) to segment individual buildings and urban areas. Heitzler and Hurni (2020) segmented individual buildings with U-Net (Ronneberger, Fischer, and Brox 2015) and automatically vectorized the building footprints afterward. However, all these approaches are susceptible to false positives of symbols with similar appearances to buildings (black spots), like borders, roads, texts, and railways (Heitzler and Hurni 2020). To distinguish symbols of similar appearances considering the graphic similarity between rivers and lakes specifically in our case, we incorporate multi-scale contexts in the belief that the local context implies details, whereas the global context summarizes the content of features. In addition, information gathered in a larger spatial context can help to improve the continuity of predictions, especially for overprinted features. Like the work of He et al. (2019), we integrate atrous spatial pyramid pooling (ASPP) (Chen et al. 2017) into U-Net to capture multi-scale information of map features.

As has been pointed out by Guo et al. (2017), although being trained successfully, modern neural networks might not be well calibrated for their predictions. As a result, the networks often yield incorrect predictions, even though the associated classification probability is high. Modeling uncertainty can encourage the network to identify peculiar situations where the predictions might be unreliable, for example, when an exotic input is given. Considering inevitable graphic flaws in scanned historical maps, modeling uncertainty is especially useful for us to interpret, correct, and use the extracted information conditionally. Data-dependent uncertainty, known as aleatoric uncertainty (Küreghian and Ditlevsen 2009; Gal 2016; Kendall and Gal 2017), has been researched in the field of computer vision to model observation noise inherent in data samples, for example object boundaries (Gal 2016), occlusions (Gal 2016), far objects (Gal 2016), and noisy labels (Hu, Sclaroff, and Saenko 2020). By learning uncertainty in the framework of Bayesian deep learning (Gal 2016; Kendall and Gal 2017; Cipolla, Gal, and Kendall 2018; Gurevich and Stuke 2018; DeVries and Taylor 2018a; Hu, Sclaroff, and Saenko 2020), the model is enabled to recognize and indicate observation noise while at the same time it reduces the impacts of those noisy samples during training to achieve better performance. Works like Kendall and Gal (2017), Cipolla, Gal, and Kendall (2018), and Gurevich and Stuke (2018) have represented uncertainty in a probability distribution (e.g. Gaussian) over the model output, whereas the work of DeVries and Taylor (2018a) encouraged confidence estimation by interpolating between the targeted and the predicted distribution, with the estimated confidence controlling the interpolation degree. The confidence estimates in their approach show better performance in identifying failed segmentation, comparing with the uncertainty modeled with Gaussian distributions (DeVries and Taylor 2018b). In our work, we apply the concept of DeVries and Taylor (2018a, 2018b) into segmenting features from historical maps.

In our work, we target at semantic segmentation of hydrological features (rivers, lakes, streams, and wetlands) from scanned historical maps, as they were rarely addressed in past research. However, the framework we propose is general enough to deal with common challenges in digital map processing and thus can be used to support other segmentation tasks (e.g. buildings, roads, and vegetation) from scanned historical maps. The main contributions of our paper are two-fold:

First, we leverage uncertainty estimation in our segmentation task to capture noise inherent in maps and reduce their impacts during training. Apart from semantic segmentation, our model yields pixel-wise uncertainty maps at the same time, to indicate the reliability of segmentation. The two tasks interact through an uncertainty-aware dice loss function. We incorporate ASPP that involves multi-scale information to distinguish features with similar symbolization (e.g. rivers and lakes), and to improve continuity of extracted features.

Second, we investigate potential uses of the predicted uncertainty. The predicted uncertainty acts as a reference for us to interpret, correct, and use the extracted features conditionally from scanned historical maps for future GIS analyses.

Materials and methods

Data preparation

In our experiment, we make use of the Topographic Atlas of Switzerland (“Siegfried map”), a Swiss national map series that was published between 1870 and
1949. Its sheets were scanned and stored in a digital archive by the Federal Office of Topography Swisstopo, the Swiss national mapping agency, and georeferenced using the methodology proposed by Heitzler et al. (2018). In total, the Siegfried map series contains 4079 map sheets, each of which is 7000 pixels wide and 4800 pixels high. 3192 sheets of scale 1:25,000 were scanned with a spatial resolution of 1.25 m/pixel, mainly covering the Swiss plateau; 887 sheets of the scale 1:50,000 were scanned with a resolution of 2.5 m/pixel, mainly covering the Swiss Alps. In our research, we vectorized 439 map sheets of scale 1:25,000 around the year 1880 semi-automatically for four feature classes: streams (lines), rivers (polygons), wetlands (polygons), and lakes (polygons).

We convert the vectorized features into raster maps as ground-truth annotations. For each map sheet, the corresponding pixel-wise annotation map has four binary channels, each corresponding to the existence of one of the mentioned feature classes. For line features (streams), we created line buffers of 2 m to approach the average thickness of stream symbols before rasterization. For polygon features (rivers, wetlands, lakes, and stream buffers), we convert them into raster using GDAL (GDAL/OGR contributors 2021). We split the 439 map sheets into training (351), validation (44), and testing (44). Due to the memory limit, we sampled both map sheets and the annotation maps into small tiles of 256 × 256 pixels (320 m × 320 m). Considering the fact that the features of interest usually only occupy small areas of the whole map sheets, sampling sheets by regular grids might lead to an overwhelming percentage of negative examples. This could make the model biased toward negative samples in the learning phase. As an alternative, we created a buffer zone of 160 m (half of the window size) around each vector feature and randomly sampled points within those buffer zones.

We sampled roughly the same number of points for each feature class. Thereafter, we cropped map sheets as well as annotations around those sample points. This resulted in 31,911 training samples and 3387 validation samples. For testing, we divided the whole map sheets into regular grids of 256 × 256 pixels.

Our network structure is illustrated in Fig. 2. The network structure. We integrate ASPP into our adapted U-Net for semantic segmentation and add uncertainty estimation as an additional task. The two tasks interact through an uncertainty-aware dice loss function.

**U-Net + ASPP**

Proposed by Ronneberger, Fischer, and Brox (2015), U-Net is a light, effective, and widely used Encoder-Decoder structure for image segmentation. It extracts features through the encoder and restores the original resolution through the decoder with skip connections to gather information from lower layers in the
encoder. By comparison, a normal fully convolutional network (FCN) cannot restore the resolution of the input, and a normal Encoder-Decoder architecture does not have skip connections that pass low-level information to the decoder. Figure 2

To consider multi-scale contextual information, we integrate ASPP after the bottleneck of U-Net. ASPP employs parallel atrous convolution with different rates to effectively capture information at different Field of Views (FoVs) without increasing the kernel parameters. It’s equivalent to down-sampling the images by the factor of the atrous convolution rate, performing standard convolution, and then recovering them to the original resolution. To reduce the computation complexity, Chen et al. (2018) proposed depth-wise separable convolution that performs atrous convolution independently for each input channel, followed by point-wise convolution to combine those outputs. The reasons why we don’t directly use the original Encoder-Decoder network DeepLabV3+ (Chen et al. 2018) where ASPP was first introduced are: first, DeepLabV3+ has a heavily loaded structure to handle daily scenes, which might be redundant for maps with sparse objects; second, the decoder of DeepLabV3+ only restores 1/4 of the input resolution, followed by a direct up-sampling process. This will cause delineation issues at object boundaries.

In our proposed network in, the encoder and decoder repetitively execute the convolutional blocks (in green). In each convolutional block, a $3 \times 3$ convolution layer, followed by a batch normalization layer and an ELU activation layer is repeated twice, with a drop-out layer in between. In the encoder, each convolutional block (except the last one before ASPP) is followed by a $2 \times 2$ max-pooling layer to half the size of feature maps. The features extracted by the encoder are passed to an ASPP module with a point-wise $1 \times 1$ convolution layer to capture the local information, three parallel $3 \times 3$ depth-wise separable atrous convolution layers with rates of 6,12 and 18, respectively, to capture multi-scale contexts, and a global average pooling layer to extract the whole image content. The outputs of the ASPP module are concatenated, followed by a $1 \times 1$ convolution layer, a batch normalization layer, an ELU activation layer, and a dropout layer, before being passed to the decoder. The convolutional block in the decoder is the same as in the encoder, and each is followed by an up-sampling layer to double the size of feature maps. Finally, the feature maps are passed to a $1 \times 1$ convolution layer with Sigmoid activation and cropped to half. We use Sigmoid activation instead of Softmax to allow for multi-class predictions when features intersect. For example, a stream can pass through a wetland. Cropping is necessary for our experiment since limited contexts for objects on the border might not be sufficient for the network to make reasonable predictions of those objects.

**Uncertainty estimation for semantic segmentation**

For many segmentation tasks, noise inevitably exists in the annotated pixels. For example, the annotated pixels could have inaccurate labels, or might not act as representative or meaningful samples for a neural network. Forcing the network to learn from noisy pixels could make the network skew toward noise and thus decrease the generalizability of the network. To increase the robustness of the network, we leverage uncertainty estimation based on Learned Confidence Estimates (LCE) proposed by DeVries and Taylor (2018a) to generate pixel-wise uncertainty maps. Similarly, we add uncertainty estimation as an additional task in parallel with semantic segmentation. The uncertainty and the segmentation accuracy are expected to be closely correlated based on the assumption that problematic segmentation should have less confidence. For uncertainty, we also use Sigmoid activation to yield a bounded value between 0 and 1 as a confidence indicator. In practice, since two branches are highly correlated and share the same activation function, we simply add another output channel for each class. In total, we have eight output channels, four for per-class segmentation and four for per-class uncertainty estimation.

In LCE, the confidence/uncertainty estimates are encouraged by interpolating between the predicted segmentation results and the targeted outputs, while the interpolation degree is controlled by the uncertainty values:

$$p'_{ij} = (1 - \sigma_{ij}) \cdot p_{ij} + \sigma_{ij} \cdot y_{ij}$$

for the prediction of class $j$ at the pixel $i$, where $\sigma_{ij}$ is the predicted uncertainty, $p_{ij}$ is the original segmentation probability and $y_{ij}$ is the binary ground truth. If the network is not confident about the results
(\sigma_{ij} \rightarrow 1), the network will receive the correct label \((p'_{ij} \rightarrow y_{ij})\) to avoid penalizing the uncertain cases. Otherwise, the network will try to push the prediction closer to the target probabilities to reduce the overall loss.

As we mentioned before, typical geographic features often only occupy a small region of maps. Common loss functions like cross-entropy could make the network be trapped in local minima where the predictions are strongly biased toward the background. Consequently, the network will misclassify the foreground objects or generate implausible predictions. Some researchers resort to re-weighting strategies to assign foreground samples with higher importance than background samples.

In our work, we make use of dice loss proposed by Milletari, Navab, and Ahmadi (2016) based on the dice coefficient that measures the overlapping degree between the prediction and the ground truth. In the training phase, instead of the original prediction, we use the interpolated prediction in Equation 1 to embed uncertainty-awareness into the loss function:

\[
DC = \frac{2 \sum_{i}^N p'_i y_i}{\sum_{i}^N p'_i^2 + \sum_{i}^N y_i^2 + 1}
\]

\[
\mathcal{L}_d = 1 - \sum_{j}^M W_j DC_j
\]

where \(M\) is the number of feature classes, \(W_j\) is the weight of class \(j\) and \(DC_j\) is the dice coefficient of class \(j\). The dice coefficient \(DC\) is calculated on the sample-level. In this formulation, the background pixels where both the prediction and the ground truth are zero will not contribute to the loss. In the testing phase, the dice coefficient is calculated from the original prediction.

To prevent the network from always predicting high uncertainty to minimize the loss, we add a log penalty as the regularization term \(\log(\sigma)\) into the loss function:

\[
\mathcal{L}_u = \frac{1}{N} \sum_{i}^N \sum_{j}^M W_j \log \sigma_{ij}
\]

\[
\mathcal{L} = \mathcal{L}_d + \lambda \mathcal{L}_u
\]

Experimental results and discussion

Experiments and results

In our paper, we have conducted an ablation study on the proposed methods to verify the effectiveness of using (a) dice loss for sparse objects (b) ASPP integration for multi-scale contexts (c) uncertainty estimation (LCE) to model noise in maps and adjust its impacts during training. We ran each experiment five times with randomly initialized models. Considering the time limits, we trained each model for 100 epochs with patience of 25 epochs for early stopping. We use the Adam optimizer that could automatically adapt the learning rate after each iteration. We set the initial learning rate as 0.001 and decrease it by 10% after each epoch.

For the uncertainty-aware model, we tested \(\lambda\) of 1, 3, 5, respectively, in Equation 5. We found that when \(\lambda\) is too small (\(\lambda = 1\)), the regularization is not strong enough so that the model lazily opts for free label predictions through predicting large uncertainty values, whereas when \(\lambda\) is too large (\(\lambda = 5\)), the penalization is too strong, leading to the generated uncertainty masks being constantly empty. We finally set \(\lambda = 3\).

We weight four classes equally in Equation 3 since the loss is calculated on the sample level and the amounts of samples are equal. By comparison, the regularization loss in Equation 4 is calculated on the pixel level, so we weight different classes considering the difference between their areas: streams (1), rivers (1), wetlands (0.15), lakes (0.1). The quantitative results of different experiments are shown in Table 1.

From Table 1 we can see, replacing binary cross-entropy loss with dice loss leads to better performance. The results are improved much further by integrating ASPP, especially for area features (i.e. wetlands, rivers, and lakes). By incorporating uncertainty estimation, the model experiences another improvement overall. In general, incorporating uncertainty estimation makes the model generate fewer false positives (higher precision rates), yet the model tends to have more false negatives (lower recall rates). In terms of the training efficiency, we can see that the integration of ASPP has accelerated training (faster
The quantitative results of different experiments. All the experiments are based on the same U-Net structure and vary from: using binary cross-entropy loss (BCE)/using dice loss/adding ASPP/leveraging uncertainty estimation (LCE). We also checked the performance of DeepLabV3+ on our dataset.

| Experiment (average training time/average training time per epoch) | DICE\textsubscript{avg} | F1\textsubscript{avg} | Precision\textsubscript{avg} | Recall\textsubscript{avg} | DICE\textsubscript{stream} | F1\textsubscript{stream} | Precision\textsubscript{stream} | Recall\textsubscript{stream} | DICE\textsubscript{river} | F1\textsubscript{river} | Precision\textsubscript{river} | Recall\textsubscript{river} | DICE\textsubscript{wetland} | F1\textsubscript{wetland} | Precision\textsubscript{wetland} | Recall\textsubscript{wetland} | DICE\textsubscript{lake} | F1\textsubscript{lake} | Precision\textsubscript{lake} | Recall\textsubscript{lake} |
|---------------------------------------------------------------|----------------|-----------------|-----------------|----------------|----------------|-------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| U-Net + BCE                                                  | 0.506          | 0.705           | 0.404           | 0.397          | 0.517          | 0.774            | 0.568           | 0.557          | 0.459          | 0.543          | 0.674          | 0.811           | 0.430            | 0.414          | 0.577          | 0.517          | 0.774          | 0.568          | 0.557          | 0.459          |
| (12.66 h/0.19 h)                                             | 0.826          | 0.806           | 0.838           | 0.849          | 0.811          | 0.772            | 0.468           | 0.309          | 0.459          | 0.701          | 0.539          | 0.768           | 0.489            | 0.307          | 0.593          | 0.539          | 0.768           | 0.489           | 0.307          | 0.593          |
| U-Net + DICE + ASPP                                          | 0.829          | 0.841           | 0.805           | 0.835          | 0.832          | 0.804            | 0.699           | 0.754          | 0.726          | 0.775          | 0.770          | 0.803           | 0.725            | 0.777          | 0.775          | 0.770          | 0.803           | 0.725           | 0.777          | 0.775          |
| (15.56 h/0.21 h)                                             | 0.897          | 0.884           | 0.904           | 0.902          | 0.898          | 0.811            | 0.659           | 0.792          | 0.764          | 0.857          | 0.836          | 0.845           | 0.779            | 0.858          | 0.857          | 0.836          | 0.845           | 0.779           | 0.858          | 0.857          |
| U-Net + DICE + ASPP + LCE                                    | 0.886          | 0.866           | 0.904           | 0.888          | 0.887          | 0.866            | 0.705           | 0.785          | 0.884          | 0.866          | 0.865          | 0.811           | 0.852            | 0.933          | 0.821          | 0.901          | 0.821          | 0.885          | 0.885          | 0.933          |
| (non-training tiles)                                         | 0.822          | 0.849           | 0.717           | 0.821          | 0.916          | 0.881            | 0.933           | 0.885          | 0.967          | 0.710          | 0.641          | 0.617           | 0.768            | 0.813          | 0.789          | 0.866          | 0.789          | 0.672           | 0.731           | 0.836          | 0.886          |
| DeepLabV3+                                                  | 0.751          | 0.559           | 0.864           | 0.937          | 0.830          | 0.841            | 0.810           | 0.830          | 0.840          | 0.785          | 0.785          | 0.810           | 0.830            | 0.840          | 0.830          | 0.840          | 0.840          | 0.840          | 0.840          | 0.840          |

convergence) when we compare it with the second baseline experiment. Incorporating uncertainty estimation has slightly increased the training time. For completeness, we have also tested our model performance on tiles from training map sheets excluding the training samples. This provides an upper bound of accuracy when training and testing samples come from exactly the same distribution, which is, however, unlikely in real map processing applications.

We have also applied DeepLabV3+ on our dataset after adding Sigmoid activation at the end and replacing binary cross-entropy loss with dice loss. Compared with U-Net integrated with ASPP, DeepLabV3+ has much worse performance for streams. This might be due to the delineation problem lying in the upsampling nature of DeepLabV3+ as mentioned before, of which narrow features like streams can be susceptible. Moreover, the training efficiency of DeepLabV3+ is significantly worse than the more lightweight U-Net architecture integrated with ASPP.

The qualitative comparison of experiments in our ablation study is shown in. For qualitative comparison, we check the model performances on complete map sheets by dividing a sheet into regular grids of the size 320 m × 320 m, generating predictions for each grid, and assembling the results. We crop both the input sheets as well as the predictions and scale them for illustration purposes.

As we can see from (a), (b), and (c), the continuity of features has been improved by integrating ASPP. Features of similar symbolic representations – rivers and lakes (both are comprised of parallel blue lines) can be better distinguished and segmented. We find inconsistent strokes of streams/rivers out of manual painting flaws occur frequently in the studied map sheets, which could hinder the continuity of segmentation. For example, in (c), a narrow river symbolized by close parallel lines has sometimes twisted strokes that are easily confused with stream symbols. The simple U-Net models with only small local contexts available fail to preserve the continuity of an entire feature. The combination of multi-contexts in ASPP has alleviated interruptions in the predictions. Another notable improvement by integrating ASPP is less fragmental segmentation for composite symbols like wetlands, which are a collection of individual symbols instead of a single area symbol. This can be seen from (d). After inspecting carefully, we find that
by leveraging uncertainty estimation, the network is more capable of recognizing density changes of individual wetland symbols and generating results with better delineation.

Indeed, the accuracy metrics are strongly influenced by our ground-truth annotations. For (d), we could easily spot abrupt density changes of individual symbols and carefully delineate the wetlands in the annotation. Nevertheless, in many cases, we tend to aggregate all the individual symbols into a whole continuous feature for annotation (see), even though it is more likely that the density of individual symbols is not consistent, which can be glimpsed from the segmentation results. If we tend to “over-aggregate” features in the annotation maps, the model that is capable of carefully delineating features might always be evaluated as underestimating. Hence, it is challenging to assert that the false negatives (reflected by the values of recall) are fully trustworthy. However, we find that with the help of uncertainty estimation, the

Figure 3. Semantic segmentation results of different experiments from Table 1. From top to bottom: U-Net + BCE, U-Net + DICE, U-Net + DICE + ASPP, U-Net + DICE + ASPP + LCE. For qualitative comparison, we checked the model performance on complete map sheets by diving a sheet into regular grids of 320 m × 320 m and assembling the testing result of each grid. For illustration, we scale and crop both the input sheets and the predictions (the scale bars of 100 m are presented in the corners).
model tends to have remarkably fewer false positives for graphic artifacts (e.g. scanning artifacts), which is illustrated in (f).

We have also compared our model with other state-of-art map segmentation methods, for buildings (Heitzler and Hurni 2020) and hydrology (Leyk and Boesch 2010). For building segmentation, we assembled five models and got the average predictions, the accuracy of which is already slightly better than the work of Heitzler and Hurni (2020), where they assembled eight models (two out of ten were discarded). For hydrology segmentation, we calculated the segmentation accuracy of the entire hydrological layer. Our model achieves distinctly superior accuracy than the work of Leyk and Boesch (2010). The comparison results are demonstrated in Table 2.

Table 2. The comparison of accuracy metrics among different methods. We compared our model with two state-of-art map segmentation methods for hydrology and buildings, respectively.

| Features     | Experiments          | IOU   | Recall | Precision | F1    |
|--------------|----------------------|-------|--------|-----------|-------|
| Hydrology    | Leyk and Boesch (2010) | -     | 0.76   | 0.80      | 0.78  |
|              | ours                 | 0.89  | 0.96   | 0.92      | 0.94  |
| Buildings    | Heitzler and Hurni (2020) | 0.88  | 0.93   | 0.95      | 0.94  |
|              | ours                 | 0.91  | 0.95   | 0.96      | 0.95  |

In our experiment, we find that there are three main scenarios where our model learns to indicate uncertainty:

Scenario I: Object boundaries. As we can see from (a), object boundaries have been captured in the uncertainty maps. This corresponds to our intuition that, generally speaking, object boundaries tend to contain less discriminative information and might have aliasing effects from scanning. Besides, for features with inexplicit boundaries (i.e. wetlands) where a group of individual symbols can never provide accurate boundary information, the model predicts

Figure 4. Possibly "Over-aggregated" wetlands in ground-truth annotations. From left to right: input map, ground truth, U-Net + DICE + ASPP, U-Net + DICE + ASPP + LCE.

Figure 5. Estimated uncertainty in object boundaries. From left to right: input map, ground truth, prediction, and uncertainty map of different classes (uncertainty≥0.5). Features containing explicit boundaries (e.g. lakes, streams) have smaller variances in estimated uncertainty distributions than features containing inexplicit boundaries (wetlands).
wider uncertainty distributions (larger variances). (b) shows the estimated uncertainty of wetland boundaries including the road area that interrupts the wetland, where the density of individual wetland symbols has abrupt changes.

Scenario II: Noise from drawing, aging, and scanning. In (c) (e), we have already seen that with the help of uncertainty estimation, the network becomes more robust against noisy strokes and artifacts. In fact, the network is able to indicate uncertainty lying in those kinds of noise at the same time. shows the estimated uncertainty at noisy strokes (a), painting errors (b), painting or scanning artifacts (c) and mixed colors (blue and black) (d) caused by aging and scanning. Figure 5

Scenario III: Out-of-distribution samples – rare or unseen designs. Maps have various designs from different map series. We randomly selected 10 Siegfried map sheets of scale 1:25,000 in our testing area and 10 of scale 1:50,000 (out-of-distribution samples) in the Swiss Alps, around major lake basins. Similarly, we divided the map sheets into regular grids of 256 pixels × 256 pixels (each grid represents 640 m × 640 m, since the resolution of a map sheet of 1:50,000 is 2.5 m), predicted for each grid, and assembled the results. We check the difference of estimated uncertainty for lakes under the assumption that the design of lakes has a large diversity in terms of colors and textures among different map series. We use the ground-truth lakes as masks to calculate the ratio between the number of uncertain pixels (uncertainty ≥ 0.5) and the area (in pixels) of lakes for each map sheet (see Figure 7). As we can see, for map sheets from 1:50,000 (rather rare or unseen designs), our model tends to indicate higher uncertainty on average with a larger range of values. Figure 8 presents two examples of the map sheets of 1:50,000 where the model yields high uncertainty values on lakes. Figure 6

![Figure 6](image_url)

* In predictions and annotations, red = stream, blue = river, green = wetland, pink = lake.

Figure 6. Estimated uncertainty in noise from drawing, aging, and scanning. From left to right: input map, ground truth, prediction, and uncertainty map. The uncertainty is predicted at noisy strokes (a) where the unstable strokes make a stream appear like a river, painting errors (b) where the symbol of a lake “flows out” and mix with that of a wetland, noise from the original painting (c), and mixed colors because of aging or scanning (d).
uncertain pixels (uncertainty > 0.2), and interpolate the missing values by taking into account the nearest neighbors out of the remaining confident pixels within the given range (50–80 m). The dice coefficients of different feature classes have been improved from 1% to 6% on average. It has illustrated the improved segmentation accuracy for wetlands in (a) and reduced false positives in (b).

Another potential use of the learned uncertainty lies in its ability to detect out-of-distribution samples. We can use it to locate rarely occurring map symbols or map designs, and then actively query more samples from those designs based on the principle of active learning to efficiently improve the network generalizability. Alternatively, we can treat the out-of-distribution samples separately by applying the pre-trained model to those new designs under the scheme of transfer learning.

Furthermore, the predicted uncertainty can guide us to select reliable features from historical maps of higher graphic quality for GIS analyses. Figure 10 shows different levels of uncertainty captured by our model of two temporally successive map sheets. The strokes of wetland symbols on the sheet from the year 1885 are so vague that they are hardly perceivable. By comparison, the graphic quality of the map sheet from 1891 is superior. Our model has captured this quality difference in the uncertainty maps. Hence, we

**Potential uses of learned uncertainty and discussion**

In the last section, we have illustrated that our model can capture different types of uncertainty, indicating different scenarios where our segmentation might be wrong. The learned uncertainty can be used to filter out potentially unreliable predictions. Table 3 presents the segmentation accuracy after filtering out pixels with uncertainty values higher than 0.2 and 0.5, respectively. The increased segmentation accuracy after filtering out uncertain pixels implies the ability of the estimated uncertainty to reveal implausible predictions.

We further use the predicted uncertainty to find the most uncertain map sheets in our testing area. We calculate the ratio between the number of uncertain pixels and that of the edge pixels from the segmentation map for each map sheet to obtain the level of uncertainty outside normal object boundaries. We select the 10% most uncertain map sheets, remove them from further testing, and compare the results with those sheets.

**Table 3. Effects of filtering out uncertain pixels.** During testing, we filter out pixels with uncertainty values above 0.5 and 0.2, respectively, calculate the segmentation accuracy of the rest pixels, and compare with the segmentation results without any filtering.

| Filter | Dice <sub>avg</sub> | Dice <sub>stream</sub> | Dice <sub>river</sub> | Dice <sub>wetland</sub> | Dice <sub>lake</sub> |
|--------|-----------------|-----------------|-----------------|-----------------|-----------------|
| No     | 0.827           | 0.788           | 0.802           | 0.920           | 0.798           |
| Uncertainty > 0.5 | 0.864     | 0.871           | 0.824           | 0.940           | 0.819           |
| Uncertainty > 0.2 | 0.880     | 0.890           | 0.853           | 0.947           | 0.831           |

![Figure 7. Percentages of uncertain pixels for lakes from different map series.](image) We calculate the ratio between the number of uncertain pixels (uncertainty ≥ 0.5) and the area (in pixels) of lakes per map sheet. The boxplot shows the minimum and the maximum, the quartiles, the median (Orange lines), and the mean (green dots).

**Figure 8.** Two testing map sheets of scale 1:50,000 and their predicted uncertainty maps for lakes. The uncertainty value increases from red (0) to blue (0.9). The scale bars of 1000 m are shown in the corners.
could refer to the predicted uncertainty to select reliable features among temporally successive map sheets and editions for potential GIS analyses Figure 9.

One major drawback of our model lies in the limited detection window size. Special features like elongated narrow lakes can require a much larger spatial context for the network to distinguish them correctly from rivers. However, incorporating much larger contexts can increase memory usage, which is not always feasible due to insufficient hardware capacity. Small detection windows that provide inadequate contexts can inhibit a model from making consistent predictions, or potentially push it to overfit certain attributes of data. Figure 11 shows a failed case of our model where lakes with changing roundness are misclassified as rivers at the slimmer parts. Our model is likely to overfit the roundness of features to make predictions and fails to indicate any uncertainty in this case.

Figure 9. Refined predictions after replacing values of uncertain pixels with the interpolation results from confident pixels. From left to right: input map, ground truth, prediction, and refined segmentation. In (a), the segmentation quality of wetlands has been improved (fewer false negatives). In (b), false positives from artifacts have been reduced.

Figure 10. Different levels of graphic quality can be indicated by the predicted uncertainty. From left to right: map of the year 1885 with inferior graphic quality of wetlands and its predicted uncertainty, map of the year 1891 with superior graphic quality and its predicted uncertainty.

Figure 11. A failed case of our model. From left to right: input map, ground truth, prediction and estimated uncertainty. The slimmer parts of both lakes are misclassified as rivers without any uncertainty indicated.
Conclusions

Our work has proposed a novel approach for segmenting features from scanned historical maps that incorporates multi-scale contexts and handles map noise by integrating ASPP and uncertainty estimation into our adapted U-Net architecture. The multi-scale feature pyramids constructed by ASPP help to improve the continuity of segmented features and to distinguish features with similar symbolization. Leveraging uncertainty estimation benefits the network with better delineation of segmentation and less sensitivity to artifacts. Meanwhile, the model is capable of recognizing uncertainty lying in object boundaries, map noise, and out-of-distribution samples. Our model shows superior performance compared with the state-of-art architecture DeepLabV3+ and two state-of-art map processing methods. The predicted uncertainty can act as an indicator for unreliable predictions. We thereby can refine our predictions by filtering out unreliable pixels and interpolating the missing values with reliable pixels. Since our model can also indicate implausible graphic quality by yielding large uncertainty values, we are enabled to select reliable features from maps of superior graphic quality for future GIS analyses. Furthermore, we can locate out-of-distribution map designs for which our model will predict larger uncertainty values. This makes it possible to explore potential schemes like active learning to actively query more samples from those special maps, or transfer learning to transfer the pre-trained model to new designs. In the future, we will investigate how to increase the input contexts when confronted with memory limits.

Data availability

In this study, we made use of the Topographic Atlas of Switzerland (“Siegfried map”) of the scale 1:25,000 provided by Federal Office of Topography (Swisstopo) https://www.swisstopo.admin.ch/en/geodata/maps/historical/siegfried25.html. The dataset is not available for free download and should be requested from Swisstopo. The samples of map sheets and manually vectorized ground-truth data could be provided upon request, only after the explicit permission from Swisstopo. The models and codes generated from this study are available upon request to the corresponding author.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was supported and funded by Schweizerischer Nationalfonds zur Förderung der Wissenschaftlichen Forschung (SNF)[No. 188692].

ORCID

Magnus Heitzler http://orcid.org/0000-0002-9021-4170
Lorenz Hurri http://orcid.org/0000-0002-0453-8743

References

Bin, D., and W. K. Cheong. 1998. “A System for Automatic Extraction of Road Network from Maps.” In Proceedings. IEEE International Joint Symposia on Intelligence and Systems, Rockville, Maryland, 359–366.
Chen, L.-C., G. Papandreou, F. Schroff, and H. Adam. 2017. “Rethinking Atrous Convolution for Semantic Image Segmentation.” arXiv, 1706.05587.
Chen, L.-C., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. 2018. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation.” In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. eds., Computer Vision – ECCV 2018, 833–851. Cham: Springer International Publishing.
Chiang, Y. Y., and C. A. Knoblock. 2009. “Extracting Road Vector Data from Raster Maps.” In The 8th IAPR International Workshop on Graphics Recognition (GREC), La Rochelle, 93–105.
Chiang, Y. Y., and C. A. Knoblock. 2013. “A General Approach for Extracting Road Vector Data from Raster Maps.” International Journal on Document Analysis and Recognition (IJDAR) 16 (1): 55–81. doi:10.1007/s10032-011-0177-1.
Chiang, Y. Y., S. Leyk, and C. A. Knoblock. 2011. “Integrating Color Image Segmentation and User Labeling for Efficient and Robust Graphics Recognition from Historical Maps.” In The 9th IAPR International Workshop on Graphics Recognition (GREC), Seoul, 1–4.
Chiang, Y. Y., S. Leyk, and C. A. Knoblock. 2014. “A Survey of Digital Map Processing Techniques.” ACM Computing Surveys 47 (1): 1–44. doi:10.1145/2557423.
Chiang, Y. Y., S. Leyk, and C. A. Knoblock. 2013. “Efficient and Robust Graphics Recognition from Historical Maps.” In Kwon, Y.-B., and Ogier, J.-M. eds., Graphics Recognition. New Trends and Challenges 742 : 25–35. Berlin, Heidelberg: Springer Berlin Heidelberg.
Chiang, Y. Y., W. Duan, S. Leyk, J. H. Uhl, and C. A. Knoblock. 2020. Using Historical Maps in Scientific Studies: Applications, Challenges, and Best Practices. Cham: Springer International Publishing.
Cipolla, R., Y. Gal, and A. Kendall. 2018. “Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics.” In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, 7482–7491.
DeVries, T., and G. W. Taylor. 2018a. "Learning Confidence for Out-of-Distribution Detection in Neural Networks." arXiv, 1802.04865.

DeVries, T., and G. W. Taylor. 2018b. "Leveraging Uncertainty Estimates for Predicting Segmentation Quality." arXiv, 1807.00502.

Dhar, D. B., and B. Chanda. 2006. "Extraction and Recognition of Geographical Features from Paper Maps." International Journal on Document Analysis and Recognition 8 (4): 232–245. doi:10.1007/s10032-005-0010-9.

Gal, Y. 2016. "Uncertainty in Deep Learning," PhD diss., University of Cambridge.

GDAL/OGC contributors. 2021. "GDAL/OGC Geospatial Data Abstraction Software Library." Open Source Geospatial Foundation. https://gdal.org

Guo, C., G. Pleiss, Y. Sun, and K. Q. Weinberger. 2017. "On Calibration of Modern Neural Networks." In International Conference on Machine Learning, Sydney, 1321–1330.

Gurevich, P., and H. Stuke. 2018. "Pairing an Arbitrary Regressor with an Artificial Neural Network Estimating Aleatoric Uncertainty." arXiv, 1707.07287.

He, H., D. Yang, S. Wang, S. Wang, and Y. Li. 2019. "Road Extraction by Using Atrous Spatial Pyramid Pooling Integrated Encoder-Decoder Network and Structural Similarity Loss." Remote Sensing 11 (9): 1015. doi:10.3390/rs11091015.

Heitzler, M., C. Gkonomos, A. Tsorlini, and L. Hurni. 2018. "A Modular Process to Improve the Georeferencing of the Siegfried Map." e-Perimetr 13 (2): 96–100.

Heitzler, M., and L. Hurni. 2020. "Cartographic Reconstruction of Building Footprints from Historical Maps: A Study on the Swiss Siegfried Map." Transactions in GIS 24 (2): 442–461. doi:10.1111/tgis.12610.

Hu, P., S. Sclaroff, and K. Saenko. 2020. "Uncertainty-Aware Learning for ZeroShot Semantic Segmentation." In Advances in Neural Information Processing Systems, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, 21713–21724. Vol. 33. New York: Curran Associates.

Iosifescu, I., A. Tsorlini, and L. Hurni. 2016. "Towards a Comprehensive Methodology for Automatic Vectorization of Raster Historical Maps." e-Perimetr 11 (2): 57–76.

Kendall, A., and Y. Gal. 2017. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In Proceedings of the 31st International Conference on Neural Information Processing Systems, 5580–5590. NIPS’17. Long Beach, California, New York: Curran Associates.

Khotanzad, A., and E. Zink. 2003. "Contour Line and Geographic Feature Extraction from USGS Color Topographical Paper Maps." IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (1): 18–31. doi:10.1109/TPAMI.2003.1159943.

Kuireghan, A. D., and O. Ditlievesen. 2009. "Aleatory or Epistemic? Does It Matter? Risk Acceptance and Risk Communication." Structural Safety 31 (2): 105–112. doi:10.1016/j.strusafe.2008.06.020.

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. "Backpropagation Applied to Handwritten Zip Code Recognition." Neural Computation 1 (4): 541–551. doi:10.1162/neco.1989.1.4.541.

Leyk, S., and R. Boesch. 2009. "Extracting Composite Cartographic Area Feature in Low-quality Maps." Cartography and Geographic Information Science 36 (1): 71–79. doi:10.1559/152304009787340115.

Leyk, S., and R. Boesch. 2010. "Colors of the Past: Color Image Segmentation in Historical Topographic Maps Based on Homogeneity." Geoinformatica 14 (1): 1–21. doi:10.1007/s10707-008-0074-z.

Leyk, S., R. Boesch, and R. Weibel. 2006. "Saliency and Semantic Processing: Extracting Forest Cover from Historical Topographic Maps." Pattern Recognition 39 (5): 953–968. doi:10.1016/j.patcog.2005.10.018.

Leyk, S. 2009. "Segmentation of Colour Layers in Historical Maps Based on Hierarchical Colour Sampling." In Ogier, J.-M., and Liu, W., eds., International Workshop on Graphics Recognition (pp. 231–241). Berlin Heidelberg: Springer Berlin Heidelberg.

Liu, Y. 2002. "An Automation System: Generation of Digital Map Data from Pictorial Map Resources." Pattern Recognition 35 (9): 1973–1987. doi:10.1016/S0031-384X(00)00154-6.

Miao, Q., P. Xu, T. Liu, Y. Yang, J. Zhang, and W. Li. 2012. "Linear Feature Separation from Topographic Maps Using Energy Density and the Shear Transform." IEEE Transactions on Image Processing 22 (4): 1548–1558. doi:10.1109/TIP.2012.2233487.

Milletari, F., N. Navab, and S. A. Ahmadi. 2016. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation." Proceedings - 2016 4th International Conference on 3D Vision, Stanford, 565–571.

Pezeshk, A., and R. L. Tutwiler. 2008. "Contour Line Recognition & Extraction from Scanned Colour Maps Using Dual Quantization of the Intensity Image." In IEEE Southwest Symposium on Image Analysis and Interpretation, Santa Fe, 173–176. IEEE.

Pezeshk, A., and R. L. Tutwiler. 2010. "Extended Character Defect Model for Recognition of Text from Maps." In IEEE Southwest Symposium on Image Analysis Interpretation (SSIAI), Austin, Texas, 85–88.

Pezeshk, A., and R. L. Tutwiler. 2011. "Automatic Feature Extraction and Text Recognition from Scanned Topographic Maps." IEEE Transactions on Geoscience and Remote Sensing 49 (12): 5047–5063. doi:10.1109/TGRS.2011.2157697.

Ronneberger, O., P. Fischer, and T. Brox. 2015. "U-net: Convolutional Networks for Biomedical Image Segmentation." In International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, 234–241.
Samet, R., and E. Hancer. 2012. "A New Approach to the Reconstruction of Contour Lines Extracted from Topographic Maps." Journal of Visual Communication and Image Representation 23 (4): 642–647. doi:10.1016/j.jvcir.2012.02.005.

Uhl, J. H., S. Leyk, Y. Chiang, W. Duan, and C. A. Knoblock. 2017. "Extracting Human Settlement Footprint from Historical Topographic Map Series Using Context-based Machine Learning." In The 8th International Conference of Pattern Recognition Systems (ICPRS 2017), Madrid, 1–6.

Uhl, J. H., S. Leyk, Y. Y. Chiang, W. Duan, and C. A. Knoblock. 2018. "Spatialising Uncertainty in Image Segmentation Using Weakly Supervised Convolutional Neural Networks: A Case Study from Historical Map Processing." IET Image Processing 12 (11): 2084–2091. doi:10.1049/iet-ipr.2018.5484.

Uhl, J. H., S. Leyk, Y. Y. Chiang, W. Duan, and C. A. Knoblock. 2020. "Automated Extraction of Human Settlement Patterns from Historical Topographic Map Series Using Weakly Supervised Convolutional Neural Networks." IEEE Access 8: 6978–6996. doi:10.1109/ACCESS.2019.2963213.