Unpaired Learning of Roadway-Level Traffic Paths from Trajectories

Weixing Jia[1,2], Guiling Wang[1,2(✉)], Xuankai Yang[1,2], and Fengquan Zhang[1,2]

[1] School of Information Science and Technology, North China University of Technology, Beijing 100144, China
wangguiling@ncut.edu.cn

[2] Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, North China University of Technology, Beijing 100144, China

Abstract. Traffic path data can be used as the basis for traffic monitoring and other technologies, which is essential for developing traffic-related technologies. Traditional methods of traffic path data extraction can no longer meet the needs because they cannot solve the problem of lacking standard benchmark data that may exist in the traffic field. Deep learning-based path extraction methods using large-scale data are a class of promising approaches. However, most of the deep learning-based path extraction methods are supervised and rely on paired training data. This paper proposes an unpaired learning method for fine-grained roadway-level paths from trajectory data based on CycleGAN. The method constructs spatio-temporal features based on HSV color space from trajectories which can enhance the model's ability to recognize the roadway details. It transforms the features using convolutional layers, which can preserve the spatio-temporal information of the features, thus making the extraction results more accurate. We conduct experiments using urban and maritime traffic trajectory data and compare the proposed method with the state-of-the-art methods. The results of our model have more roadway level details, higher precision and F1 score than the other existing unsupervised traffic path learning methods.

Keywords: Deep learning · Unpaired learning · Trajectory mining · Roadway-level path extraction

1 Introduction

Traffic paths are the carriers of transportation and traffic, and traffic paths data can be used as the basis for traffic monitoring and other technologies. Therefore, traffic path extraction is essential for the research in the field of transportation. A traffic path is the route of the traffic facility that a moving object follows

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 406, pp. 171–190, 2021.
https://doi.org/10.1007/978-3-030-92635-9_11

when traveling or the regular path of many moving objects. The fineness level of the traffic path extraction can be divided into three categories: center line level, roadway level and lane level. The traffic paths of center line level are represented by their center lines; the traffic paths of roadway level are represented by the center lines of their roadways, i.e., the sub-paths of different driving directions; the traffic paths of lane level are represented by their lanes. These three traffic path extraction levels have an increasing level of refinement. The more fine-grained of the traffic path extraction, the more traffic monitoring accuracy we can guarantee. Traditional traffic path extraction methods mainly include manual measurement and extraction methods based on high-resolution remote sensing image processing [1–4]. However, not all traffic paths in traffic-related domains are constructed manually, e.g., maritime traffic paths are the optimal navigation paths inducted from the sailing habits of ships and vessels. Therefore, in order to achieve traffic path extraction in traffic fields, the traditional manual measurement methods and remote sensing image processing methods are no longer applicable.

As presented in Sect. 2, many methods have been proposed to extract traffic paths on land or at sea. Among these methods, the most promising is the deep learning-based approaches because they do not rely on many empirical parameters and make fair use of the prior knowledge in the available map data [5–7]. With the development of computer technology, global positioning technology, GPS, and modern communication technology, a large amount of GPS data collected from moving objects provides an opportunity to extract the traffic paths. In maritime traffic, it is also possible to obtain and collect massive amounts of ship trajectory data with the help of the movement status and spatial position tracking capabilities of maritime transport ships. The Automatic Ship Identification System (AIS) is the successful application of these technologies and makes it possible to extract maritime traffic paths. The large amount of spatio-temporal data generated by the AIS system provides essential raw data for deep learning models to generate maritime traffic paths.

However, most of the deep learning-based methods for traffic path extraction are supervised, which require high-quality paired trajectories and traffic path data for model training. These methods cannot be implemented in many real-world scenarios such as maritime traffic path extraction, which lacks real paths corresponding to the trajectory data for training. In our previous study, we proposed T2I-CycleGAN [5], an unsupervised traffic path learning method based on CycleGAN [8]. To the best of our knowledge, it is the only existing unsupervised traffic path learning method from unpaired data so far to fit in maritime traffic scenarios. But there's a fly in the ointment: it can extract maritime paths of center line level and only skeletons of the traffic paths will be extracted.

In this paper we aim to go further and extract traffic paths of roadway level. The key to extract roadway level paths is constructing features for model training to effectively distinguish different roadways. The trajectory data of a moving object contains spatio-temporal attributes related to its navigation status, such

as latitude and longitude coordinates, velocity and direction. If these spatio-temporal attributes can be used to construct features for different roadways, the model's ability to distinguish different roadways can be improved, and the extraction of fine-grained paths can be realized.

The key contributions of this paper can be summarized as follows.

1) In this paper, we propose an unsupervised learning method based on CycleGAN for extracting fine-grained paths of roadway level from unpaired data. The cycle consistency of CycleGAN achieves traffic path extraction without paired training data. We construct the point and line features of the trajectory data, use the HSV color space [9] to convert the directions, speed into the color information of the trajectory, and then merge the features in the model. From the results, we can see that our feature construction method and our proposed model can achieve roadway-level path extraction.

2) In the proposed method, we add a convolutional layer to the input part of CycleGAN to get spatio-temporal information of the input trajectory features. Compared with the dense layer added in our previous research [5], the convolutional layer can target the spatially closely connected localities in the trajectory features for extraction, which can be more effective for model training.

The rest of this paper is organized as follows. Section 2 summarizes and analyzes the related work. Section 3 introduces two important preliminary concepts. In Sect. 4, an overview of the approach is presented. Section 5 presents the features to be extracted and the structural matrix for converting the extracted orientation, speed information into RGB colors. In Sect. 6, we describe the selection and modification of the network structure in the CycleGAN-based model. Section 7 gives the parameter details and evaluates the extraction method of this paper and compares it with other methods and Sect. 8 concludes the paper.

2 Related Work

Traffic path extraction methods can be divided into two categories: image-based method and trajectory-based method, according to the different types of data used. The former uses remote sensing images or gray-scale imagery to extract the traffic paths, as in [10]. This type of methods require a lot of computing resources to process image data in the early stages of the extraction process. The latter generates traffic paths based on trajectory data collected by sensors or mobile devices. In recent years, different kinds of trajectory-based methods have been proposed, which can be classified into two main categories: method based on trajectory point clustering and method based on density estimation. There are also some hybrid methods combining more than one strategy.

1) **Method based on trajectory point clustering.** This method takes trajectory points as input, uses clustering methods such as k-means to obtain traffic path segments that are finally connected to form a complete path [11–17]. Ma Wenyao [11] proposed a trajectory similarity measure based on one-way distance. Spectral clustering was used to cluster and analyze the trajectories to obtain the behavior patterns of vessels. Junwei Wu [12] clustered the converging points in the traffic road network and got the intersection positions to solve

the problem of low-quality GPS point extraction. Rade Stanojevic [14] proposed an offline algorithm that clustered GPS points for graph construction and an online algorithm that can create and update the road network. Xingzhe Xie [15] proposed to detect the intersections by finding the common sub-tracks of the GPS traces. Alireza Fathi [16] introduced an intersection detector that uses a localized shape descriptor to represent the distribution of GPS traces around a point. Jing Wang [17] used circular boundaries to separate all GPS traces into road intersections and road segments and then built the road paths. These algorithms improved the tolerance of noise and isolated points in the data. However, because clustering algorithms such as DBSCAN do not consider the overall shape characteristics of the trajectory and divide the trajectory in the clustering process, it is difficult to extract the whole trajectory effectively. And when the density of data distribution varies greatly, the results of point clustering methods tend to be poor.

2) **Method based on geometric morphometrics.** This category of method partitions the region into a set of grids, computes or estimates the density of the trajectory points, converts them into grid densities, classifies the grids into two types (those grids on road areas and those are not) and then constructs the cent lines or road boundaries using geometric morphometrics methods. Wang [18], Ahmed [19,20], Mariescu [21] and Kuntzsch [22] transformed the trajectory data into grid data according to the density. Morphological methods were used to extract grid data, and then vectorization was performed to obtain vector road network data. Tang Luliang [23], Yang Wei [24,25], Tang [26] established a constrained triangulation network to extract the road network through trajectory lines. In the extraction process, because the boundary of the obtained surface data was not smooth, the extracted road had weak folds and excessive redundant data was generated. Li [27] improved the Delaunay triangulation based extraction method by grid merging and sliding local window filtering before triangulation. Among them, the treatment of trajectory density difference has a very big impact on the extraction effect.

The above methods require many empirical parameters and are vulnerable to noisy data. Compared with these methods, the deep learning based road extraction approaches are promising. Jiang Xin et al. [28] extracted roads from high-resolution remote sensing images using the DenseUNet network Ruan et al. [6] proposed DeepMG to extract features from trajectories in both spatial view and transition view and used a convolutional deep neural network to learn road centerlines. Chuanwei Lu et al. [7] used cGAN deep learning network for automatic extraction of road network. This method can extract two-way lanes on the road network. However, it relies on paired labelled training data. To the best of our knowledge, only [5] proposed unsupervised road network extraction methods not relying on paired labelled training data. But it is an extraction method of center line level, there are still some shortcomings that need to be addressed. For example, accuracy and fineness of the results are still not so satisfactory.

Algorithm 1. HSV to RGB

Given: h on domain [0,360], s and v each on domain [0,1].
Desired: The equivalent r, g, and b, each on range [0,255]

$h := h/60$;
Let $i :=$ floor(h); $f := h$ -i;
Let $m := v$*(1-s);
    $n := v$*(1-(s*f));
    $k := v$*(1-(s*(1-f)));
Switch on i **into**
    case 0: $(r, g, b) := (v, k, m)$*255;
    case 1: $(r, g, b) := (n, v, m)$*255;
    case 2: $(r, g, b) := (m, v, k)$*255;
    case 3: $(r, g, b) := (m, n, v)$*255;
    case 4: $(r, g, b) := (k, m, v)$*255;
    case 5: $(r, g, b) := (v, m, n)$*255;

Remarks: 1) Floor(x) is the integer just less than or equal x. 2) Only one case is executed in the switch statement. 3) The expression $(r, g, b) := (x, y, z)$*255 abbreviates r:=x*255; g:=y*255; b:=z*255.

3 Preliminary Concepts

The HSV color model is used in our proposed approach to preserve the "direction" information of trajectories. And as introduced in Sect. 1, our proposed learning method is based on CycleGAN [8]. Therefore, in this section, we will introduce two preliminary concepts: the HSV color model and the CycleGAN model.

The HSV color model is a color space in which colors are expressed in terms of hue (h), saturation (s), and value (v). Hue is the dimension with points on it normally called red, yellow, blue-green, etc. Saturation measures the departure of a hue from achromatic, i.e., from white or gray. Value measures the departure of a hue from black, the color of zero energy. In the original literature [9], the range of h is from 0 to 1, but we need to map the orientation of trajectories to h, so the range of h is set from 0° to 360°. In this paper, the HSV representation of trajectory features should be converted to RGB model for later processing. The Algorithm 1 [9] shows how the HSV color space can be converted to an RGB model.

CycleGAN can model the data distributions in two domains separately by using a consistent cycle structure and the cycle consistency loss along with the adversarial loss of GAN [29], as shown in Eqs. 1, 2, 3, so it does not require data to be pairs of samples corresponding to each other at the same location. In Eq. 1, where G tries to generate images $G(x)$ that look similar to images from domain Y, while D_Y aims to distinguish between translated samples $G(x)$ and real samples y. In Eq. 2, for each image x from domain X, the image translation cycle should be able to bring x back to the original image, i.e. X→G(x)→F(G(x))≈x. Unlike other GANs, CycleGAN consists of two generators and two discriminators. The generator is used to generate target distribution, and the discriminator

is used to determine the difference between the generated target distribution and the real one. Through the adversarial process between the generator and the discriminator, the dynamic balance between the models is realized, which means that the generator can generate data that can make it difficult for the discriminator to distinguish between real and fake.

$$L_{GAN}(G, D_Y, X, Y) = E_{y \sim P_{data}(y)}[\log D_Y(y)] + E_{x \sim P_{data}(x)}[\log(1 - D_Y(G(x))] \tag{1}$$

$$L_{cyc}(G, F) = E_{x \sim P_{data}(x)}[\| F(G(x)) - x \|_1] + E_{y \sim P_{data}(y)}[\| G(F(y) - y \|_1] \tag{2}$$

$$L_{total} = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F) \tag{3}$$

4 Overview

This section first presents some basic definitions, defines our problem formally, and then analyzes the problems and introduces the overview approach.

4.1 Definition

Definition 1 (Trajectory). Given a moving object v_i, a raw trajectory of v_i is a sequence of positions of it over a period of time. For a moving vessel, v_i can be the Maritime Mobile Service Identify (MMSI). Thus, a raw trajectory can be represented as $T_{v_i} = (v_i, (p_0, p_1, ..., p_n))$, where $p_j = (x_{i,j}, y_{i,j}, t_j)$ indicates the position of the moving object at a certain moment, where t_j is the sampling (or collecting) time of the position, $x_{i,j}$, and $y_{i,j}$ represents the longitude and latitude of v_i at t_j.

Definition 2 (Grid). A grid is represented as $Grid = (Code, Points, Lines, Orientation, Speed)$, where $Code$ is the geographic hash code of the grid, while $Points$ is the point density of the grid, we use one of the three equal quantiles of point densities; $Lines$ represents the number of line segments in the grid ; $Orientation$ is the moving direction of the grid, which is the mean orientation angle of all points in the grid; and $Speed$ is the average moving speed of points in the grid.

Definition 3 (Feature Gridding). Given an area, we divide it horizontally $I - 1$ times and vertically $J - 1$ times. Feature gridding is getting $I \times J$ grids of the target area, where I and J indicates the number of rows and columns of the feature matrix, and $Points$ of all grids in the target area make up to a point feature, same as $Lines$, $Orientation$ and $Speed$ in line feature, orientation feature and speed feature, and these features are denoted as $F_{point} \in \mathbb{R}^{I \times J \times 1}$, $F_{line} \in \mathbb{R}^{I \times J \times 1}$, $F_{orientation} \in \mathbb{R}^{I \times J \times 1}$, $F_{speed} \in \mathbb{R}^{I \times J \times 1}$, respectively.

Definition 4 (Feature Merging). Feature merging concatenates features, i.e. point, line, orientation and speed, into one single feature matrix, with the size

of $I \times J \times 5$, to act as the input of the proposed model. Instead of concatenating features directly and simply, feature merging of this paper first transforms $F_{orientation}$ and F_{speed} into a new feature, denoted as $F_{orientation\_speed} \in \mathbb{R}^{i \times j \times 3}$, using HSV color space to preserve the roadway details. The specific experimental procedure is described in Sect. 5.2. Then F_{point}, F_{line} and $F_{orientation\_speed}$ are concatenated into the input of the model as $F_M \in \mathbb{R}^{i \times j \times 5}$.

Definition 5 (Fine-Grained Path). Fine-Grained Paths are the output of the proposed method, and are represented as a grid image with the size of $I \times J$, which is the same size as the trajectory features. Every line in the grid image of the fine-grained paths is a roadway of the traffic rather than the center line of the road. And every pixel in the image can be converted to its corresponding coordinates in the map.

Problem Statement. Given a set of trajectories $T = \{T_{v_1}, T_{v_2}, ..., T_{v_n}\}$, infer its underlying fine-grained paths P.

4.2 Problem Analysis and Approach Overview

In order to learn paths from trajectories without unpaired training data, we design our learning method based on CycleGAN. For urban traffic or maritime traffic, the supervised learning model cannot achieve traffic path extraction when there is no paired training data. In contrast, CycleGAN can handle unpaired training data and thus achieve traffic path extraction. When there is a lack of training data, CycleGAN can expand its training data by certain transformations because it does not need to perform one-to-one matching.

The key to learn fine-grained paths of roadway level is how to extract the roadway level features of trajectories. The directions of the adjacent roadways on the same path are usually opposite or significantly different, so the problem of extracting the roadway level features is how to represent the direction feature of trajectories. It is not appropriate to represent the direction feature directly with the angle value in the range $[0°, 360°)$, for example, the difference between 1° and 359° is very large but their directions are very close. In the HSV color space, the hue is represented with color wheel, which is consistent with the value range and change pattern of the orientation angle, for example, the HSV colors with the hue of 1° and 359° at the RGB color wheel are very similar. Inspired by the different representations of color space, if we represent the trajectory direction angle with "hue" dimension in HSV color space, the roadways in the opposite directions have very different colors in RGB color space. Therefore, in our method, we encode the direction angle value using HSV and transform it into RGB values to better represent the trajectory direction feature.

Figure 1 gives the overall process of learning the roadway level traffic paths from the unpaired traffic trajectory data and path data. As shown in the figure, the model consists of two components: the convolutional layers and the CycleGAN structure. This structure makes the model able to retain the spatio-temporal information of trajectory points and extract the roadway-level paths without standard paired reference data. In the training process, two kinds of

data sets, i.e. Trajectory Data and Path Data (note they can be unpaired), are used as the input of the model, as shown in Fig. 1. To achieve the conversion from trajectory data to roadway-level path data, spatio-temporal information are firstly obtained by feature extraction, and then feature gridding and feature merging are utilized to get the input of the proposed model. And at the same time path data is also gridded to get the sample of the target data distribution for the model.

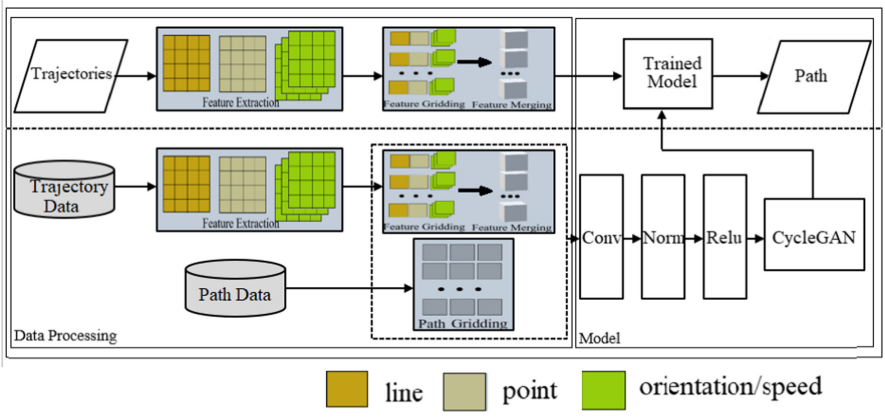


Fig. 1. The process of extracting fine-grained paths

5 Trajectory Data Transition

5.1 Feature Extraction

Given an area, We partition it into a grid with $I \times J$ cells and for each grid cell, then extract some features from the trajectories. As mentioned in Sect. 4.1, features extracted in this paper include: 1) Point, which is the most straightforward indicator of the underlying roadways; 2) Line, which can help with recovering the roads when points are sparse; 3) Orientation, this feature is helpful for distinguishing two parallel roads in opposite directions. And in Sect. 5.2, how orientation information are transformed into HSV represents is described; 4) Speed, it also helps to model the complete roadway, since, on the same road, the speed usually does not change that much on the same roadway.

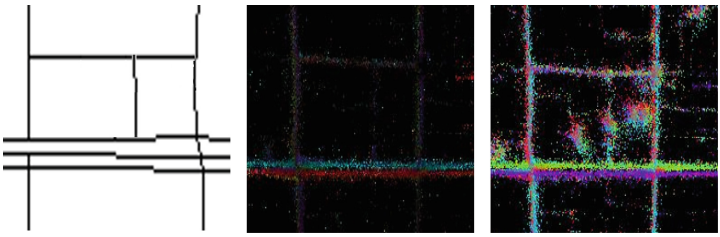
5.2 Orientation Converted to Color Information

To get the input of our CycleGAN-based method, feature matrices are organized more effectively rather than concatenated simply and directly. We adopt the HSV

color model to transform orientation feature and speed feature of trajectory point into RGB color model.

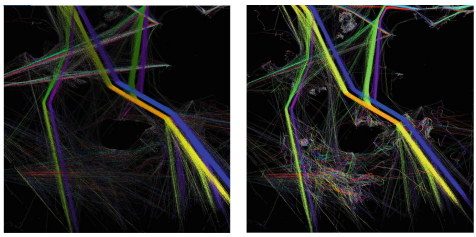
The orientation of each trajectory point can be calculated by the adjacent trajectory points in a trajectory data, i.e. the azimuth (angle with due north). And then the azimuth value, also denoted as (d), can be converted into the (h) of the HSV color space. As for s and v, if we set them as $s = 1, v = 1$ according to [7], the result of GPS trajectory transformation when $h = d, s = 1, v = 1$ is shown in Fig. 2(c). It is notable that, taking the ground truth of the same area as a reference, there are many noise points when $h = d, s = 1, v = 1$ that may affect the path extraction. Therefore, we encode the normalized value of speed of the trajectory points to the HSV to reduce the saturation of the converted trajectory points outside the normal driving speed range, so that the model ignores these noise points. We set $h = d$, s=1, $v = speed$ and the result is shown in Fig. 2(b), there are obviously less noise points than in (c), and details of the roadways are retained. The comparison of AIS trajectory transformation when $h = d$, $s = 1$, $v = speed$ and $h = d, s = 1, v = 1$ is presented in Fig. 3, it is easy to find that adding the normalized value of speed to HSV can reduce noise points.

We denoted the result of feature transformation as $F_{orientation\_speed} \in \mathbb{R}^{i \times j \times 3}$, and the input of our CycleGAN-based method is achieved by feature merging of $F_{orientation\_speed}$ with F_{point} and F_{line}.



(a) Ground Truth (b) v=speed (c) v=1

Fig. 2. Transformation of GPS trajectory features to RGB models



(a) v=speed (b) v=1

Fig. 3. Transformation of AIS trajectory features to RGB models

6 Training Model

We intend to use the CycleGAN model as a training model for trajectory extraction through the introduction in Sect. 1. As CycleGAN is an image-to-image learning model, we need to convert the input data into three-channel data. In Sect. 5.1 and Sect. 5.2, the F_{point}, F_{line}, $F_{orientation\_speed}$ is a five-channel data, so in order to solve this problem, the model needs to be modified. There are two modification methods:

1) Add a fully connected layer before the input of the generator.

2) Add a convolutional layer before the input of the generator as shown in Fig. 4.

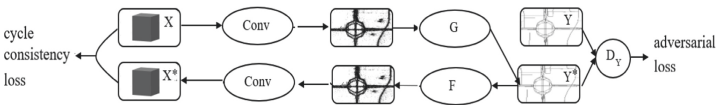


Fig. 4. Training pipeline of the proposed method

When dealing with multi-dimensional data, the convolutional layer has the following advantages over the fully connected layer: The fully connected layer flattens the input data into a vector. The adjacent elements on the input data may no longer be adjacent due to the flattening operation, and it is difficult for the network to capture local information. The design of the convolutional layer naturally has the ability to extract local information, and has less parameters. That is because the convolutional layer can capture more spatially correlated input features; and when the input shape is (c_1, h_1, w_1) and the output shape is (c_2, h_2, w_2), with h and w representing the height and width of the convolutional kernel, the number of parameters of a convolutional layer would be $c_1 \times c_2 \times h \times w$ without considering the bias, and the number of parameters of a fully connected layer would be $c_1 \times c_2 \times h_1 \times w_1 \times h_2 \times w_2$.

After transform the F_{point}, F_{line}, $F_{orientation\_speed}$ into three channels by the added convolutional layers (Gconv1 shown in Fig. 5), we borrow from the design of the CycleGAN [8] network. We adopt ResNet9blocks, a jump-connected encoder-decoder structure. It is able to extract information about data features and accurately reconstruct the road network structure. The overall network structure proposed based on CycleGAN is shown in Fig. 5. The network architectures settings are shown in Tables 1 and 2. In G_x, G_y, D_x, D_y, the convolutional layers are denoted as Gconv1, Gconv2, Gconv3, Gconv4, Gconv8, Dconv1, Dconv2, Dconv3, Dconv4, Dconv5. Gconv5_x denotes a residual block which contains two convolutional layers with the same number of filters in both layers are the same. The fractionally-strided convolutions layers are denoted as Gconv6, Gconv7. For the sake of brevity, the activation function, reflection padding and normalization layers are not shown. Input_x and Input_y represent the source domain data and the target domain data, respectively. And the

training process is as follows, where Input_x, Generated_x and Cycli_x represent the transformed feature matrix, and Input_y, Generated_y and Cycli_y represent the roadway-level paths. Input Input_x into G_x to generate Generated_y. Input Input_y into G_y to generate Generated_x. Input Generated_y into G_y to generate Cycli_x. Input Generated_x into G_x to generate Cycli_y. Input_x and Cycli_x are used as training for forward cycle-consistency loss. Input_y and Cycli_y are used as training for backward cycle-consistency loss. Here, k, n, and s denote the kernel size, the number of feature maps, and the stride, respectively.

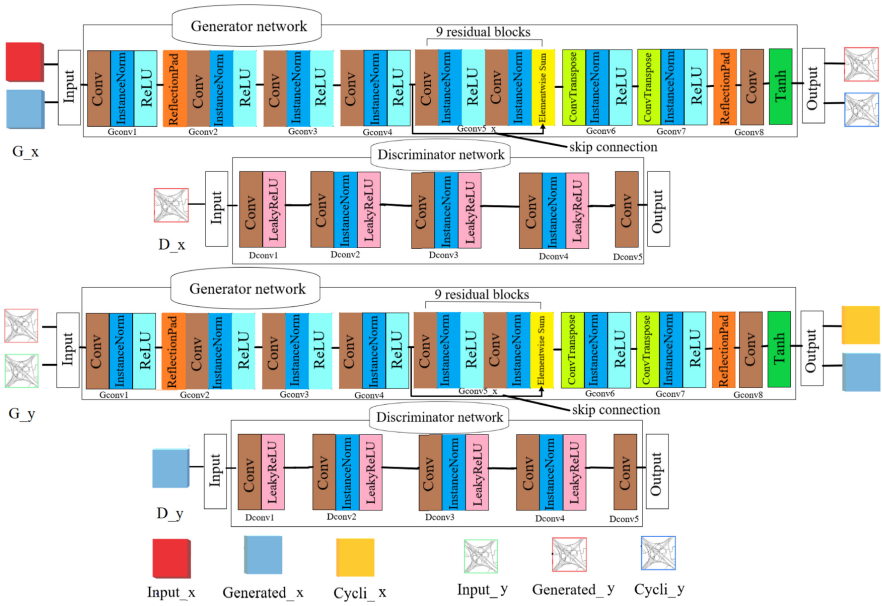


Fig. 5. Network architectures of the generator and the discriminator

7 Experiment and Analysis

7.1 Dataset and Experimental Environment

Our CycleGAN-based model achieve conversion from one data distribution to another, and the two data distributions are trajectory features and roadway-level traffic paths, respectively. We call data sets that have the same data distribution as roadway-level traffic paths target domain data. Note that the target domain data don't need to be paired with the input data.

As mentioned in Sect. 1, it is more challenging to extract traffic paths from maritime AIS trajectory data because there is a lack of standard road network data set in maritime traffic. This also means that it is difficult to quantitatively

Table 1. Model setting of the generator

Layer	Output	G_x	G_y
Gconv1	256*256	k3n3s1	k3n3s1
Gconv2	256*256	k7n64s1	k7n64s1
Gconv3	128*128	k3n128s2	k3n128s2
Gconv4	64*64	k3n256s2	k3n256s2
Gconv5_x	64*64	k3n256s1	k3n256s1
Gconv6	128*128	k3n128s2	k3n128s2
Gconv7	256*256	k3n64s2	k3n64s2
Gconv8	256*256	**k7n3s1**	**k7n5s1**

Table 2. Model setting of the discriminator

Layer	Output	D_x	D_y
Dconv1	128*128	k4n64s2	k4n64s2
Dconv2	64*64	k4n128s2	k4n128s2
Dconv3	32*32	k4n256s2	k4n256s2
Dconv4	31*31	k4n512s1	k4n512s1
Dconv5	30*30	k4n1s1	k4n1s1

evaluate our proposed method because of the lack of benchmark. Therefore, in order to quantitatively evaluate the effectiveness of the method more fairer, we conduct experiments both on maritime AIS trajectory data and on urban GPS trajectory data, since roadway benchmark can be achieved in urban traffic.

For the maritime traffic data, to prepare the target domain data, the Delaunay based method proposed in our previous study [30] is firstly used to extract the paths, and then the path center line data is extracted according to the path data. We also manually process the results based on the trajectory density heat map for fine-grained maritime traffic paths. And the urban traffic paths data is obtained from the open source website OpenStreetMap (OSM).

The maritime data we used in the experiment is the AIS data of cargo ships in Danish waters in March 2021 (ftp://ftp.ais.dk/ais_data/). AIS data includes attributes such as ship's name, call sign, MMSI, IMO, ship type, destination, ship width, and other static information, as well as dynamic information such as UTC, latitude, longitude, direction, and speed. The experiment uses four columns of MMSI, UTC, longitude, and latitude from the data set. The total amount of AIS data for cargo ships in Danish waters is 17.3GB. As shown in Table 3, the data contains travel records for more than 5,930 vessels and more than 7 million trajectory points.

We also used cab GPS(1) trajectory data from February 2, 2008 to February 8, 2008 in Beijing (https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/)and GPS(2) trajectory data from Chengdu taxi on August 3, 2014. The dataset contains four attributes, such as cab ID, date, longitude, and latitude. Data details are shown in the Table 3. The land-based trajectory data are introduced here for the later quantitative analysis.

The experiment was run in a Ubuntu 18.04 environment. Model implementation and result analysis based on Python 3.6 and Pytorch 1.4.0 deep learning open source library. The model is trained with a single NVIDIA GeForce RTX 2080Ti GPU and has a memory of 11 GB.

Table 3. Data descriptions

Dataset	Days	Ships/ Vehicles	Points	Lng(min/max)	Lat(min/max)	Amount
AIS	30	5930	7126586	2.5229/17.5107	52.9517/59.1222	17.3 GB
GPS(1)	7	10336	1713331	116.2810/116.4684	39.8451/39.9890	1.40 GB
GPS(2)	1	13605	53045405	103.2696/104.6096	30.2906/31.0324	3.64 GB

7.2 Parameter Setting and Data Division

After obtaining the original data of a rectangular area, it is then divided into 8192×8192 grid areas. The size of each grid area is 256×256, and a total of 1024 grid areas are generated. The number of grids in the test set is 100. The ratio of the training set to the validation set is 9:1, the number of grid regions in the training set is 831, and the number of grid regions in the validation set is 93. Then extract the features in each grid, each grid contains three feature data: F_{point}, F_{line}, and $F_{orientation\_speed}$. Then merge the features in each grid area into an 5 channel feature matrix. Finally, the data of the training set, validation set, and test set are 831, 93, and 100 respectively. AIS data and GPS data are divided in the same way.

During the training phase, we leverage Adam [31] to perform network training with a learning rate $2e{-}4$ and batch size 1. We keep the same learning rate for the first 100 epochs and linearly decay the rate to zero over the next 100 epochs. The slope of LeakyReLU in the negative part is set to 0.2. More parameters of the model are shown in the Tables 1 and 2.

To show the effectiveness of cycle consistency loss function, we evaluated our model at different λ, as in Eq. 3, settings from 7 to 17. The results are shown in the line graphs in Fig. 6. We find that the network achieves the best performance on both datasets when $\lambda = 16$. A smaller λ or a larger λ reduces the inferred performance of the road network. As shown in Fig. 6.

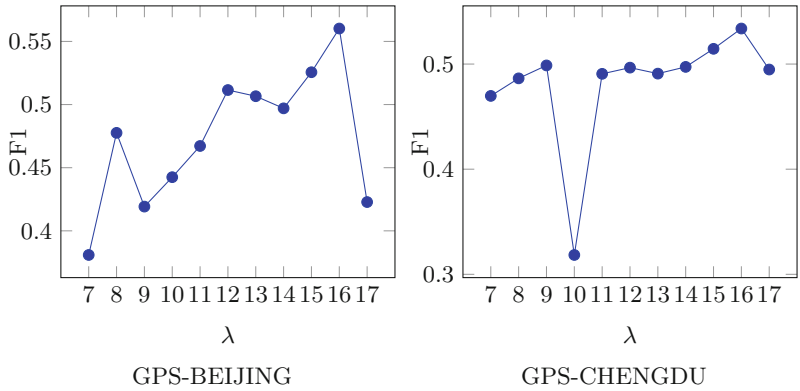


Fig. 6. Effect of λ

7.3 Results and Performance Comparison

Figure 7 shows the extraction result of the fine-grained maritime traffic paths in Danish regional waters. In order to compare and evaluate the extraction results of this experiment, this paper compares and evaluates the extraction methods of T2I-CycleGAN [5], Simplified DeepMG [6].

As introduced in Sect. 1, T2I-CycleGAN [5] is the only comparable unpaired learning method for traffic paths in the existing studies. The DeepMG framework is based on a supervised learning method, it uses OSM data as a reference for model training, its training purpose is more clear than the proposed unsupervised learning. In fact, it is unfair to compare an unpaired learning method with a supervised learning method, because their training sets are different. In general, supervised learning methods should be better than the unpaired learning approach because they the data they use for training must be paired. But here we also compare our approach with DeepMG, the purpose is to show that our method is comparable to the supervised learning method.

Simplified DeepMG: The DeepMG framework uses a CNN-based supervised training method for road network extraction. The trajectory data used in our experiments, when these data are used for T2RNet training, will result in overfitting. In addition, we have not yet implemented the topology construction part of the DeepMG framework, and will implement it in future work. The result is shown in Fig. 7. In the case of using the same test data, the results obtained by the simplified DeepMG are more detailed.

T2I-CycleGAN: By adding a fully connected layer to the CycleGAN generator, the possible sparse features are converted into dense grid images for training. The extracted features include points, lines, speed, direction and spatial features. The final extraction results are shown in the Fig. 7.

In terms of experiments on urban trajectories, we use the road network from OSM map data of Beijing area as the benchmark. The following Fig. 8 and Fig. 9 show the standard road network map and the extracted traffic paths using different approaches.

We can observe that in Fig. 8, for the areas marked by the red dotted box, our method can extract roadway level paths while the other two methods can not achieve accurate roadways. Note that all the methods can not extract roadways in the areas marked by the green box, it is only because that there is only one roadway on the actual path. Even in this case, our method can also extract the path with less density and better connectivity as shown in Fig. 9. In the extraction method, the road network of Simplified DeepMG is more straight. That is because DeepMG is based on a pix2pix method that can extract roads one-to-one, but it cannot handle the lack of data and unpaired data as in our experiments.

We use the de-facto standard for measuring the map's quality to measure the geometric and topological similarity of maps, proposed in [32]. The main idea [20] is that we look for the pixel cell within its corresponding range on the image generated by the OSM data for every location on the road, and we set the matching range to 2 pixel. Before matching, we perform pixel value filtering

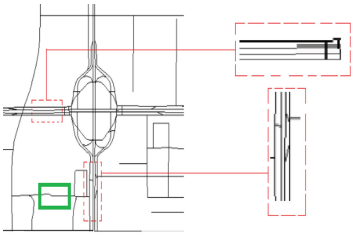
(a) Original AIS track point map

(b) Simplified DeepMG

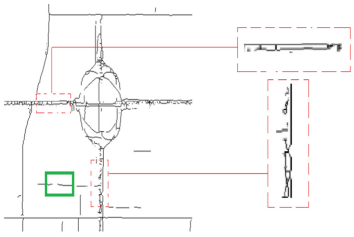
(c) T2I-CycleGAN

(d) Our Approach

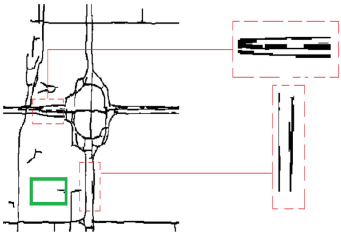
Fig. 7. Comparison of the results of different extraction methods for AIS data



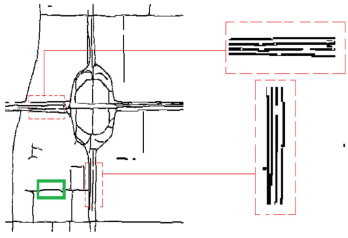
(a) Standard road network map



(b) Simplified DeepMG



(c) T2I-CycleGAN



(d) Our Approach

Fig. 8. Comparison of the results of different extraction methods for GPS data

(a) Standard road network map (b) Simplified DeepMG (c) T2I-CycleGAN (d) Our Approach

Fig. 9. Comparison of the results of different extraction methods for GPS data

on all the resulting images to remove dark spots. We set the pixel value filtering range from 150 to 255. We report the F1 score, recall, and precision in the whole result image.

1) Precision, the ratio of the correctly extracted road length to the total length of the extracted paths.

2) Recall, the ratio of the correctly extracted road length to the total length of paths of the benchmark data.

3) F1 score, weighted Harmonic mean of precision and recall, as show in calculation Eq. 4.

$$F_1 = \frac{2P \cdot R}{P + R} \tag{4}$$

The evaluation of different feature combinations is shown in Table 4, where Point, Line, Orientation_Speed means F_{point}, F_{line}, and $F_{orientation\_speed}$ as input, respectively. It can be found that F_{point} has higher metrics than F_{line}, which is consistent with our common knowledge. The results also demonstrate that the $F_{orientation\_speed}$ can improve the Recall, Precision and F1 scores of the model.

We evaluate the model structure of the generator in our approach by comparing the precision, recall and F1 for the test set of seven network structures as shown in Table 5. ResNet9blocks used in CycleGAN and in our approach has the best performance in the form of precision, recall and F1 score and is able to reduce the overfitting problem whether it is performed on the maritime or the urban data set, when comparing with seven network structures as shown in Table 5.

As shown in Table 6, in the unsupervised learning domain, for the evaluation metrics of road network extraction, the Precision, Recall and F1 scores of the extraction method in this paper are higher than those of T2I-CycleGAN. In the supervised and unsupervised learning neighborhoods, the Precision, Recall and

Table 4. Effect of different features

Feature	Precision	Recall	F1 score
Point	0.4498	0.3767	0.4100
Line	0.3739	0.2363	0.2896
Orientation_Speed	0.5597	0.4136	0.4757

Table 5. Generator model comparisons on different datasets

Model	#Params	GPS-BEIJING			GPS-CHENGDU		
		Precision	Recall	F1 score	Precision	Recall	F1 score
ResNet9blocks [33] (adopted in our approach)	11.378M	**0.4779**	0.6767	**0.5602**	**0.4101**	**0.7643**	**0.5338**
UNet256 [34]	54.410M	0.4458	0.5436	0.4897	0.3756	0.5454	0.4448
FCN [35]	20.104M	0.2784	**0.6809**	0.3952	0.2535	0.6235	0.3604
LinkNet [36]	11.534M	0.2831	0.6782	0.3994	0.2206	0.6124	0.3243
T2RNet [6]	63.082M	0.3174	0.3365	0.3266	0.3011	0.4102	0.3472
D-LinkNet [37]	31.097M	0.3386	0.5131	0.4032	0.2938	0.4803	0.3645
ResNet6blocks [33]	7.838M	0.4501	0.6264	0.5238	0.3875	0.6904	0.4963

Table 6. Comparison of evaluation indicators

Model	GPS-BEIJING			GPS-CHENGDU		
	Precision	Recall	F1 score	Precision	Recall	F1 score
Simplified DeepMG	0.4325	0.6629	0.5235	0.4091	0.5227	0.4587
T2I-CycleGAN	0.3567	0.5613	0.4362	0.3428	0.7289	0.4663
Our approach	**0.4779**	**0.6767**	**0.5602**	**0.4101**	**0.7643**	**0.5338**

F1 Score of the road network extracted by the method in this paper is higher than that of Simplified DeepMG.

8 Conclusion

By proposing an deep learning method based on CycleGAN and using HSV color space to reinforce the spatial features of moving trajectories, this paper achieves roadway-level traffic path extraction without paired standard reference data. In this paper, experiments are conducted with maritime ship AIS data as an example. By referring to the extraction and pre-processing methods of urban traffic networks, we initially processed the collected AIS data and filtered out the noise data. Then based on the CycleGAN model, we proposed an unpaired learning model and realized the extraction of maritime traffic paths by converting the direction feature of the trajectory data into RGB matrix data through HSV

color space. We use the OSM data of urban traffic as the reference standard for the results to conduct a quantitative analysis comparing the model and the feature construction method with some baseline methods, and the results show that the proposed method in this paper can extract more accurate traffic paths. For further optimization research of path extraction, on the one hand, we can increase the sample data types sothat the model can learn to generate more path types; on the other hand, the model can be further optimized to integrate more trajectory feature information, so that it can extract richer path information.

Acknowledgements. This work is supported by National Natural Science Foundation of China (Grant No. 61832004) and Projects of International Cooperation and Exchanges NSFC (Grant No. 62061136006).

References

1. Wang, W., Yang, N., Zhang, Y., Wang, F., Cao, T., Eklund, P.: A review of road extraction from remote sensing images. J. Traffic Transp. Eng. (English Edition) **3**(3), 271–282 (2016)
2. Liu, J., Qin, Q., Li, J., Li, Y.: Rural road extraction from high-resolution remote sensing images based on geometric feature inference. ISPRS Int. J. Geo Inf. **6**(10), 314 (2017)
3. Bae, Y., Lee, W.H., Choi, Y., Jeon, Y.W., Ra, J.B.: Automatic road extraction from remote sensing images based on a normalized second derivative map. IEEE Geosci. Remote Sens. Lett. **12**(9), 1858–1862 (2015)
4. Dai, J., Ma, R., Gong, L., Shen, Z., Wu, J.: A model-driven-to-sample-driven method for rural road extraction. Remote Sens. **13**(8), 1417 (2021)
5. Yang, X., Wang, G., Yan, J., Gao, J.: T2I-CycleGAN: a CycleGAN for maritime road network extraction from crowdsourcing spatio-temporal AIS trajectory data. In: Gao, H., Wang, X., Iqbal, M., Yin, Y., Yin, J., Gu, N. (eds.) CollaborateCom 2020. LNICST, vol. 350, pp. 203–218. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67540-0_12
6. Ruan, S., Long, C., Bao, J., Li, C., Zheng, Y.: Learning to generate maps from trajectories. Proc. AAAI Conf. Artif. Intell. **34**(1), 890–897 (2020)
7. Lu, C., Sun, Q.: Road learning extraction method based on vehicle trajectory data. acta geodaetica et cartographica sinica. J. Surveying Mapp. **49**(06), 26–36 (2020)
8. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), October 2017
9. Smith, A.R.: Color gamut transform pairs. ACM Siggraph Comput. Graph. **12**(3), 12–19 (1978)
10. Liu, B., Wu, H., Wang, Y., Liu, W.: Main road extraction from zy-3 grayscale imagery based on directional mathematical morphology and vgi prior knowledge in urban areas. PLoS ONE **10**(9), e0138071 (2015)
11. Ma, W., Z., W.: One-way distance-based spectral clustering for ship motion pattern recognition. J. Chongqing Jiaotong Univ. (Natural Sci. Edition) **34**(5), 130–134 (2015)
12. Wu, J., Zhu, Y., Ku, T., Wang, L.: Detecting road intersections from coarse-gained GPS traces based on clustering. J. Comput. **8**(11), 2959–2965 (2013)

13. Aronov, B., Driemel, A., Kreveld, M.V., Löffler, M., Staals, F.: Segmentation of trajectories on nonmonotone criteria. ACM Trans. Algorithms (TALG) **12**(2), 1–28 (2015)
14. Stanojevic, R., Abbar, S., Thirumuruganathan, S., Chawla, S., Filali, F., Aleimat, A.: Robust road map inference through network alignment of trajectories. In: Proceedings of the 2018 SIAM International Conference on Data Mining, pp. 135–143. SIAM (2018)
15. Xie, X., Liao, W., Aghajan, H., Veelaert, P., Philips, W.: Detecting road intersections from GPS traces using longest common subsequence algorithm. ISPRS Int. J. Geo Inf. **6**(1), 1 (2017)
16. Fathi, A., Krumm, J.: Detecting road intersections from GPS traces. In: Fabrikant, S.I., Reichenbacher, T., van Kreveld, M., Schlieder, C. (eds.) GIScience 2010. LNCS, vol. 6292, pp. 56–69. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15300-6_5
17. Wang, J., Rui, X., Song, X., Tan, X., Wang, C., Raghavan, V.: A novel approach for generating routable road maps from vehicle GPS traces. Int. J. Geogr. Inf. Sci. **29**(1), 69–91 (2015)
18. Wang, S., Wang, Y., Li, Y.: Efficient map reconstruction and augmentation via topological methods. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1–10 (2015)
19. Ahmed, M., Karagiorgou, S., Pfoser, D., Wenk, C.: Map construction algorithms. In: Map Construction Algorithms, pp. 1–14. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25166-0_1
20. Ahmed, M., Karagiorgou, S., Pfoser, D., Wenk, C.: A comparison and evaluation of map construction algorithms using vehicle tracking data. GeoInformatica **19**(3), 601–632 (2015)
21. Mariescu-Istodor, R., Fränti, P.: Cellnet: inferring road networks from GPS trajectories. ACM Trans. Spatial Algorithms Syst. (TSAS) **4**(3), 1–22 (2018)
22. Kuntzsch, C., Sester, M., Brenner, C.: Generative models for road network reconstruction. Int. J. Geogr. Inf. Sci. **30**(5), 1012–1039 (2016)
23. Furiang, T., Zhang, L., Xue, Y., Zihan, K., Qingquan, L., Kun, D.: A spatio-temporal trajectory fusion and road network generation method in accordance with cognitive laws. J. Surveying Mapp. **44**(11), 1271 (2015)
24. Yang, W., Ai, T.: Road centerline extraction based on multi source trajectory data. Geogr. Geog. Inf. Sci. **32**(003), 1–7 (2016)
25. Yang, W., Ai, T.: Using constrained delaunay triangulation to extract road boundary from multi-source trajectories. J. Surveying Mapp. **046**(002), 237–245 (2017)
26. Tang, L., Chang, R., Zhang, L., Li, Q.: A road map refinement method using delaunay triangulation for big trace data. ISPRS Int. J. Geo Inf. **6**(2), 45 (2017)
27. Li, Z., Wang, G., Meng, J., Xu, Y.: The parallel and precision adaptive method of marine lane extraction based on QuadTree. In: Gao, H., Wang, X., Yin, Y., Iqbal, M. (eds.) CollaborateCom 2018. LNICST, vol. 268, pp. 170–188. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12981-1_12
28. Xin, J., Zhang, X., Zhang, Z., Fang, W.: Road extraction of high-resolution remote sensing images derived from denseunet. Remote Sens. **11**(21), 2499 (2019)
29. Goodfellow, I., et al.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
30. Wang, G., Meng, J., Li, Z., Hesenius, M., Ding, W., Han, Y., Gruhn, V.: Adaptive Extraction and Refinement of Marine Lanes from Crowdsourced Trajectory Data. Mobile Networks and Applications, pp. 1392–1404 (2020)

31. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
32. Biagioni, J., Eriksson, J.: Inferring road maps from global positioning system traces: survey and comparative evaluation. Transp. Res. Record J. Transp. Res. Board **2291**, 61–71 (2014)
33. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
34. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
35. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
36. Chaurasia, A., Culurciello, E.: Linknet: exploiting encoder representations for efficient semantic segmentation. In: 2017 IEEE Visual Communications and Image Processing (VCIP), pp. 1–4. IEEE (2017)
37. Zhou, L., Zhang, C., Wu, M.: D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 182–186 (2018)