



Label Placement Challenges in City Wayfinding Map Production—Identification and Possible Solutions

Lars Harrie¹ · Rachid Oucheikh¹ · Åsa Nilsson² · Andreas Oxenstierna² · Pontus Cederholm¹ · Lai Wei¹ · Kai-Florian Richter³ · Perola Olsson¹

Accepted: 16 May 2022 / Published online: 25 May 2022
© The Author(s) 2022

Abstract

Map label placement is an important task in map production, which needs to be automated since it is tedious and requires a significant amount of manual work. In this paper, we identify five cartographic labeling situations that present challenges by causing intensive manual work in map production of city wayfinding maps, e.g., *label placement in high density areas*, *utilizing true label geometries in automated methods*, and *creating a good relationship between text labels and icons*. We evaluate these challenges in an open source map labeling tool (QGIS), provide results from a preliminary study, and discuss if there are other techniques that could be applicable to solving these challenges. These techniques are based on quantified cartographic rules or on machine learning. We focus on deep learning for which we provide several examples of techniques from other application domains that might have a potential in map label placement. The aim of the paper is to explore those techniques and to recommend future practical studies for each of the identified five challenges in map production. We believe that targeting the revealed challenges using the proposed solutions will significantly raise the automation level for producing city wayfinding maps, thus, having a real, measurable impact on production time and costs.

Keywords Map labeling · Map production challenges · City wayfinding maps · Automated cartography · Deep learning · Image synthesis · Generative adversarial networks

Introduction

Label placement is an important task in map production that requires a substantial amount of manual work and time. To reduce this time and enhance visual, informative, and esthetic quality of the maps, numerous studies have been carried out on automatic map labeling (see Wolff and Strijk (2009) for an overview of early studies). Even though several labeling problems have satisfying solutions (such as how to find optimal solutions for point placement on small scale maps), the automation level of map labeling in production is still low. This low automation rate is likely due to several

reasons. Firstly, adequate methods may not have been developed to solve the labeling challenges that occur in a production environment. Secondly, current label placement tools might not implement the best methods available. Thirdly, map producers might not entirely utilize the capability of the map labeling tools. Fourthly, the data structures used for the cartographic data are not sufficient to support the best methods/tools available. Most likely, the current low degree of automation in map labeling in production is caused by a combination of these reasons.

In map production, text labels and icons are often placed simultaneously since there are dependencies between how they are placed. Therefore, in this study, we include placement of both text labels and icons. In the paper, the terms *labels* and *map labeling* include both (placement of) text labels and icons.

Most research in map labeling is based on quantifying rules found in the cartographic literature, often based on seminal work such as Imhof (1975) and Wood (2000). This approach has been successful in the sense that rule-based systems and optimization techniques have been developed

✉ Lars Harrie
lars.harrie@nateko.lu.se

¹ Physical Geography and Ecosystem Science, Lund University, Sölvegatan 12, 223 62 Lund, Sweden

² T-Kartor, Olof Mohlins Väg 12, 291 62 Kristianstad, Sweden

³ Department of Computing Science, Umeå University, SE-90 187, Umeå, Sweden

and implemented in tools, but it has not solved several challenges in map production. In an era of increasing use of machine learning in many application domains, an obvious question is if and how machine learning and precisely deep learning could be applied to map labeling. This question boils down to whether we can utilize cartographic knowledge of map labeling implicitly present in map examples to train, e.g., a neural network to perform map labeling of good quality, or at least to evaluate if a map labeling is appropriately conducted. This is, in our view, still an open question that we in this study address and give insight to but not fully answer.

The paper has two main aims: the first is to identify cartographic labeling challenges, occurring in a map production environment that cannot be solved by current label placement tools. The second aim is to discuss whether there are published methods that might be useful to solve these challenges and/or if deep learning methods could be applicable. Based on this, we formulate recommendations for further studies. The paper starts with describing rules of map labeling in a production environment, with a focus on city wayfinding maps. Then, follows an introduction to deep learning and its potential use in map labeling. In the following sections, some challenging cartographic labeling situations, occurring in production of city wayfinding maps, are described. This part also includes descriptions of several methods that potentially could be useful for solving the labeling challenges, including rule-based and deep learning methods. The paper ends with some concluding remarks.

Map Labeling Rules for City Wayfinding Maps

General Label Placement Rules

Map labeling rules concern the whole labeling process which includes the following: (1) the choice of labels to show and their classification, (2) determination of font characteristics, and (3) label placement (Yoeli, 1972). In this study, we are mainly interested in the placement of labels, for which several general rules must be obeyed (for more details, see Imhof 1975; Wood 2000; van Dijk 2002; Rylov and Reimer 2015):

- Legibility: a label is not allowed to overlap another label.
- Association: it should be easy to interpret which map object a label refers to, hence avoid placing labels too close to other objects.
- Map readability: If the labels must be placed on top of map objects, they should not cover important features of those objects (and ideally only overlap homogenous

- areas and less important objects). Furthermore, the map objects should not disturb the interpretation of the labels.
- Esthetics: The labeling should contribute to an overall esthetic map.

These rules are applicable for map labeling of all types of maps. To fulfill these rules, as well as other cartographic aspects, there are more specific rules defined for a specific cartographic product. In this study, we focus on city wayfinding maps.

Production Rules for Label Placements in City Wayfinding Maps

In this study, we focus on city wayfinding maps¹ in London. City wayfinding maps provide directional information in complex urban environments in such a way that they can be easily interpreted by pedestrians and cyclists. The rules considered for label placement are based on design standards produced by Transport for London² as well as internal labeling rules from the mapping company T-Kartor.³ Even though the cartographic rules are for a specific cartographic product (city wayfinding map for London), the main content is largely generally applicable (and generally follows recommendations found in, e.g., Imhof 1975). The intention here is not to provide a complete list of label placement rules, rather to provide an outline of the rules as a base for discussion about limitations in the available algorithms/tools (see Appendix 1 for detailed rules). In short, the following rules (and their exceptions) apply.

- Point feature labeling: generally, point feature labels should be horizontal and ideally above to the right of the point (see, e.g., Slocum et al., 2005). In city wayfinding maps, most point objects are in fact represented by icons and some types of these icons are not allowed to be moved. If there is not enough space, callouts are used (Fig. 1).
- Line feature labeling: line feature labels, e.g., for roads, are to be placed within the road area. Straight parts of a road are preferable for labels due to readability; if not possible, the label shape needs to adapt to the shape of the feature. Labels can also be wrapped into two (or more) lines, or shortened, to make them fit. For long line features, labels are repeated.

¹ <https://www.t-kartor.com/our-services/city-wayfinding/>

² Transport for London (2009) Street map design standard—Issue 1. Transport for London (2011) Street map design standard—Issue 2.

³ <https://www.t-kartor.com/>



Fig. 1 The taxi icons are an example of point feature label placement, where the point is not visible, and the icon is placed on the point location; alternatively, a callout is used as for the top left taxi station icons. © Copyright Transport for London

- Hyphenation and other text manipulations: For a label text, the following priorities should be used: (1) complete text in one unit, (2) shortened text in one unit, (3) text divided into two units, and (4) text divided into two or several rows. For city wayfinding maps, there is a list of allowed abbreviations that can be used. Also, under several restrictions, font size may be changed to obtain optimally looking labels.
- ## Map Labeling Based on Deep Learning
- In this section, we discuss the potential of deep learning methods for map labeling. After a brief introduction to deep learning and its applications, we provide a more general outlook on how deep learning may contribute to achieving the key elements in good label placement. Deep learning may possibly also be utilized for improving the evaluation step in label placement, especially for evaluating cartographic rules that are difficult to quantify, e.g., map readability.
- ### Introduction to Deep Learning
- Machine learning techniques have experienced a prosperous development in recent years in several application fields such as image recognition (Ohri et al. 2021), image classification (Zhao and Du 2016), and robot technology (Levine et al. 2018). Classical machine learning techniques may achieve acceptable performance but require tedious feature engineering, in contrast to deep learning techniques, and particularly convolutional neural networks (CNN) and the learning mechanisms such as attention, adversarial, and spatial transformation. CNN include convolutional layers stacked on top of each other and each layer is capable of recognizing more sophisticated features and generating feature maps. The fully connected networks are prone to overfitting if not regularized as each neuron in one layer is connected to all neurons in the next layer. With CNN, regularization is achieved by exploiting the hierarchical patterns in their input data by employing increasingly complex filters or kernels on the data with increasing network depth. Much research has gone into optimizing the network design to increase the performance of learning specific tasks and to solve some technical issues such as overfitting, vanishing gradient problem, and under-specification. This leads to efficient model architectures such as Faster-R-CNN, U-Net, YOLO, SSD, FPN, or Inception (Dhilon and Verma 2019).
- One type of deep learning models increasingly applied in many learning applications and of interest to map labeling is generative adversarial networks (GAN). A GAN includes two networks trained in contest: the generative network generates new samples and learns to map from a latent space to a given data distribution, while the discriminative network
- Area feature labeling: preferably, area labels should be completely placed within the polygon feature they represent, wrapping text into several lines if necessary. But if unavoidable, area labels may cross the polygon boundary. Labels should be horizontal and aligned according to their relation to the polygon feature (e.g., left alignment if placed more to the right of the feature). City wayfinding maps also contain area labels for administrative areas (e.g., neighborhoods), using large fonts, opacity, and large space between characters. Ideally, these labels do not overlap other labels, but in practice, this is hardly avoidable and, thus, overlap is allowed as long as it does not harm map readability.
 - Icons: most icons relate to a specific location on a street (e.g., a bus stop). Icons come with an arrow that needs to point to the true location on the street. Ideally, icons are placed in a 90-degree angle to the corresponding street, but other angles are possible if necessary for avoiding overlaps with map features or other labels. Icons should not overlap roads, but may overlap buildings if necessary. Icons representing (parts of) area features should align with the parts they represent and text labels of these features, respectively. Exceptions are possible if there is no other solution.
 - Label overlap and removal: In short, the first rule is that no text labels and icons may overlap, and the second rule is that it is not allowed to remove a text label or icon. Clearly, these rules often result in conflicts that require exceptions, e.g., for text labels to overlap icons or buildings they do not represent as long as it is still clear which building each label corresponds to.

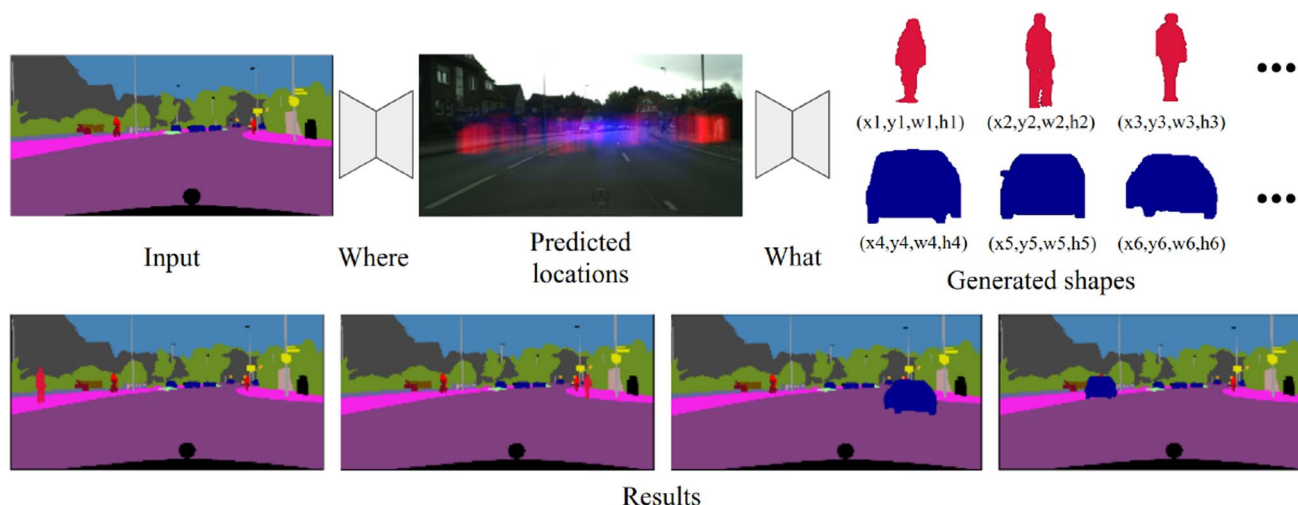


Fig. 2 Context-aware placement of objects (cars and pedestrians). By example images, an extended GAN network is trained to learn the context of where cars and pedestrians (foreground objects) are

located in the background image. The network can then produce the synthetical images shown as result in the bottom row. Source: Lee et al. 2018

evaluates the generated samples and distinguishes them from the true data distribution (Goodfellow et al. 2014). These two networks play a minimax game which, if its equilibrium is reached, results in very good performance, e.g., generating highly realistic looking images.

GAN are relevant in this context because the problem of placing labels on maps lies in the intersection of vision and language and can be formulated as an image synthesis problem. The two most interesting approaches for image synthesis are image composition and image translation. Image composition aims to synthesize new images by placing foreground objects into an existing background image (Lee et al. 2018; Fig. 2). The foreground objects in our case are the labels that should be placed in the background image, i.e., the map, at semantically sensible regions. To achieve synthesis realism and to generate labeled maps similar to the manually labeled dataset, some techniques and networks mentioned below can be used to learn and control certain parameters such as text locations within the background image, geometric transformation of the foreground texts, and blending between the foreground text and background image. On the other hand, image-to-image translation aims to find a mapping from one visual domain to another and to learn the required transformations to perform on images from one domain so they have the features of images from another domain.

There are, however, some inherent problems of using many deep learning techniques in map labeling since they rely on image-to-image translations. These translations only focus on the synthesis of appearance features (here the label) by learning the style of images of the target domain. Generally speaking, a solution to the label placement problem should include both synthesis realism in the geometry

domain (alignment, etc.) and the appearance domain (the text itself as well as the relation to the background map). A geometry synthesizer needs to learn the local geometry of background images (maps) consisting of the roads, buildings, etc. on which the labels representing our foreground objects (labels) can be transformed and placed. To which extent this is possible is further elaborated on below.

Earlier Studies on Machine Learning in Label Placement

Pokonieczny and Borkowska (2019) utilized machine learning to determine feature labeling in topographic maps. They trained a network with input terrain coverage data and labels from several maps to determine in which rectangle a label should be placed around a feature. They achieved up to 80% correctly placed labels which made it possible to reduce manual editing by 50%.

Li et al. (2020) developed a deep learning methodology for placing area feature labels. A common strategy in 43, implemented in several GIS programs, is to place the label on top of the centroid of the polygon that defines the area. However, for many polygonal shapes, this strategy is not cartographically satisfying, and in map production, cartographers manually select other positions. Furthermore, it is difficult to formalize what is a good position for an arbitrary polygon shape. Li et al. (2020) utilized data to train a stacked hourglass network to produce a heatmap that indicates a good position of the area label. The methodology was applied to map labeling of property units in a cadaster map and yielded relatively good results.

It should be noted here that neither of these two studies is concerned with conflicting labels and overlap of other

map features only plays a very minor role here; in other words, they treat quite simple labeling tasks. Therefore, their methodologies are likely not extendable to more general and/or more complex labeling situations.

Potential Deep Learning Techniques for Automated Label Placement

In this section, we describe some deep learning techniques that are related to the key elements of good label placement. The approach is to formulate the problem of text placement as a learning task, and then to explore deep learning techniques from other domains that share similar issues and, this way, to identify the appropriate approaches that may be pursued further.

Legibility

Legibility in map labeling mainly concerns avoiding labels to overlap, which can be facilitated by a saliency-based method (Vilaplana 2015). The saliency model computes a saliency map for a given image such that homogeneous image regions usually have lower saliency. Then, a predefined threshold on the resulting saliency map will determine the appropriate locations for text placement. This saliency guidance helps to find the right locations for texts within the semantically sensible regions or at least to improve the identified candidate locations while avoiding collisions with other objects.

Another deep learning method that may be interesting for label placement is the image text quality assessment (ITQA for short) which aims to evaluate the image quality with a focus on text as it computes the quality score of an image through predicting the degree of degradation at textual regions. Furthermore, Li et al. (2018) proposed a method based on ResNet to perform image text quality assessment, which is composed of three stages: text detection, text quality prediction, and weighted pooling of the quality of all detected text lines. Other methods can learn from ranked datasets such as user rankings (Liu et al. 2017). Siamese networks are trained on ranked sets and transfer this learning to a CNN that performs the absolute legibility assessment. Another related application is image stitching in which the overlapped objects should be detected so that they can be stitched and generate a wide field of view image. Lyu et al. (2019) claimed in their survey that feature-based methods have dominated image stitching and that learned CNN features are more flexible, and more potential matched candidates could be extracted from images with wide baseline or low-texture regions.

Association

There is a challenge to model associations in deep learning applied to raster maps. The raster map features alone might be insufficient to capture the relation between objects and their labels. The same applies to the practical level since learning object-centric representations from pixels is not efficient for complex tasks in which it is required to encode fine-grained locations, orientations, and complex composition of objects. However, several semantic and context-based methods have been developed in the deep learning domain that could be applicable for label placement.

Lee et al. (2018) developed a model for context-aware synthesis and placement of object instances that can simultaneously determine locations to place an object in a scene, and its appearance, i.e., scale and shape, or pose, given a semantic mask. They used an architecture that consists of two GAN modules and spatial transformation networks (STN). An STN is a special type of CNN capable of making geometric transformations on images and generating realistic looking ones by limiting the space of possible outputs to a low-dimensional geometric transformation of real images. Using only GAN can produce images of remarkable complexity and realism but may potentially ignore the explicit spatial interaction between multiple entities present in the image. That is the reason for introducing ST-GAN and using it for image composition tasks in both paired and unpaired settings (Lin et al. 2018). Volokitin et al. (2020) developed a method for the automatic determination of plausible locations for object placement into images using masked convolutions which compute feature maps for left, right, top, and bottom contexts just once per image and thus learn the spatial context of different image regions.

Readability

Map readability can be evaluated by detecting the occlusion (overlap) in the final maps. By using de-occlusion techniques which aim to recover and complete the invisible parts of occluded objects, we can ensure that no important features are hidden. In addition, saliency models could be useful to identify the attention points or regions that people would focus on and important objects that should be not occluded. Saliency feature learning was used to increase readability of posters, which are very informative, but they are usually viewed only for a few seconds (Fang et al. 2020). The used data are collected from eye-tracking experiments and the evaluation is done using specific metrics such as time to first fixation and observation length. The same techniques are used for natural scenes data in order to identify the most noticeable objects which attract human attention. Fang et al. (2020) evaluated the capabilities of six state-of-the-art models on natural scene content (i.e., text or characters)

to find salient regions and generate saliency maps. The use of custom loss functions can enhance the readability of the obtained maps. For example, using a repulsion loss can help to keep away the labels from each other by penalizing the generated samples with small spaces between the labels.

Esthetics

There are some examples of deep learning studies in cartography addressing esthetics, mainly focusing on cartographic generalization (Zhou and Li 2017; Touya et al. 2019; Feng et al. 2019; Courtial et al. 2020). For example, Courtial et al. (2020) explored deep learning techniques for mountain road generalization, where a U-Net network was trained on raster images of road objects in the Alps. The authors conclude that the network achieves smoothing, enlargement, and caricature operations on the mountain road objects in most of the cases, but they mentioned that the result is not as good as the reference data (i.e., the production data at IGN, the French mapping authority).

In art, Cetinic et al. (2019) investigated scoring artistic images according to three subjective aspects of human perception: esthetic valuation, received sentiment, and memorability. Their experiments were performed using different decision trees and CNN models on image features related to the content, composition, and color of digitized fine art collections. For each concept, they evaluated several different CNN models trained on various natural image datasets and select the best performing model based on the qualitative results and the comparison with existing subjective ratings of artworks. They conclude that CNN models pre-trained on natural images can learn and extract meaningful esthetic, memorability, and sentiment features in art images.

Some Challenges in Label Placement

In the production of city wayfinding maps (at T-Kartor), map labeling is a substantial part of the manual handling. Some tools have been evaluated to increase the automation level, but so far, no satisfying solution has been found. One reason might be that the requirements of the city wayfinding maps are somewhat unique and therefore hard to automate using standard tools. This situation is also worsened by the fact that the best cartographic solution is sometimes a violation of one or several of the requirements (simply because it is impossible to place all labels adhering to the complete list of requirements). We do, however, believe that the challenges for the labeling of city wayfinding maps are to a large degree shared with the labeling of other types of high-quality maps with dense information content.

In the following sections, we describe some labeling challenges that cause much interactive work in the production of

city wayfinding maps. These challenges have been identified together with cartographers at T-Kartor. We also look into and discuss if there are map labeling methods and/or deep learning techniques that potentially could be useful in these situations, as well as perform some tests. To illustrate the label placement challenges, we use two types of city wayfinding maps. The first type are production maps created by the company T-Kartor. These maps are produced in an ESRI ArcGIS environment using the Maplex label engine and substantial manual label (annotation) editing both in the ArcGIS environment and in the publishing tool Adobe Illustrator. The second type are maps created by us in the open source program QGIS⁴ or in the Maplex label engine with the same input data as for the production maps. Details of the QGIS map labeling tool are given in Appendix 2 (see also Ertz et al. 2009). The Maplex label engine is a rule-based system that is integrated into the ESRI environment.⁵ Maplex is extensively used and has shown to produce good results for several map types (see, e.g., the evaluation in Kern and Brewer 2008).

Challenge 1: Label Placement in High-Density Areas

Problem Identification

High-density areas are characterized by a scarcity of space for both map features and labels (Figure 3). To cope with this, cartographers often manually find solutions that are a compromise between wanted properties of the map. One particular challenge in high-density areas is to define priorities between the labels, especially since it is not possible to state that one label type always should be in priority over another label type. Referring to Figure 3a, we can identify that the area label (representing the landmark building) *The Original London Visitor Centre* has been prioritized (by the cartographer) before the line label *Cockspur street* (that was divided into two lines which is not an optimal solution according to the labeling rules). On the other hand, the area label *The Ambassadors Theatre* (in Figure 3b) is moved from its ideal placement where the main part of the label is in fact placed on another side of a street (which is not recommended from an association perspective) to allow space for the line labels *West St* and *Tower Ct*.

Figure 4a shows text labels placed automatically in QGIS, while the icons are placed manually (identical to Fig. 3a). The label placement is affected by how the parameters are

⁴ <https://qgis.org/>

⁵ <https://desktop.arcgis.com/en/arcmap/latest/map/working-with-text/working-with-the-maplex-labeling-engine.htm>

Fig. 3 a, b Map samples of high-density areas. Labels are added manually. The maps are produced by T-Kartor. © Copyright Transport for London.

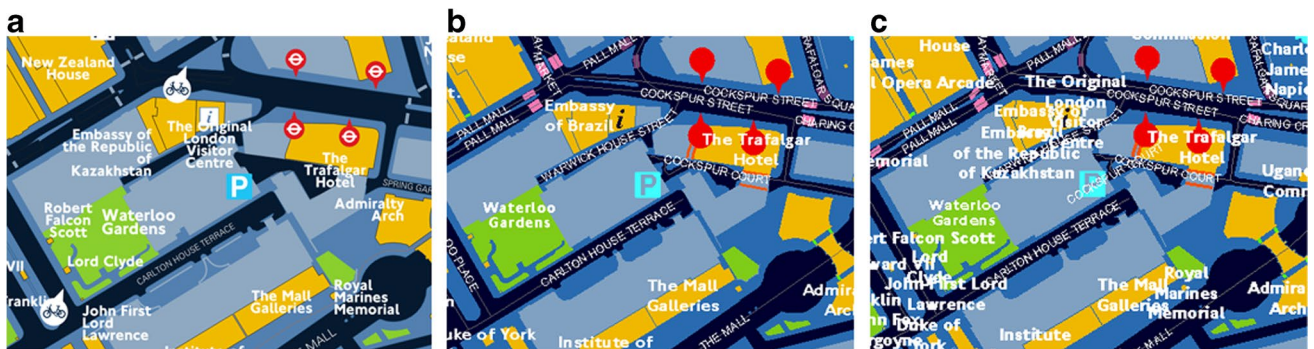
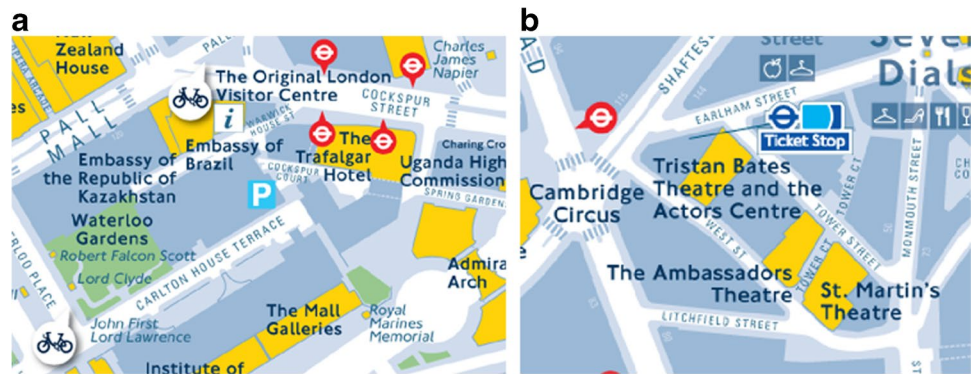


Fig. 4 Map samples of high-density areas. Labels are placed automatically in QGIS (a) and in ESRI Maplex Label Engine (b, c). In c, overlapping labels have been allowed; this type of map is used as the starting point for the manual editing by T-Kartor. The area is the same as in Fig. 3a

set in QGIS (e.g., which type of area labels are used, priorities of labels, if labels are allowed to overlap other objects). For us, it turned out to be difficult to find a set of parameters that utilize the available space in such a good way that was done manually in Fig. 3a. Some shortcomings in the map in Fig. 4a overlap between text labels and icons and that some text labels (e.g., *The Original London Visitor Centre*) had a fixed form that made it not possible to find a better location that could allow also other labels to be shown (e.g., *Embassy of Brazil*). Also, the parameter setting used was not optimal to show all the road labels. Figure 4b shows the same area where the map labeling is conducted by the Maplex tool. Both QGIS and Maplex create satisfying labeling in terms of readability. The main problem is that the tools are not capable of placing all the labels. This omission could be acceptable in many map services, but not in the city wayfinding map which has a requirement that all labels are present. The question then boils down to whether QGIS and Maplex are useful tools for placing a majority of the labels and the rest then being placed manually. In the production environment for city wayfinding maps, they have concluded that, at least in dense areas, the labeling tools do not provide good enough solutions. In other words, the proposed solutions (in Fig. 4a,b) do not provide any time savings in map production. The little support from the automatic tools can be

illustrated by comparing Fig. 4a and Fig. 4b with the manually made labeling in Fig. 3a; there are quite a few labels that are not moved and/or changed (more lines) between these maps. Instead, T-Kartor produces a labeling solution where all labels are present and then start the manual work from there. In Fig. 4c, such a map is generated in Maplex (where overlap between labels has been allowed). From this map, some placements of road labels are saved but almost all other labels have to be moved (and in some cases also divided into several lines).

Rule-Based Techniques

Labeling dense areas is a well-known challenge in automated label placement. Early studies of Doerschler and Freeman (1992) aimed at improving rule-based systems for automatic label placement to cope with high-density maps but it turned out to be difficult to utilize the available space for the labels. One improvement was the introduction of the slider model which allowed a more flexible label placement, not restricted by a fixed number of solutions (van Kreveld et al. 1999; Strijk and van Kreveld 2002). Also, optimization techniques (e.g., developed by Christensen et al. 1995 and Zoraster 1997) have shortcomings to deal with high-density areas. Much of the development of optimization techniques

have concentrated on finding a solution with most added (point) labels, and have not addressed the cartographic challenges in high-density maps concerning, e.g., the association property; see, e.g., Rylov and Reimer (2014) who address this issue utilizing a multicriteria optimization technique. Haunert and Wolff (2017) argue that the association criteria must be strengthened in their development of a new integer linear programming approach. However, it should be noted that both Rylov and Reimer (2014) and Haunert and Wolff (2017) only deal with point labels, which is the usual case for label placement optimization research. This is not adequate for labeling a high-density map as in Fig. 3a. What we can see in this map, especially in the upper part, is that the point and line labels have to fight for the same space, and therefore, it is almost impossible to find a good cartographic solution if the label types are treated independently. To improve this situation, Lu et al. (2019) developed a unified framework for placing all types of labels; this framework is based on a hybrid algorithm combining discrete differential evolution and genetic algorithms. However, as far as we know, there is no available tool, commercial or open source, that has a common framework for all label types.

To circumvent adding labels to high-density areas, a leader approach could be utilized. In this approach, the label is placed outside the area and a leader connects the label to the feature, as done for the Ticket shop icon in Fig. 3b. The labels could then be placed either in the map or just outside the border of the map. For the latter case, Kindermann et al. (2015) developed an efficient algorithm that creates a planar solution (guaranteeing no overlaps of the leaders) where the labels are allowed to be placed along two borders.

Another approach in high-density areas would be to perform a selection of data that should be labeled. We have not found any research on this for city wayfinding maps, but for other types of maps. For example, Brewer et al. (2013) provided an automated method for adaptive thinning of road features and road labels suitable (for multiscale design) which removes features by a feature hierarchy and network connectivity while preserving many urban/rural local density patterns. Also, Raposo et al. (2017) perform selection of labels in a multi scale context targeting summits (point data) in hydrological datasets. The latter study is using a tessellation approach (where restrictions are set for the labels in each cell in the tessellation) which could be of interest for a city wayfinding map.

Furthermore, label placement, of, e.g., high-density areas, could utilize an automated evaluation step. This could be implemented by computing several candidate solutions in the first step, and then in the evaluation step, select the best one according to certain criteria. But even the best identified solutions could include some labels that are not placed satisfactorily, and in these cases, the evaluation step could identify which of the labels need interactive improvement.

From a practical perspective, this identification would save much labor time since cartographers would not be required to manually inspect all labels from the automated solution (see, e.g., Klute et al. (2019) for a practical implementation of semi-automated map labeling). Analytical evaluation of map labeling was studied by van Dijk (2002) who quantified several map labeling rules to form a label quality function used for evaluation (for a practical use of a similar framework, see Kern and Brewer (2008)).

Deep Learning Techniques

In the deep learning domain, there are some interesting techniques that could be applied for high-density areas. As mentioned above, Lee et al. (2018) developed a model for context-aware synthesis and placement of objects. Closer to that, Volokitin et al. (2020) developed a method to automatically determine plausible locations for object placement into images considering the surrounding context. Such approaches can be useful to simultaneously determine the location to place the labels on the map, and their appearances, i.e., font and shape so as to avoid occlusion and overlaps.

Associations are important in map labeling in high-density areas. Association is linked to the concept of semantic coherence, since both concerns that the text should be placed at semantically sensible regions within the background images. To learn this pairing, Zhan et al. (2021) used semantic image segmentation datasets to classify image regions into two lists where one list includes only image regions that are semantically sensible for text embedding and the other include those which are not semantically sensible for text embedding. However, most current image composition systems deal only with one foreground object, while map labeling in dense areas deals with multiple foreground objects (labels). To include multiple foreground objects, hierarchical composition techniques have been developed (see, e.g., Zhan et al. (2021)).

If a GAN is used for map labeling in high-density areas, the formulation of the adversarial loss function (that models the difference between the original target image and the generated one) is important. In order to measure the error of the automatic label placement relatively to the original target image (manually labeled), the objective criteria used for the evaluation of the automatic segmentation methods can be used. Applicable loss functions in this case can be the overlap-based losses such as the Dice similarity coefficient or Jaccard index, or spatial distance-based ones such as mean boundary distance or Hausdorff distance (Wang et al. 2020a). In addition, the core network for the GAN discriminator should be well chosen. The basic discriminator is trained as a binary classification model to predict the probability that a given image is real. However, in a WGAN, the

output is a score of “realness” for a given image. Therefore, instead of playing the role of classifier and using loss functions such as binary cross-entropy, the WGAN model uses a new loss function that pushes the discriminator to predict a precise score.

Preliminary Study: Using Deep Learning for Label Placement Evaluation

We created and assessed an evaluation framework for map labeling of high-density city wayfinding maps using deep learning (Fig. 5) (see Wei (2020) for details). The deep learning part was implemented in GoogLeNet (Szegedy et al. 2015) and trained by manually created map labeling examples. The map examples were of size 256×256 pixels (as required by GoogLeNet) in scale 1:2250 tailored for the learning task (Fig. 6). All the training map samples were manually classified into three quality classes (good, moderate, and bad) based on the categories *legibility*, *disturbance*, and *association*. In total, 2400 map samples were used, 1500 for training and 900 for validation (with an equal amount for all three quality classes). The trained network was then used to evaluate map samples where the map labeling had been automatically generated by QGIS (for details, see Cederholm 2020).

The idea was then that the trained network should be able to evaluate map samples with automated map labeling conducted in QGIS. However, it turned out that the framework was not able to perform an acceptable evaluation of the test map samples, but rather that the framework identified all input images as poor quality. It turned out that this initial test of performing map labeling evaluation using deep learning has at least four shortcomings:

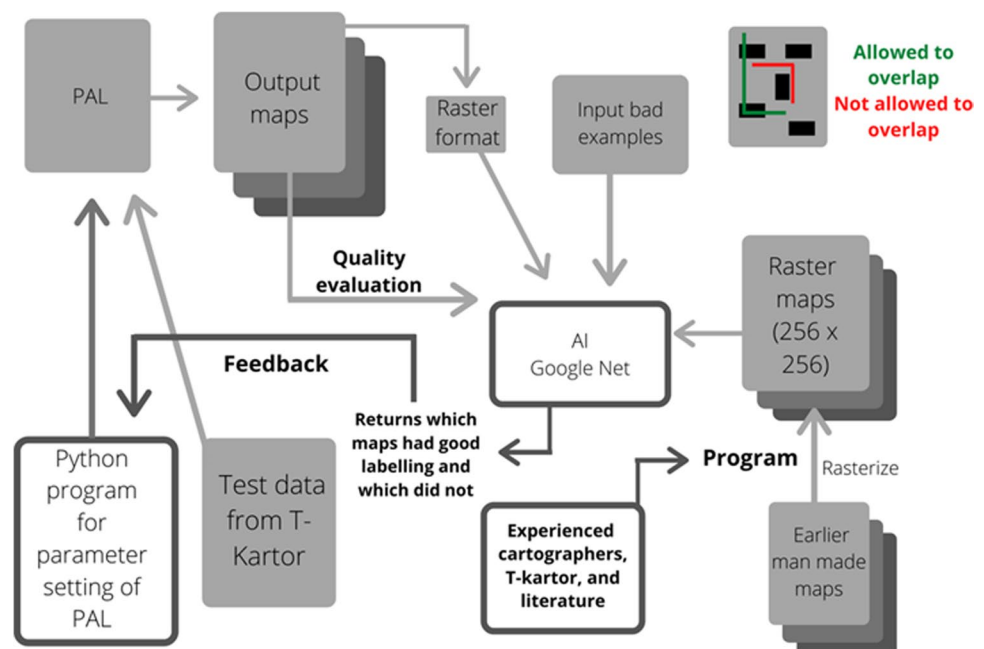
- 1) The evaluation schema is too complex for the neural network to learn. The map samples contained several labels and each label was manually classified according to the three categories (legibility, disturbance, and association). If only one of these labels was defined to be bad in one single category, the whole map sample was classified as “bad map labeling.”
- 2) The map samples were based on a single raster file. This implies that no information about what type of features that were hidden by the labels was learned in the training by the neural network. The solution for this would be to use several raster maps for a single map sample (as done in some other deep learning image applications), e.g., with one specific raster map that only contains the labels.
- 3) Due to hardware restrictions, only 800 iterations were performed in the training of the GoogLeNet network, which likely is too little training.
- 4) The sample size, i.e., number of map examples, is likely too small for both training and validation.

Challenge 2: Utilizing True Label Geometries in Automated Methods

Problem Identification

Automated labeling methods and tools generally utilize simplified geometries for the labels, most commonly minimum bounding rectangles. This works fine in many situations, but may entail shortcomings in others. Below we illustrate some shortcomings connected to large transparent text labels and icons.

Fig. 5 Overall conceptual framework. Raster maps are provided as input to a GoogLeNet classifier. After being trained, the model is used for evaluation of raster images received from PAL (Wei 2020, p. 17)



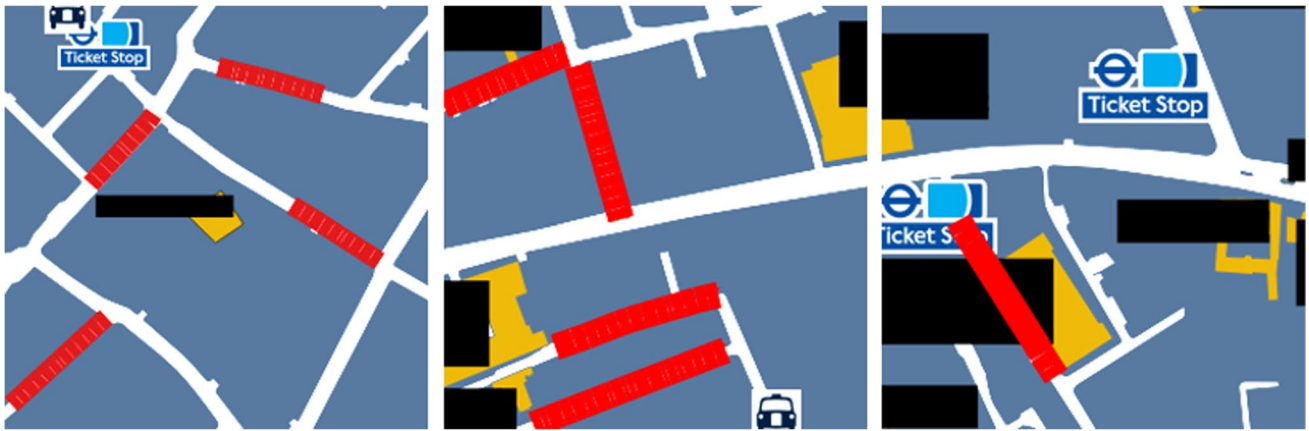


Fig. 6 Map examples created for training the network. Black boxes are landmark labels and red boxes are street labels. The leftmost was manually classified as good label placement, the middle as moderate, and the rightmost as bad

As noted above, the city wayfinding maps contain large-font labels for administrative areas such as villages and neighborhoods (see Fig. 7). These labels are difficult to place automatically due to that their large size makes overlap with other labels almost impossible to avoid. However, in many cases, these overlaps are acceptable from a map reading perspective. In the example in Fig. 7, the neighborhood label *St. Paul's* overlaps both icons and a road label without disturbing readability. In this solution, the label *St. Paul's* is placed in a central position of the neighborhood. Such a label placement is difficult to perform in QGIS, as well as in other tools, since they do not allow overlaps of the bounding boxes of the labels. In the tests we have made in QGIS, we have not been able to show road labels and the neighborhood layer at the same time in the area around *St. Paul's*.

A similar problem with the usage of simplified geometries for label placements is found for icons. In evaluating the ideal position of icons, QGIS (as well as some other programs) treats icons as a rectangle (with the extent of the minimum bounding rectangle of the icon). This implies that the program may identify non-existent label overlaps. E.g., QGIS would treat the icon and the two closest text labels in Fig. 8 as overlapping and therefore would not find the good cartographic solution that the cartographer did. This problem is

exaggerated by the low degree of freedom for placement of some icons. The *pointer* of the bus stop icon in Fig. 8 must be placed at exactly the actual location of the bus stop, implying that the only degree of freedom is the rotation of the icon. This is especially challenging in high-density areas where all available space needs to be utilized (and hence, it is not adequate to utilize an enlarged simplified geometry for the icons).

Rule-Based Techniques

There are several recommendations of which typography that should be used for maps to make the labels more readable (e.g., Slocum et al. 2005; Guidero 2017). But, to our knowledge, there have not been any studies that specifically address the issue of readability of large opaque text with large space between letters. Map readability of icons is studied in information visualization and in cartography but most of the studies concern readability due to cluttering/overlap of icons (e.g., Bereuter and Weibel 2013; Korpi and Ahonen-Rainio 2013) and comparatively few studies concern readability issues of the background map. One study that addressed the latter was conducted by Harrie et al. (2004), where they use a search strategy to place icons so that the icons overlap as few break

Fig. 7 Placement of the neighborhood label *St. Paul's*. The landmark label has been moved to give space to the neighborhood label. © Copyright Transport for London

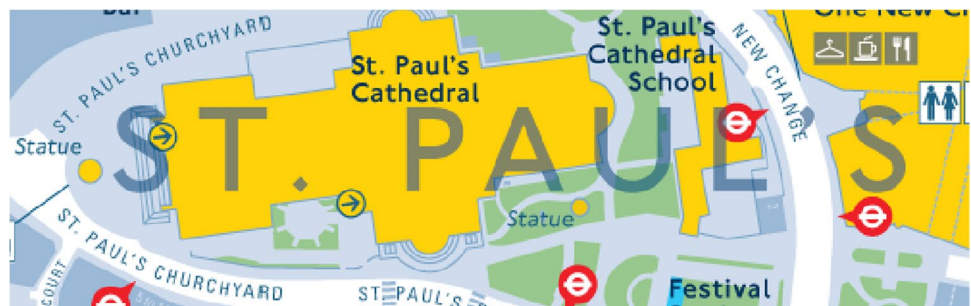




Fig. 8 A bus stop icon placed manually by a cartographer which creates good readability of the map. A cartographer manually identified that there is no overlap and that the map readability is good. © Copyright Transport for London

points (of the map features) as possible (following ideas from cognitive science, see, e.g., Biederman 1985). In their study, they only used quadratic labels, but the method does allow arbitrary shapes of the icons to be used. Van Kreveld et al. (2004) developed algorithms for diagram placement on maps that are also applicable for icon placement (especially for those cases when the icon represents an area feature, e.g., a landmark building). Besides the typically used centroid placement, they derive algorithms for, e.g., maximum self-overlap (with the area the diagram/icon represents) and minimum border overlap. The readability of labels is also linked to map complexity, which has been evaluated by, e.g., Harrie et al. (2015). They studied how single measures and composites of measures (e.g., based on linear regression or support vector machine) could describe map readability. User studies identified that foremost, the amount of information and spatial distribution are of interest, properties that likely also influence readability of areas including large font text in city wayfinding maps.

Deep Learning Techniques

To place large font labels, we need to use a unique model which can be either a part of the full labeling model or a

separate model. These large font labels should be treated as their own layer, and in doing so, we may apply some of the methods described in challenge 1 above. However, there are three problematic issues. The first issue is that the letters need to be represented with true geometries which entails that there will be a variety of area features as letters have different sizes, shapes, and orientations. The second issue is that the sample data representing these labels are very few and thus data augmentation techniques need to be used in order to generate data volume sufficient for the convergence of the model. Transformation of the original data by different operations such as cropping, resizing, rotating, grayscaling, and flipping can yield richer data and thus help the model to generalize its learning (Khalifa et al. 2022). In addition, the overlap loss function can be dropped in the training of this sub-model in order to allow overlapping with the other labels. The third issue concerns the label interaction (overlap avoidance) both towards the base map and the smaller labels. Additionally, overlaps are sometimes allowed for the large font layers (especially if they are transparent) which makes the modeling even more difficult.

Challenge 3: Creating a Good Relationship Between Text Labels and Icons

Problem Identification

Text labels and icons are commonly treated as separate objects. In QGIS, for example, icons are visual representations of point objects and text labels are generated based on attribute values of point, line, and area objects. This implies that the placement of text labels and icons are independent from each other, which causes problems. One challenge concerns finding a good placement of text labels in those cases the text labels should be placed in close relation to non-moveable icons. The examples shown in Fig. 9 and Fig. 10 illustrate how the station icons should be aligned (in this case left and right justified) with the station text labels. To perform this, the icon and text labels need to be combined into a common label that is then jointly placed. To our knowledge, no cartographic tool enables to do this in an

Fig. 9 A high-density area where there needs to be a relation between an icon (the large underground icon) and a text label (*Chancery Lane*). © Copyright Transport for London





Fig. 10 Aligned text labels and icons. © Copyright Transport for London

automated fashion. What is required instead is to manually combine the text labels and the icon into a common *icon* and then place the *icon* interactively.

Rule-Based Techniques

Combined icon and text label placement have, e.g., been studied by Zhang and Harrie (2006). Their approach was to compute candidate positions for the text labels and icons and then perform a combinatorial optimization step to identify the best possible combination of label positions. Also, the framework of Lu et al. (2019) could be extended to include icons.

Deep Learning Techniques

One technique that can enhance the learning of relations between the labels and icons and the objects they represent is attentional visual transformers. In general, the attention mechanism manipulates the hidden states (usually of encoders) in such a way that the attention is focused on the most relevant parts of the given data. Attentional visual transformers allow to learn through position embeddings the importance of a pixel with respect to the other parts of the image and also to model the local relations of highlighted objects. Another possible technique is semantic pooling: if any metadata is provided or the images are annotated, semantic pooling can be used to capture the importance of a set of pixels for these textual data.

Challenge 4: Adjustments of Text Labels to Make Them Fit Available Space

Problem Identification

To facilitate label placement in high-density areas, the labels can be divided into several rows (using left, center,

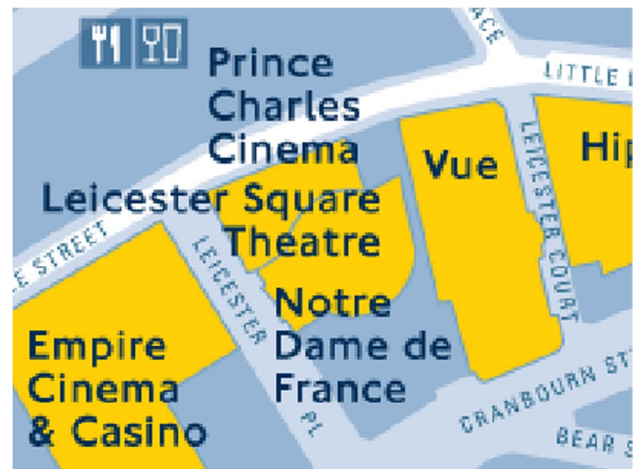


Fig. 11 A map with manually placed landmark buildings and road labels. © Copyright Transport for London

or right justification), text can be divided into several units (see, e.g., road label *Leicester Pl* in Fig. 11), and abbreviations can be utilized (e.g., *Pl* for *Place*). QGIS has the capability of dividing text strings into several lines (based on, e.g., a wrapping character stored in the string and maximum number of letters on a line), different justifications (left, right, center) if the text is divided into two or more lines, abbreviations (using label text substitutes), etc. That is, the available text manipulation methods are adequate. What is missing is the flexibility of tailored solutions for each label (for some labels, one line is to be preferred, sometimes two depending on the space available, etc.) and an evaluation routine that can answer which of the possible solutions is best in a certain situation. Our tests in QGIS have provided labeling as in Fig. 12, which do not have the same quality as the manually made map in Fig. 11.

Deep Learning Techniques

Deep learning methods have been developed to ensure that embedded texts are agreeable to the surrounding objects. Zhan et al. (2021) developed a Wasserstein GAN-based model to determine the contextual object borders suitable for text placement in scene images and then align the text accordingly and find its appropriate style and shape. Wu et al. (2019) faced similar challenges in free text editing. The issue was to replace or modify a text in the source image with another one which can have a different shape and length while keeping consistency with the background.

In addition, deep learning techniques have been developed for automatic assessment of clutter in images which can be useful in two ways. First, it can help to decide based on the size of (the minimum bounding box of) the label which text that should be shortened or which font size to use. Second, clutter

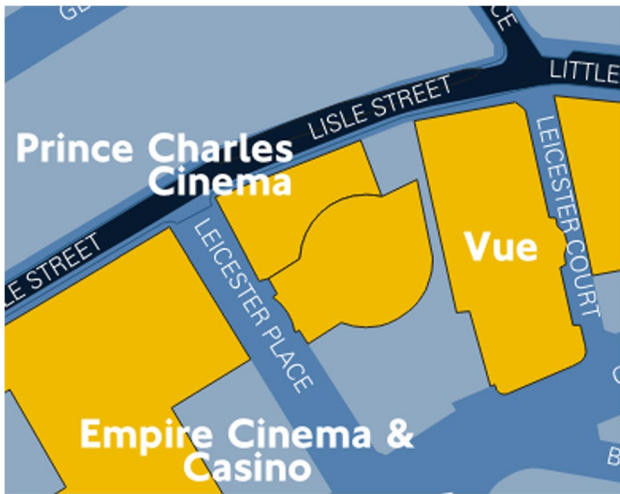


Fig. 12 The same area as in Fig. 11, but with map labeling automatically performed in QGIS

detection can help to apply priorities to which features should be labeled. Work in this domain has been performed by, e.g., Tezcan et al. (2018) who used ResNet to estimate the rate of clutter for any given input image.

Challenge 5: Placement of Road Labels

Problem Identification

Above road labels have been introduced under the heading *Line feature labeling*. However, they could also be regarded as a type of area feature labels of a long and thin area. The rule that the street label should be within the road area is also a typical area feature rule.

Rule-Based Techniques

Chirié (2000) interviewed cartographers about road label placement and based on these interviews has implemented a rule-based system (denoted PANR) where he divided the process into candidate positions, position evaluation, and position selection. PANR managed to place road labels in a medium dense road network where no other labels (e.g., landmarks) were present. Gemsa et al. (2014) developed an optimization algorithm for road label placement. The algorithm maximizes the number of labeled road segments where the label placement (somewhat simplified) should follow the rules in Chirié (2000). Gemsa et al. prove that the general road label placement problem is NP-hard, but that it can be solved in polynomial time if the (topological line) road network can be defined using a tree structure.

QGIS and Maplex label engines both provide good results for road labels (based on rule-based techniques) as long as

there are no other labels present. The problem seems to be the priority between labels and identification of good solutions if also other types of labels are present, such as landmark text labels and icons; in such cases, it seems as several labels cannot be placed in high-density areas (cf. Figure 4).

Data Representation Techniques

One specific challenge in the city wayfinding maps, as for many other urban maps, is the representation of road data. In the case of the London data, as used in the examples here, the road areas are mainly defined by the built-up area and pavement data, i.e., simply put the road area is the area not covered by built-up areas or pavements (or other area features). Therefore, road names are not linked to the actual road areas. To enable automated text setting for the road, “objects” line data is used, where each road line represents a lane. The challenge is then that the automatically placed line labels should fit well into the road areas (cf. Chirié 2000). This is especially difficult if a road consists of several lanes and/or the lanes are not straight lines (Fig. 13). This road label placement is a typical example of when the available data is not adequate for reaching the potential of automated text labeling. A possible solution could be to create road area objects based on the available area objects and then derive the straight skeleton of this area (cf. Haunert and Sester 2008). Ideally, from a map producer perspective, data producers should use a better data representation for roads, similar to the one in the recently approved CityGML ver. 3.0 standard where various LoD (level of detail) representations contain (linked) linear and area representations of road objects (see Fig. 14 and Beil et al. 2020 for details).

Deep Learning Techniques

Using raster data, the road labeling can be considered as a labeling of specific long and narrow areas and thus



Fig. 13 A road text label that is not straight due to that it is constructed based on lane data. The lane data (red line) is shown here only for illustration

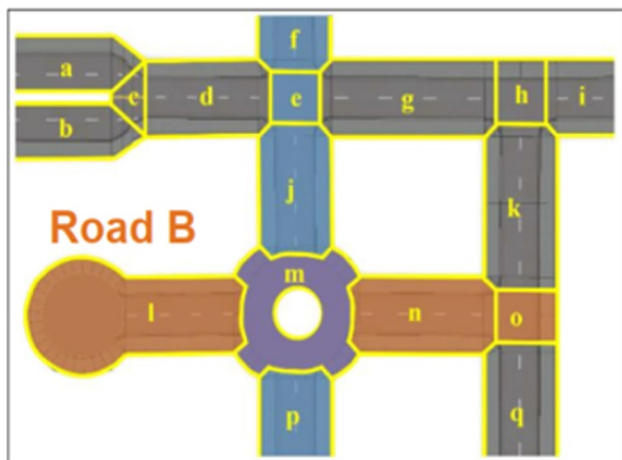


Fig. 14 Representing road areas in CityGML ver. 3.0. The road and junctions are represented by several topologically connected area objects where each area object could be linked to a line (skeleton) representation. Source: Beil et al. (2020, p. 14) (CC BY 4.0)

overcomes the problem of the data structure and graph issues. The task for the deep learning model is to learn the shape of these areas (length, width, and curvature) and adhere to it by selecting the best position inside the area for label placement. In addition, a subnetwork can be used to recognize the intersections and thus learn from the ground truth data to place the labels far away from them.

One of the solutions that can be used for the road label placement is deep reinforcement learning (cf. Wang et al. 2020b). Reward functions can be designed using the labeling rules so that the labeling agent can be rewarded if it follows the quantified rules and punished if the label is misplaced, e.g., placed on an intersection. A fatal error that can abort the training episode is to place the label outside the road. Similar learning is, for example, applied in the case of placement of objects by autonomous robots (Harada et al. 2014). The object placement is constrained by several rules which impose the pose of the object placed on the environment. These rules can be, for instance, the position and orientation of the object. However, the application of deep reinforcement learning for map labeling is costly in terms of the data and the computation it requires in addition to its dependence on the appropriate quantification of the cartographic rules.

Concluding Remarks

In this study, we identified five challenges in map labeling that currently cause intensive manual work in the production of city wayfinding maps. To address these challenges, several actions should be taken. Based on our investigations, we recommend the approaches below for the identified challenges.

Challenge 1: Label Placement in High-Density Areas

For this challenge, there are two deep learning techniques that are promising. The first technique is the context-aware synthesis and placement of objects (labels) (e.g., Lee et al. 2018; Volokitin et al. 2020). An open question here is whether it is possible to train a network to “understand” the mapping context for the label placement. The second technique would be to apply an evaluation strategy, where candidate solutions are computed using rule-based methods (or the deep learning method above). The evaluator network would then need to be trained to distinguish between poor and good candidate solutions using a large number of map samples with manually placed labels. We have earlier tested this approach (Wei 2020) with a low level of success, but as noted above, we have identified several aspects where this approach can be improved.

Challenge 2: Utilizing True Label Geometries in Automated Methods

The key issue here is to model the text and icons with their true geometry and not a simplified one (e.g., minimum bounding box) which is almost exclusively used today. For icons, the search space for adequate placement is narrow which means that quite simple rule-based methods (as the one suggested by Harrie 2004) would most likely be applicable. It is more difficult to establish methods for texts with large fonts (as in Fig. 7). Here, the interplay with other text labels and icons (as well as the background map) is more complicated. Perhaps, some kind of rule-based system followed by an evaluation strategy would be a feasible solution. There are several aspects that makes the deep learning techniques difficult to handle for true label geometries as used in, e.g., large font labels, especially to establish enough training data for the model.

Challenge 3: Creating a Good Relationship Between Text Labels and Icons

This is a difficult challenge where we have not been able to find a good candidate method. A possibility would perhaps be to use a brute force approach where, e.g., all possible alignments of text and icons (as in Fig. 10) are created and placed in several positions to generate candidate solutions. Then, an evaluation strategy would be applied to select the best candidate, e.g., based on overlap. In any case, to address this challenge, we cannot continue to deal with text label placement and icon label placement as two separate processes.

Challenge 4: Adjustments of Text Labels to Make Them Fit Available Space

For this challenge, it would be interesting to evaluate the deep learning techniques developed for text placement on

images (similar to Zhan et al. 2021, Wu et al. 2019). It is plausible that this type of methods could be modified to be used for providing recommendations regarding which geometric shape the text labels should have and hence how the adjustment of the text labels should be conducted.

Challenge 5: Placement of Road Labels

At least for the city wayfinding maps in this study, good road label placement would benefit from that both linear (skeleton) and area representations of the roads are available, and that these representations are linked (knowing which line corresponds to which area object). What should be studied in future studies are how to generate the road area data and the skeleton data and how to link them (utilizing the data models in, e.g., CityGML3). If this data is available, we anticipate that good road label placement could be performed as long as only road labels are present, but that there are still challenges if other labels are in the same area, especially if it is a high-density area (challenge 1). When area representation are available, some of the deep learning techniques for challenge one could be applied but also other techniques such as deep reinforcement learning.

The recommendations above are just starting points for our practical tests, and it is likely that these practical tests will result in changes in the recommendations. But what will certainly not change is that a mix of diverse methods needs to be used. As we see it, there will not be a single technique that can deal with all the identified production challenges. This is also in line with experience of earlier studies in automated map labeling. There have been many good solutions to solve particular tasks in labeling, but no method has been applicable to solve all the tasks, or not even the majority of these tasks. Therefore, when a new technique emerges, such as deep learning, we should not expect that it will completely automate map labeling, but we believe that it has a good potential to solve some specific tasks and, thus, significantly raise the automation level for production of city wayfinding maps as well as other types of maps.

Appendix 1. Production rules for label and icon placements in city wayfinding maps

The rules below are based on design standards produced by Transport for London and internal labeling rules from the mapping company T-Kartor. The text is based on Cederholm (2020), where you also could find more details.

Point Feature Labeling

Point feature labels should be placed according to traditional cartographic rules; all labels should be horizontal and with

the priority position above to the right of the point (see, e.g., Slocum et al. 2005). In a city wayfinding map, most point objects are presented by icons (and few labels are used). Some icons are not allowed to be moved. If there is not enough space, callouts are used (Fig. 1).

Line Feature Labeling

The line features that should be labeled in a city wayfinding map are, e.g., roads, rivers, and creeks. They all have similar rules for placement, here illustrated with road labels. Road labels are always placed within the road area, which means that they must adhere to the shape of the road. Texts are easier to read if the roads are straight; therefore, straight parts of roads are generally preferable for text labels. But if the entire road has a significant curvature, labels cannot be entirely straight, since they need to adhere to the shape of the road. If a label does not fit within its road, it could be wrapped into two lines, if possible; if not, the label should be shortened. A label is repeated on the line if the road is long. Road labels should be placed at the center of a road, and not at the end of a road. The labels should never end or start at an intersection, since one cannot tell whether the road continues after the intersection, or if there is another road name after the intersection. For this reason, a label should be placed across an intersection (when the same road name continues after the intersection). Furthermore, road labels should avoid overlapping pedestrian crossings. It is preferable if labels of parallel roads are aligned in a row, as the road labels in Fig. 15.

Area Feature Labeling

An example of area feature labeling is illustrated in Fig. 16. Preferably, area labels should stay within the boundaries of the polygon feature that represents the area, but they may cross the boundary if necessary. However, this should preferably be avoided by wrapping the text into several lines. Area labels should be horizontally placed and aligned depending on their placement in relation to the polygon feature they belong to. Generally, if a label needs to be placed more to the right of its polygon, then left alignment should be used, and vice versa. The aligned part of the text, or at least a fraction of it, should always be inside the polygon.

City wayfinding maps also contain area labels for administrative areas such as villages and neighborhoods. These labels have a large font and an opacity of 80–85% (see Fig. 16, Fig. 7) and should ideally not overlap other labels, but in practice, this is hard to fulfill and therefore label overlap is allowed as long as it does not violate map readability (see Fig. 7). The opacity and the letter spacing are key to enable readability of the overlapping labels.

Fig. 15 Examples of how road labels can be placed. © Copyright Transport for London



Fig. 16 An example of area feature label placements. © Copyright Transport for London

Icons

In a city wayfinding map, most icons are associated to a feature on or close to a street (cycle hire, bus stop, taxi, etc.). The rules for placing these icons are similar and are here illustrated with rules for the cycle hire icons (Fig. 17). These icons have an arrow that must point to the true position of a hire station. The icons should preferably be placed in a 90-degree angle to its corresponding street. Other angles are permitted if necessary. Generally, the cycle hire icons should not be placed over roads (there are only a few exceptions to this). The icons are allowed to overlap buildings if necessary.

Several other icons should ideally be placed such that their center coincides with the labeled feature. This is, e.g., the case for entrance symbols. In some cases, the symbols have to be slightly moved to avoid ambiguities in their interpretation. One such example is the placement of the Ticket Stop label in Fig. 18. In this case, the ticket stop is (likely) situated in the corner of the icon but if it would have been



Fig. 17 Bus stop and cycle hire station icons. Note that some of the icons overlap streets, which is permitted if no better solution is found. © Copyright Transport for London

placed in the middle, it would be more difficult to see on which side of the road the Ticket Stop is (and in this particular case, it would also overlap with a road label).

There is an interplay between labels and icons. A basic rule is, of course, that they are not allowed to overlap. But there are also rules of how they should be placed in relation to each other (Fig. 19). For example, the station icons should be aligned to the station text labels either left or right depending on the graphic situation (Fig. 19, Fig. 10). Retail



Fig. 18 Placement of icons. © Copyright Transport for London

unit icons are always placed in their own row, under the text labels; they are usually centered, but not always. Important buildings from a wayfinding perspective, such as stations, should also have icons that illustrate the entrance/exit of the buildings. These icons must partly overlap the building and be in their true position, i.e., where the entrance/exit actually is in the building.

Label Overlapping and Removal

The first rule is that no labels and icons may overlap and the second rule is that it is not allowed to remove a label or icon. These two rules often result in conflicts between text labels and icons that require exceptions. One exception is that the large, transparent labels for neighborhoods, boroughs, and villages may overlap other text and icons. In exceptional cases, text labels are allowed to overlap icons, if it is judged to be the best possible solution.

There are also rules for overlap of labels and map features. Generally, text labels are not allowed to overlap building and road objects if they do not represent that particular object. However, it is allowed for an unrelated text label to partially cover a building, if it is clear to which building each label corresponds to, and that the building's own label is clearly visible. This is the case in Fig. 1, where the building label *Thistle Piccadilly Hotel* partly covers the neighboring building *Prince of Wales Theatre*. Furthermore, the latter label is overlapping the road object which is generally not advisable. But in this case, there is no other space available, and it is quite clear which building the text label *Prince of Wales Theatre* represents. Lastly, cycle hire icons and similar icons are allowed to overlap buildings if necessary.



Fig. 19 Relationship between text labels and icons. The Waterloo label is left aligned to the station icons. The entrance/exit are represented with arrow icons. © Copyright Transport for London

Hyphenation and Other Text Manipulations

For a text label, the following priorities should be used: (1) complete text in one unit, (2) shortened text in one unit, (3) text divided into two units (see, e.g., *Mepham Street* in Fig. 19), and (4) the text is divided into two or several rows (see, e.g., *Waldorf Hilton Hotel* in Fig. 16). There are also rules for how text is allowed to be split into several units or rows. Examples of such rules are as follows: Never split after “The” and never split between words that are tightly connected such as the words *Youth Hostel*. Furthermore, Transport for London provides a list of all abbreviations that are allowed to be used (e.g., DR for *Drive* and AVE for *Avenue*). It should also be noted that city wayfinding maps allow, under several restrictions, a change of font size to obtain optimally looking labels.

Appendix 2. Map labeling in QGIS

Some years ago, QGIS integrated the open source map labeling library PAL.⁶ This library contains algorithms that use a combinatorial optimization method that includes two steps (Ertz et al. 2009). In the first step, several candidate positions are created for each point, line, and polygon label. In the next step, the optimization is performed. The placement cost is computed for each candidate label position based on (1) label placement in relation to the feature; (2) eventual overlapping of another map feature, or the distance to another map feature; and (3) label suppression. For more details on PAL, see Ertz et al. (2009).

QGIS utilizes most functionality of PAL, and has also extended the functionality substantially to generate a freely available map labeling toolbox that can also be reached from a Python API. Here, we just describe some features of this tool that is of interest for our study (the text is based on QGIS Development Team 2021, see also Cederholm 2020 for more details).

In QGIS (version 3.20), there are three placement options for generating label candidates for point features: *Cartographic*, *Around point*, and *Offset from point*. The placement options for line feature candidates are *Parallel*, *Curved*, and *Horizontal*. For the *Parallel* and *Curved* options, one can select whether the candidates should be placed *Above*, *On*, or *Below* line, or have a *Line orientation dependent position*. There are seven placement options for area feature labels. Two of those are centroid-based: *Around centroid* and *Offset from centroid*. With the options *Using perimeter* and *Using perimeter (curved)*, the perimeters of polygons are treated as lines which labels follow. Candidates can also be placed

⁶ <http://pal.heig-vd.ch/>

inside or outside polygons, either tilted or horizontal (QGIS Development Team 2021). The label candidate placement cost for polygons increases with decreasing distance to the boundary of the polygon or to other features. Of the many candidates produced for a feature, a portion is removed due to bad placements. These can be labeled candidate positions causing overlapping, and other positions with high costs (Ertz et al. 2009). It is also possible to add blocking to a feature type, implying that no other features can overlap features of this type.

Finally, it should be noted that the labeling is performed in real time in QGIS, which is not really a requirement in map production.

Acknowledgements Thanks to Transport of London and T-Kartor for providing the map data and to QGIS project for providing a labeling tool. Also, thanks to Kalle Åström, Lund University, for advice.

Funding Open access funding provided by Lund University.

This study was funded by eSENCE@LU 7:1: “Data-driven automation of map labelling – enabling affordable high-quality maps” as part of the strategic research area eSENCE (financed by the the Swedish research council). Financial support has also been provided by Lund University, Umeå University and T-Kartor.

Declarations

The research conducted for this paper has been in full compliance with ethical standards at Lund University.

Ethics Approval This study follows the ethical rules at Lund University.

Informed Consent Not applicable.

Conflict of Interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bereuter P, Weibel R (2013) Real-time generalization of point data in mobile and web mapping using quadrees. *Cartogr Geogr Inf Sci* 40(4):271–281. <https://doi.org/10.1080/15230406.2013.779779>
- Biederman I (1985) Human image understanding: recent research and a theory. *Computer Vision, Graphics, Image Processing* 32(1):29–73
- Brewer CA, Stanislawski LV, Buttenfield BP, Sparks K, McGilloway J, Howard MA (2013) Automated thinning of road networks and road labels for multiscale design of The National Map of the United States. *Cartogr Geogr Inf Sci* 40(4):259–270. <https://doi.org/10.1080/15230406.2013.799735>
- Cederholm P (2020) Automatic label placement for city maps with the labelling library PAL. M.Sc. thesis series INES nr 521, Lund University, Sweden
- Cetinic E, Lipic T, Grgic S (2019) A deep learning perspective on beauty, sentiment, and remembrance of art. *IEEE Access* 7:73694–73710. <https://doi.org/10.1109/access.2019.2921101>
- Chirié F (2000) Automated name placement with high cartographic quality: city street maps. *Cartogr Geogr Inf Sci* 27(2):101–110. <https://doi.org/10.1559/152304000783547902>
- Christensen J, Marks J, Shieber S (1995) An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics (TOG)* 14(3):203–232
- Courtial A, El Ayedi A, Touya G, Zhang X (2020) Exploring the potential of deep learning segmentation for mountain roads generalisation. *ISPRS Int J Geo Inf* 9(5):338. <https://doi.org/10.3390/ijgi9050338>
- Dhillon A, Verma GK (2019) Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress Artificial Intelligence* 9(2):85–112. <https://doi.org/10.1007/s13748-019-00203-0>
- Doerschler JS, Freeman H (1992) A rule-based system for dense-map name placement. *Commun ACM* 35(1):68–79. <https://doi.org/10.1145/129617.129620>
- Ertz O, Laurent M, Rappo D, Sae-Tang A, Taillard E (2009) PAL-A cartographic labelling library. *Position IT July 2009*: 56–61
- Fang Y, Zhu L, Cao X, Zhang L, Li X (2020) Visual saliency: how text influences. In: Meiselwitz G (ed) *Social Computing and Social Media. Design, Ethics, User Behavior, and Social Network Analysis. HCII 2020. Lecture Notes Computer Sci* 12194. Springer, Cham. https://doi.org/10.1007/978-3-030-49570-1_4
- Feng Y, Thiemann F, Sester M (2019) Learning cartographic building generalization with deep convolutional neural networks. *ISPRS Int J Geo Inf* 8(6):258. <https://doi.org/10.3390/ijgi8060258>
- Gemsa A, Niedermann B, Nöllenburg M (2014) Label placement in road maps. *EuroCG 2014, Ein-Gedi, Israel, March 3–5, 2014*.
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Ghahramani Z et al. (ed) *International Conference on Neural Information Processing Systems*. Montreal, Canada, pp. 2672–2680.
- Guidero E (2017) *Typography. The Geographic Information Science & Technology Body of Knowledge (3rd Quarter 2017 Edition)*, John P. Wilson (ed.). <https://doi.org/10.22224/gistbok/2017.3.2>
- Harada K, Tsuji T, Nagata K, Yamanobe N, Onda H (2014) Validating an object placement planner for robotic pick-and-place tasks. *Robot Auton Syst* 62(10):1463–1477. <https://doi.org/10.1016/j.robot.2014.05.014>
- Harrie L, Stigmar H (2015) Djordjevic M (2015) Analytical estimation of map readability. *ISPRS Int J Geo-Inf* 4:418–446. <https://doi.org/10.3390/ijgi4020418>
- Harrie L, Stigmar H, Koivula T, Lehto L (2004) An algorithm for icon placement on a real-time map. In: Fisher P (ed) *Development in Spatial Data Handling*. Springer, pp 493–507
- Hauert JH, Sester M (2008) Area collapse and road centerlines based on straight skeletons. *GeoInformatica* 12:169–191. <https://doi.org/10.1007/s10707-007-0028-x>
- Hauert JH, Wolff A (2017) Beyond maximum independent set: an extended integer programming formulation for point labeling. *ISPRS Int J Geo Inf* 6(11):342
- Imhof E (1975) Positioning names on maps. *The American Cartographer* 2(2):128–144
- Kern JP, Brewer CA 2008 Automation and the map label placement problem: a comparison of two GIS implementations of label

- placement. *Cartographic Perspectives* 60:22–45. <https://doi.org/10.14714/CP60.230>
- Khalifa NE, Loey M, Mirjalili S (2022) A comprehensive survey of recent trends in deep learning for digital images augmentation. *Artif Intell Rev* 55:2351–2377. <https://doi.org/10.1007/s10462-021-10066-4>
- Kindermann P, Niedermann B, Rutter I, Schaefer M, Schulz A, Wolff A (2015) Multi-sided boundary labeling. *Algorithmica* 76:225–258. <https://doi.org/10.1007/s00453-015-0028-4>
- Klute F, Li G, Löffler R, Nöllenburg M, Schmidt M (2019) Exploring semi-automatic map labeling. In *Proc. 27th SIGSPATIAL*. ACM, 13–22
- Korpi J, Ahonen-Rainio P (2013) Clutter reduction methods for point symbols in map mashups. *Cartogr J* 50(3):257–265. <https://doi.org/10.1179/1743277413Y.0000000065>
- Van Kreveld M, Schramm E, Wolff A (2004) Algorithms for the placement of diagrams on maps. *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, November 2004 Pages 222–231
- Lee D, Liu S, Gu J, Liu MY, Yang MH, Kautz J (2018) Context-aware synthesis and placement of object instances. In: Bengio S et al. (ed) *Conference on Neural Information Processing Systems*, 32nd ed. Montreal, CANADA, pp. 10393–10403.
- Levine S, Pastor P, Krizhevsky A, Ibarz J, Quillen D (2018) Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int J Robotics Res* 37(4–5):421–436
- Li Y, Sakamoto M, Shinohara T, Satoh T (2020) Automatic label placement of area-features using deep learning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B4-2020*:117–122. 10.5194/isprs-archives-xliii-b4-2020-117-2020
- Li H, Zhu F, Qiu J (2018) DeepITQA: deep based image text quality assessment. *Lect Notes Comput Sci* 397–407. https://doi.org/10.1007/978-3-030-04224-0_34
- Beil C, Ruhdorfer R, Coduro T, Kolbe TH (2020) Detailed streetspace modelling for multiple applications: discussions on the proposed CityGML 3.0 Transportation Model. *ISPRS Int. J. Geo-Inf.* 9:603. <https://doi.org/10.3390/ijgi9100603>
- Lin CH, Yumer E, Wang O, Shechtman E, Lucey S (2018) ST-GAN: spatial transformer generative adversarial networks for image compositing. In: Brown MS et al. (ed) *Conference on Computer Vision and Pattern Recognition*. Salt Lake City, USA, pp. 9455–9464. <https://doi.org/10.1109/cvpr.2018.00985>
- Liu X, Van De Weijer J, Bagdanov AD (2017) RankIQA: learning from rankings for no-reference image quality assessment. In: Ikeuchi K et al. (ed) *IEEE International Conference on Computer Vision*, 16th ed. Venice, Italy, pp. 1040–1049. <https://doi.org/10.1109/iccv.2017.118>
- Lu F, Deng J, Li S, Deng H (2019) A hybrid of differential evolution and genetic algorithm for the multiple geographical feature label placement problem. *ISPRS Int J Geo-Inf* 8:237
- Lyu W, Zhou Z, Chen L, Zhou Y (2019) A survey on image and video stitching. *Virtual Reality Intelligent Hardware* 1(1):55–83. <https://doi.org/10.3724/sp.j.2096-5796.2018.0008>
- Ohri K, Kumar M (2021) Review on self-supervised image recognition using deep neural networks. *Knowl-Based Syst* 224:107090. <https://doi.org/10.1016/j.knsys.2021.107090>
- Pokonieczny K, Borkowska S (2019) Using artificial neural network for labelling polygon features in topographic maps. *GeoScape* 13(2):125–131. <https://doi.org/10.2478/geosc-2019-0012>
- QGIS Development Team (2021) QGIS 3.16 Documentation, <https://www.qgis.org/en/docs/index.html>
- Raposo P, Brewer CA, Stanislawski LV (2013) Label and attribute based topographic point thinning: *Proceedings, 16th ICA Workshop on Generalization and Multiple Representations*, August 23–24, 2013. Dresden, Germany, p 8
- Rylov M, Reimer A (2014) A comprehensive multi-criteria model for high cartographic quality point-feature label placement. *Cartographica* 49(1):52–68
- Rylov MA, Reimer AW (2015) Improving label placement quality by considering basemap detail with a raster-based approach. *GeoInformatica* 19:463–486
- Slocum TA, McMaster RB, Kessler FC, Howard HH (2005) *Thematic cartography and geovisualization*, second edition. Upper Saddle River, NJ, USA: Pearson Prentice Hall. ISBN 9780132298346
- Strijk T, Van Kreveld M (2002) Practical extensions of point labeling in the slider model. *GeoInformatica* 6(2):181–197
- Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. *IEEE conference on computer vision and pattern recognition*, Boston, USA, pp. 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
- Tezcan MO, Konrad J, Muroff J (2018) Automatic assessment of hoarding clutter from images using convolutional neural networks. *2018 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. <https://doi.org/10.1109/ssiai.2018.8470375>
- Touya G, Zhang X, Lokhat I (2019) Is deep learning the new agent for map generalization? *International Journal of Cartography* 5(2–3):142–157
- Van Dijk S, Van Kreveld M, Strijk T, Wolff A (2002) Towards an evaluation of quality for names placement methods. *Int J Geogr Inf Sci* 16(7):641–661
- Van Kreveld M, Strijk T, Wolff A (1999) Point labeling with sliding labels. *Comput Geom* 13(1):21–47
- Vilaplana V (2015) Saliency maps on image hierarchies. *Signal Processing: Image Communication* 38:84–99. <https://doi.org/10.1016/j.image.2015.07.012>
- Volokitin A, Susmelj I, Agustsson E, Gool L, Van Timofte R (2020) Efficiently detecting plausible locations for object placement using masked convolutions. *Lect Notes ComputSci* 252–266. https://doi.org/10.1007/978-3-030-66823-5_15
- Wang Z, Wang E, Zhu Y (2020) Image segmentation evaluation: a survey of methods. *Artif Intell Rev* 53:5637–5674. <https://doi.org/10.1007/s10462-020-09830-9>
- Wang HN, Liu N, Zhang YY, Feng D, Huang F, Li D, Zhang Y (2020) Deep reinforcement learning: a survey. *Front Inform Technol Electron Eng* 21:1726–1744. <https://doi.org/10.1631/FITEE.1900533>
- Wei L (2020) An artificial intelligence method for text placement evaluation in maps. M.Sc. thesis series INES nr 524, Lund University, Sweden
- Wolff A, Strijk T (2009) A map labeling bibliography. Available at <https://i11www.itl.kit.edu/~awolff/map-labeling/bibliography/>
- Wood CH (2000) A descriptive and illustrated guide for type placement on small scale maps. *Cartogr J* 37(1):5–18. <https://doi.org/10.1179/caj.2000.37.1.5>
- Wu L, Zhang C, Liu J, Han J, Liu J, Ding E, Bai X (2019) Editing text in the wild. *Proceedings of the 27th ACM International Conference on Multimedia*. <https://doi.org/10.1145/3343031.3350929>
- Yoeli P (1972) The logic of automated map lettering. *Cartogr J* 9(2):99–108. <https://doi.org/10.1179/caj.1972.9.2.99>
- Zhan F, Zhu H, Lu S (2019) Spatial fusion GAN for image synthesis. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3653–3662
- Zhan F, Huang J, Lu S (2021) Hierarchy composition GAN for high-fidelity image synthesis. arXiv preprint. doi.org:<https://doi.org/10.13140/RG.2.2.19889.51041>
- Zhang Q, Harrie L (2006) A real-time method of placing text and icon labels simultaneously. *Cartogr Geogr Inf Sci* 33(1):53–64

- Zhao W, Du S (2016) Spectral–spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach. *IEEE Trans Geosci Remote Sens* 54(8):4544–4554
- Zhou Q, Li Z (2017) A comparative study of various supervised learning approaches to selective omission in a road network. *Cartogr J* 54(3):254–264

- Zoraster S (1997) Practical results using simulated annealing for point feature label placement. *Cartography and Geographic Information Systems* 24(4):228–238

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.