IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 14, 2021, accepted October 25, 2021, date of publication October 28, 2021, date of current version November 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123942

CE-Net: A Coordinate Embedding Network for Mismatching Removal

SHIYU CHEN[1,2,3], JIQIANG NIU[1,2], CAILONG DENG[4], YONG ZHANG[5], FEIYAN CHEN[1,2], AND FENG XU[1,2]

[1]School of Geographic Sciences, Xinyang Normal University, Xinyang 464000, China
[2]Henan Engineering Research Center for Big Data of Remote Sensing and Intelligent Analysis in Huaihe River Basin, Xinyang Normal University, Xinyang 464000, China
[3]Key Laboratory for National Geographic Census and Monitoring, National Administration of Surveying, Mapping and Geoinformation, Wuhan University, Wuhan 430079, China
[4]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[5]Visiontek Inc., Wuhan 430205, China

Corresponding author: Cailong Deng (dcl@whu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 41901402 and Grant 4171438BS123456, in part by the Key Scientific Research Projects of Higher Education Institutions of Henan under Grant 20A420005, in part by the Key Laboratory for National Geographic Census and Monitoring through the National Administration of Surveying, Mapping and Geoinformation under Grant 2018NGCM04, in part by the Nanhu Scholars Program for Young Scholars of Xinyang Normal University (XYNU) under Grant 2019037, and in part by the Science and Technology Innovative Research Team in Higher Educational Institutions of Henan Province under Grant 22TRTSTHN010.

ABSTRACT Mismatching removal is at the core yet still a challenging problem in the photogrammetry and computer vision field. In this paper, we propose a coordinate embedding network (named CE-Net). We consider the mismatching problem as a graph node classification problem, and generate node descriptors by embedding point coordinates and aggregating geometric information from neighboring nodes based on self-attention and cross-attention mechanism. Finally, a binary classifier is used to separate node descriptors into two classes, namely matching inliers and outliers. Benefiting from the attention mechanism, firstly the node descriptors can get geometric information from ''good neighbors'' (i.e., matching inliers) and keep away from ''bad neighbors'' (i.e., matching outliers), improving the exactness of the descriptors; secondly the node descriptors can contain the information from both intra-graph and inter-graph, improving their distinctiveness. Experiments in testing datasets show that our proposed CE-Net achieves the state-of-the-art performance with a precision of 0.972, an outlier recall of 0.984, and an inlier recall of 0.963. Furthermore, CE-Net also outperforms the compared methods in real mismatching removal tasks in terms of positional accuracy, dispersion, and number of remaining point pairs, showing great potentials in practical applications. Our codes and data are available on https://github.com/csyhy1986.

INDEX TERMS Coordinate embedding, node classification, attention mechanism, inter-graph, intra-graph, binary classifier.

I. INTRODUCTION

Obtaining reliable matching points between image pairs is at the core in photogrammetry and computer vision task [1], and yet mismatches caused by illumination condition changes, different camera viewpoints, and object occlusion are inevitable, which hinders the subsequent applications of the matching results [2], [3]. Thus, a mismatch removal process should serve as a necessary step to ensure correct matches and improve the accuracy [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Mingjun Dai.

Mismatch removal (i.e., removing the mismatched point which is not from the same object) is still a challenging problem because one should find an invariant transformation between matching inliers which is immune to matching outliers. This is a chicken-and-egg problem, as finding the transformation first requires the identification of all the inliers (i.e., simultaneously maximize inlier recall and outlier recall) and vice versa [5]. Thus, most of the mismatch removal algorithms are heuristic. Based on the way to find the optimal transformation, existing mismatch removal methods can be classified as handcrafted methods and learning-based methods. Thereinto, handcrafted methods

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

have two most representative implements, one is based on RANdom SAmpling Consensus (RANSAC) [6], and the other is based on local deformation consistency.

Handcrafted methods aim to iteratively find mathematical models that are invariant to local or global transformation between matched points. RANSAC is one of the most representative handcrafted methods which works through iteratively sampling to find an optimal transformation matrix (such as affine, homography, and fundamental matrix) [7], then uses the computed matrix to determine whether a pair of matched points is an inlier or not. Most variants of RANSAC, such as MLESAC [8], Guided-MLESAC [9], and PROSAC [10], improve the sampling strategy either to obtain a more reliable result or to accelerate the computing speed. Nevertheless, global transformation obtained by RANSAC is not suitable for matching points from multi-consistency [11] or non-rigid images [12].

Unlike RANSAC-type algorithms that intend to find a global mathematical model, other handcrafted methods argue that the deformations of local image patches are consistent and utilize a series of local models to approach the globally optimal transformation. For example, Locality Preserving Matching (LPM) [13] assumes that the local neighborhood structures of potential matches do not vary freely due to the physical constraints, and preserves these invariances by a set of cost functions, and finally gets inliers by minimizing these functions. Grid-Based Motion Statistics (GMS) [14] gives a more relaxed assumption than LPM, it introduces smoothness constraint into a statistic framework to separate matching inliers and outliers, and uses grid-based implementation to speed up computing. Whereas ''bad neighbors'' (i.e., matching outliers) are essentially noise and ''good neighbors'' (i.e., matching inliers) are actually useful information in data mining, these handcrafted mining methods cannot differentiate ''good neighbors'' from ''bad neighbors.'' Thus, if matching inliers are surrounded by outliers, which often happens in clustered matching points, then mining local neighbor structures is not sufficient to separate inliers and outliers.

Learning-based methods explore another way to solve the mismatch removal problem. Inspired by GMS and LPM, a potential match is closely related to its neighbors, therefore a match can be described by a vector (named as descriptor) which embeds the geometric information of its neighbors. Then the descriptor can be fed to a classifier, such as support vector machine or random forest, to determine whether the match is an inlier or not. For example, learning for mismatch removal (LMR) [15] first embeds neighborhood topology (such as lengths and angles) of putative matches into descriptors, and then uses labeled descriptors to train a two-class classifier and applies the trained classifier to separate inliers and outliers. While LMR has a small receptive field and cannot mine global geometric information which is vital for mismatching removal of clustered matching points.

Recently, deep learning [16] is flourishing in computer vision field and has made dramatic progress in semantic segmentation [17], object recognition [18] and image classification [19]. If coordinates of two matched points are concatenated together to form a four-dimensional point, all matched points will form a set of four-dimensional point clouds, thus a semantic point cloud segmentation algorithm (such as PointNet [20]) can be applied in solving the mismatching removal problem. Learning to find good correspondences (LFGC) [21] network improves PointNet by applying a context normalization in a convolutional neural network (CNN) [22] to process each data point independently, and it is a multi-task learning process that simultaneously minimizes the classification loss and fundamental matrix regression loss. Therefore, global and local geometric information can be imbued in the network, leading to better results than PointNet.

Based on CNNs and graph embedding, graph neural network (GNN) [24] is proposed to extract spatial features and aggregate information from irregular graph data, and it shows reliable performance in node classification [42]. Neighbor mining network (NM-Net) [23] adopts a similar network architecture to LFGC, it mines the structures of subgraphs of k nearest compatible neighbors to generate matched point descriptors. Although the structures only capture local geometric information, NM-Net can also grasp the global information because of its multi-layer architecture. Specifically, initial layers concentrate on local structures, while last layers can get global information by using an aggregator (e.g., mean, sum).

Though deep learning based methods can obtain local and global information, they process a pair of matched points as a whole, that is, two coordinates of the matched point pairs are concatenated together to feed to a network. Whereas the two matched points in the left and right images are conjugated, if they are truly matched, then they will have similar neighbors; on the contrary, their neighbors are far different. Thus, geometric information of inner and cross images should be synchronously included, because intra-image information can give a precise description of the matched points like LFGC and NM-Net, meanwhile inter-image information will improve the distinctiveness of the description.

Therefore, if we consider matching points in the left and right images as nodes of two graphs, the idea of embedding both inner and cross image information in node descriptors can be easily implemented by means of node embedding paradigm in graph neural network (GNN) [24]. Taking the advantage of graph node embedding paradigm, such as graph attention (GAT) network [25], graph convolutional network (GCN) [26], Graph sample and aggregate (GraphSAGE) [27], matching coordinate geometric information can be easily embedded into node descriptors. GAT is one of the best choices because it can aggregate intra-graph information by self-attention and inter-graph information by cross-attention [28], and is capable of both parallelizability and interpretability.

Given the precise node descriptors, the mismatching removal problem will be converted into a graph node classification problem [29], and it can be easily solved by numerous deep learning-based classifiers. Whereas, to the

best of our knowledge, it is not common to deem mismatching removal problem as a graph node classification problem, and embed both intra-graph and inter-graph information into node descriptors in existing studies.

To solve the mismatching removal problem and explain how the network works, a coordinate embedding network (CE-Net) is proposed in this paper. Compared with traditional methods, the main contributions are in the following four folds.

(1) We deem mismatching removal problem as a node classification problem, and use GAT to embed geometric information of matched points (Section II). Experiments (Section III.B) and applications (Section III.C) show the proposed method (CE-Net) outperforms the state-of-the-art methods such as RANSAC, LPM, GMS, PointNet, LFGC and NM-Net in terms of precision, recall, and positional accuracy.

(2) To improve the distinctiveness of the node descriptors, we use cross-attention as a supplement for self-attention in producing node descriptors (Section II.C), and ultimately improve the recall of inliers and outliers.

(3) We give a concise explanation of GAT (Section II.C) and visualize the process of how CE-Net takes effect (Section III.B).

(4) The proposed CE-Net can be regarded as a generalization of local deformation consistency based method (e.g., GMS and LPM), and CE-Net can generate exact node descriptors by aggregating geometric information of "good neighbors" and ignoring that of "bad neighbors."

II. METHOD

Given a matching point set $M = \{m_i = (x_i, y_i, x_{i'}, y_{i'})|1 <= i <= n\}$, where $c_i(x_i, y_i)$ and $c_{i'}(x_{i'}, y_{i'})$ are located in the left image and right image respectively. The mismatching removal problem is to find a classifier C to divide the point set M into an inlier set R and an outlier set S with the constraints $R \cap S = \emptyset$ and $R \cup S = M$. If we consider m_i as a graph node, then the mismatching problem turns into a node classification problem and it can be solved by a deep learning algorithm without any strain.

As a necessary process in GNN-based node classification algorithms, node embedding plays a vital role in differentiating one node from another [30]. Especially in mismatching removal problem, we want to learn a descriptor for every node which comprises geometric invariance by aggregating intra-graph and inter-graph geometric information. As the flow chart shows in Fig. 1, if we want to know whether c_5 and $c_{5'}$ is a matching outlier or not, we should first construct two graphs whose nodes represent the putative point correspondences, then utilize GAT to embed their neighbor geometric information from intra-graph and inter-graph into node descriptors for node 5 and 5′. Once the learned node descriptors are obtained, a certain classifier can be used to divide the matching point set M into two subsets (i.e., an inlier set and an outlier set), and consequently the mismatching removal problem is solved.

In section II, we detailedly explain how to use attention mechanism to embed geometric information of matched points. Section II are organized as follows: 1. Network architecture of CE-Net is shown (Section II.A); 2. The node embedding paradigm is introduced (Section II.B); 3. We describe coordinate embedding in detail (Section II.C), namely using self-attention and cross-attention to embed intra-graph and inter-graph information of point coordinates into node descriptors, Section II.C includes 1) MLP

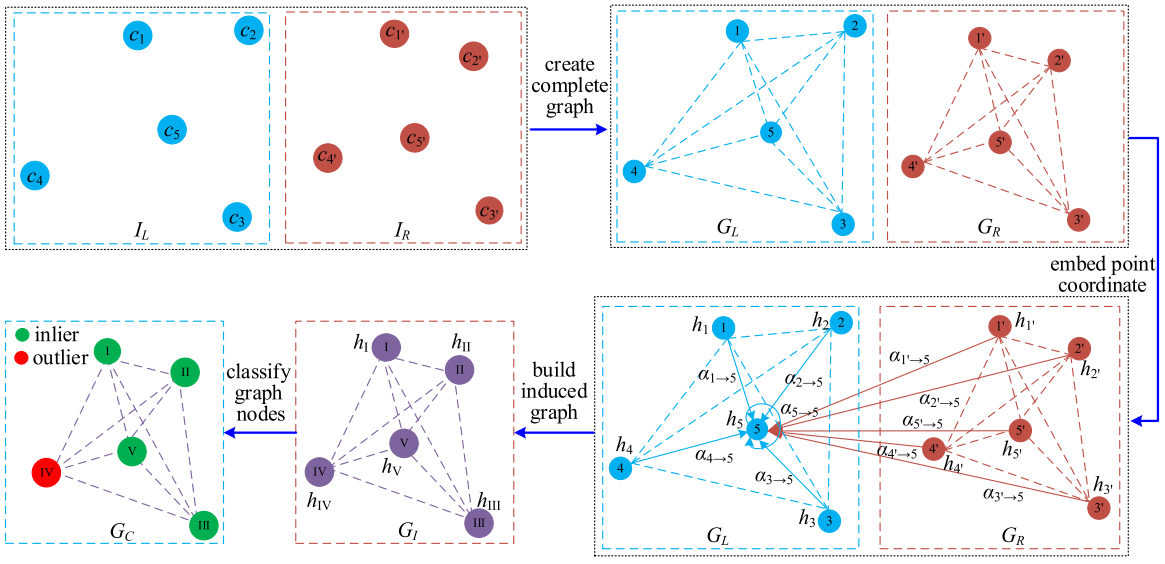


FIGURE 1. Workflow of our proposed method (CE-Net). I_L and I_R are the left and right images; c_i and $c_{i'}$ is a matched point pair; G_L and G_R are two complete graphs constructed by the matched points located in the left and right images; i and i' are the graph nodes; h_i and $h_{i'}$ represent node descriptors; $\alpha_{j\to i}$ is the directed edge weight of edge $j \to i$; G_I is the induced graph; and G_C is the node-classified graph.

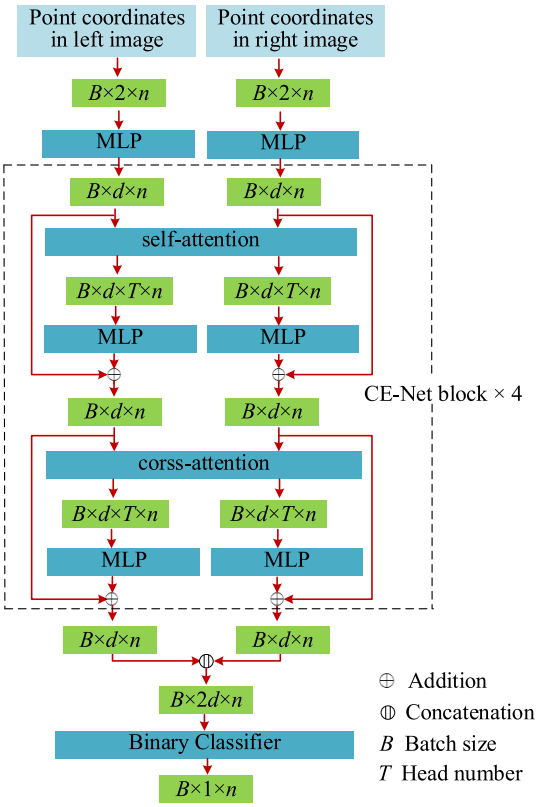


FIGURE 2. Network architecture of CE-Net, followed by a binary classifier using in mismatching removal. The input of the network is labeled point pairs, and the output is the predicated labels.

layer, 2) self-attention layer, 3) multi-head attention, 4) cross-attention layer, 5) attention in batched graphs; 4. A node classifier is used to separate graph nodes (Section II.D); 5. Loss function is presented (Section II.E).

A. NETWORK ARCHITECTURE

As shown in Fig. 2, the proposed network is composed of two main parts: (1) a matching point coordinate embedding net that gives every graph node a descriptor, and (2) a node classification net that outputs a predication of the graph nodes indicating inliers and outliers. In the following section, we will firstly introduce the concept of graph node embedding, then present a detailed description of coordinate embedding, namely how to use attention mechanism to embed point coordinates into node descriptors, and finally how to use a binary classifier to separate graph nodes.

B. NODE EMBEDDING

The node embedding paradigm in GNN can be defined as an iteratively updating step $(t+1)$ for the computation of node-wise descriptor and edge-wise weight [31]:

$$\alpha_{j\to i}^{(t+1)} = \phi(h_i^{(t)}, h_j^{(t)}, \alpha_{j\to i}^{(t)}) \tag{1}$$

$$h_i^{(t+1)} = \psi(h_i^{(t)}, \rho(\alpha_{j\to i}^{(t+1)})) \tag{2}$$

where i is the destination node, j is a neighboring node of i; $\alpha_{j\to i}$ is the directed edge weight of edge $j \to i$; h_i and h_j are the descriptors of node i and j; ϕ is an information aggregation function defined on each edge to generate a weight by combining the initial edge weight with the descriptors of its incident nodes; ψ is an update function defined on each node to update the node descriptor by aggregating its incoming edge weight using the reduce function ρ (namely aggregator). For the information aggregation function ϕ, some deep GNNs, such as GAT, GCN, GraphSAGE, are the commonly chosen; for reduce function ρ, mean and summation functions are among the chosen list; and update function ψ is usually a weighted summation function.

C. COORDINATE EMBEDDING

The point coordinates are 2D vectors that can be viewed as projections from higher vector space. Since projecting to lower vector space may lose the expressivity, we aim to learn the back-projective matrices which can recover the 2D vectors' geometric information in a way of back-projecting these vectors to a higher dimensional space. That is to say, by using these learned back-projective matrices we can embed coordinate geometric information into higher dimensional vectors. The proposed coordinate embedding algorithm follows the node embedding paradigm, meanwhile, it considers the characteristics of image matching point coordinates. That is, *matched points have similar neighbors while neighbors of unmatched points are far different*. Thus, on the one hand, we learn node descriptors containing the intra-graph coordinate information of node neighbors and these descriptors determine whether a matched point pair is an inlier or not; on the other hand, we embed inter-graph coordinate information into the descriptors to improve the distinctiveness, which can be easily separated by a classifier.

Technologically, we use self-attention and cross-attention to embed intra-graph and inter-graph coordinate information respectively. Here attention mechanism [28] performs as a double check for inliers and outliers, therefore it increases the accuracy and recall of the proposed CE-Net. Specifically, the attention, which depends on the learnable parameters, can effectively prevent the generated node descriptors from being contaminated by outliers. In the following section, we concentrate on attention mechanism and give a detailed explanation of the network architecture (shown in Fig. 2).

1) MULTILAYER PERCEPTRON (MLP) LAYER

The initial matched points are in $\mathbf{R}^2$ space, to improve their expressivity, MLP layers are applied to expand the matched points to $\mathbf{R}^d$ space [32].

2) SELF-ATTENTION LAYER

Let h_i be the descriptor of node i in G_L, W_H be a learnable matrix, and $q_i = W_H h_i$, $K = [k_1\ k_2\ \dots\ k_j] = W_H[h_1\ h_2\ \dots\ h_j]$, where $h_1\ h_2\ \dots\ h_j$ are the node descriptors of i's neighbors (note: node i is also one of i's neighbors) in G_L. Then we can use self-attention mechanism to generate

directed edge weights [25], [28]:

$$\underset{j\in N_i}{\alpha_{j\to i}} = \text{softmax}\left(\frac{q_i^T K}{\sqrt{d}}\right) = \text{softmax}\left[\frac{q_i^T k_1}{\sqrt{d}} \quad \cdots \quad \frac{q_i^T k_j}{\sqrt{d}}\right] \tag{3}$$

where N_i is the neighboring node set of node i, $\underset{j\in N_i}{\alpha_{j\to i}}$ are the directed edge weights (from i's neighbors to i), $\sqrt{d}$ is the scaling factor and d is the dimension of h_i. Finally, we can use the directed edge weights and the neighboring node descriptors to generate an updated descriptor of node i [25], [28]:

$$h_i = \sum_{j\in N_i} \alpha_{j\to i}\, q_j = \sum_{j\in N_i} \alpha_{j\to i}\, W_H h_j \tag{4}$$

Here we can give a concise explanation of the directed edge weight $\alpha_{j\to i}$. From Equation (3), we can get $\underset{j\in N_i}{\alpha_{j\to i}} = \text{softmax}_j(q_i^T k_j/\sqrt{d})$ (note $q_i = W_H h_i$, $k_j = W_H h_j$), that is to say, $\alpha_{j\to i}$ measures the "correlation" of node i and j in a W_H projected linear space which is originally spanned by h_i and h_j. The "correlation" can also be expressed as "*attention that node i gets from node j*," and how much attention there depends on the learnable parameter W_H which is essentially determined by the learning task. It is the learned "attention" that determines how to get useful information from "good neighbors" and get rid of noise from "bad neighbors."

To make self-attention focus on as many parts of the graph as possible, the proposed CE-Net is built on an 8-layer graph attention network, and to prevent from gradient vanishing, the network backbone is a residual connection network (shown in Fig. 2).

3) MULTI-HEAD ATTENTION

Analogous to the convolution channels in CNN, multi-head attention is also applied in the CE-Net. Multi-head attention allows the network to jointly attend to information from different representation subspaces at different positions [28], namely, multi-head attention can learn more information from point coordinates because different heads can learn different geometric invariance (e.g., distance invariance, angel invariance, and so on). As illustrated by Fig. 3, there is a multi-head attention with 3 heads by node 5 on its five neighbors (node 5 is its own neighbor), and we can take one head t as an example to deduce how multi-attention takes effect.

Let W_H^t be the learnable matrix corresponding to head t, then node descriptor h_i will be transformed into a new space by a linear projection of W_H^t:

$$q_i^t = W_H^t h_i \tag{5}$$

Similar to the single head attention described in Equation (3), we can obtain an attention (i.e., directed edge weight) in head t from node j to i:

$$\underset{j\in N_i}{\alpha_{j\to i}^t} = \text{softmax}_j\left[\frac{(q_i^t)^T k_j^t}{\sqrt{d}}\right] \tag{6}$$

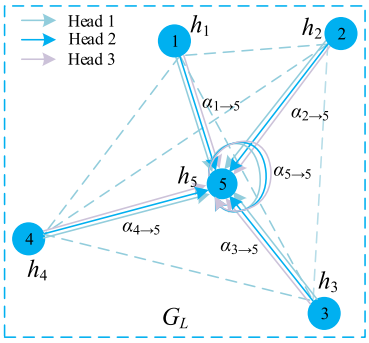


FIGURE 3. An illustration of multi-head attention, different arrow colors denote independent attention computational heads.

Aggregating neighboring information of node i, we can get head t attention:

$$h_i^t = \sum_{j\in N_i} \alpha_{j\to i}^t\, q_j^t = \sum_{j\in N_i} \alpha_{j\to i}^t\, W_H^t h_j \tag{7}$$

Concatenating all heads attention results in the final embedding of point coordinates [25]:

$$h_i = \Big\Vert_{t=1}^{T} h_i^t \tag{8}$$

where $||$ denotes vector concatenation in a column order. It can be easily verified that if the input dimension of node descriptor is d, and the number of attention heads is T, then the output dimension of multi-head attention is $d \times T$. Specially, if we want the output dimension to be d', then some additional layers, such as MLP, should be used to project the node descriptors to $\mathbf{R}^{d'}$ space [32].

4) CROSS-ATTENTION LAYER

The matching nodes of the two graphs are constructed by matching points in the image pair, thus the two matched nodes are conjugated and their neighbors have similar geometric distribution. Inter-graph attention, namely cross-attention, can be applied for gathering geometric information across two graphs. Analogous to multi-head self-attention aggregating intra-graph coordinate information (shown in Equation (7)), a multi-head cross-attention (as illustrated in Fig. 4) can generate a node descriptor by gathering the neighboring node descriptors:

$$h_i^t = \sum_{j'\in N_{i'}} \alpha_{j'\to i}^t\, W_H^t h_{j'} \tag{9}$$

where i' is the matched node of i located in graph G_R; $N_{i'}$ is the neighboring nodes of i' in graph G_R; j' is one of neighbors of i'; $\alpha_{j'\to i}^t$ is the cross-attention (i.e., directed edge weight) in head t from node j' to i.

Then, concatenating all the descriptors by Equation (8) generates final result. CE-Net alternately uses cross-attention and self-attention, thus giving an accurate and distinctive description of nodes both in graph G_L and G_R.

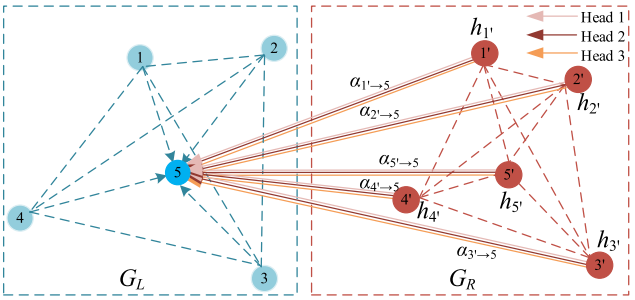


FIGURE 4. An illustration of multi-head cross-attention (3 heads) by node 5 on the neighbors of its matched node 5′.

5) ATTENTION IN BATCHED GRAPHS

Equation (8) gives the computing of a single node descriptor h_i by multi-head attentions, and it can be seen clearly that h_i is a $d \times T$ matrix. If a graph has n nodes, then stacking n node descriptors obtains a $d \times T \times n$ tensor. If we batch B graphs together in training, the output of multi-head attention is a $B \times d \times T \times n$ tensor. The detailed process can be illustrated in Algorithm 1, using Python syntax and PyTorch deep learning library.

Algorithm 1 Coordinate Embedding by Using GAT

Input: C_L, C_R (C_L and C_R are coordinates of matched points in the left and right image, respectively), L_s (sequences of attention layer notations)
Output: updated node descriptors of G_L and G_R

def *attention* (Q, K, H):
 $s =$ **torch.einsum** (*'BdTn, BdTn->BTnn,'* Q, K)/$d^{**}.5$
 $alpha =$ **torch.nn.functional.softmax** (s) # alpha is $B \times d \times T \times n$ since the graph is complete and the head number is T
 return torch.einsum (*'BTnn, BdTn->BdTn,' alpha*, H)
$H_L, H_R =$ **torch.nn.Linear** (C_L), **torch.nn.Linear** (C_R) # using MLP to improve the expressivity
for L **in** L_s:
 if $L ==$ *'cross'*:
  $S_0, S_1 = H_R, H_L$
 if $L ==$ *'self'*:
  $S_0, S_1 = H_L, H_R$
 $d_0, d_1 =$ *attention* (H_L, S_0, S_0), *attention* (H_R, S_1, S_1)
 $d_0, d_1 =$ **torch.nn.Linear** (d_0), **torch.nn.Linear** (d_1)
 $H_L, H_R = (S_0 + d_0), (d_1 + S_1)$
return H_L, H_R

D. NODE CLASSIFIER

CE-Net gives an accurate description of nodes in the two graphs, while before a classifier is applied in node classification, the two graphs should be fused into one graph. As shown in Fig. 1, we induce a graph G_I with an identical structure to G_L and G_R [2], and its node descriptors are the concatenation of the matched node descriptors. Formally, if i and i' are a pair of matched graph nodes coming from G_L and G_R respectively, then the two matched nodes induce a node I of G_I, and node I has a descriptor of $\|(h_i, h_{i'})$. If i and j are connected in G_L (i' and j' are connected in G_R), I and J are connected in G_I. Once the induced graph is constructed, we use a binary node classifier to classify the graph nodes, i.e., partition the nodes into two sets, with one only containing outliers and another only containing inliers. The architecture of the binary classifier is as below.

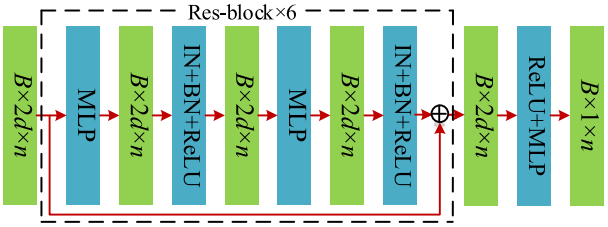


FIGURE 5. The architecture of the binary classifier. IN and BN are short for instance normalization [33] and batch normalization [34] respectively, ReLU is rectified linear unit [35].

By using the binary classifier shown in Fig. 5, every node descriptor is reduced to a number: if a number is smaller than 0, then its corresponding node which represents a putative point correspondence indicates an outlier, otherwise, it indicates an inlier.

E. LOSS FUNCTION

The proposed network is trained in a supervised manner, and binary cross-entropy loss function is used to calculate the deviation between output predication of node m_i and the ground truth label l_i [41]:

$$L(\omega) = \frac{1}{n}\sum_{i=1}^{n} p_i E(l_i, F(m_i, \omega)) \tag{10}$$

where ω is the learnable parameters; p_i is an adaptive weight to balance the number of inliers and outliers; F is the proposed network, and $F(m_i, \omega)$ outputs the predication of m_i; E is a binary cross entropy function. As is shown next, the simple while effective way can achieve even better results.

III. EXPERIMENTS

To demonstrate the effectiveness of CE-Net, we firstly implement the algorithm through PyTorch deep learning library [36], and then train and test the network in a simulated matching point set. Finally, we apply CE-Net and learned parameters in practical mismatching removal tasks, and give comparison results of the current state-of-the-art methods such as RANSAC, LMP, GMS, LFGC, NM-Net and PointNet.

A. LEARNING DATA

The learning data is derived from aerial triangulation tasks. The points are first extracted and matched by SIFT [1], and then these matched points are used to compute external orientation elements of images by a least square method. To guarantee all the matched points are inliers, these matched

point pairs are filtered by a back projection threshold (0.25 pixel) using external orientation elements. We view these inliers as a part of the learning dataset, and simulate outliers by considering geometric distribution pattern of outliers which arise from feature-based matching.

In the feature-based matching results, most matching outliers lie near clusters of matching points. Clustered matching points mean poor image textures, and the matching results are prone to be contaminated by noise, discontinuity and occlusion [3]. To simulate the actual situation as much as possible, we first randomly choose a number (0.35-0.60) as the outlier rate, and then divide outliers into two categories, one near the matching point clusters (clustered outliers for short), and another uniformly distributed (uniform outliers for short).

Algorithm 2 Simulating Training Data (Python Style)

Input: a pair of error free matched point set P_L and P_R, image area IA
Output: training data sets C_L and C_R

$C_L, C_R = P_L, P_R$
$n = length\ (P_L)$ # number of matching point pairs
$n\_to = random\ (0.35, 0.60) * n$ # number of total outliers
$n\_co = random\ (0.60, 0.75) * n\_to$ # number of clustered outliers
$n\_uo = n\_to$ - n_co # number of uniform outliers
$k = random\ (15, 30)$ # number of clusters to be partitioned
$Cs = Kmeans\ (P_L, k)$ # clusters partitioned by K-means algorithm
$C_S, A_S = filter\ (Cs, 0.02 * IA)$ # filtering clusters with a threshold of MCC area of $0.02 * IA$, and returning the filtered clusters and their corresponding MCC areas
$T_A = sum\ (A_S)$ # summation of the MCC areas
for c_l, a **in** $zip\ (Cs, A_S)$:
  $s\_n = (a/T_A) * n\_co$ # number of simulated outliers determined by MCC area
  $c_1 = gen\_uni\_outliers\ (s\_n, c\_l)$ # generating uniform outliers within MCC of c_l
  $c\_r = get\_matched\_points\ (c\_l, P_R)$ # getting matched cluster of c_l in P_R
  $c_2 = gen\_uni\_outliers\ (s\_n, c\_r)$ # generating uniform outliers within MCC of c_r
  $C_L.append\ (c_1)$
  $C_R.append\ (c_2)$
$c_1, c_2 = gen\_uni\_outliers\ (n\_uo, P_L), gen\_uni\_outliers\ (n\_uo, P_R)$
$C_L.append\ (c_1)$
$C_R.append\ (c_2)$

To add clustered outliers, we first randomly choose a number as the rate of clustered outliers (0.60-0.75), and then use the K-means algorithm [37] to detect 15-30 matching clusters (the number of matching clusters is also randomly chosen). For every matching cluster, we calculate its minimum circumscribed circle (abbreviate to MCC) and the circle's area, if the area is smaller than 1/50 of the image area, we add evenly distributed outliers within the circle (the number of outliers is proportional to the area of the circle).

For the uniform outliers, we simply add outliers uniformly distributed within the image pairs. The simulating algorithm is presented in Algorithm 2.

To enhance the generalization ability of neural networks and avoid overfitting, we collect aerial image matching results of 38 surveying areas, more than 40 thousand stereo images and about 8 million matching points. These images are captured at different angles, and contain common topographies such as mountains, hills, buildings, rivers, farmlands and so on (details are shown in Table 1). Then the learning dataset is shuffled and divided into three irrelevant groups (the division ratio is 8:1:1), that is, training dataset, validating dataset and testing dataset. We use training dataset to train the learnable parameters of the network, use validating dataset to adjust hyper parameters such as the choice of optimizer, learning rate, training batch size, and so on, and use the testing dataset to verify the effectiveness of the proposed CE-Net.

TABLE 1. Learning dataset classification.

Classification by ways of collecting images			
No.	Collecting way	Number of image pairs	Number of point pairs
1	Oblique	8,255	1,651,000
2	Vertical	31,003	6,200,600
Classification by image topographies			
No.	Main image contents	Number of image pairs	Number of point pairs
1	Mountains	1,507	301,400
2	Hills	2,945	589,000
3	Buildings	12,777	2,555,400
4	Rivers & Lakes	4,926	985,200
5	Farmlands	17,103	3,420,600

B. TRAINING AND TESTING

For training parameters of the proposed CE-Net, the training optimizer is Adam [38], the learning scheduler is a cosine annealing schedule. In the cosine annealing schedule, the initial learning rate is 0.001, the warm restart period is 60, and the minimum learning rate is set to 10^{-5}. The training batch size is set to be 32, and the training epoch is 60. The head number of the attention is 2, and the number of GAT layers is 4, and parameter d used in MLP layer is 64. At the final epoch of the training, the learned model is saved. Then, the model is loaded in the network and tested in the testing dataset. For the compared algorithms, the parameters settings are shown in Table 2.

We adopt three commonly used criteria to verify the effectiveness of the proposed method:

$$\begin{aligned} p &= cp/n \\ ri &= ci/ni \\ ro &= co/no \end{aligned} \tag{11}$$

TABLE 2. Details of the comparison method settings.

Method	Source code website	Parameter settings
RANSAC	https://github.com/opencv/opencv	Fundamental matrix, back projection error: 1 pixel.
LMP	https://github.com/jiayi-ma/LPM	λ_1=λ_2=0.3, N_1=N_2=4
GMS	https://github.com/JiawangBian/GMS-Feature-Matcher	Grid size: 20×20, number of neighbors: 9
PointNet	https://github.com/fxia22/pointnet.pytorch	As in [20]
LFGC	https://github.com/vcg-uvic/learned-correspondence-release	As in [21]
NM-Net	https://github.com/sailor-z/NM-Net	As in [23]

where p is the precision, cp is the number of correct predications of inliers and outliers (note: if the label of a point pair is an inlier, and the predication output is greater than 0, then we count this predication as a correct one. Likewise, if the output predication of an outlier is less than 0, then the predication is also a correct one), n is the number of total point pairs; ri is the recall of the inlier, ci is the number of correct predications of inliers, ni is the number of total inliers; ro is the recall of outliers, co is the number of correct predications of outliers and no is the number of total outliers. Precision is used to evaluate the overall effectiveness, ri and ro are used to evaluate the recalling ability of inliers and outliers, respectively. The comparison results are listed in Table 3.

As shown in Table 3, apart from RANSAC, deep learning-based methods generally outperform hand-crafted methods, and our proposed CE-Net ranks the first in all the three criteria.

RANSAC utilizes random sampling technology to generate the minimal samples and find the maximal consensus, and the epipolar constraint computed by RANSAC is a universal geometric constraint in two view geometry of a pinhole camera. Thus, RANSAC can distinguish most of the true correspondences from false ones and obtain the precision higher than 92%, inlier recall higher than 96% and outlier recall higher than 90%.

Both LPM and GMS basically belong to neighbor-supported methods, LPM finds neighbors by Euclidean distance while GMS by a predefined grid. Therefore, the two methods have similar results in precision (about 0.62) but different performances in inlier recall and outlier recall. GMS may confront with classification problem when processing images captured at different angles, as matched points sometimes are not in a same grid and cannot give supports to their neighbors, resulting in a lower inlier recall than LPM (about 0.38, the worst result among the compared methods). Moreover, LMP has the problem in differentiating mismatches with small displacements, leading to a lower outlier recall (about 0.42, the worst outlier recall in the experiments).

TABLE 3. Comparison results of CE-Net and RANSAC, LMP, GMS, PointNet, LFGC and NM-Net (the best result in each column is shown in green, and the worst in red).

Method	Precision	Inlier recall	Outlier recall
RANSAC	0.928	0.962	0.905
LMP	0.623	0.801	0.378
GMS	0.620	0.416	0.920
PointNet	0.877	0.879	0.874
LFGC	0.961	0.945	0.983
NM-Net	0.952	0.934	0.980
CE-Net	0.972	0.963	0.984

LFGC performs better than NM-Net though these two methods are both derived from PointNet (LFGC is about 1% higher than NM-Net in precision and inlier recall, and they are equal in outlier recall). LFGC tries to find the optimal parameters by minimizing both binary classification error and regression error of fundamental matrix, and it is a multi-task learning scheme. As fundamental matrix is suitable for describing geometric relationship of matched points of aerial images, LFGC has better performances. NM-Net has slightly worse results than LFGC though it applies some new measures to overcome the drawbacks of Point-Net, such as mining geometric information of neighboring points, and using a series of residual connected convolution layers to expand its receptive field.

Next, we randomly choose an image pair from the testing dataset, and visualize self-attention and cross-attention weights in several selected layers of CE-Net. As shown in Fig. 6, the opacity of edges between matching points is directly proportional to the self-attention edge weight $\alpha^t_{j\to i}$ in Equation (6) and the cross-attention edge weight $\alpha^t_{j'\to i}$ in in Equation (9).

CE-Net achieves the best results among all the compared algorithms (1% higher in precision and approximately 2% higher in inlier recall than that of LFGC) for the usage of multi-head attention in embedding point coordinates. Through observation of the visualization of attention patterns across layers (Fig. 6), we can find the reasons of these improvements as follows: (1) Attention mechanism tries to aggregate geometric information of "good neighbors," which can get rid of adverse impacts of "bad neighbors." As it can be seen from Fig. 6, regardless of a checked point is an inlier or not, most of its connected neighbors are in green color. This means most of its attentive neighbors are inliers, thus the generated descriptors avoid being contaminated by irrelevant outliers. Since attention mechanism is governed by the learnable projection matrices, these matrices can project descriptors of inliers to a vector space which leads to a highly similarity of a vector. On the contrary, descriptors of outliers are randomly distributed and cannot be aligned by any projection matrix. (2) Cross-attention enhances the expression of self-attention. As shown in Fig. 6, self-attention gets information within images, meanwhile, cross-attention

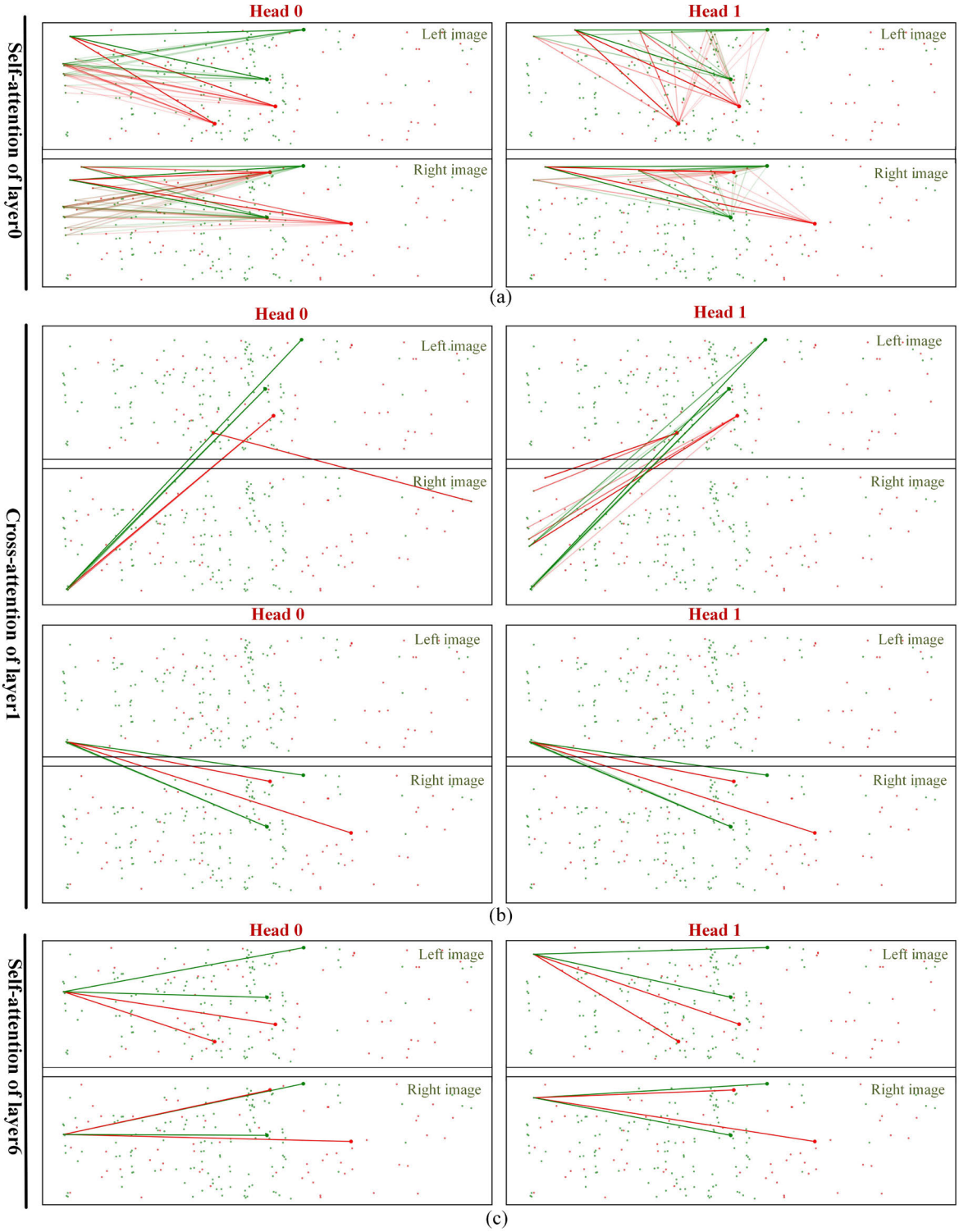


FIGURE 6. Visualization of attention patterns across layers. For an image pair randomly chosen from the testing dataset, we mark all the outliers with red dots and all inliers with green dots. We check the attention of four pairs of matched points (two are outliers marked with big red circles and others are inliers marked with big green circles) located in left and right images respectively. We visualize self-attention (within images) and cross-attention (cross images) weights of the selected layers, varying the edge opacity with $\alpha^t_{j\to i}$ and $\alpha^t_{j'\to i}$. (a)~(d) indicate Layer0(Self-attention), Layer1(Cross-attention), Layer6(Self-attention) and Layer7(Cross-attention), respectively.

obtains information across images and improves the distinctiveness of the node descriptors, which increases the outlier recall. (3) Different heads focus on different parts of images and form different geometric invariance, improving the performance of the proposed CE-Net. Taking self-attention of layer 0 as an example, head 0 looks at the left direction while head 1 look at the upper direction, and other heads also have this similar characteristic. (4) The attention initially attends

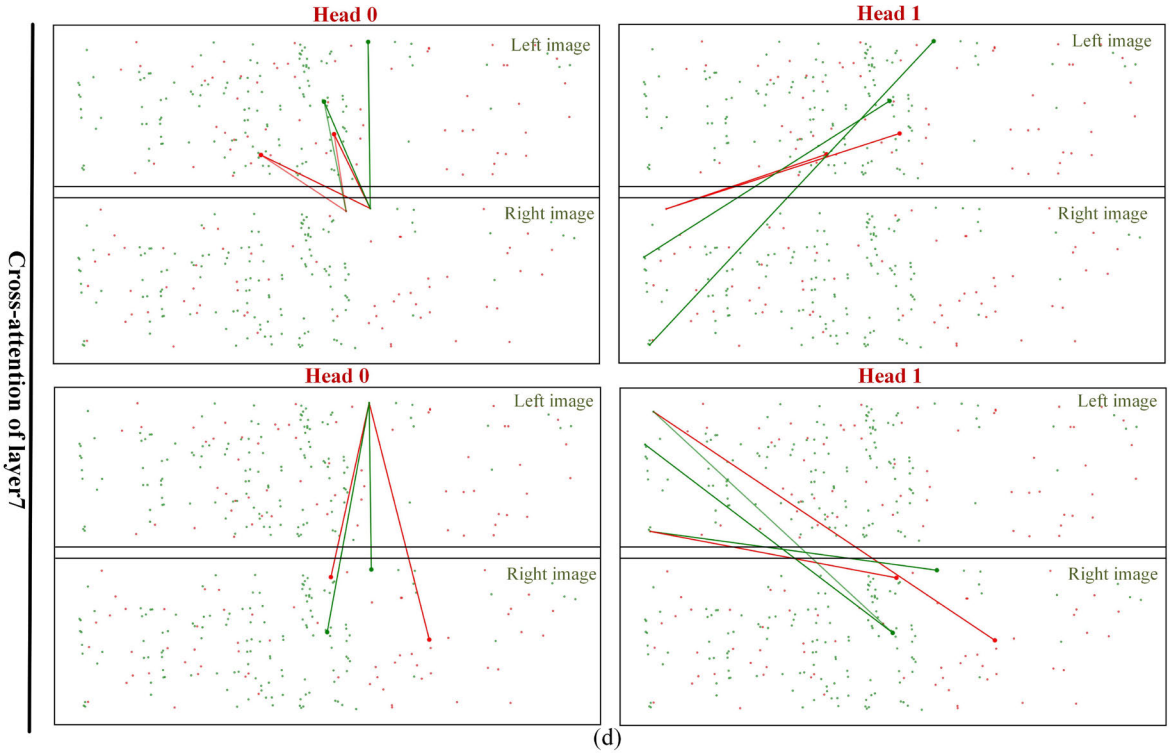


FIGURE 6. *(Continued.)* **Visualization of attention patterns across layers. For an image pair randomly chosen from the testing dataset, we mark all the outliers with red dots and all inliers with green dots. We check the attention of four pairs of matched points (two are outliers marked with big red circles and others are inliers marked with big green circles) located in left and right images respectively. We visualize self-attention (within images) and cross-attention (cross images) weights of the selected layers, varying the edge opacity with $\alpha^t_{j\to i}$ and $\alpha^t_{j'\to i}$. (a)∼(d) indicate Layer0(Self-attention), Layer1(Cross-attention), Layer6(Self-attention) and Layer7(Cross-attention), respectively.**

numerous points all over the image (as seen in self-attention of layer 0 and cross-attention in layer1), and gradually focuses on specific neighbors (as seen in self-attention of layer 6 and cross-attention in layer 7). This progressive process enables node descriptors to contain geometric information from coarse to fine, improving the precision of the mismatching removal. In conclusion, all these four aspects contribute to the improvements of CE-Net.

C. APPLICATIONS

We apply our proposed CE-Net algorithm in real mismatching removal tasks, the following is a demonstration of the detection results of one image pair.

Fig. 7 shows mismatching removal results of a randomly chosen pair of oblique images in one of our aerial photometric tasks. SIFT keypoints are first extracted in the image pairs, then these keypoints are matched by nearest neighbor searching without the distance ratio test [1]. As shown in Fig. 7, among these compared algorithms, the proposed method achieves precision of 0.960 (192/200), inlier recall of 0.920 (81/88), and outlier recall of 0.991 (111/112), which is similar to the results shown in testing dataset.

We also select aerial image matching results of four surveying areas as the real task dataset. As shown in Table 4, to make the dataset more representative, we choose aerial images that include two main collecting way of images (i.e., oblique and vertical), and four main contents (i.e., hills, farmlands, lake and urban buildings). The images are matched with SIFT algorithm without taking any mismatching removal measures apart from ratio test. There are more than 6,300 pairs of images and more than one million pairs of points in the real task dataset, and it is sufficient for evaluating the comparison methods.

TABLE 4. **Details of the real task dataset.**

Name	Collecting way	Main contents	Number of image pairs
BA	Vertical	Hills	2,242
MSC	Vertical	Farmlands	1,385
RIVER	Vertical	Lake	1,956
OBL_45	Oblique	Buildings	723

To demonstrate the effectiveness of the proposed CE-Net, we compare our method to RANSAC, LPM, LFGC, NM-Net. GMS and PointNet are not included, for there are no enough point pairs in about 10% of the filtered results to conduct the subsequent comparisons. For the filtered matching results, we compute their dispersion, point pairs numbers, and positional accuracy in every image pair. Dispersion is estimated as in [39] and the lower, the better. Positional accuracy is the

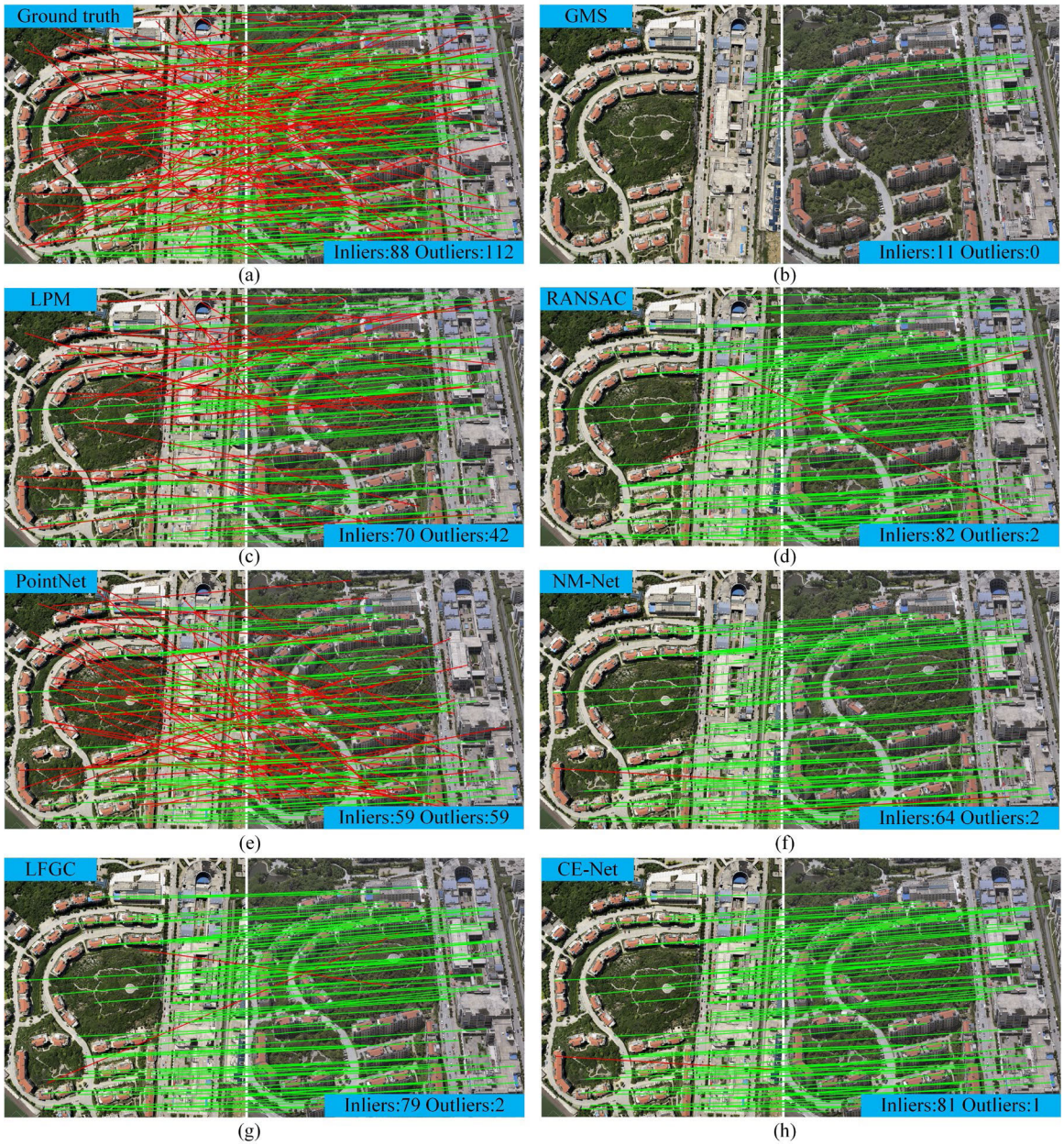


FIGURE 7. Demonstration of mismatching removal results. The matched point pairs linked by green lines are inliers, and point pairs linked by red lines are outliers. (a)∼(h) indicate Ground truth, RANSAC, LPM, GMS, PointNet, LFGC, NM-Net and CE-Net, respectively.

maximal epipolar distance [7] and computed as follows: first, we calculate a fundamental matrix by eight-point method [7]; then, we use the matrix to compute the epipolar distances of the matched points in the left and right images; finally, we adopt the maximal distances as the positional accuracy. We use boxplots to show the reliability and stability of these methods, and results are shown in Fig. 8.

It can be seen clearly that our proposed CE-Net and RANSAC dominate the mean positional accuracy (MPA), and all these methods output approximately equal dispersion and number of remaining point pairs. Specifically, CE-Net has the best MPA qualified with competitive dispersion and number of remaining point pairs in the first two experiments. As can be seen from Fig. 8(a) and 8(b), the MPA of CE-Net is 0.36 pixel and 0.38 pixel in the first two experiments respectively (note: the average pixel size of the experimental images is 0.005 millimeter), and the short interquartile range of the boxplot shows a small accuracy variation, which indicates the stability of the proposed CE-Net. In the last two experiments RANSAC performs slightly better than CE-Net. The reason is that there is a small number of matching point pairs in the third experimental images (main contents are lake), and these matched points are clustered in lake islands. Our proposed learning-based method therefore cannot aggregate sufficient

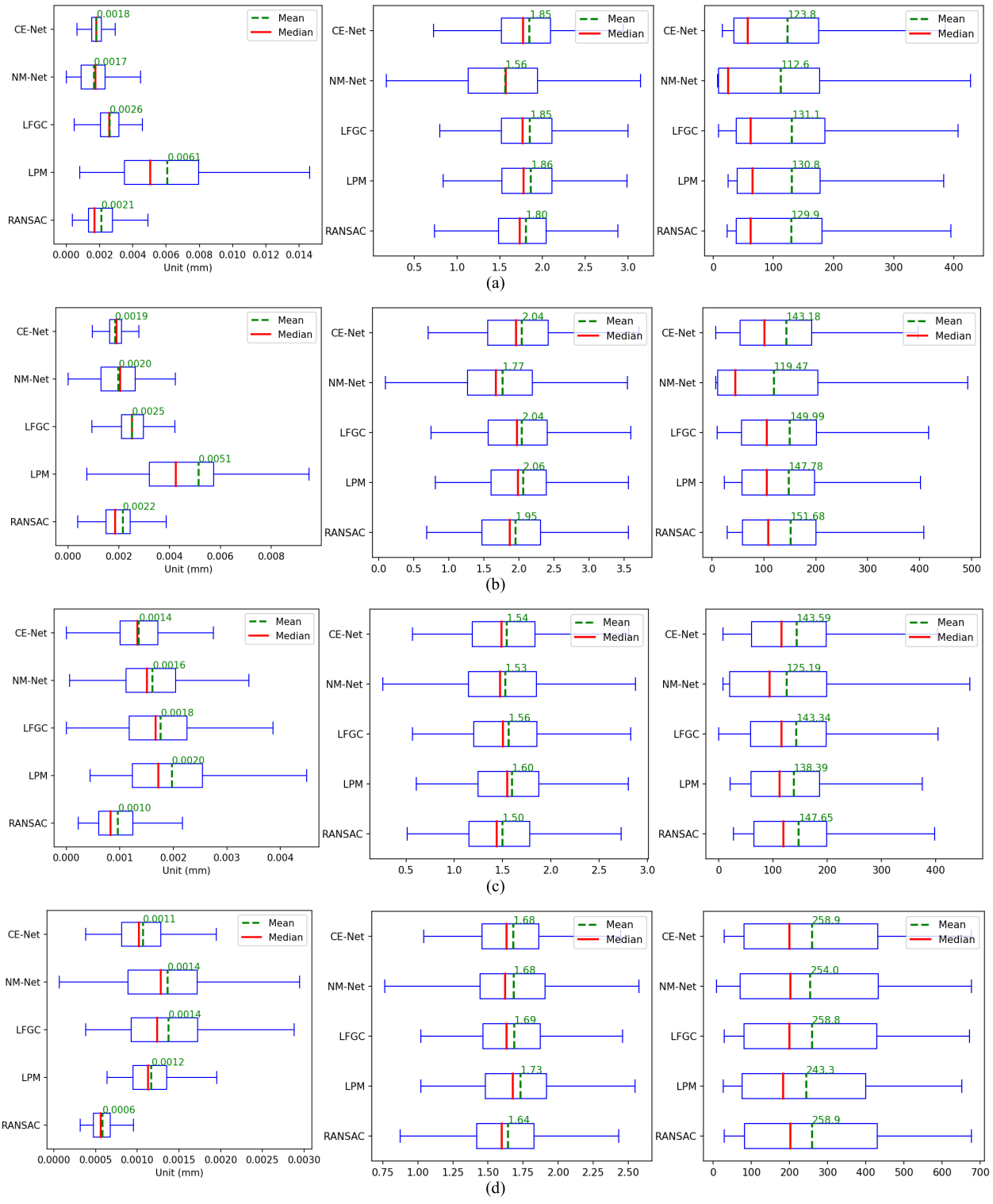


FIGURE 8. Performance of the mismatching removal algorithms in terms of positional accuracy, dispersion and number of remaining points. The left-most figure in every row shows positional accuracy, the middle shows dispersion, and the right-most shows number of remaining point pairs. The numbers in green are mean values of the corresponding items. The results are (a) BA, (b) MSC, (c) RIVER and (d) OBL_45.

geometric information to produce precise node descriptors, which leads to a slightly lower performance compared to RANSAC. Besides, the overall matching precision of SIFT is higher than 0.75 [40] in typical image pairs, which is very suitable for processing by RANSAC. In addition, the images used in the fourth experiment are oblique images, whereas,

most of the training point pairs are vertical images, which has a slightly negative influence on the generalization of CE-Net. Overall, the proposed method is still reliable as MPA of SIFT is about 0.5 pixel [40] and CE-Net promotes the MPA to about 0.2 pixel (0.001 millimeter).

D. ABLATION EXPERIMENTS

To clarify the performance of self-attention and cross-attention in mismatching removal, we use a self-attention-only network and a cross-attention-only network. In the coordinate embedding network (CE-Net), the two networks are modified by replacing the cross-attention layer with self-attention layer and replacing self-attention layer with cross-attention layer, respectively. And the two networks are tested by the same datasets in Section III.A and the same setups in Section III.B, the experimental results are listed below:

TABLE 5. Comparisons of different attention networks (the best result in each column is shown in green, and the worst in red).

Method	Precision	Inlier recall	Outlier recall
self-attention-only	0.969	0.962	0.973
cross-attention-only	0.964	0.950	0.979
CE-Net	0.972	0.963	0.984

It can be seen clearly that the alternate use of self-attention and cross-attention (i.e., our proposed CE-Net) obtains the best results. The self-attention-only network has the worst outlier recall, and the cross-attention-only network has the worst precision and inlier recall. The comparison results are in accord with the conclusions drawn in Section II.C, that is, the self-attention aggregates intra-graph information to determine whether a matched point pair is an inlier or not (as shown in Table 5, the self-attention-only network has a better inlier recall); the cross-attention aggregates inter-graph information to improve the distinctiveness of node descriptors, thus nodes can be more easily separated into inlier and outlier by a classifier, which increases the outlier recall (as shown in Table 5, the cross-attention-only network has a better outlier recall).

To examine the influence of diversity and amount of the training data on the performance of the proposed method, we collect a bigger training dataset, this dataset has more than 63,000 image pairs (i.e., about double amount of the original training dataset in Section III.A). Because almost all the image pairs rise from vertical photogrammetry, and the main contents are farmlands and hills, thus diversity of the dataset does not increase. By following the testing setups of Section III.B and using the same testing dataset in Section III.A, the result is as follows:

We also apply the newly trained network in real mismatching removal tasks of Section III.C, and the following is the result of positional accuracy:

It can be seen from Table 6 and Fig. 9 that more training data can really improve the overall precision and recalls of

TABLE 6. Experimental result of CE-Net with the new training dataset and the same testing dataset.

Method	Precision	Inlier recall	Outlier recall
CE-Net	0.989	0.991	0.986

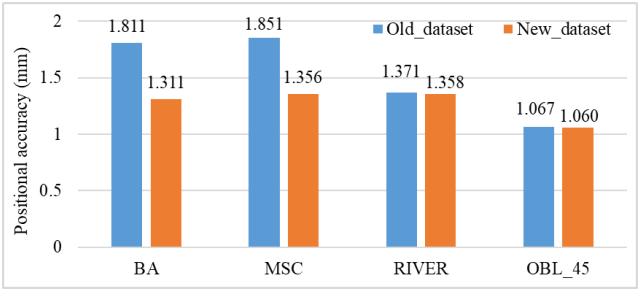


FIGURE 9. Comparisons of positional accuracy in real tasks.

CE-Net, and the training dataset diversity also has enormous influence on the performance. The new dataset contains more vertical images of hills and farmlands, therefore the positional accuracies of BA and MSC are further improved; while for RIVER and OBL_45, because of additional training data containing fewer lake images and oblique images, their positional accuracies have limited improvement.

IV. CONCLUSION

We have presented a deep learning based method for mismatching point pair removal. The method converts the mismatching problem into a node classification problem, and encodes both intra-graph and inter-graph information into node descriptors based on node embedding paradigm. Owing to self-attention and cross-attention used in the process of node embedding, the proposed method (CE-Net) outperforms RANSAC, LPM, GMS, PointNet, LFGC, NM-Net in precision, outlier recall, and inlier recall in the testing dataset. Furthermore, it also yields a better performance than the compared methods in real task dataset in terms of positional accuracy, dispersion, and number of remaining point pairs.

Since our method is a data-driven method, its performance is limited by the diversity and amount of the training data. Our training dataset is not perfectly balanced as the number of vertical images is about four times more than that of oblique images, which results in a small defect in mismatching removal of oblique images. In addition, attention-based networks sometimes have many layers that require a time and memory-consuming training. In the future, we will focus on collecting a more balanced dataset and training a shallow network without weakening the effectiveness.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[2] S. Chen, X. Yuan, W. Yuan, J. Niu, F. Xu, and Y. Zhang, "Matching multi-sensor remote sensing images via an affinity tensor," *Remote Sens.*, vol. 10, no. 7, p. 1104, Jul. 2018.

[3] X. Yuan, S. Chen, W. Yuan, and Y. Cai, "Poor textural image tie point matching via graph theory," *ISPRS J. Photogramm. Remote Sens.*, vol. 129, pp. 21–31, Jul. 2017.

[4] J. Ma, J. Jiang, H. Zhou, J. Zhao, and X. Guo, "Guided locality preserving feature matching for remote sensing image registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4435–4447, Aug. 2018.

[5] R. Ranftl and V. Koltun, "Deep fundamental matrix estimation," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 284–299.

[6] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Compute Vision*, 2nd ed. Cambridge, U.K.: CUP, 2004.

[8] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 138–156, Apr. 2000.

[9] B. J. Tordoff and D. W. Murray, "Guided-MLESAC: Faster image transform estimation by using matching priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1523–1535, Oct. 2005.

[10] O. Chum and J. Matas, "Matching with PROSAC-progressive sample consensus," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 220–226.

[11] C. Zhao, J. Yang, Y. Xiao, and Z. Cao, "Scalable multi-consistency feature matching with non-cooperative games," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 1258–1262.

[12] W. R. Crum, T. Hartkens, and D. L. G. Hill, "Non-rigid image registration: Theory and practice," *Brit. J. Radiol.*, vol. 77, no. 2, pp. S140–S153, Dec. 2004.

[13] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, "Locality preserving matching," *Int. J. Comput. Vis.*, vol. 127, no. 5, pp. 512–531, Sep. 2019.

[14] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M. M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 4181–4190.

[15] J. Ma, X. Jiang, J. Jiang, J. Zhao, and X. Guo, "LMR: Learning a two-class classifier for mismatch removal," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4045–4059, Mar. 2019.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[18] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Hamburg, Germany, Oct. 2015, pp. 681–687.

[19] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Apr. 2019.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 652–660.

[21] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 2666–2674.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[23] C. Zhao, Z. Cao, C. Li, X. Li, and J. Yang, "NM-Net: Mining reliable neighbors for robust feature correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 215–224.

[24] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Dec. 2009.

[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[27] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2017, *arXiv:1706.02216*.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[29] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," in *Social Network Data Analytics*. Boston, MA, USA: Springer, 2011, pp. 115–148.

[30] S. Cavallari, V. W. Zheng, H. Cai, K. C.-C. Chang, and E. Cambria, "Learning community embedding with community detection and node embedding on graphs," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, Nov. 2017, pp. 377–386.

[31] *Deep Graph Library (DGL)*. Accessed: Jul. 22, 2021. [Online]. Available: https://docs.dgl.ai/guide/message.html

[32] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 4937–4946.

[33] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*. [Online]. Available: https://arxiv.org/abs/1607.08022

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167

[35] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, Fort Lauderdale, FL, USA, Apr. 2011, pp. 315–323.

[36] PyTorch. Accessed: Jul. 22, 2021. [Online]. Available: https://pytorch.org/

[37] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

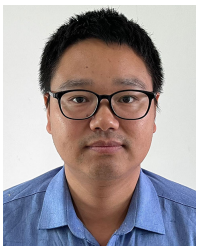
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[39] Q. Zhu, B. Wu, and Z.-X. Xu, "Seed point selection method for triangle constrained image matching propagation," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 2, pp. 207–211, Apr. 2006.

[40] A. Lingua, D. Marenchino, and F. Nex, "Performance analysis of the SIFT operator for automatic feature extraction and matching in photogrammetric applications," *Sensors*, vol. 9, pp. 3745–3766, May 2009.

[41] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT, 2012.

[42] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.



SHIYU CHEN was born in Xinyang, Henan, China, in 1986. He received the B.S. degree in remote sensing science and technology from Information Engineering University, Zhengzhou, Henan, in 2010, and the M.S. and D.E. degrees from the School of Remote Sensing and Engineering, Wuhan University, Wuhan, China, in 2014 and 2017, respectively. He is currently a Lecturer with Xinyang Normal University. His research interests include computer vision and precise agriculture, and devote himself to pest control via remote sensing methods.

JIQIANG NIU received the B.S. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2001 and 2010, respectively. He was a Postdoctoral Researcher with Henan University, from 2012 to 2017, and an Academic Visitor with the University of North Carolina at Charlotte, from 2015 to 2016. He has been working with Xinyang Normal University, since 2001, and became a Professor, in 2018. He is currently the Vice President of the Geographical Society of Henan Province. His research interests include spatial information mining, spatial data reliability theory and resource, and environmental assessment and planning.

CAILONG DENG received the B.S. degree in geomatics engineering from Wuhan University, Wuhan, China, in 2012, and the M.S. degree in environmental engineering from The First Institute of Oceanography, Qingdao, China, in 2015. He is currently pursuing the Ph.D. degree in photogrammetry and remote sensing with Wuhan University. His current research interests include image processing and multi-sensor integration and fusion.

YONG ZHANG received the Ph.D. degree from the School of Remote Sensing and Engineering, Wuhan University, Wuhan, China, in 2017. He has been working at VisionTech Research, since 2000, and became a Senior Research Scientist, in 2021. His research interests include computer vision, multispectral remote-sensing, and 3-D reconstruction from images.

FEIYAN CHEN received the B.S. degree from Xinyang Normal University, Xinyang, Henan, China, in 2005, and the Ph.D. degree in land resource management from Wuhan University, Wuhan, Hubei. She is currently an Associate Professor with Xinyang Normal University. Her research interests include resource and environmental assessment and planning and land use and regional sustainable development.

FENG XU was born in Xianfeng, Hubei, China, in 1978. She received the B.S. degree in land resource management from Wuhan University, Wuhan, Hubei, in 2001, and the M.S. degree from the State Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, in 2006. She is currently an Associate Professor with Xinyang Normal University. Her research interests include multi-scale representation of spatial data and its uncertainty.