Deep Learning for Enrichment of Vector Spatial Databases: Application to Highway Interchange

GUILLAUME TOUYA and IMRAN LOKHAT, Université of Gustave Eiffel, LASTIG, Univ Gustave Eiffel, ENSG, IGN

Spatial analysis and pattern recognition with vector spatial data is particularly useful to enrich raw data. In road networks, for instance, there are many patterns and structures that are implicit with only road line features, among which highway interchange appeared very complex to recognize with vector-based techniques. The goal is to find the roads that belong to an interchange, such as the slip roads and the highway roads connected to the slip roads. To go further than state-of-the-art vector-based techniques, this article proposes to use raster-based deep learning techniques to recognize highway interchanges. The contribution of this work is to study how to optimally convert vector data into small images suitable for state-of-the-art deep learning models. Image classification with a convolutional neural network (i.e., is there an interchange in this image or not?) and image segmentation with a u-net (i.e., find the pixels that cover the interchange) are experimented and give better results than existing vector-based techniques in this specific use case (99.5% against 74%).

CCS Concepts:
• Computer systems organization → Geographic information systems;
• Computing methodologies → Neural networks;

Additional Key Words and Phrases: Spatial data enrichment, deep neural networks, highway interchange, map generalization

ACM Reference format:
Guillaume Touya and Imran Lokhat. 2020. Deep Learning for Enrichment of Vector Spatial Databases: Application to Highway Interchange. ACM Trans. Spatial Algorithms Syst. 6, 3, Article 21 (April 2020), 21 pages.
https://doi.org/10.1145/3382080

1 INTRODUCTION
Spatial analysis of vector data remains very complex because of the diversity of configurations and the heterogeneity of datasets. Among other applications, map generalization (i.e., the process to simplify map information when scale is reduced) is particularly dependent on vector pattern recognition [14] to make explicit the implicit structures of the map (e.g., buildings aligned along a road). This vector pattern recognition task is very complex, and the existing techniques are not completely satisfying [27]. Pattern recognition in images was recently revolutionized by deep convolutional neural networks (CNNs) and more generally deep learning techniques. Even if CNNs cannot directly work on vector data for now, can they also revolutionize vector spatial analysis?

Deep learning has already been thoroughly used on spatial information but generally on raster spatial information or on 3D point clouds: for style transfer (e.g., a Google Earth image rendered as
a Google Maps image) [10, 17], for image segmentation (i.e., finding buildings, roads, crosswalks, or other features in images) [2, 9, 12], or for image classification [3, 4, 18]. But it is also possible to use deep learning techniques when vector spatial data is concerned, with images generated from representations of the vector data: the machine learning model learns to process the images of the vector data and not the vector data themselves. Applications to the assessment of OpenStreetMap data quality [30], car trajectory analysis [16], or map generalization [23, 29] were recently proposed. This work follows the same principles (i.e., generating images of the vector data) to explore the possibilities offered by deep learning for vector spatial analysis. The workflow is the following: first convert vector data to raster optimally, then process the raster data with deep learning techniques, then finally reinject the results into the vector dataset. We focus on a use case that is particularly difficult to tackle with vector data analysis: automatically identifying the road sections that belong to a highway interchange.

This article is structured as follows. Section 2 describes the use case and the past attempts using vector-based or geometrical spatial analysis. Then, Section 3 presents an image classification model that identifies images containing highway interchanges. Section 4 details an image segmentation model that identifies the pixels of the image containing highway interchanges. Then, Section 5 discusses the optimal generation of training images based on vector spatial data. Finally, Section 6 draws some conclusions and discusses future research.

2 HIGHWAY INTERCHANGE DETECTION IN VECTOR DATASETS

Usually, roads are modeled in geographic datasets with minimal semantics. But they are important features of most maps, not only topographic maps, because of their key role for transportation. This is why road network enrichment by spatial analysis has been an important topic for years in geographic information science [7, 8, 21, 26, 31, 32]. Past research focused on various types of road structures and patterns: continuous road sections or strokes [26], complex crossroads [8, 21], ring roads [7, 32], grid-like patterns [7, 31, 32], and dual carriageways [31], among others.

Among the patterns and structures that are implicit in a network of road sections, highway interchanges are particularly interesting (Figure 1). Highway interchange road sections are the roads that connect a highway with other highways or other simple roads. Highway interchange patterns can be very diverse, and even if most of them can be classified to well-known patterns [5], local modifications of the patterns make them hard to recognize. Why do we need to detect highway interchange roads? The main application is map generalization, because highway interchange can be represented very differently across scales (Figure 2), and a generalization process requires the identification of these patterns prior to their graphical abstraction [15, 27]. The recognition of highway interchange is also useful to enrich data in car navigation applications [5].

To detect the roads belonging to an interchange, road directions can be effectively used as slip roads and highway sections are mainly one-way directed [25]. The shape and the curvature of the road sections can also be used to detect interchanges [21]; however, in this work, the road network is already limited to highway sections plus slip roads thanks to an attribute value in the road dataset.

When there is no semantic information on road direction or road function, two other methods exist in the literature to recognize the road sections that belong to a highway interchange, the second one being a specialization of the first one [15, 27]. The method is based on the classification of each junction between two road sections according to their shape and their connectivity. For instance, y-shaped junctions are connected with three road sections that form angles similar to the shape of letter y. The first step of the detection method is to recognize the y-shaped or fork-shaped junctions. Then, the second step is to cluster the close detected junctions using the distance in the
Fig. 1. Illustration of the use case: finding the road sections that belong to a highway interchange (in red here).

Fig. 2. (a) Highway interchange fully represented with road sections at the 1:10k scale. (b) The same interchange topologically represented at the 1:100k scale. (c) The same interchange represented with a point symbol at the 1:250k scale.
Fig. 3. The initial vector dataset for the use case: detailed roads, with an importance value conveyed by the color in the image (red is the most important). There are also points for a small part of the highway interchanges.

road network as the clustering distance. Next, all roads intersecting the convex hull of the large clusters are considered as belonging to a highway interchange.

In this use case, the road data we work with is produced by the French national mapping agency ($^1$IGN) and covers the whole French territory. In addition to the polylines, the road sections are labeled with an importance value that distinguishes highways from less important roads, but the sliproads that connect the network to the highway have varying importance values, which prevents from using this semantic information in the detection (Figure 3). In this dataset, we can also find an interesting geographic layer with the large complex junctions, which include some highway interchange, being represented with a point geometry (Figure 3). This multiple representation of highway interchange is a key feature to automatically derive training datasets for machine learning [$^2$]. But only approximately half of the French highway interchanges are considered as large junctions, so this information cannot really be used in the vector-based detection method.

Figure 4 shows some results on the use case dataset with the existing vector-based method from Touya [$^2$]. The results are generally unsatisfying. Around 40% of the interchange are correctly delineated, as in Figure 4(a), but most interchange instances are identified but incorrectly delineated, as in Figure 4(b); there is even a significant part of the detected interchange instances that are in fact not highway interchange, as in Figure 4(c). These results confirm that a better method is required, and the paradigm shift brought by CNNs gives that better method, as shown in the following section.

3 DETECTION BY IMAGE CLASSIFICATION
3.1 Classification of Road Network Images
The initial breakthrough of deep learning models was in the domain of image classification, so our first idea was to classify small images of the road network to classify them into two classes:

$^1$IGN: http://www.ign.fr.
Deep Learning for Enrichment of Vector Spatial Databases

Fig. 4. Results of the method from Touya [27]. (a) Interchange correctly delineated. (b) Interchange identified but incorrectly delineated. (c) Interchange identified where there is no interchange. The symbols represent the classified crossroads.

Our problem is reduced to an image classification problem, so we decided to use a deep CNN that proved successful for such problems. We empirically selected a network close to the LeNet-5 network that was proposed for hand-drawn digits recognition [13]. The version of the network we used is described in Figure 5. It can be noted that we added a dropout layer that reduces overfitting, which was particularly important for the no-interchange images that can be more diverse than the ones with interchange, so the training images might not represent the diverse patterns with enough instances. A dropout layer randomly drops out some of the units of the network to prevent co-adaptations during the early steps of training [24]. Table 1 shows the parameters used in the CNN.

3.2 Generating a Training Dataset
As mentioned earlier, the dataset for this use case is composed of road line sections that cover the whole French territory, and of 2,835 points that correspond to the most significant highway

interchange, which means that there is at least one instance of interchange in the image, and no interchange, which means that there is no instance of interchange in the image. Such a modeling of our problem does not provide the road sections that belong to an interchange. But if the image is small enough, it should limit the parts of the network that are processed with the vector-based method and could improve its results. At least it should avoid the false-positive instances like the one in Figure 4(c).

Our problem is reduced to an image classification problem, so we decided to use a deep CNN that proved successful for such problems. We empirically selected a network close to the LeNet-5 network that was proposed for hand-drawn digits recognition [13]. The version of the network we used is described in Figure 5. It can be noted that we added a dropout layer that reduces overfitting, which was particularly important for the no-interchange images that can be more diverse than the ones with interchange, so the training images might not represent the diverse patterns with enough instances. A dropout layer randomly drops out some of the units of the network to prevent co-adaptations during the early steps of training [24]. Table 1 shows the parameters used in the CNN.

3.2 Generating a Training Dataset
As mentioned earlier, the dataset for this use case is composed of road line sections that cover the whole French territory, and of 2,835 points that correspond to the most significant highway

interchange, which means that there is at least one instance of interchange in the image, and no interchange, which means that there is no instance of interchange in the image. Such a modeling of our problem does not provide the road sections that belong to an interchange. But if the image is small enough, it should limit the parts of the network that are processed with the vector-based method and could improve its results. At least it should avoid the false-positive instances like the one in Figure 4(c).

Our problem is reduced to an image classification problem, so we decided to use a deep CNN that proved successful for such problems. We empirically selected a network close to the LeNet-5 network that was proposed for hand-drawn digits recognition [13]. The version of the network we used is described in Figure 5. It can be noted that we added a dropout layer that reduces overfitting, which was particularly important for the no-interchange images that can be more diverse than the ones with interchange, so the training images might not represent the diverse patterns with enough instances. A dropout layer randomly drops out some of the units of the network to prevent co-adaptations during the early steps of training [24]. Table 1 shows the parameters used in the CNN.

3.2 Generating a Training Dataset
As mentioned earlier, the dataset for this use case is composed of road line sections that cover the whole French territory, and of 2,835 points that correspond to the most significant highway

interchange, which means that there is at least one instance of interchange in the image, and no interchange, which means that there is no instance of interchange in the image. Such a modeling of our problem does not provide the road sections that belong to an interchange. But if the image is small enough, it should limit the parts of the network that are processed with the vector-based method and could improve its results. At least it should avoid the false-positive instances like the one in Figure 4(c).

Our problem is reduced to an image classification problem, so we decided to use a deep CNN that proved successful for such problems. We empirically selected a network close to the LeNet-5 network that was proposed for hand-drawn digits recognition [13]. The version of the network we used is described in Figure 5. It can be noted that we added a dropout layer that reduces overfitting, which was particularly important for the no-interchange images that can be more diverse than the ones with interchange, so the training images might not represent the diverse patterns with enough instances. A dropout layer randomly drops out some of the units of the network to prevent co-adaptations during the early steps of training [24]. Table 1 shows the parameters used in the CNN.

3.2 Generating a Training Dataset
As mentioned earlier, the dataset for this use case is composed of road line sections that cover the whole French territory, and of 2,835 points that correspond to the most significant highway
Table 1. Model Architecture of the Proposed CNN

| Layer                | Kernel or Pool Size | Strides |
|---------------------|--------------------|---------|
| Convolution 2D      | 32 × 32            | 3 × 3   |
| Maximum pooling     | 2 × 2              | —       |
| Convolution 2D      | 64 × 64            | 3 × 3   |
| Maximum pooling     | 2 × 2              | —       |
| Convolution 2D      | 128 × 128          | 3 × 3   |
| Maximum pooling     | 2 × 2              | —       |
| Convolution 2D      | 128 × 128          | 3 × 3   |
| Maximum pooling     | 2 × 2              | —       |
| Flatten             | —                  | —       |
| Fully connected     | 64                 | —       |
| Fully connected     | 64                 | —       |
| Dropout             | 0.5                | —       |
| Fully connected     | 1                  | —       |

H and W are the height and the width of the input image.

Fig. 6. Four sample images containing a highway interchange, used to train the model to recognize images.

interchange of the country. We also have access to a complete topographic dataset, and we will see later that it will be useful to automatically generate the training dataset for our model.

The first step is to generate the instances of the first class (i.e., the interchange images). We use the 2,835 points to generate the same number of images of the interchange class. We generate images with 256 × 256 pixels with a scale of 1.2 × 1.2 km per image (Figure 6). These two values for image size and scale gives a good compromise between images that should be big enough to contain all interchange roads but small enough to make the vector-based detection effective.

The images are centered on the interchange point and then randomly slightly deviated from the center, as the images that the model will predict after training might contain interchanges near their border. We decided to generate black and white images with the background in black and the roads in white, with a 1 pixel width as the road symbol. All of these choices to generate the images are discussed in Section 5.

Then, the second step is to generate the training examples of the second class, no interchange. We need to generate images that do not contain any interchange. Rather than picking some random points in our road network, a more controlled process was used. First, tunnel points available in the dataset were extracted; tunnels in France are mostly located in mountainous areas where there are very few highways. Only the tunnel points that do not have an interchange point around were kept. Images were generated with the same process as with interchange points (Figure 7). The number of these points was not big enough, and the road network around tunnels, mostly located in mountainous areas, did not represent well the diversity of network structures. Then,
to obtain more images of urban networks, we selected school points, as they are abundant and mostly located in urban areas. Once again, only the school points that do not have an interchange point around were used (Figure 8). Using these points, 3,410 images were generated for the class no interchange.

In terms of implementation, scripts using the Mapnik library\(^2\) were developed to interact with the data stored in a PostGIS database. Mapnik is a library to generate tiled raster maps from geographic information and can be easily hacked to generate images for deep learning.

To process the vector data that is classified in the interchange class, keeping track of the original vector data, related to a given image, is necessary. Several options are possible: generating a geotiff image (i.e., a geolocated image) or keeping track of the coordinates of the extent of each image in a separated file. We chose a third option and generated a geojson file containing the vector data related to each training image, using the same Mapnik library. The created training dataset is made openly available, as well as the scripts and the model, in the DeepMapGen platform (https://github.com/umrlastig/DeepMapGen).

3.3 Results

To test the model and the training dataset, we separated the images of each class randomly, keeping a little more than 1,500 images in each class for the training phase, 500 in each class for testing during the training phase, and another 500 images in each class for a further assessment of the trained model. The presented results in this section are all from these last 500 images of interchange and 500 without interchange. The model was implemented with the Keras Python library, with a TensorFlow core. We performed the random split of our dataset several times, and the results were extremely similar with different images in the training, validation, and test samples.

---

\(^2\)https://mapnik.org.
Table 2. Confusion Matrix of the Best Classification Result Obtained with the Proposed Model Trained with Our Training Examples

| Classification                                 | Label          | Interchange | No Interchange |
|------------------------------------------------|----------------|-------------|----------------|
| Percentage of interchange classification       |                | 99.8%       | 0.8%           |
| Percentage of no-interchange classification    |                | 0.2%        | 99.2%          |
| Number of images classified as interchange     |                | 499         | 4              |
| Number of images classified as no interchange  |                | 1           | 496            |

Table 3. Confusion Matrix of the Classification Results Obtained with the Vector Baseline

| Classification                                 | Label          | Interchange | No Interchange |
|------------------------------------------------|----------------|-------------|----------------|
| Percentage of interchange classification       |                | 81%         | 33.8%          |
| Percentage of no-interchange classification    |                | 9%          | 66.2%          |
| Number of images classified as interchange     |                | 405         | 169            |
| Number of images classified as no interchange  |                | 95          | 331            |

The best results obtained with this dataset were ceiling around 96% of overall accuracy on the test data, so we added some data augmentation with random rotation of the input images, which yielded better results with 99.6% overall accuracy, with a loss of 0.036, on the test data. The detailed results presented in Table 2 show that images containing interchange instances are correctly classified with an even higher rate of 99.8%, which means that images without interchange are classified as false positive 0.8% of the time. This result is interesting because the aim of this classification model was not to provide an automated result for interchange detection, but just an optimized subset of the road network to improve the vector-based method, and this is what is achieved with this model.

The best results were obtained with a batch size of 64 and 100 epochs. To select these optimal values, experiments were carried out with batch sizes ranging from 16 to 128 and epochs ranging from 20 to 500.

To assess how good this model performed, there was no existing baseline, so we defined two ourselves. The first one is based on the vector-based method: the vector-based interchange detection is triggered on each vector extract corresponding to an image of the validation dataset, and the extract is classified as interchange when there is at least one interchange instance detected and no interchange when there is none. The results of this baseline are presented in Table 3. The CNN classifier is clearly better than the vector baseline, and the difference is even bigger, as expected, on the images that do not contain any interchange.

The second baseline is also analyzing the vector data and not its image, and uses a random forests classifier. We derived a set of descriptors of the vector road sections in each extract of the dataset. We used the following descriptors in our baseline:

- the number of intersections (as interchanges contain a high density of intersections);
- the total length of roads in the extract (to describe the global density of the network);
- the mean and the standard deviation of the road section lengths;
- the mean and standard deviation of the road section sinuosity (the proxy for sinuosity used here is the distance between the extreme points of the polyline divided by the length of the polyline); and

ACM Transactions on Spatial Algorithms and Systems, Vol. 6, No. 3, Article 21. Publication date: April 2020.
Table 4. Confusion Matrix of the Classification Results Obtained with the Random Forests Baseline

| Classification                           | Interchange | No Interchange |
|------------------------------------------|-------------|----------------|
| Percentage of interchange classification | 87.4%       | 10.8%          |
| Percentage of no-interchange classification | 12.6%       | 89.2%          |
| Number of images classified as *interchange* | 437         | 54             |
| Number of images classified as *no interchange* | 63          | 446            |

Fig. 9. Four sample heat maps that visually illustrate what the last layer of the CNN "sees," based on the Grad-CAM algorithm [22].

- the total length of the road sections with maximum importance (as highways are mainly labeled with a maximum importance).

The random forests classifier was trained with the same train and test datasets and then assessed on the same validation dataset. This second baseline gives an accuracy of 88.3% on the 500 extracts of the validation dataset, and the confusion matrix is presented in Table 4. The results of the second baseline are already way better than the first one, but once again our proposed CNN model clearly outperforms this baseline. We believe that the ability of the CNN to detect the cluttered areas of the image explain the performance difference, as the descriptors used in this baseline are not really able to convey this local clutter caused by the rendering of interchange road sections.

We also used the Grad-CAM algorithm [22] to visually assess what was learned by the CNN. Figure 9 shows heat maps that correspond to what “sees” the last layer of the CNN, and it clearly learns to highlight the locations of the interchange, which confirms the good classification results.

3.4 Detecting Interchange Roads in Predicted Images

Now that the classification gives a subset of the road network that is very likely to contain highway interchange instances (i.e., the roads contained in an image classified as *interchange*), we need to check that restricting the vector-based method to this subset of roads improves the vector-based method results. We processed all extracts that were related to an image classified as *interchange* and analyzed the results. Although there is a major improvement compared to the initial processing of the complete road network, the results are still unsatisfying (Figure 10).

It is pretty clear that both classification results and vector-based detection could be improved by adapting the method to process small extracts of the network that do contain an interchange instance. We believe that the way to optimize our results is to both filter even more the roads to process in the vector method and to adapt the vector method to these filtered roads. In this work, we decided to first focus on filtering even more the roads on which to apply the vector-based
Fig. 10. Vector-based detection algorithm used on the roads extracted from an image classified as a highway interchange: the interchange is detected but with two different parts, the middle sections being left undetected.

Fig. 11. Architecture of the U-Net used for the segmentation of interchange images. The arrows show how the concatenate layers are connected to previous layers of the model.

method. We propose to use an image segmentation model, and this segmentation method is presented in the following section.

4 DETECTION BY IMAGESEGMENTATION

4.1 Segmentation of Road Images

Image segmentation by deep learning has been a very active field in the past 10 years, and models are now able to delineate features in photographs, scientific images, or videos. The images generated from the road network are not close to photographs that attract most of the attention of the researchers but are closer to the scientific images processed in the works introducing the U-Net architecture [20]. U-Nets were already used with images generated from vector spatial data [16, 23]. U-Nets provide a classification probability for each pixel of an input image and are based on a sequence of down convolutions, as in CNNs, and then up convolutions (Figure 11). The compactness of the features segmented by the model is assured by so-called concatenate layers that are connected to the neurons of down convolution layers.

We also briefly tried other network architectures dedicated to segmentation problems [1, 19], but there was no clear improvement compared to our U-Net, so we decided to limit the investigations...
Fig. 12. Four examples used to train the segmentation model. In the input images on top, the colors are related to the importance value of each section (red is most important, and black is least important). In the label images below, a black square is generated around the interchange points.

with these networks. Future work is required to know if a finer architecture can improve our results.

4.2 Generating a Training Dataset
In this case, a training example consists of an input image of the road network and a label image showing the pixels of the input image that belong to an interchange (Figure 12). We first used the same images as the ones in the classification model, but the results were not optimal so we changed the color model only, switching from black and white images to RGB images, with a white background and roads with colors corresponding to their importance value in the dataset. As highway interchanges are usually around important roads, it helped improve the segmentation results a lot.

Regarding the label images, they are black and white images, where a white pixel means that the pixel does not belong to an interchange, and a black one means that the pixel belongs to an interchange. The label images were generated from the interchange points used to train the classification model, and a black square was generated around each point. The extent of an interchange can vary a lot from small to very complex interchanges, but to keep the dataset generation automatic, we had to choose a fixed size for the black square. The size of the square is kept rather small to make sure it does not cover areas where there is no interchange. The drawback is that it does not completely cover all of the roads of the interchange. Label images that do not contain any interchange are left totally white.

4.3 Results
The experiment setup was the same as the one for image classification: similarly to the classification experiment, we separated the images of each class randomly into train, test, and validation datasets, and the U-Net model was also implemented with the Keras Python library, with a TensorFlow core.
The best results obtained after 30 epochs are the following: an accuracy of 97.8% and a loss value of 0.0696 on the training data, and an accuracy of 94.3% and a loss value of 0.3773 on the validation data.

Figure 13 shows some good results on four of the interchange images of the test data. The pixels with the higher probabilities (with darker shades of grey in the image) are clearly the ones that intersect the roads of the interchange. Even the very small interchange on the right of the image is correctly segmented. And even when the highway is not colored in red because the importance value is not the one usually applied to such roads (maybe an error in the data), the interchange is segmented (second image from the left). Similar good results appear on a large majority of the tested samples: sometimes the pixels with a high probability cover a little more than the interchange, and sometimes they cover a little less, but as the raster-based segmentation is just a first step to filter the roads to include into the vector-based method, the results really correspond to what was expected. Figure 14 shows four other examples that confirm the good results even with very complex or unusual structures. The image from the right shows the correct segmentation of two different interchanges in the same image. And the third image shows that there is nothing segmented when there is no interchange in the image.

However, both figures show that the segmented area is always mostly a square due to the shape we gave to our training masks. In the examples of Figures 13 and 14, the square grossly captures the interchange location, but in some rare cases the delineation is visually not correct. We compared the segmentation results with interchange delineations performed manually in Figure 15. Sometimes the segmented square is not located on the interchange (in the left image, the model segmented the roundabout instead of the small interchange at the bottom left). And sometimes the square is off-center (in the other three images).

5 DERIVING TRAINING IMAGES FROM VECTOR SPATIAL DATASETS

The previous two sections presented two deep learning models to detect highway interchanges in road networks that we trained with images derived from vector data. The way images are generated was determined empirically, and it might be sub-optimal. There are different variables on image generation: the scale and the resolution of the image, the style of the rasterized vector data,
data selection, and the way the label area is created in the segmentation use case. In this section, we discuss different alternatives for these variables to generate the training images from vector data and how they perform in our use cases.

5.1 Scale and Image Resolution

The first two variables to set when generating images for the vector data are the scale and the resolution of the image. The scale is simply the ratio between the width (or height) of the image and the length of the same geographic extent on the ground. The image resolution is the ratio between the width (or height) of the image and the number of pixels.
Table 5. Classification Results with Training Images of Different Sizes for a Same Geographic Extent

| Resolution | Accuracy | Loss | Interchange Ratio | No-Interchange Ratio |
|------------|----------|------|-------------------|---------------------|
| 128 × 128  | 98.6%    | 0.098| 496/500           | 491/500             |
| 256 × 256  | 99.8%    | 0.036| 499/500           | 496/500             |
| 512 × 512  | 95.6%    | 0.328| 486/500           | 466/500             |

To assess the importance of those two variables on the effectiveness of the images, we tested two types of variations:

- *Scale variations:* Same image size and resolutions but different ground extents.
- *Resolution variations:* At constant scale, different image resolutions by changing the number of pixels (128 × 128, 256 × 256, 512 × 512).

Table 5 shows the results obtained with resolution variations with the image classification model for a same scale (images cover 1.2 × 1.2 km). The resolution we used (256 × 256 pixels) gives the best results. A smaller resolution gives worse results, but they are still very good. On the contrary, a larger resolution gives much worse results, which suggests that images should not be larger than 256 × 256 pixels. A smaller resolution than 128 × 128 pixels for such a scale was not tested, but we believe that it would give bad results, as many highway interchanges would be covered by very few pixels, making them hard to recognize in the image.

The scale variation was also tested, but the range for scale variation in our use case is not that big. Hence, a highway interchange can be quite large, and a too large scale might cause incomplete interchanges in the images, which is a problem. However, too small scales might cause too much coalescence between the road symbols, except if the resolution is high enough (but we have seen just before that a high resolution gives bad results). This is why only slight variations of the scale were tested: Figure 16 shows images that are 1.2 km wide and images that are 0.8 km wide. The results with these two scales are extremely similar, so we conclude that if scale remains in a reasonable range, this variable has a low impact on the effectiveness of the deep learning models.

5.2 Styling Vector Data

Besides scale and resolution, another key variable is the way vector data is styled prior to rasterization. In the experiments described in the previous sections, we tested two different styles:

- a black background and white symbols for roads (1 pixel width), and
- a white background and roads colored according to importance (from black to red, with 1 pixel width).

The first style has the advantage of simplicity, as the color information is coded with only two values (0 or 1), whereas a colored image is encoded with a triplet of values between 0 and 256. The drawback of this simple style is that it does not use the semantic information about road importance, whereas highways are coded as important roads. It was enough to get great classification results but not enough for the segmentation model.

The second style makes a complete use of the convolutional layers, as the model learns that interchange roads often lie around roads colored in red (the color of the most important roads). However, we have seen in the results that not all interchange instances contain roads coded as important (Figure 14), so we have to make sure that colors do not introduce a learning bias.

Another way to modify style is to change the width of the symbols applied to roads rather than the colors. Figure 17 shows an example of an interchange represented with a symbol width of 1 pixel on the left and 2 pixels on the right. It is visually clear that a symbol width of 2 pixels introduces symbol coalescence, and this coalescence might blur the location of interchanges.
Fig. 16. Two series of images with a different extent for the same image size: the extent is larger in the top images than in the bottom images.

Fig. 17. Two images generated with a road symbol width of 1 pixel on the left and 2 pixels on the right. Test on the classification model with 2 pixel symbols confirmed our doubts, with much lower classification results. However, this quick test does not dismiss the usefulness of using larger widths for other use cases where a larger scale is possible.

5.3 Data Selection

The last variable for the generation of input images is selection of the features of the dataset to render in the image. First, if we only focus on the roads to display in the image, there are different possibilities:

- display only the \( n \) closest roads to the interchange point (this is the option chosen in our classification experiment with \( n = 200 \)),
- display all of the roads that fall into the envelope of the image, or
- display only the most important roads.
Figure 18 shows images where we selected only the closest roads with a different number of roads. The threshold of 200 roads was determined empirically, as the classification model trained with these images gave the best classification results. It can be explained by a balance between a good description of the network structure and the complexity of the image.

We did not test the last option of data selection (i.e., selecting only the important roads). Indeed, a simple filter on the importance attribute was often filtering too many roads, even when only removing the least important roads (the black ones in the colored examples presented in this article). Road network generalization provides methods to select the most important and salient roads, going beyond a basic semantic filter [27]. It would be very interesting to test the training of the segmentation models with images where the road network was generalized to get rid of the minor roads of the network.

If we broaden the scope of data selection, it is possible to display other types of geographic features in the image. For instance, buildings are rarely built inside interchanges, so we could display the buildings in the image. We did not carry out any experiment with buildings for time reasons, but it would be interesting to verify if it can improve the segmentation results, as the classification results are already very good.

5.4 Segmentation Labels
Regarding the segmentation problem, we tried different types of outputs to learn, all derived from the interchange point in our initial dataset. Figure 19 shows some of the labels we tested before selecting the one presented in Section 4.2. All of the methods that use the roads in the label were the least effective compared to the more abstract ones.
We also compared the automatically created labels to labels segmented manually on a very small number of examples (Figure 20). On this small number of examples, the manually segmented labels allowed much better results than our best automatic method (a small black square on a white background). Although such results cannot be generalized without experiments, we believe that using a large amount of images that were manually segmented, would improve the results of the segmentation model. But creating such a large manual training dataset is costly.

5.5 Data Augmentation

This last section is not strictly about variables to generate training images but is about the possibilities to augment the dataset to have enough training examples. Contrary to photographs, it is
very easy to augment a dataset derived from vector by simple translations and rotations during
the rasterization. Figure 21 shows an illustration with three images generated for the same inter-
change point thanks to three different slight translations of the image center from the interchange
point.

6 CONCLUSION AND FUTURE WORK

The experiments presented in this article show that deep learning greatly improves the detection of
highway interchanges in vector road data. First, a simple image classification CNN model greatly
reduces the number of false-positive interchange detection when coupled with the vector-based
detection method. Then, the U-Net segmentation model clearly improves the delineation of the
interchange roads, even with an automatically generated training dataset. Beyond the progress on
this specific issue, the main contribution of this research is the exploration of different methods to
generate images adapted to deep learning models from vector datasets. It opens up new perspec-
tives on the use of deep convolutional networks for pattern recognition with vector spatial data.

There is a lot to do to go further. Regarding the use case on highway interchange, a first step
would be to improve the training datasets used for the segmentation model by data augmentation.
Another way to improve the training dataset is to use the vector-based method to find interesting
areas in the network: it could add new interchange instances that do not have an interchange point
in our dataset, and also add false-positive cases (i.e., areas recognized as interchange by the vector-
based method but are not), because it would help the model learn how to process such cases. An-
other way to improve the training dataset is to add new instances. We already used all instances
in the French national mapping agency dataset, but we can use the worldwide OpenStreetMap
dataset, where the features with the tag value “motorway_junction” for the tag “highway” (Fig-
ure 22). There are different features than the ones used in this article, but images centered on
such features would contain the interchange roads. By the time of this writing, there were 171,784
instances\(^3\) of motorway junctions that could be used to generate new examples.

We already mentioned it several times in the article, but another obvious future work is to
improve the vector-based method to obtain optimal results once the roads have been properly
filtered by the deep learning models. Our idea is to combine our existing method with the one
using road shapes and curvature [21] that also process filtered roads, a semantic filter in their
case.

\(^3\)https://wiki.openstreetmap.org/wiki/Tag:highway%3Dmotorway_junction.
More generally, we can improve the segmentation model by going deeper in the experiments on the model architecture. First, it could be interesting to test residual U-Nets [33] as an improvement of the segmentation model: residual U-Net outperformed traditional U-Nets on different binary segmentation problems. Regarding the loss function, we used a basic binary cross-entropy function, and we plan to test more functions, such as the one proposed for the seminal U-Net [20] or overlap measures. To segment geometrically realistic delineations of the interchanges, it could also be interesting to test generative adversarial networks that perform well to generate realistic map images [10, 11]. We also plan to test the model on other regions of the world that might present other types of interchange patterns. In addition to these tests on other regions, we want to try the model on other datasets, such as OpenStreetMap, to check if there are some transfer learning issues.

A straightforward follow-up to this work would be to use deep learning techniques to recognize other types of patterns in road structures, such as complex crossroads, ring roads, grid-like patterns, or dual carriageways, as existing vector-based methods [7, 8, 21, 31, 32] can really be improved.

A similar approach could also be used for other types of vector analysis problems. For instance, urban block classification is quite useful for map generalization purposes, but the current approaches based on classical supervised learning techniques fall quite short because the optimal descriptors to classify a building block are not easy to identify. Another interesting example is the inference of landmarks from a set of polygon buildings, i.e. finding the ones that are visually salient during a navigation or a map reading task [6, 28]; once again, classical machine learning techniques give perfectible results because it is extremely complex to define the descriptors of what makes a landmark for the human brain.

Finally, one last track for future research would be to generalize the experiments and discussions of Section 5.1: more experiments could help us define some general guidelines on the variables for image generation (scale, resolution, styling, data selection).
REFERENCES

[1] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. 2017. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence 40, 4 (2017), 834–848. http://arxiv.org/abs/1606.00915.

[2] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. 2017. Joint learning from earth observation and Open-StreetMap data to get faster better semantic maps. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW’17). IEEE, Los Alamitos, CA, 1552–1560. DOI: https://doi.org/10.1109/CVPRW.2017.199

[3] R. F. Berriel, A. T. Lopes, A. F. de Souza, and T. Oliveira-Santos. 2017. Deep learning-based large-scale automatic satellite crosswalk classification. IEEE Geoscience and Remote Sensing Letters 14, 9 (Sept. 2017), 1513–1517. DOI: https://doi.org/10.1109/LGRS.2017.2719863

[4] Jiaoyan Chen and Alexander Zipf. 2017. DeepVGI: Deep learning with volunteered geographic information. In Proceedings of the 26th International Conference on World Wide Web Companion (WWW Companion ’17). ACM, New York, NY, 771–772. DOI: https://doi.org/10.1145/3041021.3054250

[5] Ahmet Ozgur Dogru, Nico Van de Weghe, Necla Ulugtekin, and Philippe De Maeyer. 2007. Classification of road junctions based on multiple representations: Adding value by introducing algorithmic and cartographic approaches. In Proceedings of the International Cartographic Conference.

[6] Birgit Elias. 2003. Extracting landmarks with data mining methods. In Spatial Information Theory: Foundations of Geographic Information Science. Lecture Notes in Computer Science, Vol. 2825. Springer, 375–389. DOI: https://doi.org/10.1007/978-3-540-39923-0_25

[7] Frauke Heinzle and Karl-Heinrich Anders. 2007. Characterising space via pattern recognition techniques: Identifying patterns in road networks. In The Generalisation of Geographic Information: Models and Applications, W. A. Mackaness, A. Ruas, and T. Särkijäskoki (Eds.). Elsevier, 233–253.

[8] Frauke Heinzle and Karl-Heinrich Anders. 2005. Graph based approaches for recognition of patterns and implicit information in road networks. In Proceedings of the International Cartographic Conference (ICA’05).

[9] Z. Huang, G. Cheng, H. Wang, H. Li, L. Shi, and C. Pan. 2016. Building extraction from multi-source remote sensing images via deep deconvolution neural networks. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS’16). 1835–1838. DOI: https://doi.org/10.1109/IGARSS.2016.7729471

[10] Philipp Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR’17). http://arxiv.org/abs/1611.07004

[11] Yuhao Kang, Song Gao, and Robert E. Roth. 2019. Transferring multiscale map styles using generative adversarial networks. International Journal of Cartography 5, 2–3 (2019), 115–141. DOI: https://doi.org/10.1080/23729333.2019.1615729

[12] Loïc Landrieu and Martin Simonovsky. 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’18).

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 11 (Nov. 1998), 2278–2324. DOI: https://doi.org/10.1109/5.726791

[14] William A. Mackaness and Geoffrey Edwards. 2002. The importance of modelling pattern and structure in automated map generalisation. In Proceedings of the Joint ISPRS/ICA Workshop on Multi-Scale Representations of Spatial Data. 7–8. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.91.1311

[15] William A. Mackaness and Gordon A. Mackenzie. 1999. Automating the detection and simplification of junctions in road networks. Geoinformatica 3, 2 (1999), 185–200. DOI: https://doi.org/10.1023/a:1009807927991

[16] Y. Meñéroux, V. Dizier, M. Margolle, M. D. Van Damme, Y. Kato, A. Le Guilcher, and Guillaume Saint-Pierre. 2018. Convolutional neural network for road sign inference based on GPS traces. In Spatial Big Data and Machine Learning in GIScienceWorkshop at GIScience 2018, Melbourne. Leibniz International Proceedings in Informatics. Schloss Dagstuhl, Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany, 4.

[17] Torben Peters and Claus Brenner. 2018. Conditional adversarial networks for multimodal photo-realistic point cloud rendering. In Spatial Big Data and Machine Learning in GIScience, Workshop at GIScience 2018, Melbourne. Leibniz International Proceedings in Informatics. Schloss Dagstuhl, Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany, 48–53. http://spatialbigdata.ethz.ch/wp-content/uploads/2018/07/12_Peters_Conditional-Adversarial-Networks-for-Multimodal-Photo-Realistic-Point-Cloud-Renderring-revised.pdf.

[18] Tristan Postadjian, Arnaud Le Bris, Hichem Sahbi, and Clément Mallet. 2018. Domain adaptation for large scale classification of very high resolution satellite images with deep convolutional neural networks. In Proceedings of the 2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS’18). 3623–3626. DOI: https://doi.org/10.1109/IGARSS.2018.8518799
Deep Learning for Enrichment of Vector Spatial Databases

[19] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR’16). arxiv:1506.02640 http://arxiv.org/abs/1506.02640

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. arXiv:1505.04597.

[21] Sandro Savino, M. Rumor, M. Zanon, and I. Lissandron. 2010. Data enrichment for road generalization through analysis of morphology in the CARGEN project. In Proceedings of 13th ICA Workshop on Generalisation and Multiple Representation.

[22] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2016. Grad-CAM: Visual explanations from deep networks via gradient-based localization. arXiv:1610.02391.

[23] M. Sester, Y. Feng, and F. Thiemann. 2018. Building generalization using deep learning. In ISPRS—International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-4. Leibniz Universität Hannover, 565–572. DOI: https://doi.org/10.5194/isprs-archives-XLII-4-565-2018

[24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html.

[25] Stuart Thom. 2005. A strategy for collapsing OS integrated transport network dual carriageways. In Proceedings of the 8th ICA Workshop on Generalisation and Multiple Representation. http://aci.ign.fr/Acoruna/Papers/Thom.pdf.

[26] Robert C. Thomson. 2006. The ‘stroke’ concept in geographic network generalization and analysis. In Progress in Spatial Data Handling, A. Riedl, W. Kainz, and G. A. Elmes (Eds.). Springer, 681–697.

[27] Guillaume Touya. 2010. A road network selection process based on data enrichment and structure detection. Transactions in GIS 14, 5 (2010), 595–614. DOI: https://doi.org/10.1111/j.1467-9671.2010.01215.x

[28] Guillaume Touya and Marion Dumont. 2017. Progressive block graying and landmarks enhancing as intermediate representations between buildings and urban areas. In Proceedings of 20th ICA Workshop on Generalisation and Multiple Representation.

[29] Guillaume Touya, Xiang Zhang, and Imran Lokhat. 2019. Is deep learning the new agent for map generalization? International Journal of Cartography 5, 2–3 (2019), 142–157.

[30] Yongyang Xu, Zhanlong Chen, Zhong Xie, and Liang Wu. 2017. Quality assessment of building footprint data using a deep autoencoder network. International Journal of Geographical Information Science 31, 10 (Oct. 2017), 1929–1951. DOI: https://doi.org/10.1080/13658816.2017.1341632

[31] Bisheng Yang, Xuechen Luan, and Qingquan Li. 2010. An adaptive method for identifying the spatial patterns in road networks. Computers, Environment and Urban Systems 34, 1 (Jan. 2010), 40–48. DOI: https://doi.org/10.1016/j.compenvurbsys.2009.10.002

[32] Q. Zhang. 2004. Modelling structure and patterns in road network generalization. In Proceedings of the ICA Workshop on Generalisation and Multiple Representation. http://aci.ign.fr/Leicester/paper/Zhang-v2-ICAWorkshop.pdf.

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

Received May 2019; revised August 2019; accepted February 2020