Instance Segmentation, Body Part Parsing, and Pose Estimation of Human Figures in Pictorial Maps*

Raimund Schnürer a, A. Cengiz Öztireli b, Magnus Heitzler a, René Sieber a and Lorenz Hurni a

aDepartment of Civil, Environmental and Geomatic Engineering, ETH Zurich, Zurich, Switzerland; bDepartment of Computer Science and Technology, University of Cambridge, Cambridge, UK

ABSTRACT
In recent years, convolutional neural networks (CNNs) have been applied successfully to recognise persons, their body parts and pose keypoints in photos and videos. The transfer of these techniques to artificially created images is rather unexplored, though challenging since these images are drawn in different styles, body proportions, and levels of abstraction. In this work, we study these problems on the basis of pictorial maps where we identify included human figures with two consecutive CNNs: We first segment individual figures with Mask R-CNN, and then parse their body parts and estimate their poses simultaneously with four different UNet++ versions. We train the CNNs with a mixture of real persons and synthetic figures and compare the results with manually annotated test datasets consisting of pictorial figures. By varying the training datasets and the CNN configurations, we were able to improve the original Mask R-CNN model and we achieved moderately satisfying results with the UNet++ versions. The extracted figures may be used for animation and storytelling and may be relevant for the analysis of historic and contemporary maps.

Ces dernières années, les réseaux de neurones convolutifs (CNN) ont été appliqués avec succès pour reconnaître des personnes, des parties du corps et des points de pose saillants dans des photos et des vidéos. Le transfert de ces techniques vers des images créées artificiellement est peu exploré, bien que difficile puisque ces images sont dessinées dans différents styles, proportions corporelles et niveaux d’abstraction. Dans ce travail, nous étudions ces problèmes appliqués à des cartes illustrées où nous identifions des figures humaines avec deux CNN consécutifs : nous segmentons d’abord des figures individuelles avec Mask R-CNN, puis analysons leurs parties du corps et estimons leur pose simultanément avec quatre différentes versions UNet++. Nous entrainons les réseaux de neurones avec un mélange de...
Introduction

Digital ‘view-only’ maps, such as scanned historical maps or modern maps made with graphic editors, include a multitude of information. To process this information in a machine-readable manner, the content of these maps needs to be extracted – in a best case fully automatically. At present, however, maps stored in a raster image format have been mostly manually annotated with metadata on social media websites and additionally georeferenced in digital map libraries, but information about their actual content is largely missing. In this work, we have a closer look at one particular content element of maps, namely human figures (Figure 1). This object type frequently occurs

Figure 1. Exemplary pictorial map with human figures on the sides (source: https://i.pinimg.com/736x/16/36/0b/16360bbad0f6ed13e56ff7215e7590dc.jpg).
as a decoration in pictorial maps (Child, 1956). After a successful detection, the following two use cases are thinkable: Firstly, historians (e.g. Davies, 2016) may be interested in the ethnos and clothing of figures, as well as in certain rites or common activities. Offering an additional search filter option in digital map catalogues for human figures in maps would therefore be highly beneficial. Secondly, figures could be animated to act as storytellers or guides in maps or paintings, for instance for museum visitors (e.g. Liu et al., 2020). For the latter usage scenario, it is additionally required to identify their body parts and pose keypoints.

A promising technology to tackle the above-mentioned tasks are convolutional neural networks (CNNs). Recent experiments have shown that it is feasible to extract labels (Weinman et al., 2019), road intersection points (Saeedimoghaddam & Stepinski, 2020), or building footprints (Heitzler & Hurni, 2020) from maps with CNNs. By conducting our research, we like to extend this list by human figures. To our knowledge, this object class has not been retrieved with CNNs in maps yet, only in natural images such as photos. Here, architectures like Mask R-CNN (He et al., 2017) or PANet (Liu et al., 2018) were developed to segment individual objects. The task of parsing object parts, such as body parts, was approached with configurations like an adapted fully convolutional network (FCN) (Oliveira et al., 2016) or DeepLab (Chen et al., 2018). Convolutional Pose Machines (Wei et al., 2016) and Stacked Hourglass (Newell et al., 2016) enable to detect pose keypoints of single humans, whereas newer networks (e.g. Cao et al., 2017; Sun et al., 2020) are capable of registering multiple persons. Body part segmentation and key-point detection were also combined, for example, in a FCN including a conditional random field (Xia et al., 2017) or JPPNet (Liang et al., 2019).

Data

Compared to real persons in photos, annotated datasets of human figures in maps are non-existing. Therefore, we created the data for training and testing our CNNs by ourselves. The data partly originates from a Pictorial Map Classification Dataset. This dataset consists of 3100 maps harvested from Pinterest accordingly to Art. 24d of the Swiss Copyright Act, among 1500 are pictorial maps. A manual classification by ourselves revealed that roughly half of the pictorial maps include human figures. We included humanoid representations (e.g. Statue of Liberty, Christ the Redeemer; snowmen, robots) but we excluded humanoid animals (e.g. King Kong) in our classification. Fifty two of those maps, which include 387 larger figures, were manually annotated using a web application (Figure 2). The average size of maps is $577 \times 581$px and of figures is $44 \times 78$px. The low resolution of the map images to be fed into CNNs is due to memory limitations of current graphic boards. Annotating one figure took about nine minutes (incl. corrections), resulting in a total working time of about 60 h. Six body parts (head, torso, two arms, and two legs) were masked and skeletons consisting of 16 keypoints (head, neck, thorax, two shoulders, two elbows, two wrists, pelvis, two hip joints, two knees, and two ankles) were created. Since the number of annotated maps would be too little to train CNNs, these 52 maps will serve as testing data for the CNNs.

For training the CNNs, we produced a synthetic dataset, what is a common approach in machine learning (e.g. Varol et al., 2017). It has the advantage that larger amounts of annotated data can be created in less time compared to manually annotated data.
However, the variability of data may not be as large (i.e. the styles of the figures in our case) and training times may need to be reduced so that the CNN does not specialise in the synthetic data. Our synthetic data consists of background maps, on which persons and objects from photos, generated figures, and icon objects are placed (Figure 3). By this mixture of real and abstract representations of humans, we hope

Figure 3. Real and synthetic entities randomly scaled and placed on a map (source: https://i.pinimg.com/736x/11/f1/eb/11f1ebb9bcf3d20690cb0f3d8fcf5119–city-maps-terra.jpg). Map images (source data, left) and silhouettes (target data, right) of figures serve as training data for instance segmentation.
that the CNN interpolates between them to recognise pictorial figures. Since not only human figures but also other objects (e.g. means of transportation, animals) are an integral part of pictorial maps, we included them as well so that the network learns to distinguish between figures and objects.

The background maps were also derived from the *Pictorial Map Classification Dataset* by selecting only those without any human figures. Thirty-three anthropomorphic maps (e.g. Europa Prima Pars Terrae In Forma Virginis, The Avenger – An Allegorical War Map) were excluded because their content is interwoven with humanoid figures. One map of them was added due to balancing reasons. The selection resulted in 2306 background maps, among which about one third are pictorial maps without any persons and two thirds are non-pictorial maps. To enrich these maps, 4558 real persons (Figure 4(a)), including skeletons and body part masks, and 5169 real objects (Figure 4(b)) from 19 different categories were firstly taken from the *PASCAL-Part Dataset*,3 which contains annotations for photos. Secondly, 4558 synthetic human figures (Figure 4(c)) were generated in a custom web application, initially as scalable vector graphics (SVG) which are finally rasterised to images. The selection of real persons and the generation of synthetic figures are based on the frequency of occurrence of certain body part configurations in the test dataset (Table 1). The postures of the synthetic figures are derived from skeleton annotations of the *MPII Human Pose Dataset*.4 At the corresponding joints of each skeleton, ellipses were drawn for the head, whereas polygons – partly with rounded corners – were drawn for torso, arms and legs. Shapes for hats, hair, glasses, eyebrows, eyes, noses, mouths, hands, and shoes were additionally attached to the synthetic figures. Shapes, colours, fill patterns, body part sizes, and stroke widths were randomly varied. Pose keypoints, originating from the skeletons, and body part masks, derived from the overlays, were generated aside from the synthetic figure images. Thirdly, 4759 medium sized, non-circular icon objects (Figure 4(d)) from 44 categories were retrieved from *Iconfinder*.5 In a last automated step, a random number of zero to 15 real and

![Figure 4.](a) A real person, its body part mask and skeleton as well as (b) a real object from the PASCAL-Part Dataset; (c) A synthetic person, its body part mask and skeleton from our SVG figure generator; (d) an icon object from the search engine ‘Iconfinder’. Figures like (a) and (c) serve as training data for body part parsing and pose estimation.)
synthetic persons as well as objects are scaled randomly between 20 and 120px and placed randomly on the background maps. In case of overlaps, person masks covering an area of less than 50px, and corresponding keypoints, were excluded.

We tested how the following training datasets, varying in real and synthetic entities (= persons and objects), affect the accuracy of the CNN targeted at instance segmentation:

- **Real**: 2304 maps with real entities.
- **Synthetic**: 2304 maps with synthetic entities.
- **Separated**: 1152 maps with real entities and 1152 maps with synthetic entities.
- **Mixed**: 2304 maps with real and synthetic entities.
- **Separated-Mixed**: 768 maps with real entities, 768 maps with synthetic entities, and 768 maps with real and synthetic entities.

For training the CNN targeted at body part detection and pose estimation, 4558 real persons, 4558 synthetic figures, and a combination of 2279 real persons and 2279 synthetic figures (i.e. the Separated dataset) are taken into account. A larger selection of pictorial maps and figures, real persons and objects, synthetic figures and icon objects, and background and training maps can be found in Figures 9–16 of the Supplemental data.

**Methods**

We follow a top-down approach (e.g. Lin et al., 2020) by first segmenting instances of human figures on maps and then body parts and pose keypoints. In the first step, we try to identify silhouettes of individual characters on pictorial maps with the established Mask R-CNN (He et al., 2017) architecture. This CNN is targeted at recognising objects, such as persons, from photos at a pixel-level. Mask R-CNN is an extension of Faster R-CNN (Ren et al., 2017), a network, which is able to detect bounding boxes of objects. Similar to Faster R-CNN, a series of convolution and downscaling operations are initially applied inside a backbone network in Mask R-CNN to extract specific image features. The output, so-called feature maps, are processed next in two stages: Firstly, objectness scores, denoting the likelihood that a region contains an object, and offsets for anchors, that are rectangles differing in size and aspect ratio, which are distributed equally in a grid covering the feature maps, are predicted in a region proposal.
network. Secondly, the regions of interest are further refined and a score for the potential object class is predicted. As an addition to Faster R-CNN, Mask R-CNN predicts a binary mask in this second stage, where each pixel of the mask corresponds to a probability. Only one channel is required for the mask since the separation of a potentially contained object from the background is determined by a threshold.

Mask R-CNN is included in the TensorFlow Model Garden where different CNN architectures are pre-implemented and accessible via a Python API. In our experiment, we retrained the model based on the COCO dataset (Lin et al., 2014), which comprises segmentations for 500,000 masked objects on photos, such as persons. For transfer learning, we take the five training datasets (i.e. Real, Synthetic, Separated, Mixed, Separated-Mixed) described in the previous chapter. As a backbone network for feature detection, we use ResNet with 101 layers (He et al., 2016) and atrous convolutions, which has a good accuracy-speed balance compared to the other three available TensorFlow models. Atrous (aka dilated) convolutions ‘enlarge the field of view of filters to incorporate larger context, which [has been] shown to be beneficial’ (Chen et al., 2018, p. 4). We set the anchor stride to eight, which has been favourable to detect smaller objects (e.g. Schnürer et al., 2020). We vary the four sizes for the anchors (minimum: 0.0625, maximum: 2.0) and retain their three aspect ratios (i.e. 2:1, 1:1, and 1:2). Since the architecture requires much graphics memory, images could be fed in batches of one (i.e. single images) on a NVIDIA GeForce GTX 1080 Ti. On this graphics boards, one epoch of learning (i.e. 2304 steps) takes about 30 min. Other parameters have not been modified from the given Mask R-CNN configuration file. For evaluation, we calculated the average precision (AP) for masks with the COCO API. As given in the configuration for the original model, we set the threshold of confidence scores to larger than 0.3, which means that detections below the threshold will be discarded.

After having recognised human silhouettes, we parse body parts and detect keypoints simultaneously in our own network consisting of four different versions (Figure 5). Other networks have been proposed, which perform both tasks at the same time, but those do not have any code available (e.g. Xia et al., 2017) or classify different types of body parts (e.g. Liang et al., 2019). Therefore, we cannot report any baselines for pre-trained models on real persons. Mask R-CNN would be also able to indicate keypoints; however, this

Figure 5. Our custom CNN architectures (see arrows) to parse body parts and estimate poses simultaneously.
functionality was not part of the code. Therefore, we decided to implement and test our own network configurations, inspired by a simple deconvolution head network (Xiao et al., 2018) and UNet++ (Zhou et al., 2020). In contrast to the simple deconvolution head network (to be called ‘Simple Deconv’ in the following) but similarly to UNet++, we used a decreasing number of filters for the deconvolution operations. Diverging from UNet++ but similarly to Simple Deconv, we do not include the loss of intermediate layers and we do not perform any convolution operations after having upsampled the feature maps. Opposed to both networks but similarly to other networks like Stacked Hourglass (Newell et al., 2016), we perform an ‘Add’ operation to merge layers instead of concatenating them and we do not upsample the image to the full resolution. More details of the architectural decisions and their alternatives are given in the Discussion chapter.

We implemented our body part parsing and pose estimation networks with TensorFlow’s high-level keras API.8 We use ResNet with 50 layers (He et al., 2016) pre-trained on ImageNet weights as a backbone network. We feed square RGB images encoded in the JPEG format with a size of 128²px into ResNet, which is smaller than the default image input size of 224²px, but the reduced size better conforms to our data. An additional one-strided convolution is performed to adjust the number of channels to 128 of the output layer after the first two-strided convolution in ResNet. This step is not necessary for the ResNet output feature maps of the second to the fourth stage. We omit the lowest stage of ResNet because figures on maps do not have as many details as persons on photos. The results of the backbone network are passed to a head network, where we test four different versions (Figure 5):

- Simple Deconv: The output feature map of the fourth ResNet stage (X³.0) is upsampled three times (X².1, X¹.2, and X⁰.3) by two-strided convolutions.
- Simple UNet: Supplementary to Simple Deconv, outputs of the third (X².0), second (X¹.0), and first (X⁰.0) ResNet stage are added to the upsampled feature maps – one at a time.
- Simple UNet+: The output feature maps of the third and second ResNet stages are upsampled and added to the outputs of the second and first stage. The first result (X¹.¹) from the previous addition is upsampled and added to the second result (X⁰.1). The output of the third ResNet stage, the first and the third result (X¹.²) are added to the upsampled feature maps of Simple Deconv.
- Simple UNet++: Supplementary to Simple UNet+, the following skip connections are inserted: X¹.⁰ before X¹.², X⁰.⁰ before X⁰.², X⁰.⁰ before X⁰.³, and X⁰.¹ before X⁰.³.

Xᵐ⁻ⁿ is the notation from UNet++: X symbolises the tensor, m corresponds to the down-sampling level, and n denotes how many upsampling (i.e. deconvolution) operations have been performed. In our head networks, each deconvolution and adding operation is followed by a batch normalisation operation and a Rectified Linear Unit activation function. Kernels of convolutional layers are initialised by a truncated normal distribution centred at zero (i.e. ‘he_normal’). The final convolution operation is followed by a sigmoid activation function so that we have a 64²px image with 24 channels in the end. The channels correspond to six body parts and one channel for the background, and to 16 keypoints and one channel containing the inverted image of the summed keypoints. In the ground truth data, body part pixels have a value of one and other pixels a value of zero. The keypoints
are represented by a 2D gaussian kernel, similarly to Convolutional Pose Machines (Wei et al., 2016), having a probability of one in the centre and gradually decreasing values around the centre (Figure 6). The loss is split between body parts and keypoints, and reduced in both cases using the categorial cross entropy function and the RMSprop optimiser during training. The background channel for body parts is ignored in the loss function so that the network is not biased towards the white background of the images with human figures. We fed images in batches of 15 into the network and trained for 15 epochs, which took about 10 min in total.

Results

We trained each Mask R-CNN configuration (i.e. same training data and same anchor scales) five times, disregarded the highest and lowest score, and averaged the remaining three scores to reduce the variability in the comparison. Our evaluation procedure is similar to Zhang et al. (2019), who additionally calculated the standard deviation but did not discard the extreme scores. Quantitative results (Table 2) show that the highest

Table 2. Averaged COCO metrics of retrained Mask R-CNN models (backbone: ResNet101 with atrous convolutions) for different datasets and scales.

| Data         | Anchor scales | AP    | AP$_{50}$ | AP$_{75}$ | AP$_{small}$ | AP$_{medium}$ | AP$_{large}$ |
|--------------|---------------|-------|-----------|-----------|--------------|---------------|--------------|
| COCO         | 1/4, 1/2, 1/1, 2/1 | 15.57% | 32.18%    | 14.08%    | 6.97%        | 21.54%        | 16.37%       |
| Synthetic    | 1/4, 1/2, 1/1, 2/1 | 2.85%  | 7.67%     | 1.48%     | 0.54%        | 4.86%         | 0.00%        |
| Real         | 1/8, 1/4, 1/2, 1/1 | 2.97%  | 8.55%     | 1.37%     | 0.78%        | 5.21%         | 2.23%        |
|              | 1/16, 1/8, 1/4, 1/2 | 2.64%  | 6.83%     | 1.95%     | 0.65%        | 4.51%         | 0.00%        |
| Separated    | 1/4, 1/2, 1/1, 2/1 | 4.87%  | 12.04%    | 3.23%     | 2.12%        | 6.95%         | 8.42%        |
|              | 1/8, 1/4, 1/2, 1/1 | 3.63%  | 9.12%     | 2.66%     | 1.81%        | 5.25%         | 2.97%        |
|              | 1/16, 1/8, 1/4, 1/2 | 6.07%  | 15.49%    | 3.28%     | 2.10%        | 8.74%         | 9.90%        |
| Mixed        | 1/4, 1/2, 1/1, 2/1 | 10.40% | 27.76%    | 9.60%     | 5.32%        | 26.61%        | 19.46%       |
|              | 1/8, 1/4, 1/2, 1/1 | 18.18% | 43.10%    | 9.56%     | 5.32%        | 26.61%        | 19.46%       |
|              | 1/16, 1/8, 1/4, 1/2 | 17.54% | 41.28%    | 10.61%    | 5.43%        | 25.69%        | 19.88%       |
| Separated-Mixed | 1/4, 1/2, 1/1, 2/1 | 9.73%  | 25.62%    | 4.45%     | 3.29%        | 14.47%        | 9.90%        |
|              | 1/8, 1/4, 1/2, 1/1 | 11.07% | 27.24%    | 5.82%     | 3.49%        | 16.73%        | 10.40%       |
|              | 1/16, 1/8, 1/4, 1/2 | 10.47% | 27.24%    | 5.82%     | 3.49%        | 16.73%        | 10.40%       |
| Best run: Separated | 1/4, 1/2, 1/1, 2/1 | 14.29% | 34.01%    | 7.53%     | 4.56%        | 20.79%        | 17.39%       |
|              | 1/8, 1/4, 1/2, 1/1 | 13.32% | 33.25%    | 6.30%     | 3.68%        | 19.86%        | 18.47%       |
|              | 1/16, 1/8, 1/4, 1/2 | 14.21% | 34.42%    | 8.04%     | 3.76%        | 21.49%        | 17.79%       |

Notes: The best and the worst result of five runs have been excluded from the calculation. The first row contains the baseline metrics for the original Mask R-CNN model trained on the COCO dataset. The highest average scores are marked in bold. The last row contains the highest overall achieved result for a retrained model.
Figure 7. Comparison of Mask R-CNN results for few (a), several (b), and many (c) human figures on maps (sources: https://i.pinimg.com/736x/1a/dc/c2/1adcc2066b649919f3d28af4f2f2985b.jpg, https://i.pinimg.com/736x/00/7a/a0/007aa06e4b34aac357df25b239781ce63.jpg, https://i.pinimg.com/736x/14/d1/12/14d11277df1e2f9e66fc0d6723106c4e--illustrated-maps-manila.jpg), between the original COCO model (left) with an AP of 15.57% and the best-retrained model (right) with an AP of 19.38%. In both cases, objects are highlighted with confidence scores >0.3.
AP on average was obtained when training Mask R-CNN with the Separated dataset, where real and synthetic entities are placed on different maps. The AP is higher than the original model trained with the COCO dataset. Qualitative results (Figure 7) illustrate that more human figures could be identified on maps with the retrained model. However, the number of false positives also increased as seen on the exemplary visual results. Yet, not all figures – especially smaller ones – could be identified.

APs were lower when training Mask R-CNN with synthetic or real entities only, likewise where those entities were mixed on maps. The Separated-Mixed dataset resulted in an AP ranging in the middle between the standalone datasets. The results vary slightly for different anchor scales, however, no clear trend could be observed whether smaller or larger values are favourable. For example, the best overall AP was achieved for the largest tested anchor scale values with the Separated dataset, whereas the highest AP on average was measured for the smallest anchor scale values when training with the Mixed dataset. For the other datasets, sometimes larger, intermediate, and smaller anchor scales led to higher accuracies on average.

As training times for our CNNs on body parsing and pose estimation were lower, we could run each configuration (i.e. same dataset, same architecture) 20 times, disregarded the four highest and lowest scores, and averaged the remaining 12 scores. Quantitative results (Table 3) indicate that the highest AP on average for both tasks and the best overall accuracy was achieved for Simple UNet trained with real and synthetic persons. Simple UNet+ and Simple UNet++, which are architectures with more connections,

| Data       | Architecture | AP  | AP50 | AP75 | APsmall | APmedium | APlarge |
|------------|--------------|-----|------|------|---------|----------|---------|
| Synthetic  | Simple Deconv| 2.27%| 5.94%| 1.46%| 2.38%   | 2.77%    | 6.59%   |
|            | Simple UNet  | 0.36%| 2.03%| 0.04%| 0.36%   | 0.12%    | 2.89%   |
|            | Simple UNet+ | 1.94%| 5.14%| 1.30%| 2.14%   | 2.01%    | 8.02%   |
|            | Simple UNet++| 0.41%| 2.28%| 0.05%| 0.38%   | 1.22%    | 3.00%   |
|            | Simple UNet++| 1.99%| 5.37%| 1.24%| 2.25%   | 2.11%    | 10.91%  |
|            | Simple UNet++| 0.46%| 2.59%| 0.05%| 0.43%   | 0.45%    | 3.16%   |
|            | Simple UNet++| 1.66%| 4.64%| 0.95%| 1.92%   | 2.04%    | 8.67%   |
|            | Simple UNet++| 0.22%| 1.27%| 0.02%| 0.17%   | 0.48%    | 1.89%   |
| Real       | Simple Deconv| 10.03%| 19.23%| 9.18%| 10.65%  | 7.27%    | 3.29%   |
|            | Simple UNet  | 6.38%| 21.49%| 2.45%| 8.98%   | 4.48%    | 18.10%  |
|            | Simple UNet+ | 10.25%| 19.56%| 9.38%| 11.05%  | 6.83%    | 8.82%   |
|            | Simple UNet++| 7.77%| 25.33%| 3.30%| 10.92%  | 5.70%    | 20.69%  |
|            | Simple UNet++| 9.61%| 18.53%| 8.76%| 10.48%  | 5.91%    | 3.63%   |
|            | Simple UNet++| 7.78%| 25.04%| 3.21%| 10.96%  | 4.35%    | 20.68%  |
|            | Simple UNet++| 8.70%| 16.90%| 7.83%| 9.68%   | 5.10%    | 2.90%   |
|            | Simple UNet++| 6.05%| 20.68%| 1.90%| 8.50%   | 4.27%    | 17.42%  |
| Separated  | Simple Deconv| **10.80%**| **21.21%**| 9.69%| 11.17%  | **8.77%**| 27.12%  |
|            | Simple UNet  | 8.91%| 30.14%| 3.24%| 11.64%  | 5.05%    | 21.94%  |
|            | Simple UNet+ | **10.13%**| **33.89%**| **3.53%**| **13.49%**| **3.35%**| **23.74%**|
|            | Simple UNet++| 10.33%| 19.99%| 9.25%| 10.85%  | 7.76%    | 26.28%  |
|            | Simple UNet++| 9.81%| 32.23%| 3.35%| 13.33%  | 3.70%    | 23.33%  |
|            | Simple UNet++| 9.84%| 19.21%| 8.85%| 10.41%  | 7.39%    | 23.39%  |
| Best run:  | Simple UNet  | **12.46%**| **23.78%**| 11.23%| **13.18%**| **7.96%**| **43.49%**|
| Separated  | Simple UNet  | **13.12%**| **42.75%**| 4.25%| **16.95%**| **3.55%**| **27.60%**|

Notes: The four best and worst results of 20 runs have been removed from the calculation. The highest average scores for body parts and pose keypoints are marked in bold. The last row contains the highest achieved accuracy of all runs.
performed slightly worse, whereas the results of Simple Deconv are comparable to Simple UNet. Training the CNNs with real data only yielded similar results for parsing body parts, whereas the addition of synthetic data led to higher average accuracies for detecting pose keypoints. Training with synthetic figures only resulted in clearly lower accuracies.

Qualitative results (Figure 8) demonstrate that common poses and some of the more difficult ones (e.g. side view, overlapping or missing body parts) could be identified satisfactorily, however sometimes more challenging cases (e.g. persons viewed from behind, similar body parts) caused classification errors. Unusual poses (e.g. during sport activities) or too small figures were not recognised very well.

Discussion

While there were several training datasets available for human parsing in photos, no datasets existed for pictorial maps to our knowledge. As supervised learning with CNNs requires many training samples though, we decided to create a mixture of real persons extracted from photos and automatically generated abstract figures. Training a CNN with data from a different domain is usually less promising, however, it is not clear at this point whether a fully manually annotated dataset in our target domain would have led to better results due to the different drawing styles. In our case, the combination of real and synthetic data paid off for segmenting instances of human figures, their body parts, and estimating their poses. However, training the CNNs with synthetic data only would be not sufficient as the result metrics show. The accuracy may be higher when more variations of shapes, body features, and clothes of synthetic figures would be included. We tried to minimise the manual effort, for example, by generating a fill pattern with random polylines, but still, it is quite different from the original folds and shadows of clothes and hair. Originally, 10 body parts (incl. lower/upper arms/legs) were classified in our training datasets, but as CNNs have already struggled to segment 6 body parts in moderately complicated poses, we trained them with this number of categories. Therefore, a post-processing step (e.g. Voronoi diagram) would be required to distinguish the upper and lower parts of the limbs.

Figure 8. Selection of success cases (a), moderate failure cases (b), and severe failure cases (c) for simultaneous body parsing and pose estimation for the best-retrained model on our Simple UNet.
The achieved scores between 10% and 20% sound low, but qualitative results look already reasonable. The original Mask R-CNN model had an AP of 33% for instance segmentation of different object categories on photos, but it is not clear where the score for persons would range. Our low scores for body parts and pose keypoints may be justified by the small image sizes so that already minor deviations have a large impact. Furthermore, the COCO metrics for keypoints are not directly comparable to those of real images since they contain pre-defined standard deviations for every pose keypoint. Body proportions of pictorial figures, however, could be largely distorted. Internally, we have calculated a simpler error metric but as these results correlate with the COCO scores, we reported only the standard metric. For a comparison of the different CNN configurations, however, we relied on average results instead of giving only the best result. This methodology may be more robust concerning outliers since a configuration can be tested only a couple of times due to the long training times. When training more often, slightly higher APs than the reported ones can be achieved.

The developed CNN configurations were already the result of many trials, but a detailed evaluation of every architectural decision would go beyond the scope of this article. Instead, we like to give a brief overview of the different factors to consider: For both CNNs, it would be possible to use a different backbone network, image input sizes, learning rates, or number of learning steps and epochs. Besides those factors and the compared anchor scales, we relied on the default hyperparameters for Mask R-CNN because they already have been fine-tuned by the authors. For our CNN versions on body part parsing and pose estimation, we listed the tested possibilities and our decisions in Table 4.

**Table 4.** Possibilities to consider and decisions made for our CNN versions on parsing body parts and estimating poses.

| Possibility                                      | Our decision                                      |
|--------------------------------------------------|---------------------------------------------------|
| **Training data**                                |                                                   |
| Different ratios of body part configurations      | Proportionally to test data                       |
| Different standard deviation of 2D gaussian for    |                                                   |
| representing keypoints                           | 2                                                 |
| Ratios between image input and output size       |                                                   |
| Image format                                     |                                                   |
| Data augmentation (e.g. mirroring, noise)        |                                                   |
| Using no pre-trained model for ResNet50 (i.e.    | Using ImageNet weights for ResNet50 'he_normal'   |
| learning from scratch)                           |                                                   |
| Different kernel initialiser of convolutional     |                                                   |
| layers                                            |                                                   |
| Different conversion of the first stage of        |                                                   |
| ResNet50                                          |                                                   |
| **Initialisation**                               |                                                   |
| No augmentation                                   |                                                   |
| **Architecture: Encoder**                         |                                                   |
| Using all stages of ResNet50                     | Excluding the last stage of ResNet50              |
| Different upsampling method (e.g. bilinear        | Two-strided convolution                           |
| interpolation)                                   |                                                   |
| **Architecture: Decoder**                        |                                                   |
| Different filter numbers (e.g. fixed) and kernel  | Filters correspond to input channels, 4 × 4 kernel|
| sizes (e.g. 3 × 3) for two-strided convolutions   |                                                   |
| Additional convolutional layers after upsampling  | No convolutional layers after upsampling          |
| 'Add' operation                                   |                                                   |
| Different layer combination operation (e.g.       | Parameters of ResNet50 are adjusted during       |
| Concatenate)                                     | training                                         |
| Freezing parameters of ResNet50                  |                                                   |
| Different loss function (e.g. mean squared error) | Categorical cross entropy                         |
| Combined loss function for body parts and keypoints| Separate loss functions for body parts and        |
| Include intermediate layers in the loss function  | keypoints                                        |
| Different optimisers (e.g. Adam)                 | Consider only the final layer for the loss       |
|                                                  | function                                          |
|                                                  | RMSprop                                           |
alternatives led to worse or only marginally different outcomes compared to our parameters. We reported the scores of Simple UNet+ and Simple UNet++ to demonstrate that increasing the architectural complexity did not help to solve our problem. However, our results suggest that we have not found an optimal solution yet, only that we reached a local maximum. Further investigations, as proposed in the next chapter, will be needed to improve the quality of the outcomes.

**Conclusion and future work**

We have made the following contributions within the scope of this article:

- Creation of publicly available training datasets including annotated body parts and skeletons of real persons and synthetic figures on maps and on single images
- Creation of publicly available test datasets including annotated body parts and pose keypoints of human figures on pictorial maps and on single images
- Application of a CNN for instance segmentation of human figures on pictorial maps
- Development of CNN architectures for simultaneous prediction of body parts and pose keypoints of human figures on single images
- Qualitative and quantitative evaluation of the results of both CNNs

We measured an increased accuracy compared to the baseline model for identifying silhouettes of human figures on pictorial maps when training the CNN with real persons and synthetic figures on separate maps. The accuracy of CNNs detecting body parts and joints of human figures simultaneously was slightly higher for the simpler architectures. Here, the combination of real and synthetic data only led to a small gain in accuracy. Qualitative results showed that many figures on maps can be found, but not all. Body parts and keypoints were satisfactorily recognised for common poses by our developed CNN architectures, however, not for special cases.

Our work offers various potentials for improvement. Eventually, other datasets than PASCAL-Part with real persons could be added; unfortunately, those of DensePose (Güler et al., 2018) and Look Into Person (Liang et al., 2019) were only partly compatible. The synthetic figure generator could be extended so that a larger range of appearances and clothes is supported. Adding a generative adversarial network (GAN) may help to reduce the domain gap between training and test data (e.g. Sankaranarayanan et al., 2018). The number of annotated maps in the test dataset could be also increased as well as a distinction could be made between visible and hidden keypoints. Ideally, only one CNN is needed to fulfil all three tasks (i.e. instance segmentation, body part parsing, pose estimation). Newer feature extractors than ResNet, for instance, HR-Net (Sun et al., 2020), may lead to a higher accuracy. Body part predictions may be smoothed by the addition of conditional random fields (Arnab et al., 2018).

The results of our work may be transferable to recognise human figures in books, comics/manga, or paintings. Extracting additional properties of figures (e.g. age, gender, skin colour, pieces of clothing, performed activity, and spatial relation to other figures) would be helpful for an even fine-grained search. Beyond, it would be interesting to identify other pictorial entities like animals or means of transportation. When figures were displaced from their original position, for example when being animated, an
empty background would remain. This is a semantic image inpainting task that could be tackled by GANs (e.g. Yeh et al., 2017).

Notes

1. http://narrat3d.ethz.ch/detection-of-pictorial-map-objects-with-cnns/
2. https://www.fedlex.admin.ch/eli/cc/1993/1798_1798_1798/en#art_24_d
3. http://roozbehm.info/pascal-parts/pascal-parts.html
4. http://human-pose.mpi-inf.mpg.de/
5. https://www.iconfinder.com/
6. https://github.com/tensorflow/models
7. https://cocodataset.org/#detection-eval
8. https://www.tensorflow.org/api_docs/python/tf/keras
9. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md
10. https://cocodataset.org/#keypoints-eval

Acknowledgements

The authors express their gratitude to Nicolas Steinmann for annotating the test data. Special thanks go to Stefan Räber for his continuous motivation while working on this article.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by ETH Research Grant ETH-11 17-1. It is an university internal grant, so it does not appear in the Open Funder Registry.

Data availability statement

The training and test datasets, the retrained models with the highest accuracies, and the source code are openly available at: http://narrat3d.ethz.ch/

Notes on contributors

Raimund Schnürer has worked as a research assistant at the Institute of Cartography and Geoinformation of ETH Zurich for the ‘Atlas of Switzerland’, the Swiss national atlas. In 2018, he started a doctorate at the same institute in the project ‘Storytelling with Animated Interactive Objects in Real-time 3D Maps’.

A. Cengiz Öztireli is a senior lecturer (associate professor) at the University of Cambridge. He worked as a research scientist at Disney Research, and a senior research associate at ETH Zurich before joining Cambridge. His research interests are in computer graphics, vision, and machine learning/artificial intelligence with a focus on 3D digital reality.

Magnus Heitzler is a postdoctoral researcher at the Institute of Cartography and Geoinformation of ETH Zurich, where he is mainly working on visualisation and machine learning related topics of cartography. During his doctorate, he worked on visualisation and simulation techniques in the context of natural disaster risk assessments of civil infrastructure networks.
René Sieber is a senior scientist at the Institute of Cartography and Geoinformation of ETH Zurich. In 1995, he became project manager of the ‘Atlas of Switzerland’ and administered four digital versions, of which the latest release was an online application. Since 2015, he is also the chair of the ICA commission on atlases.

Lorenz Hurni is a full professor of Cartography at the Institute of Cartography and Geoinformation of ETH Zurich since 2003. He is managing editor-in-chief of the ‘Atlas of Switzerland’ and the ‘Swiss World Atlas’. Currently, he is vice rector for study programmes at ETH Zurich.

ORCID

Raimund Schnürer  http://orcid.org/0000-0002-7558-3621
A. Cengiz Öztireli  http://orcid.org/0000-0002-4700-2236
Magnus Heitzler  http://orcid.org/0000-0002-9021-4170
René Sieber  http://orcid.org/0000-0003-0780-5808
Lorenz Hurni  http://orcid.org/0000-0002-0453-8743

References

Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kirillov, A., Savchynskyy, B., Rother, C., Kahl, F., & Torr, P. H. S. (2018). Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. IEEE Signal Processing Magazine, 35(1), 37–52. https://doi.org/10.1109/MSP.2017.2762355
Cao, Z., Simon, T., Wei, S., & Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1302–1310. https://doi.org/10.1109/CVPR.2017.143
Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834–848. https://doi.org/10.1109/TPAMI.2017.2699184
Child, H. (1956). Decorative maps (first edition). Studio Publications
Davies, S. (2016). Renaissance ethnography and the invention of the human: New worlds, maps and monsters. Cambridge University Press. https://doi.org/10.1017/CBO9781139568128
Güler, R. A., Neverova, N., & Kokkinos, I. (2018). DensePose: Dense human pose estimation in the wild. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7297–7306. https://doi.org/10.1109/CVPR.2018.00762
He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV), 2980–2988. https://doi.org/10.1109/ICCV.2017.322
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90
Heitzler, M., & Hurni, L. (2020). Cartographic reconstruction of building footprints from historical maps: A study on the Swiss Siegfried map. Transactions in GIS, 24(2), 442–461. https://doi.org/10.1111/tgis.12610
Liang, X., Gong, K., Shen, X., & Lin, L. (2019). Look into person: Joint body parsing & pose estimation network and a new benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(4), 871–885. https://doi.org/10.1109/TPAMI.2018.2820063
Lin, K., Wang, L., Luo, K., Chen, Y., Liu, Z., & Sun, M. (2020). Cross-domain complementary learning using pose for multi-person part segmentation. IEEE Transactions on Circuits and Systems for Video Technology, 1066–1078. https://doi.org/10.1109/TCSVT.2020.2995122
Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars
Liu, D. S.-M., Cheng, C.-I., & Liu, M.-L. (2020). Animating characters in Chinese painting using two-dimensional skeleton-based deformation. *Multimedia Tools and Applications, 79*(27), 20343–20371. https://doi.org/10.1007/s11042-020-08842-5

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8759–8768*. https://doi.org/10.1109/CVPR.2018.00913

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer vision – ECCV 2016* (pp. 483–499). Springer International Publishing. https://doi.org/10.1007/978-3-319-46484-8_29

Oliveira, G. L., Valada, A., Bollen, C., Burgard, W., & Brox, T. (2016). Deep learning for human part discovery in images. *2016 IEEE International Conference on Robotics and Automation (ICRA), 1634–1641*. https://doi.org/10.1109/ICRA.2016.7487304

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Saeedimoghaddam, M., & Stepinski, T. F. (2020). Automatic extraction of road intersection points from USGS historical map series using deep convolutional neural networks. *International Journal of Geographical Information Science, 34*(5), 947–968. https://doi.org/10.1080/13658816.2019.1696968

Sankaranarayanan, S., Balaji, Y., Jain, A., Lim, S. N., & Chellappa, R. (2018). Learning from synthetic data: Addressing domain shift for semantic segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3752–3761*. https://doi.org/10.1109/CVPR.2018.00395

Schnürer, R., Sieber, R., Schmid-Lanter, J., Öztireli, A. C., & Hurni, L. (2020). Detection of pictorial map objects with convolutional neural networks, *The Cartographic Journal, 1*–19. https://doi.org/10.1080/00087041.2020.1738112

Sun, K., Xiao, B., Liu, D., & Wang, J. (2020). Deep high-resolution representation learning for human pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 5693–5703*. https://doi.org/10.1109/TPAMI.2020.2983686

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., & Schmid, C. (2017). Learning from synthetic humans. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4627–4635*. https://doi.org/10.1109/CVPR.2017.492

Weinman, J., Chen, Z., Gafford, B., Gifford, N., Lamsal, A., & Niehus-Staab, L. (2019). Deep neural networks for text detection and recognition in historical maps. *2019 International Conference on Document Analysis and Recognition (ICDAR), 902–909*. https://doi.org/10.1109/ICDAR.2019.00149

Zhang, J., Malmberg, F., & Sclaroff, S. (2019). Salient object subitizing. In J. Zhang, F. Malmberg, & S. Sclaroff (Eds.), *Visual saliency: From pixel-level to object-level analysis* (pp. 65–93). Springer International Publishing. https://doi.org/10.1007/978-3-030-04831-0_5

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2020). UNet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging, 39*(6), 1856–1867. https://doi.org/10.1109/TMI.2019.2959609