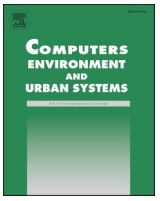
Computers, Environment and Urban Systems 113 (2024) 102174

Contents lists available at ScienceDirect

Computers, Environment and Urban Systems

journal homepage: www.elsevier.com/locate/ceus



Large-scale integration of remotely sensed and GIS road networks: A full image-vector conflation approach based on optimization and deep learning

Zhen Lei [a], Ting L. Lei [b,*]

[a] College of Automation, Wuhan University of Technology, Wuhan 430070, China
[b] Department of Geography and Atmospheric Science, University of Kansas, Kansas 66045, USA

ARTICLE INFO

Keywords:
Remote sensing
GIS
Conflation
Optimization

ABSTRACT

Road networks play an important role in the sustainable development of human society. Conventionally, there are two sources of road data acquisition: road extraction from Remote Sensing (RS) imagery and GIS based map production. Each method has its limitations. The RS road extraction methods are primarily raster-based and the extracted roads are not directly usable in GIS due to their fragmented and noisy nature, while vector-based methods cannot utilize rich raster information. Further more, the vector and raster data can have discrepancies for various reasons. Efficient road data production requires an image-vector conflation process that can match and combine raster and vector-based road data automatically.

In this study, we propose a full image-vector conflation framework that directly integrates image and vector road data by appropriately transforming extracted roads from imagery and establishing a match relation between these roads and a credible target GIS road dataset. Based on analyzing these match relations, we propose new metrics for measuring the degree of agreement between the raster and vector road data. The proposed framework combines state-of-the-art deep learning methods for image segmentation and optimization-based models for object matching. We prepared a large-scale high-resolution road dataset covering two counties in Kansas, US. Using trained models from one of the two counties, we were able to extract road segments in the other county and match them to the TIGER/Line roads.

Our experiments show that conventional performance metrics for road extraction (e.g. IoU) are insufficient for measuring the degree of agreement between image and vector roads as they are pixel-based and are too sensitive to spatial displacement. Instead, the newly defined vector-based agreement metrics are needed for image-vector conflation purposes. Experiments show that, by the vector-based metrics, nearly 90% of GIS road lengths in the study area were extracted and over 90% of extracted roads matched the target GIS roads. The new framework streamlines raster-vector conflation of roads and can potentially expedite relevant geospatial analyses regarding change detection, disaster monitoring and GIS data production, among others.

1. Introduction

Transportation networks are critical for the functioning and sustainable development of human society. Obtaining accurate up-to-date information about roads is important for a variety of reasons, including traffic management, day-to-day GIS data production, and emergency/disaster response. Maintaining road information is often difficult in practice. Road data can be collected from two primary sources. The first source is GIS-based vector data acquisition. This involves land surveying, digitization of paper maps, and field measurement using GPS. GIS and cartographic processes are labor-intensive and incur inevitable human errors, making large-scale data collection expensive and time consuming.

The second source of road data collection is remote sensing. Unlike vector data collection, remote sensing methods are relatively fast and inexpensive. With the emergence of new sensors, their positional accuracy is also high and consistent. However, remote sensing methods for road extraction have their own limitations. First, they can be disrupted by elements such as shadows, obstructions from nearby objects, and inundation (as shown in Fig. 1a), rendering roads as fragments of disconnected pieces. Second, many road extraction methods render roads as a collection of pixels. Neither the fragmented road segments or

* Corresponding author.
E-mail addresses: leizhen@whut.edu.cn (Z. Lei), lei@ku.edu (T.L. Lei).

https://doi.org/10.1016/j.compenvurbsys.2024.102174
Received 22 September 2023; Received in revised form 10 July 2024; Accepted 11 August 2024
Available online 20 August 2024
0198-9715/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

road surface pixels can be used directly by transportation and GIS analyses. Third, road data obtained from remote sensing typically do not contain attribute information. To be useful, extracted road pixels or fragments have to be merged to existing GIS datasets to have attributes (such as street names and addresses).

A more efficient approach for producing road data is image-vector conflation. Geospatial data conflation is the process of combining two spatial datasets to produce a coherent new dataset with richer information or better quality. Current conflation research mostly deal with conflating vector-format maps from different sources. The main goal of this work is to extend the capability of current conflation methods so that they can work on roads in images as well. Via image-vector conflation, remote sensing can provide frequent and accurate updates of the geometric (and spectral) information for roads, whereas GIS can manage and maintain a database of the higher level geometric, attribute, and topological information.

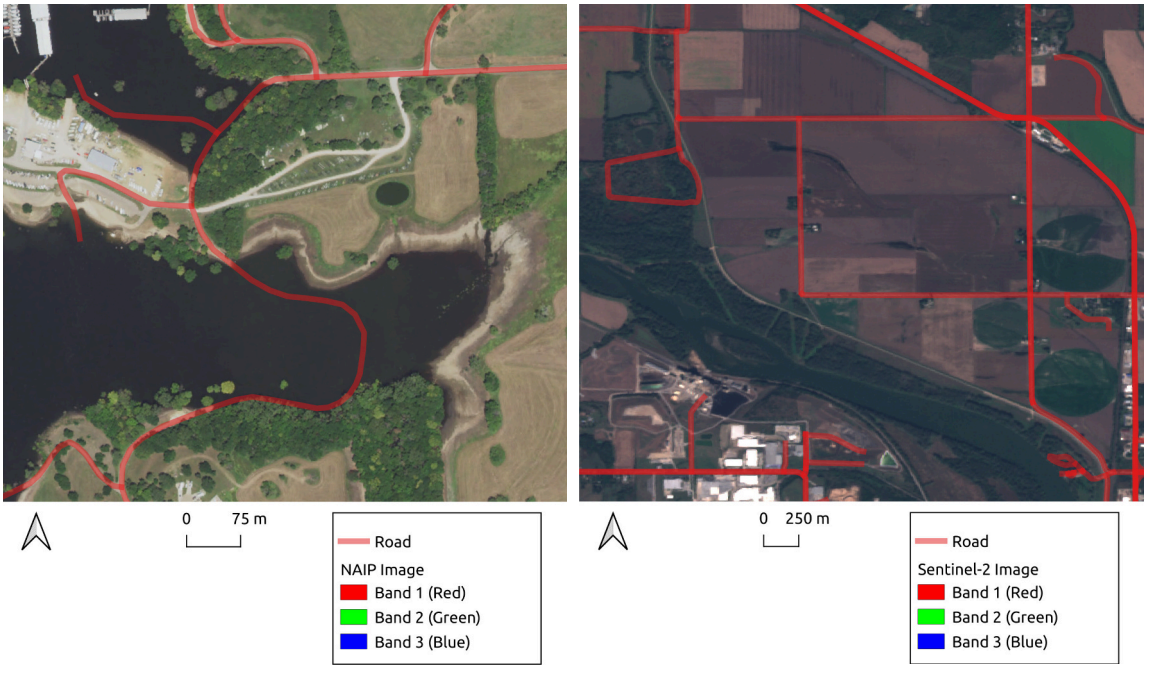
While both road extraction and vector conflation methods have been extensively studied, neither of them is sufficient for raster-vector conflation by itself. On the one hand, existing road extraction methods mostly focus on image analysis. The extracted roads have rarely been merged with any GIS databases. On the other hand, vector-based GIS data conflation lacks the capability of utilizing information from imagery. What is lacking is a methodology for full image-vector conflation.

To achieve full image-vector conflation, several issues have to be solved. Firstly, GIS and remote sensing data can exhibit discrepancies with each other (as shown in Fig. 1a, b). These include coordinate errors as well as errors introduced in the cartographic process. Such discrepancies must be correctly identified during the conflation. Existing pixel-based metrics for road extraction are not suitable for image-vector conflation purposes. As will be demonstrated later, if used for conflation, they can easily mis-classify matched roads as unmatched ones when spatial displacement exist. Secondly, road segments extracted from remote sensing images are often fragmented (as shown in Fig. 1a) and noisy. This complicates the match relation between extracted road segments from images and the GIS roads. Thirdly, image analysis and vector conflation processes need to be reconciled into one coherent process to better address the aforementioned issues.

This article proposes a new framework for full image-vector conflation that attempts to automate the integration of remote sensing and GIS road data. The new framework is based on state-of-the-art deep-learning methods, standard GIS operations and vector conflation models. And it can be used to directly conflate a road image dataset with credible GIS data source (such as TIGER/Line). In particular, we propose a data processing workflow based on modern spatial databases and show that standard GIS operations can be used to handle all of the data processing and matching problems in image-vector conflation given that the road extraction method can provide a segmentation of road pixels (as the majority of road extraction methods do).

In order to relate and compare roads in the image and roads in the vector data, we analyze the match relationship between them and represent them using JOIN operations in relational databases. Based on this representation, we propose two sets of performance metrics to gauge the degree of agreement between the roads in image and the GIS road network. The first set of metrics is based on counting the number of correctly extracted road objects. We define and use two-sided True Positives to account for the complex many-to-one correspondence between extracted roads and GIS roads. The second set of metrics are length-weighted and reflect the percentage of correctly matched roads in terms of total length.

The framework is aimed at automating image-vector conflation for transportation networks, and can be useful in different applications once the match between the image and vector data is established. The matched portion between the road image and GIS dataset is useful for data production purposes. The extracted roads typically have higher positional precision and may be used to improve the geometries of existing roads in GIS. The parts in which the road images and GIS data



a. Broken roads in a NAIP image caused by inundation

b. Discrepancies between roads from Sentinel imagery and TIGER/Line for Lawrence, KS (due to cartographic errors etc.)

Fig. 1. Examples of mismatch between roads in remote sensed image and roads in GIS data (red) in Douglas County, KS. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

differ are equally useful. The difference may be a result of: a) map errors on the GIS part due to errors in the cartographic process (as shown in Fig. 1b), or b) abnormalities in the remote sensing imagery due to obstruction, shadowing, and even disasters such as flooding. These differences can be automatically identified using image-vector conflation and then used for correcting GIS maps, change detection, road condition monitoring, and disaster relief.

In the remainder of this paper, we provide a brief review of the relevant literature in Section 2. In Section 3, we describe the image-vector conflation workflow based on Res-UNet and optimization-based conflation. We then present the experimental design and results using high-resolution NAIP imagery (0.6 m) and the recent TIGER 2020 road network. Experimental results show that the proposed framework can match most roads faithfully, as indicated by the newly proposed vector-based agreement metrics.

From the outset, it should be noted that we do not attempt to enhance existing deep learning algorithms such as the Res-UNet or propose a new one. Instead, we employ them as a component of the image-vector conflation process (with optimized conflation being the other major component). We adopt the specific DL algorithm (Res-UNet) because it is relatively new and reportedly has good performance. In principle, any modern road surface segmentation model in the deep learning literature can be adapted and used to replace the specific NN that we use as the road extraction component.

2. Background

Conflation is the process of combining two or more datasets into a new dataset with richer information and/or better quality. In the context of remote sensing, the conflation of multiple imagery data is often referred to as data fusion. To conflate the extracted roads from images with existing GIS road networks, the extracted roads should be converted to vectors and then matched to the GIS road network. In this section, we briefly review the background and related literature on vector-based data conflation and road extraction (in that order). From the outset, it should be noted that we do not aim to provide a comprehensive review of either road extraction or vector-based conflation literature. Instead, we review only the representative methods in the literature related to image-vector conflation. Interested readers are referred to (Ruiz, Ariza, Ureña, & Blázquez, 2011; Xavier, Ariza-López, & Ureña-Cámara, 2016) for comprehensive reviews of the GIS data conflation literature and to (Abdollahi, Pradhan, Shukla, Chakraborty, & Alamri, 2020a; Lian, Wang, Mustafa, & Huang, 2020) for the road extraction literature, respectively.

2.1. Optimization based road network conflation

The ultimate goal of the proposed method is to merge a remotely sensed road network with an existing GIS network. This can be conducted in vector form between the extracted road centerlines from imagery and the existing GIS roads. We will review vector-based conflation methods in this sub-section, and review the specific road extraction methods in the next subsection. Different types of vector conflation methods have been developed since the 1980s due to the common need for map conflation in GIS. In the sequel, we briefly cover each type.

2.1.1. Conflation by exact coordinates

The simplest form of vector-based conflation is conflation by coordinates. In essence, two vector GIS layers are superimposed on top of each other, and objects from the two layers that coincide or almost coincide with each other are considered as corresponding objects and merged. This gives rise to the frequently used map overlay method in multi-source GIS data analysis (Fan, Zipf, Fu, & Neis, 2014; Harvey, Vauglin, & Ali, 1998).

In principle, the overlay method is similar to raster-based image fusion (e.g., pan-sharpening). In both cases, the underlying assumption is that the data at the same location refer to the same objects on the ground. Conflation is then performed at each location (pixel-based or otherwise). A main challenge for overlay based methods is that the underlying collocation assumption does not always hold. This is especially true for vector data because the vector data model allows infinite precision in coordinates, and two objects in vector format almost never agree with each other in coordinates. Significant spatial displacement may also exist because of the land surveying and map production processes. Several measures have been proposed to address the spatial displacement.

2.1.2. Overlap measurement

One possible way of dealing with spatial displacement is to relax the co-location requirement and only require that corresponding objects have significant overlap with each other. This criterion is usually expressed as the ratio between the intersection of two objects and the sum, minimum or maximum area of the two objects (Fan et al., 2014; Harvey et al., 1998). If this ratio is greater than a certain threshold (e.g., 90%), the two objects are considered identical and matched. Apparently, the overlapping ratio only applies to polygons. Linear features and point features typically do not have non-zero overlapping areas. However, one can easily generate buffer polygons (Goodchild & Hunter, 1997) for linear and point features and then measure the overlap ratio of the buffer polygons instead. It should be noted that the overlapping ratio is essentially the same metric as the widely used Intersection over Union (IoU) measurement for the degree of measurement of raster-based representation of objects. The difference is that, in the IoU metric, there is no distinct object boundary, and the comparison is based on overlapping pixels in a gridded system.

2.1.3. Rubbersheeting

Another possible method for dealing with spatial displacement is to remove it or at least reduce it. A widely adopted process known as rubbersheeting (Saalfeld, 1985; Saalfeld, 1988) attempts to remove the spatial displacement between two GIS datasets via a divide-and-conquer strategy. The idea is to identify many control points, known as “anchors” that divide the entire study area into many triangular patches between these control points. Each triangular patch is small enough so that spatial displacement pattern is uniform. After moving anchors in one dataset to collocate with their counterparts in the other dataset, a continuous transformation (known as “rubbersheeting”) is applied to move all features within the patch to reduce displacement.

2.1.4. Greedy and heuristics based conflation methods

While overlapping and rubbersheeting methods limit and reduce the amount of spatial displacement, a more general approach to match GIS features is to measure displacement using distance metrics between objects and match object pairs with smaller distances. Commonly used distances include the Euclidean distance, the Hausdorff distance, and the Frechet distance (among others). A prototypical example of distance based methods is the nearest-neighbor join. It assigns each object to its closest counterpart in the other dataset. An advantage of such simple methods is that they are readily available in most GIS systems (spatial join operations). One main disadvantage, as pointed out in (Beeri, Kanza, Safra, & Sagiv, 2004), is that they can be inconsistent. Even for point features, (Beeri et al., 2004) demonstrated that the notion of the closest feature is not symmetric. Given two objects a, b in dataset I and an object c in dataset J, it could be the case that c is the object closest to a in J but b (rather than a) is the object closest to c in I. If a greedy closest assignment strategy is used, a would be identified with b but b would be identified with a different object c, leading to inconsistency.

An alternative method called the K-closest pair queries (KCPQ) iteratively selects the least distance pair from all candidate object pairs between the two datasets and identify them. (Beeri et al., 2004) themselves proposed a “probabilistic” measure to reconcile the inconsistency between the two one-sided nearest neighbor joins. This involves

computing a score between 0 and 1 for nearby features and assigning features based on the score.

Generally speaking, the rudimentary distance-based methods and their variants are often greedy in nature. They may be trapped in a local optimal solution based on the features they have observed thus far. However if a better solution is encountered in the later stages of the search, heuristic based methods generally lack the capability to change to better solutions.

2.1.5. Optimization based conflation modeling

Another approach, conceptualized in the early days of vector conflation research (Rosen & Saalfeld, 1985), is the optimization-based method. Starting with the "Map assignment problem" of (Rosen & Saalfeld, 1985), optimization-based methods treat GIS feature matching as an optimization problem of choosing match relations to minimize the total distance between matched features. The use of the classic assignment problem in operations research for conflation was first conceptualized in (Rosen & Saalfeld, 1985), but was not implemented or experimented with until (Li & Goodchild, 2010; Li & Goodchild, 2011). The assignment-problem based method has been the predominant method in the past few decades.

In its original form, the assignment problem is defined for optimally solving the crew scheduling problem. Given a set of workers and a equal number of jobs, the assignment problem seeks the minimum cost plan for assigning jobs to workers, assuming that each worker has a specific time cost for completing a specific job. Li and Goodchild (Li & Goodchild, 2010) used the assignment problem directly to solve the GIS feature matching problem treating one GIS dataset as the worker set and the other dataset as the job set. The Hausdorff distance between GIS features were used to represent the assignment cost. Since the assignment problem requires the worker set and job set to be equal in size, they have to relax this requirement while matching two GIS datasets (which are typically not equal in size) and only require that all features in the smaller dataset must be assigned.

However, the assignment problem formulation of conflation is flawed. As pointed out in (Lei & Lei, 2019), the assignment problem has the stringent assumption that all objects in the smaller dataset must assign. This can result in distorted matching solutions. To overcome the issues of the assignment problem (Lei, 2020; Lei & Lei, 2019) developed a set of new optimal conflation models based on the network-flow problem. The network flow problem is another classic operations research model, and subsumes other classic optimization problems, such as the assignment problem and the shortest path problem, as its special cases. Specifically, (Lei, 2020) developed two models: a fixed-charge matching (*fc-matching*) model and a fixed-charge bi-matching (*fc-bimatching*) model. The *fc-matching* model, similar to the assignment problem, optimizes one-to-one matches between two datasets. Each object, if matched to a target, is assumed to be the same as the target object. In comparison, the *fc-bimatching* model optimizes many-to-one and one-to-many matches. If an object is matched to a target, it is assumed to *belong to* the target object as a part of it. The *fc-bimatching* model optimizes the two-way many-to-one matches simultaneously in one model, and attempts to avoid the inconsistent matches described in (Beeri et al., 2004).

More advanced optimization-based conflation models exist (Lei & Lei, 2022; Lei & Lei, 2023). (Lei & Lei, 2023), for example, attempts to preserve the connectivity between adjacent edges when matching two road networks, and can therefore improve the reliability of the generated matches. These more advanced optimization models are generally based on Integer Linear Programming (ILP) formulations. Compared to the network-flow based formulations, the ILP-based models are more flexible in expressing various match conditions but are significantly more expensive computationally. This is because the network flow problems have polynomial time solution algorithms (such as the push relabel algorithm), whereas the ILP problems do not have general polynomial time algorithms.

2.2. Road extraction

Fundamentally, the success of image-vector conflation depends on the availability of reliable methods for extracting the target object (roads here) from the imagery. Over the past few decades, a plethora of road extraction methods have been developed owing to the importance of roads as corridors of movement and spatial references. Road extraction methods typically require high-resolution images from satellite or aerial imagery because they are often too thin to extract for low resolution images. Given the US interstate highways' standard lane width of 3.7 m (and similar standards elsewhere), all but the primary roads will be less than one pixel in width in lower resolution images, and therefore difficult to extract. On the other hand, high-resolution images from satellite and aerial platforms can have meter or sub-meter resolution and provide sufficient detail for road extraction. Although a large number of methods have been developed in the past decades, we classify them roughly into two categories and discuss them briefly: 1) traditional methods and 2) learning methods based on neural networks and deep learning.

2.2.1. Traditional road extraction

Traditionally, computerized road extraction methods are based on extracting a set of "features" from remote sensing images, and then using these features to estimate whether pixels are road pixels. These image features can be simple or abstract and complex, and can be defined based on photometric as well as geometric characteristics.

One of the simplest image features that is useful for road extraction is the edge feature (Jensen, 2015). It can be calculated using differential operators which determine the difference between neighboring pixels. A threshold can then be applied to obtain pixels that correspond to locations of changing pixel values, i.e. edges. More sophisticated edge detectors such as the Canny detector have additional capabilities for noisy suppression and so on.

Another classic method for extracting linear features such as roads is the Hough transform (see e.g. (Liu, Zhang, Li, & Tao, 2017)). It involves voting in a parameter space for potential linear features in an image. A target pixel is a vote to every line that passes through itself. Real linear features win, as they obviously will have many votes from its constituent pixels. More sophisticated and efficient Hough transforms exist. The reader is referred to (Liu et al., 2017) for a discussion of generalized Hough transforms.

Mathematical morphology, a technique based on geometrical structures (Haralick, Sternberg, & Zhuang, 1987), is another classic method used in road extraction. Directional mathematical morphology (DMM) operators (Talbot & Appleton, 2007) (i.e., path opening and path closing) are used to remove compact noise while preserving line-like road features in (Valero, Chanussot, Benediktsson, Talbot, & Waske, 2010). Furthermore, Liu et al. (W, Bo, & Wu, 2015) integrated DMM with OpenStreetMap data to achieve better performance.

Another commonly used feature is texture. The image texture is a set of metrics about intensities arrangement in a region used to depict local patterns and irregular sub-elements in an image. Mena and Malpica (Mena & Malpica, 2005) fused several texture measurements for image segmentation in a road extraction pipeline. Wang et al. (Wang et al., 2014) fused the texture and spectral information for road extraction.

2.2.2. Deep Learning based road extraction

In traditional object extraction methods, the human experts are responsible for choosing the image features and then designing various feature detectors. For example, when designing a typical edge detector using a moving window (a.k.a, kernel), the human expert needs to determine the appropriate weight values of the kernels based on experience and reasoning. With larger kernels and more complex features, such as textures, it is becoming increasingly difficult for the human expert to pre-define good weights and other parameter values. This is where the neural-network (NN) and deep learning (DL) based methods

shine. A Neural Network typically consists of multiple layers of Threshold Logic Units (TLUs), where each TLU is a computational unit that applies a (usually linear) transformation of its input values and a threshold to produce an output. Layers of TLUs are inter-connected in the sense that the output of one TLU may feed into the input of another. Each TLU contains a set of parameters (coefficients) for the transformation. Unlike traditional object extraction methods, these parameters are left as unknown values in neural networks, and a learning algorithm is used to learn these parameters from real world training data. This can be achieved, e.g. by penalizing mis-classifications. This data-driven approach for selecting parameters is often more effective than pre-defining parameters especially when the number of parameters is large. Several important NN/DL based methods have been applied to the road extraction problem, as follows.

2.2.3. Patch-based CNN models

The Convolutional Neural Network (CNN) (LeCun, Bottou, Bengio, & Haffner, 1998) is one of the most important neural networks for computer vision and the extraction of various objects, including roads. At the operational level, the CNN consists of many convolution kernels (sliding windows), forming a hierarchy of levels. At the lowest level, convolution is performed at each location on raw pixel values in much the same way as pre-defined convolutional operators (such as edge detectors). In addition, these lower-level kernels extract low-level features such as edges in various directions. The kernel parameters (weight values) are kept as unknowns and learned from the image and the ground truth labels. A unique feature of the CNN is that it uses a hierarchy of kernels to capture features at different scales. While lower-level kernels capture local details such as edges, higher-level kernels can capture larger features spanning a much larger area. This is accomplished by downsampling when defining the sliding windows at the higher levels, which is also known as "pooling". The CNN itself is structured as an interleaved stack of convolutional and pooling layers. The outputs of the various levels are *feature maps*, including low-level features (such as edges) and high-level features (describing larger scale structures). Owing to sampling, the image or feature map becomes smaller and smaller. At the top level, a few fully connected layers are typically used to convert the top feature map into the final prediction.

The final prediction of the CNN is usually the class probability of the entire image, for example, the likelihood of whether the image is for a dog or cat. If applied directly to a road image, the output of the CNN will be whether the image scene has any road. However, road extraction requires the extraction of all road pixels and not just one probability. Patch-based CNNs were proposed (Mnih & Hinton, 2010) to address this issue. The input road image was divided into many small sliding patches that overlapped with each other. A CNN was applied to each patch to classify whether its *center pixel* represented roads. (Mnih & Hinton, 2010) was the first to use a patch-based CNN to extract roads from remote sensing imagery. This was followed by numerous other studies (Wang, Song, Chen, & Yang, 2015, Alshehhi, Marpu, Woon, & Mura, 2017; Li et al., 2016; Saito & Aoki; Wei, Wang, & Xu, 2017)–(Abdollahi, Pradhan, & Shukla, 2020b).

2.2.4. UNet-like models

A main issue of patched-based CNNs is the much repeated computation for training in the overlapping areas. An alternative NN that does not repeat these computations is the UNet model and its various extensions. Unlike the CNNs, the UNet employs *two stacks* of convolution kernels. The first stack is similar to the CNN design and gradually transforms data from larger low-level feature maps to smaller high-level feature maps with higher dimensions. In the big picture, it encodes the original image into high-dimensional feature maps. Symmetrically, the second stack "decodes" the encoded information and gradually transforms data from the high-dimensional feature space back to the image space and produces a prediction image of the same size as the input image. Each pixel in the prediction image contains the class probability of that pixel (e.g., whether it is a road pixel). The prediction image is then compared to a similarly formatted ground truth class image to compute the difference or "loss function" value, which is used to optimize or train the unknown parameters at various levels of the UNet. The letter "U" in the term "UNet" refers to the U-shape of two stacks: one downsampling encoder stack that boils down images to high-dimensional feature maps plus a upsampling decoder stack.

Similar to the CNN, the hierarchical structure of the UNet means that both fine-grained local features and large-scale features can be captured at different levels. The various levels of features are encoded into feature maps at each level of the encoder sub-network. Ideally, these feature maps characterize the essence of the target objects and neglect noise signals. Unlike the CNN, the reverse decoder network is used in UNet to reconstruct a prediction image to match the label image. This design led to the successful application of UNet in image segmentation and road extraction (Litjens et al., 2017).

Various extensions of UNet have been proposed in literature. One important direction is to employ various "skip" mechanisms, called residual block, to solve the so-called gradient explosion and vanishing problem caused by deep network (He, Zhang, Ren, & Sun, 2016). By using residual block in UNet, ResUNet (Zhang, Liu, & Wang, 2018) further improved stability and accuracy in road extraction. More flexible skip mechanisms have also been proposed in a variant of UNet called UNet++ (Zhou, Siddiquee, Tajbakhsh, & Liang, 2020), which provided the Res-UNet implementation that is used in this article.

2.3. Road centerline extraction

One remaining task for image-vector conflation is the extraction of centerlines from predicted road images. As a result of typical image segmentation algorithms, the predicted roads are in a pixelated format and are represented as a set of road regions, or the raster equivalent of polygons. These regions still need to be converted into polylines (i.e., road centerlines). This process, which is sometimes known as thinning or skeletonization, can be tricky. Depending on the algorithm used, the extracted road regions can be fragmented teeming with spurious lines and noise. Some algorithms, such as the Canny edge detector, have built-in noise-suppression and thinning functions. In general, thinning requires a separate step. In the deep learning literature, a distinctive method for extracting centerlines is via multi-task learning (Qi, Liu, Yang, Guan, & Wu, 2017). The basic idea is to learn related tasks simultaneously with a shared representation of the multiple tasks at hand (road segmentation and centerline extraction). The interested reader is referred to the review paper by (Liu, Wang, Yang, Li, & Zhang, 2022) for further details.

Most conventional and deep learning based road centerline extraction algorithms generate centerlines in pixel form (Cheng et al., 2017; Liu et al., 2019; Lu et al., 2019; Shao, Zhou, Huang, & Zhang, 2021; Yang et al., 2019), and therefore cannot be used by GIS. Therefore, a vectorization process is necessary to convert them to discrete GIS objects for conflation. In the next section, we will present a vector-based thinning framework that directly converts road polygons to centerlines.

2.3.1. Directly extracting centerlines from images

Other than the above image segmentation based methods, there is also a class of remote sensing road extraction methods that directly extract the roads in vector form. For example, Bastani et al. (Bastani et al., 2018) proposed an iterative road centerline tracking method, called RoadTracer, which uses a window centered on a given position at each step of the tracking to determine the direction and the action of the next tracking step. Constrained by the number of origins, locations, and fixed step sizes, RoadTracer often reportedly results in incomplete extracted roads and displaced intersections. Wei et al. (Wei, Zhang, & Ji, 2019) proposed a multi-point tracing strategy, called MspTracer, in which they traced the centerline of a road starting from multiple intersections in the road network. The road network is obtained by fusing

the road traces. In order to correct the road offset caused by the fixed step size in the RoadTracer, Tan et al. (Tan, Gao, Li, Cheng, & Ren, 2020) proposed a VecRoad framework with variable step sizes, guided by the so-called trace graph and the intersections. VecRoad can obtain road maps that are closer to the real scene. While iterative road tracing is good at maintaining road connectivity, it is computationally slow due to its iterative nature. In a one-pass style, He et al. (He et al., 2020) proposed a framework for generating road graphs directly from images (Sat2Graph), which encodes road graphs into a tensor through graph tensor coding (GTE) to train a non-recursive, supervised model. The model predicts the whole road map from the input image and therefore accomplishes road extraction. Gaetan et al. (Bahl, Bahri, & Lafarge, 2022) proposed a one-pass road vector extraction method with similar principles.

It should be noted that although the above new methods can generate road centerlines directly, these centerlines still need to be conflated with GIS data. They can potentially suffer from the same incompleteness and noise issues as other road extraction methods. They can be used potentially as a **component** of our image-vector conflation workflow (just as segmentation-based methods), but cannot replace the conflation itself. We used the geometric methods to extract centerlines not because that it is the only method, but rather because we will need GIS in the conflation process eventually, and letting GIS to extract the centerline allows a wider range of road extractions methods (conventional or new) to be utilized. In principle, these road extraction methods can be used interchangeably in our framework as long as they produce road centerlines in the end.

2.3.2. Evaluation metrics

To evaluate the performance of road extraction algorithms, a commonly used performance metric is the Intersection over Union (IoU). It is defined as the ratio between 1) the size of the intersection of the set of predicted road pixels (from the algorithm) and the set of labeled road pixels (from the ground truth) and 2) the size of their union. Clearly, the larger the IoU, the higher the degree of agreement between the computer predicted roads and the labeled ground truth.

Alternatively, another commonly used set of metrics are the True Positives (TP), False Positives (FP), False Negatives (FN), and various rates derived from these numbers. Originally from the information retrieval domain, the TP is defined as the number (or set of) road pixels that are correctly classified as roads. The FP is defined as the number of non-road pixels falsely classified as road pixels. The FN is the number of road pixels incorrectly classified as non-road pixels. Based on this definition, commonly used metrics, such as the recall rate, the precision rate, and the F1-score can be computed. Existing performance metrics are pixel-based, and are routinely computed in most research articles and mainstream libraries such as TensorFlow and PyTorch. A potential issue with these pixel-based performance metrics, is that they can be disrupted by spatial displacement between the remote sensing and GIS datasets. Therefore, they cannot be directly used to measure the degree of agreement in raster-vector conflation, as will be demonstrated in later sections.

In summary, a prerequisite for automatic integration of raster and vector data is object matching or conflation. While vector-vector conflation is still an active research topic, image-vector conflation poses new challenges. This is due to the complex image scenes, inherent noises in the image data, and the fragmented nature of the extracted objects (in pixel or vector forms). The image-vector conflation problem is an area worthy of further investigation, and the problem we will address in the remainder of this article.

3. Methodology

This section presents the methodology and main workflow of the proposed image-vector conflation framework. First, we describe the extraction of road regions using the Res-UNet model. We then discuss methods for converting the pixel-based road regions to a polyline format as well as important practical issues therein. Next, we detail the use of state-of-the-art optimization-based conflation model to match the algorithm-generated roads and the ground truth roads (TIGER/Line 2020). We then discuss the issues of performance evaluation and propose a new set of vector-based metrics to complement traditional pixel based metrics.

3.1. Generation of vector road network from remotely sensed data

3.1.1. Training and predicting road pixels

In the first step, we extract road surface pixels in the high resolution remote sensing images. In the NAIP images we use, each pixel is only 0.6 m by 0.6 m, and typical roads of 10 m width measure more than 15 pixels. Since the image resolution is high, road lines appear to be large and elongated objects. To effectively extract these large road segments, we use a relatively simple but robust deep learning model called Res-UNet (Zhang et al., 2018; Zhou et al., 2020).

Similar to classical UNet model, the ResUNet model we use consists of a contracting (encoding) path, an expansive (decoding) path, and a middle part. In the contracting path, the input image is encoded into a compact representation. The number of level of blocks along the path is called the model depth. In the expansive path, which has the same number of levels of up-sampling blocks, a prediction image is generated in which each pixel is an object class (road vs. non-road). A middle part connects these encoding and decoding parts so that the whole structure forms a U-shape.

Unlike regular UNet, ResUNet uses residual units as the basic building block (Zhang et al., 2018). Typical residual unit contains two 3 $\times$ 3 convolution blocks plus an "identity mapping" that links the output of the unit with the input of the unit. Each of the two convolutional blocks consists of a Batch Normalization (BN) layer, a ReLU activation layer and a convolutional layer. The identity mapping connecting the input to the output essentially makes the residual unit train the residual signal rather than the original signal, and gives rise to the term residual learning. This "skip" link allows the use of the lower-level features at the input to reinforce larger-scale features at the higher level output, thereby reducing information loss and allowing the learning model to have a much greater depth (He et al., 2016). By combining UNet's power of recovering hierarchical contextual information and Residual learning's power of enabling deeper models, ResUNet (Zhang et al., 2018) reaches a satisfying balance between complexity and robustness that is required for large scale road extraction.

In our application of road extraction in NAIP images, we adopt a ResUNet structure for training and prediction with the following specifications based on fine-tuning. We use a UNet with residual connections and a model depth of 5. Resnet-34 is used as encoder backbone. The total number of model parameters are 24 million (24,436,369). We use tanh function as the activation function and MSE as the loss function. The initial learning rate is set to 0.0002 and the "RMSprop" optimizer is used with a weight_decay of 10^{-8}.

3.1.2. Polygonization of predicted images and extraction of medial axes

Once the road regions are predicted, we verified that the quality of these regions were sufficiently high. Although road regions are occasionally broken into disconnected parts, they are generally continuous and elongated in shape so that they can be converted to polylines. There are multiple methods to extract road centerlines from binary road images. One possibility is raster-based polygon thinning with mathematical morphology operators, in which the road regions are narrowed down to one pixel wide. One can then connect the pixels and identify the links formed by tracing contiguous pixels. Depending on the algorithm used, the thinning result may not coincide with the real centerline and may also have too many details because of the pixelated nature of the road regions.

In this study, we adopt a geometric approach by converting the road

regions in the prediction images to polygons upfront (using GDAL's gdal_polygonize command). This allows us to preserve the shape information as much as possible. We then extract and process the centerline of each road polygon. We define the center line of a road polygon as its medial axis (which can be computed using the PostGIS ST_ApproximateMedialAxis() function). In computational geometry, the medial axis of a polygon is the set of points with equal distances to two or more edges. Clearly, this is a faithful characterization of the concept of "center" line. During the road polygonization process (using GDAL), we accumulate road polygons extracted from all image tiles into a single layer and merges any adjacent road polygons. This allows us to extract continuous road segments across tile boundaries.

3.1.3. Building topology and removing twigs

While the medial axis is a natural characterization of road centerlines, the medial axis of a road polygon typically come with many unwanted branches or "twigs" as shown in Fig. 2.

In one test for Shawnee County, KS (using USDA NAIP 2019 images and TIGER/Line 2020 shapefiles), while the polygonization process generated only 7142 road polygons, the medial axis computation generated 211,996 line segments, which is approximately 30 times more than the polygons. Upon closer investigation, most of these line segments are small "twigs" only a few meters in length (Fig. 2). To reduce the number of twigs in the extracted centerlines, two approaches were adopted. First, we generalized the road polygon shape before computing medial axes (using the PostGIS ST_Simplify() function). By smoothing out the shape details (mostly from pixelated polygon boundaries), the number of twigs was reduced (see Fig. 2a,b).

Although generalization reduces the number of twigs, the medial axis computation still produces a relatively large number of twigs (Fig. 2b). As a second measure, we remove these twigs using a topological analysis (results in Fig. 2c). The main idea is to precisely characterize twigs and then filter them out. In essence, a short edge is a twig if one of its end nodes has a degree of one (dangling), and the other has a degree of at least three (branching). Here, an edge is considered "short" if its length is less than 10 m. Operation-wise, we first create the node-edge topology using the pgRouting library (the (pgr_createTopology()) function) and compute the degree of nodes for all extracted centerlines. We then remove all the edges that meet the degree and length criteria in PostGIS using SQL.

Owing to the large number of edges, the topological pre-processing above is time consuming. Initially, we attempted to compute the topology for the entire Shawnee centerlines in one run. The computation took a long time and eventually caused the PostGIS system to crash with a memory overflow (16 Megabytes). After experimentation, we found that we can reduce the computation time by dividing the centerlines into batches of 40,000 geometries each, and build the topology one batch at a time. This is possible because pgRouting's topology building function (pgr_createTopology()) allows the use of a logical condition to limit the topology-building to a subset of lines in the road table. And we keep track of the range of id numbers for each batch of roads and let pgr_createTopology() to only process those roads in the current batch. After batching, the topology was successfully computed within one hour.

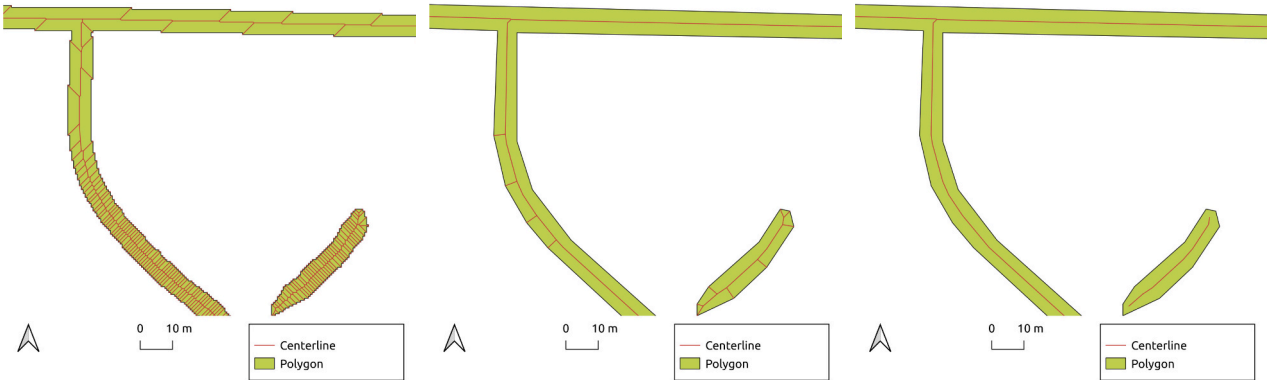
In our routine, the pruning strategy is applied twice to ensure that most twigs are removed. For the Shawnee area, the number of edges was reduced from 211,996 to 97,512 after the first round of pruning, and was further reduced to 96,280 after the second round. We then merged adjacent lines that met each other at degree-two nodes, and the number of edges was reduced to 18,982. Once the number of edges is reduced to a commensurate level to the GIS data, we proceed to the next stage to establish the match relation between the extracted edges and the existing GIS dataset. The workflow for road network generation is illustrated in Fig. 3 below.

3.2. Optimization-based conflation of extracted roads and ground truth

Once the road centerlines are extracted, we relate them to the roads in the target GIS dataset to evaluate whether or to what extent the extracted data agree with the target dataset. Many methods for matching vector datasets have been developed since the early effort on map conflation in the 1980s by the US Geological Survey (Rosen & Saalfeld, 1985; Saalfeld, 1988). In this study, we take the optimization-based approach. As discussed in the Background section, it is capable of capturing the essence of match relations and finding the optimal solutions using off-the-shelf optimization solvers.

We chose the fixed-charge bi-matching model (Lei & Lei, 2019) (*fc-bimatching*) as the main model for two reasons. First, it is much faster than other Integer Linear Programming (ILP) based models. With the *fc-bimatching* model, we were able to match two road networks with approximately 10 k or 20 k edges within half an hour. Model instances of this scale are often beyond the capabilities of comparable ILP models.

Second, in our application, the spatial offset between road geometries extracted from images and road geometries in the target GIS dataset is present, but is usually not large. This is because we used road data from the latest decennial census (2020) as the GIS database and recent NAIP images acquired in 2019. Both datasets have a relatively high positional precision owing to advancements in surveying and sensor technologies. Consequently, we do not have to deal with large spatial displacements in older GIS datasets (which could amount to a half street block). Additionally, the extracted road centerlines are often broken at places owing to obstruction, etc., and complicated conflation models



a. Original road polygons and their medial axes

b. Improved version with generalized road polygons

c. Final version after topological twig removal

Fig. 2. Road polygons and their medial axes as road centerlines: the "twigs" problem.

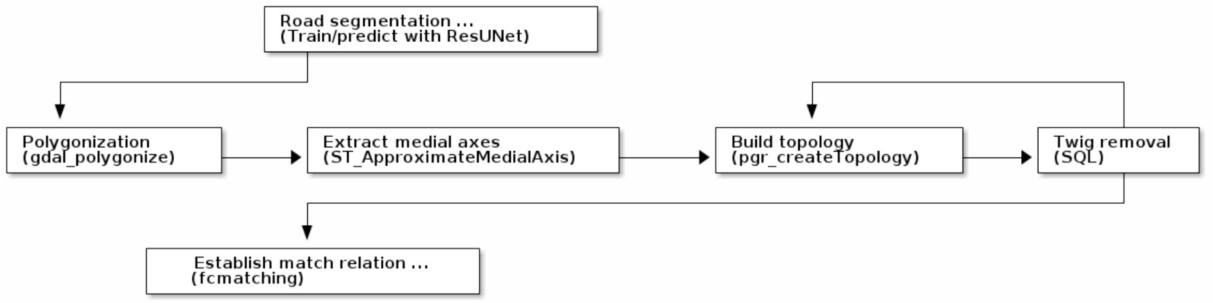


Fig. 3. Workflow for the generation of the vector road network.

with connectivity considerations may be a overkill. However, if more advanced optimization models are needed in the future, they can always be used as a drop-in replacement of the *fc-bimatching* model used in this study.

3.3. Evaluating the degree of agreement in image-vector conflation

One of the tricky issues in image-vector conflation is the effective evaluation of the degree of agreement between the raster and vector datasets. The approach we adopt is rooted in the large body of computer vision literature, in which the accuracy of object extraction is measured in terms of comparing the extracted pixels to the "ground truth". If an object in the image is correctly detected at a pixel position according to a ground truth image, it is counted as a "True Positive" (TP). If an object is present in the image at a pixel but not detected by the algorithm, it is counted as a "False Negative" (FN). If the algorithm detects an non-existing object pixel in the image, it is counted as a "False Positive" (FP).

In the context of image-vector conflation, we measure the agreement/disagreement of discrete objects (in vector form). More specifically, we apply the extraction model we trained to a target geographic region (which may be different from the region where the model is trained). In the target region, we compare the extracted roads with a published GIS road dataset. If an extracted road segment coincides with or is sufficiently close to a road in the published GIS data, this case is considered as a "True Positive". If an extracted road segment is absent from the published GIS data, it is considered a "False Positive". Here, we borrow the terms "ground truth", "True Positives" and recall rates etc. from computer vision for measuring the degree of agreement/disagreement between the raster and image datasets. And there is a subtle difference from the original meaning of these terms in that the GIS road data is not necessarily true or the "ground truth". The GIS data can have map errors too due to the cartographic process or the maps being outdated. Therefore, the metrics we compute should be interpreted as such: the degree of agreement between the roads in the raster and those in the GIS data. With this in mind, we will use the terms the "ground truth" and "target GIS data" interchangeably.

Several assumptions are made in the agreement metrics. First, we assume that the published GIS data is from a credible source and has relatively high positional accuracy. In this study, we use a road dataset from a recent decennial census (TIGER 2020). And we assume that the data quality is reasonably high given the advancement of surveying technology. Second, we assume that the remote sensing imagery has very high spatial resolution (0.6 m in our data) and positional accuracy. This is warranted by the highly calibrated imaging process of modern remote sensors. Combining these factors and based on inspecting the displacement between the images and the GIS data, we assume a cutoff distance of 30 m. If the extracted road is within this cutoff distance from a road in the published GIS data and matched to a published road segment by the matching algorithm (to be described briefly), it is considered a True Positive. We can similarly define False Positives and False Negatives, and compute the following metrics:

$$\text{recall}_0 = \frac{TP}{TP + FN}$$

$$\text{precision}_0 = \frac{TP}{TP + FP}$$

3.3.1. Matching extracted roads to target GIS data

One caveat in the above classic definition of recall and precision is that they do not really apply due to the fragmented nature of extracted roads. This is because the classic definition is based on the assumption of a one-to-one correspondence between the extracted and reference road segments. That is, the geometry of an extracted road segment should be very similar and close to the geometry of the same road in the target GIS dataset in an ideal situation. However, in reality, the extracted road geometries are often fragmented owing to presence of clouds, obstruction by tree canopies, and inundation etc. Consequently, one continuous road is often broken down into several disconnected polylines in the extracted road datasets. This is problematic for the agreement metrics because the one-to-one assumption underlying the classic definition of TP, FP, and FN no longer hold.

To remedy this issue, we have to step back and re-define the match metrics from first principles while accommodating many-to-one correspondence. In essence, recall measures the percentage of objects in the ground truth data captured by object extraction. This implies that recall should be defined in terms of the number of objects in the ground truth. In turn, this means that the True Positives (TPs) and False Negatives (FNs) should be defined in terms of objects in the ground truth. For exposition, we use the letter "r" to designate the ground truth dataset and the letter "e" to designate the extracted dataset. With this notation, recall should be defined in terms of TP_r, which is the number of correctly matched roads **in the ground truth data** *r*. Similarly, precision measures the percentage of objects in the computer generated dataset *e* that are correctly matched to their corresponding objects. Therefore, precision should be defined in terms of TP_e, which is the number of correctly matched objects **in the generated data** *e*. From this discussion, we see that it is necessary to distinguish two types of True Positives (TP): TP_e and TP_r, and define *two-sided* True Positives.

We do not need to define two False Positive (FP) numbers because FP is the number of incorrectly extracted objects in the generated dataset *e*. Likewise, we only need one False Negative number, as it is defined purely in terms of ground truth data *r*. Now that we have extended the definitions of TP, FP, and FN, we can define the recall and precision rates under (two-way) many-to-one correspondence as follows:

$$\text{recall} = \frac{TP_r}{TP_r + FN}$$

$$\text{precision} = \frac{TP_e}{TP_e + FP}$$

The commonly used F1-score is defined in terms of recall and precision, as follows:

$$F1 = \frac{2 \cdot recall \cdot precision}{recall + precision}$$

3.3.2. Computing match metrics using relation algebra

The aforementioned vector-based match metrics can be precisely and conveniently computed from the two GIS layers directly using relational algebra. The conceptual steps are as follows: First, we use the GIS layers to store the matching results. Each object has an id field (for its unique identifier) and a match_id field, which stores the identifier of the target object in the other dataset to which this object is matched. The optimization-based conflation model fills the match_id field appropriately after solving the object matching problem. Then, we can compute the metrics using a relational (join) operators, as follows:

TP_e is defined as e INNER JOIN r then projected to e. TP_r is defined as e INNER JOIN r then projected to r. FP is defined as e LEFT ANTI JOIN r. FN is defined as e RIGHT ANTI JOIN r.

The JOIN condition above is that either e's match_id value is equal to r's id value, or vice versa, r's match_id value is equal to e's id value.

3.3.3. Length weighted match metrics

The performance metrics so far are defined in terms of the **number** of correctly extracted road lines. As an auxiliary metric, we also measure the match rates in terms the length of road segments that are correctly matched. This is accomplished by weighting each road segment with the length of matched portions. For a road polyline in FP or FN, we simply use its entire length as the length that is incorrectly matched (FP) or missed (FN). For a correctly matched road polyline i in generated data e, we define its matched length as the total length of all road segments in the ground truth data r that are matched to i. Similarly, for a road segment j in r, its matched length is the total length of all those in e that are assigned to j.

3.3.4. Comparison with conventional metrics

It should be noted that the match metrics defined above are related to but different from the IoU metric widely used in the remote sensing literature for comparing predicted and ground truth images. Broadly speaking, both our match metrics and the IoU metric measure the percentage of overlap between computer generated/extracted results with some kind of reference roads. The difference between our metrics and the IoU is that we measure the overlap between extracted roads and a target GIS dataset that is not necessarily the ground truth. The GIS dataset is just another useful road dataset in the vector form that we want to merge to. Furthermore, our metrics are based on matching discrete objects (i.e. Vector-based), whereas the IoU metric is based on comparing pixels at each grid point.

As such, the IoU metrics should only be used in the road extraction context for comparing against the ground truth roads (in vector or raster form). They should not be used for general conflation purposes. This is because the IoU metric (and similarly the pixel based F1-score reported by many deep learning libraries) are sensitive to spatial displacement. If the IoU is used as a match metric for image-vector conflation and the spatial offset between extracted road segments in the GIS counterparts is greater than the road width, the IoU (and pixel-based F1-score etc.) will be zero. In practice, such coordinate errors occur frequently because the GIS datasets such as TIGER and OSM do contain coordinate errors (ranging from a few meters to 100 m in some cases). By comparison, our match metrics are suitable for image-vector conflation as they are more resistant to spatial offsets and coordinate errors.

The vector-based match metrics measure the degree of agreement between the image and vector data. They may not be 100% for two different reasons. For one, the road extraction may not be perfect; some error in prediction or vectorization may cause mis-matching, which we call "methodology-induced mismatching". In the second case, which is more interesting, there may exist some real differences between road networks in the image and in the GIS data due to various reasons such as flooding, outdated maps, or GIS data production issues. We call these differences "change-induced mismatching". These changes often provide useful information in environmental change detection and map quality checking. "False Positives" may indicate newly construction roads that are not yet reflected in the target GIS data, and "False Negatives" may indicate anomalies such as flooding, land slide, etc.

4. Experiments

4.1. Experimental settings

We collected the USDA NAIP 2019 images as the remote sensing data and TIGER roads as the GIS data for the Douglas County and Shawnee County, KS. The NAIP 2019 imagery is an aerial dataset with a spatial resolution of 0.6 m and has both natural color and pseudo-color versions. We selected the pseudo-color version with the Infrared (IR), Red (R), and Green (G) bands. We chose the pseudo color version because our initial experiments showed that the pseudo color version with the IR bands seems to have more discriminative power and consistently performs better than the natural color version (although the difference is small). The TIGER/Line road dataset consists of all named roads from the most recent decennial US census (Census 2020). The Douglas County, (1230 km^2 in area) and the Shawnee County (1440 km^2 in area) were selected as the training and testing sites, respectively (as shown in Fig. 4). As the training sets were too large for manually labeling road pixels, we used the TIGER/Line dataset as the baseline road network and used GIS packages to generate ground truth road images within Douglas County, KS, in a similar way to prior research (e.g., (Mnih, 2013)).

Binary ground truth images were generated in accordance with the format of the Deep Globe contest. More specifically, the TIGER/Line roads were converted to buffer polygons with a 5 m radius and then "burnt" into the NAIP 2019 image. In other words, pixels are assigned non-zero or zero values based on whether they are within the road buffer. The Douglas County NAIP image was divided into tiles of 1024 by 1024 pixels. By convention, the tiles are divided into three subsets: training set (70%), testing set (20%) and validation set (10%). To test the generalization capabilities of the road extraction model, we also used the entire image for Shawnee County as another testing set, which was divided into tiles in the same manner. While using a random subset of tiles as the testing dataset is a common practice in deep learning, it is more useful in practice to train the road model for one specific area and apply it in other areas.

The experiments were then carried out following the workflow

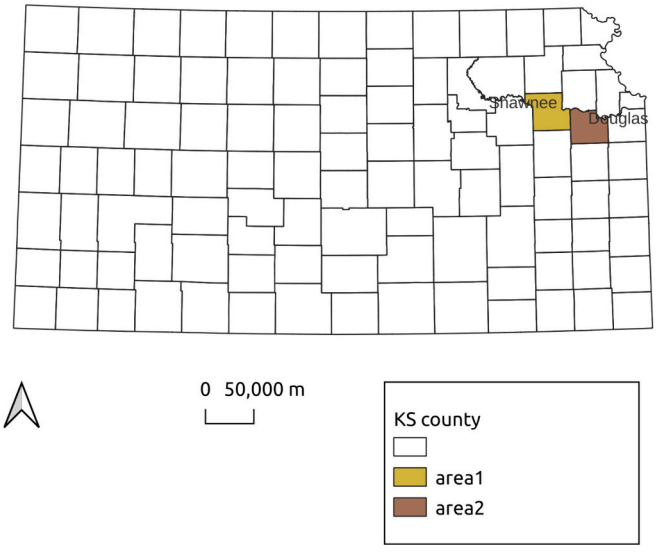


Fig. 4. The study area in Douglas and Shawnee Counties, KS.

described in the Methods section and Fig. 3. The Res-UNet based road extraction model was trained on Douglas County data and was used to extract road areas in Shawnee County, KS. After the polygonization of the predicted images, the extraction of medial axes and twig removal, the extracted roads and ground truth were conflated using the *fc-bimatching* model. The overall image-vector conflation was evaluated using the vector-based metrics described in Section 3.3. Just for comparison purposes, we also computed traditional pixel-based IoU metrics between the extracted roads and the rasterized version of the target GIS roads.

4.2. Overall performance results

In this subsection, we report both conventional pixel-based and vector-based performance metrics (with the F1-score reported for both cases). The metrics are reported for the cross-county experiment in which we trained the road extraction model over Douglas County, KS, and applied it to predict roads in the nearby Shawnee County, KS.

4.2.1. Pixel-based metrics

Overall, the average IoU of the road prediction in Shawnee County was 59.8%, and the average prediction F1-score was 71.2%. This means that nearly 60% of the predicted and rasterized GIS road overlapped. The F1-score was computed using pixel-based TP, FP, and FN numbers using the pytorch library. This suggests that the road extraction performance is adequate, but there are a large portion of pixels (over 40%) where the predicted roads and rasterized GIS roads do not match.

4.2.2. Vector-based match metrics for image-vector conflation

After the vectorization, there were 17,154 polylines in the extracted road layer, while the target GIS layer from the US Census has 10,688 roads in the Shawnee area. The fact that there are more extracted roads than the ground truth demonstrates the fragmented nature of the extracted roads. Note that we define the study area as the intersection between all image tile areas and areas with roads in the ground truth, leaving out empty tiles with no roads. In the non-empty areas, the object-level recall rate is 81.0%. This means that of all the roads in the Census data, over 80% have been matched by extracted roads with the given cutoff distance (30 m). As will be demonstrated shortly, some smaller roads were entirely missed by the road extraction algorithm. The object-level precision rate was 86.7%. This means that nearly 87% of the extracted roads matched the target GIS roads. On average, the object-level F1-score was: 83.8%.

The object-level match metrics gauge the degree of agreement between the extracted roads and the target GIS roads in terms of *the number of* correctly identified objects. In comparison, the measure-based metrics describe the degree of agreement based on the total length of the corrected identified roads. On average, the measure-based recall rate was 88.7%. This means that nearly 90% of all target roads were matched by extracted roads. The measure-based precision was 93.0%. This means that most of the extracted road segments, in terms of length, were matched to the target GIS roads. The measure-based F1-score was 90.8%.

The total computational time of the prediction-matching process was 107 min, tested on a machine with an Intel i5-12400F CPU and a Nvidia RTX 3060 GPU. The prediction step took a little more than 30 min (1915 s), and the optimization-based matching took approximately 2 min (119 s). The rest of the time was spent on the GIS processing for polygonization, medial axis extraction and twig removal.

Fig. 5 visualizes the overall degree of match between the raster and vector roads in the Shawnee County. And Fig. 5a and Fig. 5b depicts the extracted roads and GIS roads, respectively. Comparing the two subfigures, we can observe that overall, the extracted roads match the GIS roads pretty well. Most of the roads in the GIS database have been extracted, and there were not many spurious roads. Note that the blank areas in the NE and SW corners of Fig. 5a were due to missing data (with N/A pixel values) in the NAIP image. So, these areas were excluded from both the IoU and vector-based metrics. From these two figures, it is clear

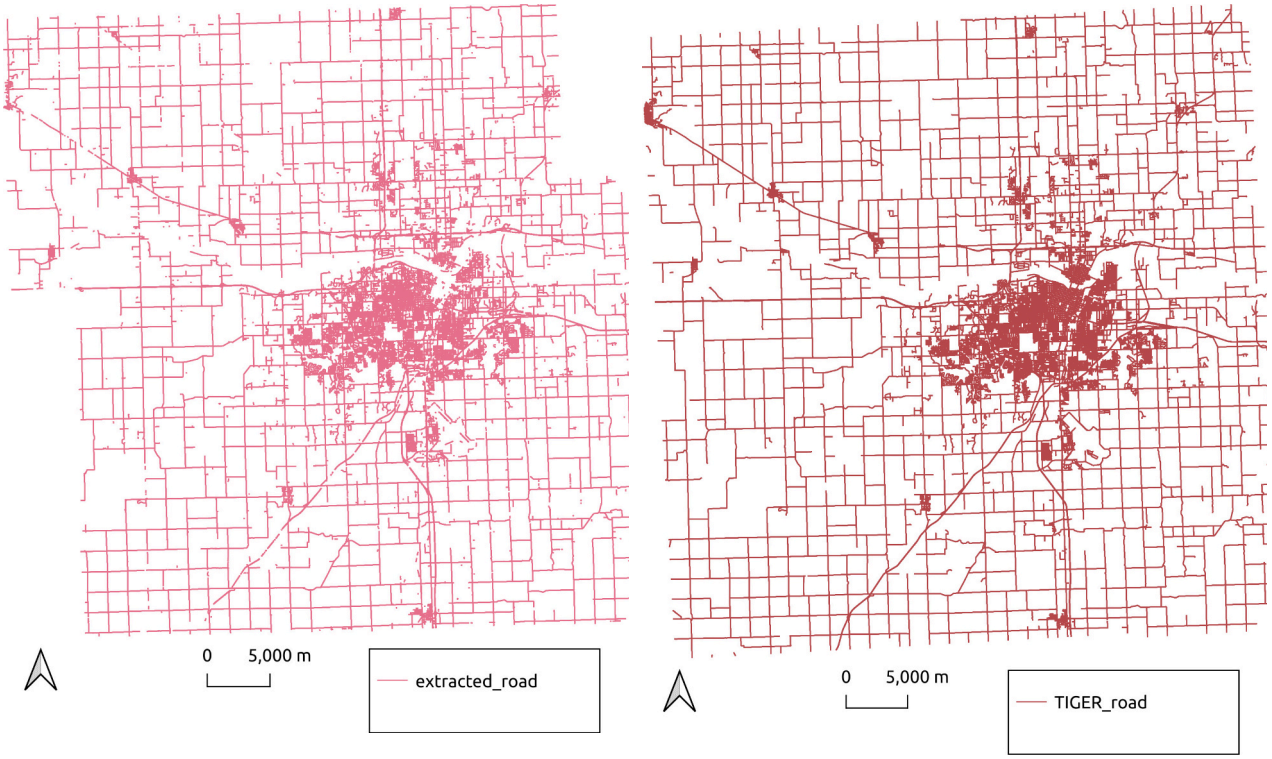


Fig. 5. Comparison between extracted roads and existing GIS roads in Shawnee County, KS.

that the conventional IoU metric is not a good characterization of the degree of match. It is unlikely that only 60% of the roads matched (in terms of mileage).

At face value, the new vector-based metrics were significantly higher than the pixel-based IoU metric. The pixel-based IoU is approximately 60% while the measure-based recall and precision rates were both around 90%. Fig. 5 shows that the pixel-based metrics are not suitable for evaluating image-vector conflation and the vector-based metrics are more appropriate. Presumably, this is because a pixel can be counted as a false positive or false negative just because it is a few pixels off in position. Given the 0.6 m resolution of the NAIP image, this means that a small spatial displacement of a meter or two may cause a pixel to be falsely counted. By comparison, the vector-based metrics speak directly to the intuition about the degree of match between the roads as objects.

4.2.3. *Discussion of the cutoff distance*

In the experiments so far, we have assumed a cutoff distance of 30 m. One natural question is the following. What is the appropriate value for the cutoff and how do different values impact the vector-based agreement metrics? In principle, the cutoff distance is not really a model parameter that should be adjusted from one image scene to the next. Instead, it should be a fixed value reflecting the maximum positional offset for the entire dataset to be conflated. Within this limit, the extracted road segment is considered to represent the same object as the road in the ground truth.

Using a cutoff value that is either too large or too small could have negative consequences. If the value is too small, we risk missing correctly detected roads. If the value is too high, we risk being over-optimistic and count off-road detections as roads. We chose a 30 m cutoff based on **estimated** maximum cutoff and a visual inspection of detected roads and their GIS counterparts (as shown in Fig. 7a,b). This value should be chosen carefully by the human expert for each data source based on the positional accuracy of both the image and vector data.

To evaluate the influence of the cutoff distance, we also repeated the above experiments for additional cutoff values at 15 m and 45 m, respectively. At 15 m cutoff, the object-level recall rate was 77.6%, which was 3.4% lower than the 30 m recall value. This is likely because some of the target GIS roads were considered unmatched due to the lower cutoff. The object-level precision rate was 82.3% (which was 4.4% lower). This is presumably because some of the correctly extracted roads fell outside the cutoff and considered as unmatched. The object-level F1-score is 79.9% (3.9% percent lower).

At 45 m cutoff, the object-level recall rate was 82.8%, which was 1.8% higher than the 30 m recall value. The object-level precision rate was 89.0% (which was 2.3% higher). The F1-score is 85.8% (2% higher). This is presumably because the larger cutoff distance led to a smaller number of unmatched roads in both the set of extracted roads and the target GIS roads. The nominal increase in agreement metrics is relatively small. On the other hand, we should be careful about these far off detected roads when counting them as true matches. And it is advisable to visually inspect them to make sure that they indeed correspond to roads in the ground truth GIS data. Overall, this test suggests that when an appropriate cutoff distance is used, the vector-based metrics are much higher than the IoU (around 60%) and renders a more accurate characterization of the degree of agreement of the overall image-vector conflation process.

4.3. *Case studies*

In this subsection, we demonstrate a number of scenes to gain a better understanding of the difference between the pixel and vector based metrics as well as the quality of the overall image-vector conflation process. Fig. 6 depicts the result of the conflation between the predicted road network and ground truth road network from TIGER-line in the Topeka area. In the figure, the thick lines indicate matched roads.

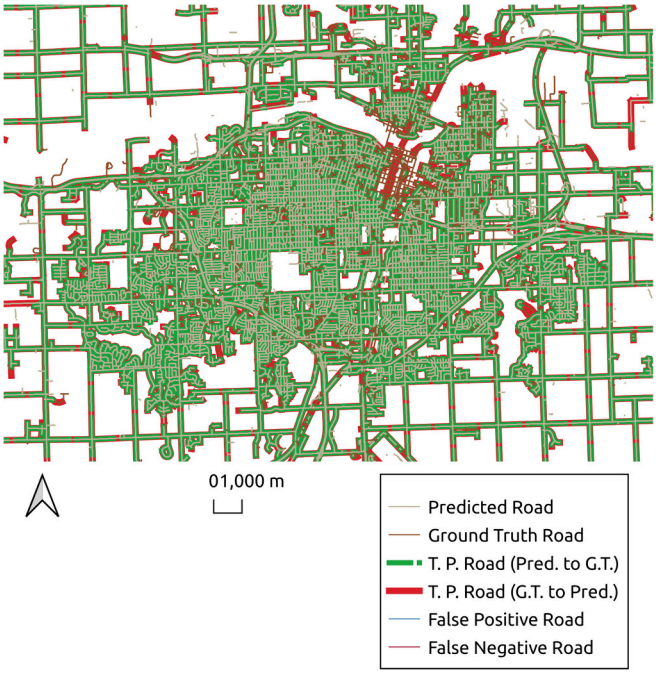


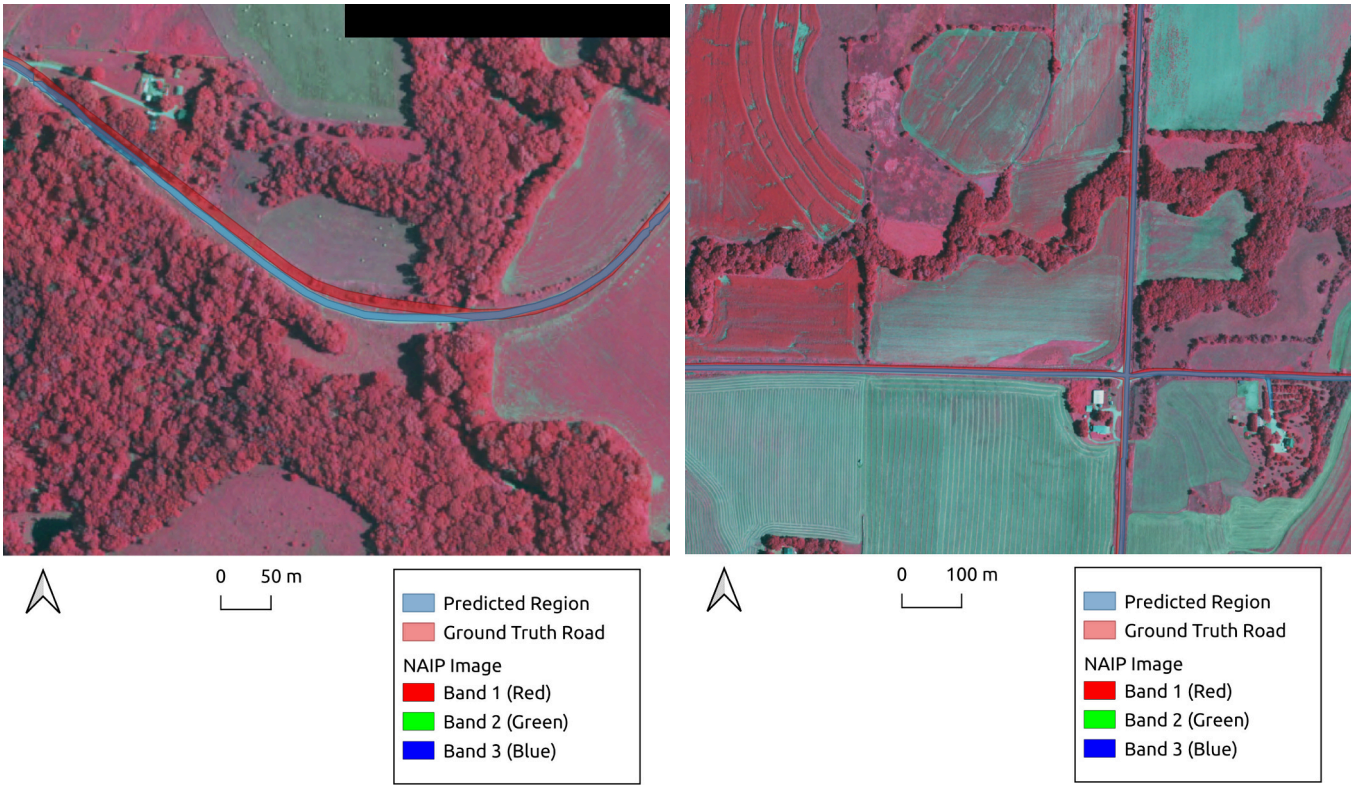
Fig. 6. True Positives (thick), False Positives (thin green) and False Negatives (thin red) in Topeka, KS. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Specifically, the thick green lines represent the extracted roads that are matched to roads in the ground truth. The thick red lines represent matched ground truth roads. Thin green lines represent extracted roads that fail to match any roads in the ground truth (False Positive), and thin red lines represent ground truth roads that are missed by the road extraction (False Negative). From the figure, we can observe that the conflation/matching was mostly correct. While there are missed/mismatched roads here and there, there are a few blocks of missed roads near US I-70 and S Kansas Ave (as will be discussed shortly). This is in accordance with the vector-based metrics presented previously.

To understand the difference between raster- and vector-based metrics, we present in Fig. 7. two example scenes demonstrating the spatial displacement between the extracted roads and the ground truth. In Fig. 7a, we can observe that in the curved portion of the road, the extracted road (in blue) is almost complete off from the ground truth road (in red). For a short length of the road, the pixel-based IoU would be effectively zero. However, any human expert can immediately tell, that the road extraction model did an excellent job and extracted the road pixels even though they are nominally outside the ground truth road area. Upon closer inspection, it can be seen that the western portion of the road suffers from the same displacement issue, although to a lesser degree.

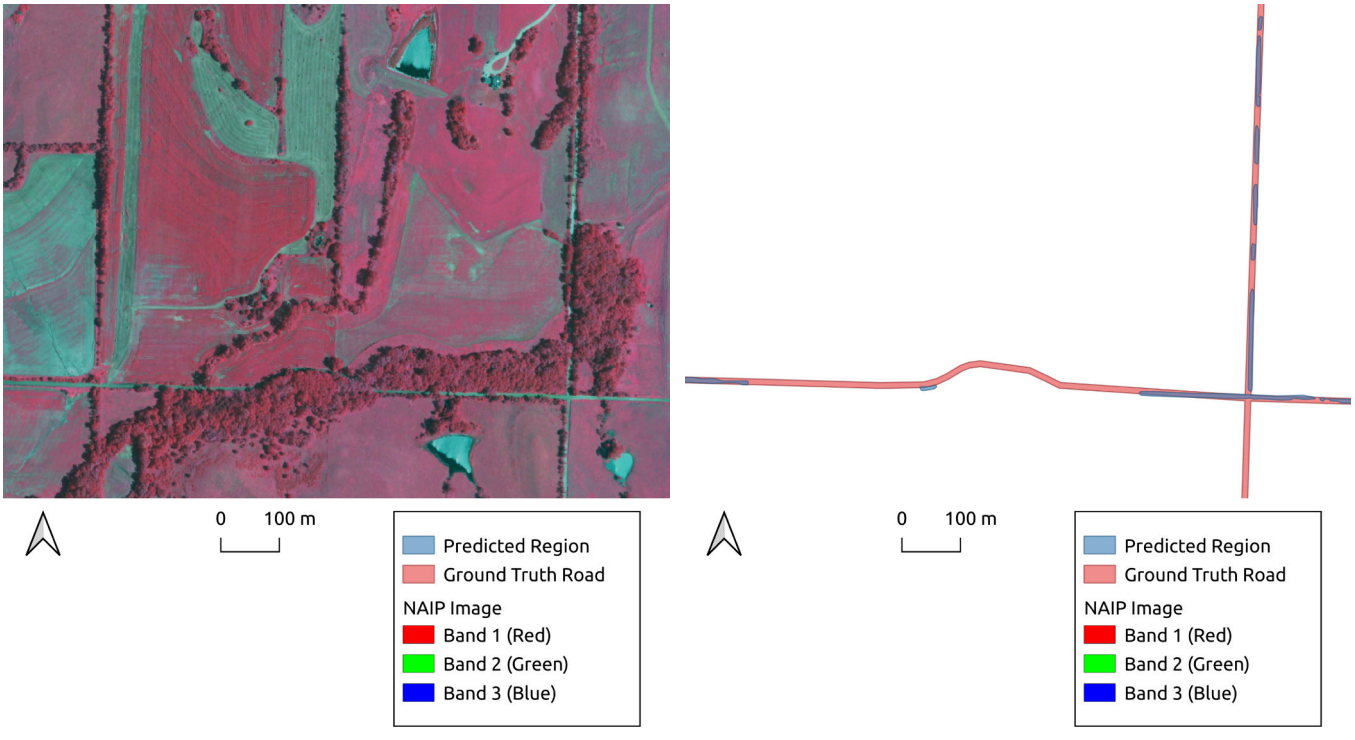
Fig. 7b demonstrates a similar problem. It can be observed that for almost the entire length of the horizontal road, the extracted road polygon is approximately half the road width off relative to the ground truth road area. Similar to the case in Fig. 7a, this would lead to a low pixel-based performance metric (in IoU or F1-score). This is a systematic underestimation of the degree of agreement, given the fact that the algorithm did a near perfect job in extracting the roads here.

Fig. 8a presents a scene in which much of the roads are either obstructed or shadowed by nearby tree crowns. Consequently, the road extraction model is disrupted by these conditions, and only fragments of the roads (blue) are extracted, which constitute a small percentage of the ground truth in red (as shown in Fig. 8b). This means that the measure-based recall for the depicted roads will be low, as a large percentage of the ground truth roads (in terms of length) have no counterparts in the extracted roads. By comparison, the measure-based precision will be



a. Zero IoU at the turn b. Systematic offset

Fig. 7. Spatial displacement between extracted roads (blue) and ground truth in GIS (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



a. Original image b. Extracted (blue) vs ground truth (red)

Fig. 8. Obstruction by tree canopy and shadows.

high (near 100%) in this scene, as almost all road lengths in the extracted data have been accounted for in the ground truth. This scene demonstrates the difference between length-based metrics and the object-count metrics. In the specific case here, the object-count metrics over-count the amount of captured roads, as both roads have some captured portions and will count as two matched roads. The measure-based metrics are a more accurate description of the degree of agreement in this case.

Fig. 9 shows some difficult scenes for road extraction. Fig. 9a shows a scene of the Topeka area near US I-70 and S Kansas Ave. We can observe that most of the roads near the bend of the highway are missed by road extraction. This is probably because this area is used commercially and industrially (with a BNSF terminal). Given a low tree canopy, the surface materials of the shopping areas and the terminal look similar to those of the roads. Another potential reason is that most of the training areas (Douglas County) are either rural areas or small towns, which lack high density land-use areas. Consequently, such roads are missing in the training dataset, and the DL network did not encounter these types of roads and their environments. Fig. 9b depicts a missing road situation in a rural area near 106th Road (horizontal) and I Road (vertical). We can observe that the horizontal road was detected. However, the vertical road (I road) was missed by the algorithm. This is probably due to the fact that the vertical road is too narrow compared to typical roads in the training data. In this study, we used a fixed buffer radius for all roads. In future work, this could be remedied by using actual road widths from the GIS dataset (if available).

5. Conclusion and future work

Conventional road extraction research mostly stops at the boundary between image and vector analyses. The extracted road segments or pixels have rarely been merged to existing GIS data. In this paper, we argue that a more effective way of road data production is to combine road data in the raster and the vector forms into one new dataset, and we propose such a framework for full image-vector conflation. In addition to the road extraction phase, we propose to use GIS and optimization to merge the extracted road centerlines to published GIS datasets by establishing a match relation between them.

To achieve full image-vector conflation, we needed to handle the complex match relation between the fragmented road extractions and the target GIS road data. Via an analysis of this relation, we defined vector-based match metrics for image-vector conflation based on comparing the extracted road centerlines and the target GIS road data. In particular, we defined and used two-sided true positives between the two datasets to account for the typical many-to-one matching between the fragmented extracted roads and the target GIS roads. We also defined length-weighted versions of the match metrics that gauge the degree of agreement in terms of the total length of matched roads.

To verify the effectiveness of the framework, we created a large-scale experiment data set covering two counties in Kansas, USA, using one county (Douglas) as data for training a Res-UNet based road extraction model. We used the other county's data (Shawnee) to test the image-vector conflation process.

On average, the object-count based recall, precision and F1-score were 81.0%, 86.7% and 83.8%, respectively. The measurement-based (i.e., length-based) recall, precision and F1-score were 88.7%, 93.0% and 90.8%, respectively. These results indicate that the degree of agreement between the roads in the imagery and those in the target GIS dataset is quite high. In terms of total lengths, nearly 90% of all roads were correctly extracted and matched, and 93% of the extract centerlines corresponded to actual roads. By comparison, conventional pixel-based metrics such as the IoU are not suitable for measuring the degree of agreement due to their sensitivity to spatial displacement.

We also detailed the major steps of the conflation framework and discussed some of the difficult cases in the Experiment section. Based on this discussion, several directions are worthy of future study. First, the centerline extraction process (especially the topology building and "twig" removal) is time consuming. Future work could improve the centerline extraction process by dividing the entire area into smaller blocks and processing each block sequentially (or even in parallel).

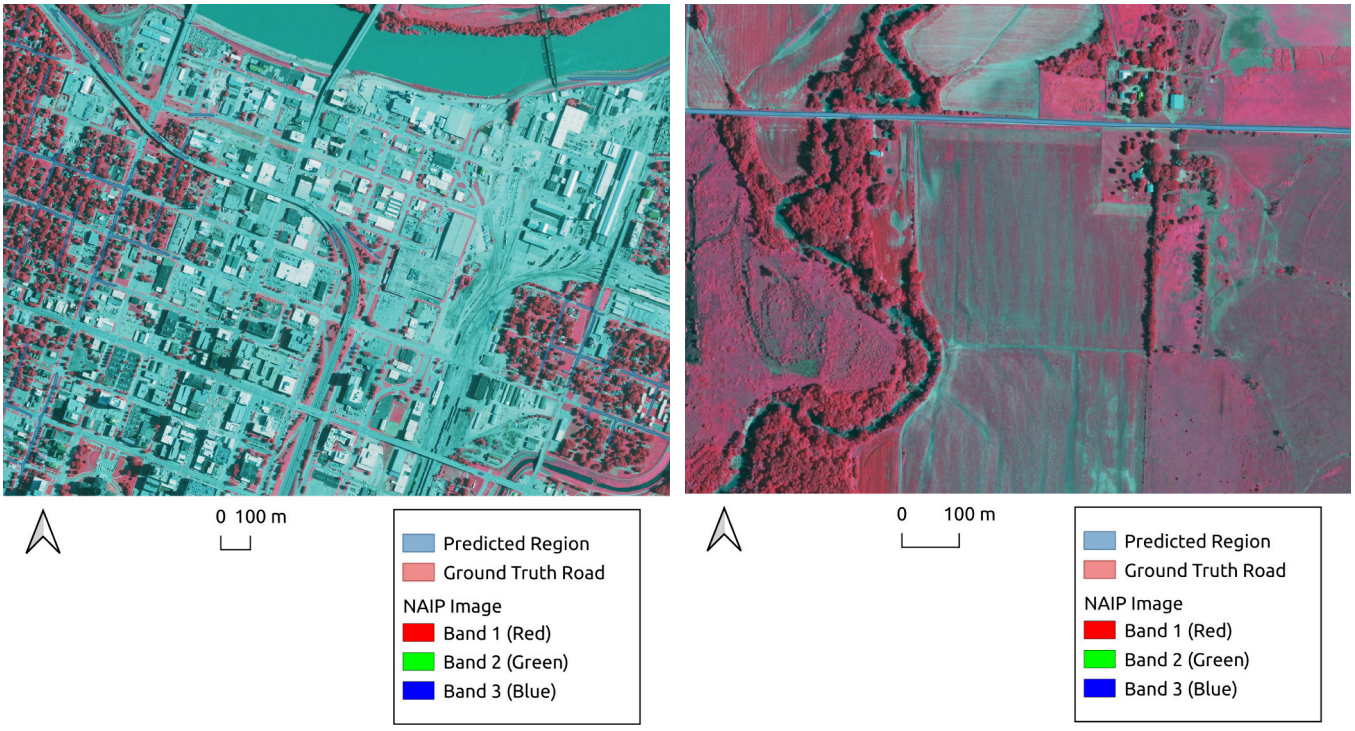


Fig. 9. Miscellaneous difficult scenes for road extraction.

Second, the case studies showed that the training data did not include certain areas (high-density commercial areas) and consequently did not recognize these types of roads in the testing area. Future work could expand the training data to include more representative environments. On the GIS side, some roads were missed, probably because of their smaller widths. In future work, the ground truth dataset could be enhanced by adopting varying buffer radii based on actual or estimated road widths (rather than a fixed buffer width).

Credit authorship contribution statement

Zhen Lei: Writing – review & editing, Writing – original draft, Software, Resources, Methodology, Conceptualization. **Ting L. Lei:** Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

None.

Acknowledgements

The authors hereby acknowledge that they have not used Generative AI tools in writing this manuscript. This research was partly supported by the National Natural Science Foundation of China (NSFC) (Grant no. 41971334). This research was partly supported by the National Natural Science Foundation (NSF) (Grant no. BCS-2215155).

References

Abdollahi, A., Pradhan, B., Shukla, N., Chakraborty, S., & Alamri, A. (2020a). Deep learning approaches applied to remote sensing datasets for road extraction: A state-of-the-art review. *Remote Sensing, 12*(9). https://doi.org/10.3390/rs12091444

Abdollahi, A., Pradhan, B., & Shukla, N. (2020b). Road extraction from high-resolution orthophoto images using convolutional neural network. *Journal of the Indian Society of Remote Sensing, 49*. https://doi.org/10.1007/s12524-020-01228-y

Alshehhi, R., Marpu, P. R., Woon, W. L., & Mura, M. D. (2017). Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing, 130*, 139–149. https://doi.org/10.1016/j.isprsjprs.2017.05.002

Bahl, G., Bahri, M., & Lafarge, F. (2022). Single-shot end-to-end road graph extraction. In *2022 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 1402–1411). https://doi.org/10.1109/CVPRW56347.2022.00146

Bastani, F., He, S., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., & DeWitt, D. (2018). RoadTracer: Automatic extraction of road networks from aerial images. *IEEE/CVF conference on computer vision and pattern recognition, 2018*, 4720–4728. https://doi.org/10.1109/CVPR.2018.00496

Beeri, C., Kanza, Y., Safra, E., & Sagiv, Y. (2004). Object fusion in geographic information systems. In *Proceedings of the thirtieth international conference on very large data bases-volume 30* (pp. 816–827).

Cheng, G., Wang, Y., Xu, S., Wang, H., Xiang, S., & Pan, C. (2017). Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing, 55*(6), 3322–3337. https://doi.org/10.1109/TGRS.2017.2669341

Fan, H., Zipf, A., Fu, Q., & Neis, P. (2014). Quality assessment for building footprints data on OpenStreetMap. *International Journal of Geographical Information Science, 28*(4), 700–719.

Goodchild, M. F., & Hunter, G. J. (1997). A simple positional accuracy measure for linear features. *International Journal of Geographical Information Science, 11*(3), 299–306.

Haralick, R. M., Sternberg, S. R., & Zhuang, X. (1987). Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI–9, no. 4*, 532–550. https://doi.org/10.1109/TPAMI.1987.4767941

Harvey, F., Vauglin, F., & Ali, A. B. H. (1998). Geometric matching of areas, comparison measures and association links. In *Proceedings of the 8th international symposium on spatial data handling* (pp. 557–568).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90

He, S., Bastani, F., Jagwani, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Elshrif, M., Madden, S., & Sadeghi, A. (Jul. 2020). *Sat2Graph: Road graph extraction through graph-tensor encoding*.

Jensen, J. R. (2015). *Introductory digital image processing: A remote sensing perspective* (pearson series in geographic information science), (4th ed.). Pearson.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Lei, T. L. (2020). Geospatial data conflation: A formal approach based on optimization and relational databases. *International Journal of Geographical Information Science, 34*(11), 2296–2334.

Lei, T., & Lei, Z. (2019). Optimal spatial data matching for conflation: A network flow-based approach. *Transactions in GIS, 23*(5), 1152–1176.

Lei, T. L., & Lei, Z. (2022). Harmonizing full and partial matching in geospatial conflation: A unified optimization model. *ISPRS International Journal of Geo-Information, 11*(7), 375.

Lei, T. L., & Lei, Z. (2023). Linear feature conflation: An optimization-based matching model with connectivity constraints. *Transactions in GIS, 27*(4), 1205–1227.

Li, L., & Goodchild, M. F. (2010). Optimized feature matching in conflation, in *Geographic information science: 6th international conference. GIScience*, 14–17.

Li, L., & Goodchild, M. F. (2011). An optimisation model for linear feature matching in geographical data conflation. *International Journal of Image and Data Fusion, 2*(4), 309–328.

Li, P., Zang, Y., Wang, C., Li, J., Cheng, M., Luo, L., & Yu, Y. (2016). Road network extraction via deep learning and line integral convolution. In *, 2016. IEEE international geoscience and remote sensing symposium (IGARSS)* (pp. 1599–1602). https://doi.org/10.1109/IGARSS.2016.7729408

Lian, R., Wang, W., Mustafa, N., & Huang, L. (2020). Road extraction methods in high-resolution remote sensing images: A comprehensive review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13*, 5489–5507. https://doi.org/10.1109/JSTARS.2020.3023549

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., … Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis, 42*, 60–88. https://doi.org/10.1016/j.media.2017.07.005

Liu, W., Zhang, Z., Li, S., & Tao, D. (2017). Road detection by using a generalized hough transform. *Remote Sensing, 9*(6), 590.

Liu, Y., Yao, J., Lu, X., Xia, M., Wang, X., & Liu, Y. (2019). RoadNet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images. *IEEE Transactions on Geoscience and Remote Sensing, 57*(4), 2043–2056. https://doi.org/10.1109/TGRS.2018.2870871

Liu, P., Wang, Q., Yang, G., Li, L., & Zhang, H. (Feb. 2022). Survey of road extraction methods in remote sensing images based on deep learning. In *, 90. PFG – Journal of Photogrammetry Remote Sensing and Geoinformation Science* (pp. 1–25). https://doi.org/10.1007/s41064-022-00194-z

Lu, X., Zhong, Y., Zheng, Z., Liu, Y., Zhao, J., Ma, A., & Yang, J. (2019). Multi-scale and multi-task deep learning framework for automatic road extraction. *IEEE Transactions on Geoscience and Remote Sensing, 57*(11), 9362–9377. https://doi.org/10.1109/TGRS.2019.2926397

Mena, J. B., & Malpica, J. A. (2005). An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery. *Pattern Recognition Letters, 26*(9), 1201–1220. https://doi.org/10.1016/j.patrec.2004.11.005

Mnih, V. (2013). *Machine learning for aerial image labeling, PhD thesis*. University of Toronto. Available https://www.proquest.com/dissertations-theses/machine-learning-aerial-image-labeling/docview/1500835065/se-2.

Mnih, V., & Hinton, G. E. (2010). Learning to detect roads in high-resolution aerial images. In *Computer vision – ECCV 2010* (pp. 210–223).

Qi, K., Liu, W., Yang, C., Guan, Q., & Wu, H. (2017). Multi-task joint sparse and low-rank representation for the scene classification of high-resolution remote sensing image. *Remote Sensing, 9*(1). https://doi.org/10.3390/rs9010010

Rosen, B., & Saalfeld, A. (Mar. 1985). Match criteria for automatic alignment. In *Proceedings of 7th international symposium on computer-assisted cartography* (pp. 1–20).

Ruiz, J. J., Ariza, F. J., Ureña, M. A., & Blázquez, E. B. (2011). Digital map conflation: A review of the process and a proposal for classification. *International Journal of Geographical Information Science, 25*(9), 1439–1466. https://doi.org/10.1080/13658816.2010.519707

Saalfeld, A. (1985). A fast rubber-sheeting transformation using simplicial coordinates. *The American Cartographer, 12*(2), 169–173.

Saalfeld, A. (1988). Conflation automated map compilation. *International Journal of Geographical Information Systems, 2*(3), 217–228.

Saito, S., & Aoki, Y. (Feb. 2015). Building and road detection from large aerial imagery. In *, 9405. Proceedings of SPIE - The International Society for Optical Engineering*. https://doi.org/10.1117/12.2083273

Shao, Z., Zhou, Z., Huang, X., & Zhang, Y. (2021). MRENet: Simultaneous extraction of road surface and road centerline in complex urban scenes from very high-resolution images. *Remote Sensing, 13*(2). https://doi.org/10.3390/rs13020239

Talbot, H., & Appleton, B. (2007). Efficient complete and incomplete path openings and closings. *Image and Vision Computing, 25*(4), 416–425. https://doi.org/10.1016/j.imavis.2006.07.021

Tan, Y.-Q., Gao, S.-H., Li, X.-Y., Cheng, M.-M., & Ren, B. (2020). VecRoad: Point-based iterative graph exploration for road graphs extraction. *IEEE/CVF conference on computer vision and pattern recognition (CVPR), 2020*, 8907–8915. https://doi.org/10.1109/CVPR42600.2020.00893

Valero, S., Chanussot, J., Benediktsson, J. A., Talbot, H., & Waske, B. (2010). Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. *Pattern Recognition Letters, 31*, 1120–1127. Available https://api.semanticscholar.org/CorpusID:1774087.

W, H. A., Bo, L., & Wu, A. N. D. (Sep. 2015). Main road extraction from ZY-3 grayscale imagery based on directional mathematical morphology and VGI prior knowledge in urban areas. *PLoS One, 10*(9), 1–16. https://doi.org/10.1371/journal.pone.0138071

J. Wang, Q. Qin, X. Yang, J. Wang, X. Ye, and X. Qin, "Automated road extraction from multi-resolution images using spectral information and texture," in 2014 IEEE geoscience and remote sensing symposium, 2014, pp. 533–536. doi:https://doi.org/10.1109/IGARSS.2014.6946477.

Wang, J., Song, J., Chen, M., & Yang, Z. (2015). Road network extraction: A neural-dynamic framework based on deep learning and a finite state machine. *International Journal of Remote Sensing, 36*(12), 3144–3169. https://doi.org/10.1080/01431161.2015.1054049

Wei, Y., Wang, Z., & Xu, M. (2017). Road structure refined CNN for road extraction in aerial image. *IEEE Geoscience and Remote Sensing Letters, 14*(5), 709–713. https://doi.org/10.1109/LGRS.2017.2672734

Wei, Y., Zhang, K., & Ji, S. (2019). Road network extraction from satellite images using CNN based segmentation and tracing. In *IGARSS 2019–2019 IEEE international geoscience and remote sensing symposium* (pp. 3923–3926). https://doi.org/10.1109/IGARSS.2019.8898565

Xavier, E. M. A., Ariza-López, F. J., & Ureña-Cámara, M. A. (2016). A survey of measures and methods for matching geospatial vector datasets. *ACM Computing Surveys, 49*(39), 1–34. https://doi.org/10.1145/2963147

Yang, X., Li, X., Ye, Y., Lau, R. Y. K., Zhang, X., & Huang, X. (2019). Road detection and centerline extraction via deep recurrent convolutional neural network u-net. *IEEE Transactions on Geoscience and Remote Sensing, 57*(9), 7209–7220. https://doi.org/10.1109/TGRS.2019.2912301

Zhang, Z., Liu, Q., & Wang, Y. (2018). Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters, 15*(5), 749–753. https://doi.org/10.1109/LGRS.2018.2802944

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