IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 28, 2019, accepted December 17, 2019, date of publication December 23, 2019,
date of current version January 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961606

Monocular Depth Estimation Based on Multi-Scale Graph Convolution Networks

JUNWEI FU [1], JUN LIANG [1], AND ZIYANG WANG [2]
[1]State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China
[2]Hangzhou Finance and Investment Group, Hangzhou 310016, China

Corresponding author: Jun Liang (jliang@zju.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant U1664264 and Grant U1509203.

ABSTRACT Monocular depth estimation is a foundation task of three-dimensional (3D) reconstruction which is used to improve the accuracy of environment perception. Because of the simpler hardware requirement, it is more suitable than other multi-view methods. In this study, a new monocular depth estimation algorithm based on graph convolution network (GCN) is proposed. The pixel-wise depth relationship is introduced into conventional convolution neural network (CNN) to make up the disadvantage of processing non-Euclidian data. And the remaining depth topological graph information on the spatial latent variables are extracted based on a multi-scale reconstruction strategy. The final results on NYU-v2 depth dataset and KITTI depth dataset demonstrate that our algorithm improves the quality of monocular depth estimation, especially there are several little objects coexisting in the scenes.

INDEX TERMS Monocular depth estimation, reconstruction strategy, graph convolution network.

I. INTRODUCTION

In the field of image processing, the deep learning networks have achieved a great success in object classification and detection [1]. The main reason is that the features extracted by deep learning model are better than artificial features. In monocular depth estimation problem, the deep learning network also has a great advantage over the traditional image algorithm. Due to lack of disparity data, the traditional algorithms can not infer depth information directly. And this kind of task can be regarded as an ill posed problem [2]. At present, the main methods aim to find depth clues from images, and estimate the depth based on the depth clues. Therefore, the accuracy of depth estimation depends largely on the quality of depth clues.

The previous researches [3]–[5] had constructed a framework to estimate depth based on monocular camera. And it was composed of an encoder network, a decoder network and a refined network. In this framework, one image as input would be projected into the sparse feature space through the encode network. Those kind of sparse features represented depth clues. The decoder network projected the sparse featrues into the dense depth space by upsampling layers. And the refined network improved the coarse results from the decoder network. Recent evidences [6]–[9] suggest that a delicate loss function based on graphic mechanism could effectively improve the training results.

Most of the depth estimation algorithms use CNN to extract feature information, but they always ignore the charactistic of depth information. Because the distribution of depth map is unconsistent with RGB images. For example, each pixel value in the depth map is not only related to the neighboring pixel values, but also to other pixels of the same depth value. CNN will be limited by the receptive field of convolution kernel. In order to solve this problem, the dilation convolution layer is proposed in [10], which expands the convolution kernal to a scale, and fills the unoccupied area with zero. The purpose of this layer is to expand the receptive field of kernel with same size of parameters. In addition, it further reduces the information loss in deep networks than the pooling operation. In this way, the model keep a balance between expanding receptive field and maintaining image size. The other method improving the receptive field is to introduce attention mechanism. Xu, *et al.* [7] proposed an attention-based CRF. Li, *et al.* [11] proposed a channel-wise attention mechanism for diverse scenes. Through the attention mechanism, the learning ability of the model for local features can be enhanced.

Recenlty, Zeller's [12] work proposed a conception about motifs which represents the regularly appearing substructures in scene graphs. They proved that the performance of the

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

tasks such as scene detection, scene classification and prediction classification can be significantly improved by the relationship information of objects. Dai, *et al.* [13] proposed a deep relational network based on object detection, pair filtering and joint recogition, which improve the accuracy of visual recognization when the visual cues are ambiguous. The scene graph is a data structure proposed by [14], which is composed of object nodes and relationship edges. They extracts the scene graph from text based description to generate scene images. Those mentioned algorithms inspired us that the relationship of pixel-level depth clues have the potential of improving the quality of monocular depth estimation. And the challenge of monocular depth estimation can be divided into two problems: 1. How to extracted the depth association information in pixel-level. 2. How to process the non-Euclidean information from the former problem.

Our main contributions are as follow:

1. A reconstruction strategy of depth topological graph is proposed to extract non-Euclidean depth information.

2. The graph convolution network has been firstly employed in the monocular depth estimation task. And a reconstructed depth topological graph loss has is proposed to constrain the training processing.

3. A multi-scale graph nerual network module has been proposed to ifmprove the accuracy of depth map.

In this paper, Sect. II introduces the related works about monocular depth estimation. Sect. III describes the method proposed in this paper, introducing the reconstruction strategy of depth topological graph and the graph nerual network in detail. Sect. IV shows the experiments and corresponding analysises. Sect. V is our conclusion.

II. BACKGROUND

A. BACKBONE NETWORK

The structure of encoder networks is mainly derived from several popular network structures in the image recognition tasks. For example, residual neural networks (ResNet) [15], which introduces the residual learning into the model to deepen the layers of network and improve the accuracy. Densely connected convolutional networks (DenseNet) [16] is proposed to alleviate the problem of gradient vanishing and reduce the number of parameters. Recently, squeeze and excitation networks (SENet) [17] give an excellent idea about learning importance of each feature map channels. Those three networks all achieve sound performance on image recognition benchmarks. And we select the above ResNet and SENet for comparison in our experiments to project high-dimensional features into low-dimensional space with ImageNet initial parameters.

B. SKIP CONNECT

The degeneration of neural networks is one of the main difficulties of training deep networks. In [18], the researchers point out that only a small number of hidden units in each layer change their activation values for different inputs, and most hidden units make the same respond to different inputs. Thus, the rank of the whole weight matrix is not high. With the increase of network layers, the whole rank becomes lower after multiply operation. In order to enhance the sensitivity of features in deep layer, the features of shallow layers are directly passed to the deep layers. It is so called skip-connect, making up the feature lost in deep layer. It is employed in the CNN encoder network and decoder network to improve the representation of low-dimensional feature maps.

C. GRAPH NEURAL NETWORK

Graph neural network(GNN) was first proposed by [19]. The representation of the target node was learned through recurrent neural network(RNN) to propagate the neighboring information, which is further illustrated by [20]. But it consumes too much computation when updating the state of each node. Bruna, *et al.* [21] proposed convolution based on spectral graph theory, which directly performs convolution on the graph structure by aggregating the information of adjacent nodes. Deferrard, *et al.* [22] employed Chebyshev polynomials to fit convolution kernels and to reduce computational complexity, and this model was called ChebNet. [23] proposed the first-order approximation method to simplify ChebNet, which verifies that the graph neural network model can be used to deal with the semi-supervised classification of nodes in graph data in a fast and scalable way. In our work, the depth graph convolution network is designed based on the first-order ChebNet.

III. PROPOSED METHODS

Monocular depth estimation is an ill-posed problem, because it loses many depth clues when 3D objects are mapping into 2D plane. In previous researches, most of researchers focus on training a model based on a deep CNN with an elaborate loss function. However, there is no method to represent the depth relationship among nearest pixels. Inspired by Johnson's work [24], we regard the pixels as the targets. It is assumed that the location relationship of targets can be obtained when we are observing a scene, the distance of the targets can be inferred according to this kind of clue. And the greatest problem is how to process the non-Euclidean location relationships.

In this section, we firstly introduce the depth topological graph as the depth clue, and then the GNN is employed to process this kind of information insteading of CNN. The principle is to estimate the depth value by calculating the adjacency matrix and eigenvector in the depth map. The detail of our model is described through the following sections:

A. DEPTH GRAPH CONSTRUCTION STRATEGY

Generally speaking, any data can establish its corresponding topological graph in the normed space. For example, an image could be projected into a regular grid in the Euclidean space, but the relationships of all pixels are in the non-Euclidean domain. The main disadvantage is that CNN is unable to handle the data of non-Euclidean domain. In the early stage, the target node information can be learned from neighbor node information by recurrent neural networks [25]. This learning process is computationally expensive and can not be applied in the complex graphs.

The previous researches have been considered that the results of encoder network play an essential role in domain translation between image and depth. It aims to represent hidden latent into a low-dimensional vector space, but node topological graph and node information are applied into encoder network hardly. In order to introduce explicit relationship into the depth estimation network, we propose a GNN module, which introduces topological structure and node features into the network through graph convolution operation to enhance the feature representing of hidden layer.

Algorithm 1 Reconstruction Strategy of Depth Topological Graph

```
Input: Coarse depth: $d_{coarse}$
Output: Adjacent matrix: $A$
function ExtractGraph($d_{coarse}, R_{scale}, m, n$)
    $d_{pool} \leftarrow Pooling(d_{coarse})$
    $d_{noised} \leftarrow Noised(d_{pool}, R_{scale})$
    $\theta \leftarrow Interval(d_{pool}, m, n)$
    $ReconGraph(d_{noised}, \theta, m, n)$ :
    $(1)Set\{N_i, N_j\} \leftarrow Paired(d_{noised}, \theta, m, n)$
    $(2)A \leftarrow Dropout(Set\{N_i, N_j\})$
    return $A$
end function
```

The biggest challenge of monocular depth estimation based on GNN is the construction of depth topological graph, because there is no gound truth for supervised learning. A strategy is proposed to generate a depth topological graph from coarse depth map d_{coarse} which is obtained by the pre-trained model. The R_{scale} is the scale of Guass noise. And m, n are the width and height of scale. The summarization of this strategy is described as Algorithm.1, which is composed of four steps:

1. Down-sampling the predicted depth map from coarse to fine, operation is $Pooling(\bullet)$ and output is d_{pool};

2. Adding the Gauss noise to improve the robustness. The Operation is $Noised(\bullet)$ and the output is d_{noised};

3. Calculating the depth interval threshold θ by $Interval(\bullet)$;

4. Generating the graph by operation $ReconGraph(\bullet)$ which is composed two sub-steps: first step is obtained the adjcent nodes $Set\{N_i, N_j\}$ by the operation $Paired(\bullet)$, and the dropout operation $Dropout(\bullet)$ is employed in the second step.

Finally, the multi-scale depth topological graphs are obtained.

1) DOWN-SAMPLING OPERATION

In the first step of the depth graph construction, the depth reconstruction error will be brought into the depth graph inevitably based on the coarse depth map. For reducing the error and obtaining different scales, there are three pooling method, max pooling [26], stochastic pooling [27] and mean pooling, are applied in down-sampling operation.

The depth map are divided into several blocks with same shape. Those kind of blocks can be regarded as nodes in the graph, whose edge information is composed of the depth values and the location of blocks. The purpose of down-sampling operation is to obtain the depth maps for graph reconstruction.

Max pooling: for each channel (assuming there are N channels), the maximum value of the feature map of the channel is selected as the representative of the channel. An N-dimensional vector representation can be obtained. In Fig.1, the block depth value is selected by the biggest value of d1, d2, d3, d4.

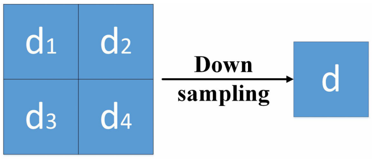


FIGURE 1. Down-sampling processing.

Stochastic pooling: the value of feature graph in a pooling window is normalized. It is selected according to the probability value of normalized feature graph. In other words, the probability of being selected with large element value is also high. In Fig.1, the probability of those four blocks are calculated depend on the depth value and the summation of depth value. The d value is chosen according to the probability.

Average pooling: all the pixel values of the feature map of the channel is averaged so that each channel gets a real value. In Fig.1, the d value is obtained by mean operator on d1, d2, d3, d4.

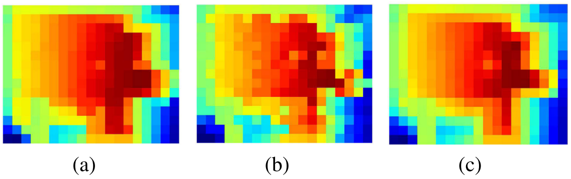

(a) (b) (c)

FIGURE 2. Depth map from different pooling method (a)Max pooling (b)Stochasitc pooling (c)Average pooling.

The following three pooling method provides a solution to get appropriate depth map in different scales, which is shown in Fig.2. According to the comparsion results of the experiment in Sect. IV-D(1), the max pooling method achieves the best performance. This method will be employed in the strategy of depth graph construction.

2) NOISE TOLERANT

Due to the reconstruction error of the coarse-grained depth map, the Gauss noise is proposed to be added into the coarse-grained depth map to avoid the reconstruction error learned by the model during training. It is well known that adding noise into the model in training can improve the generalization ability of the model [28]. Equation (1) shows the processing of ading noise. However, how to set the appropriate noise is still an unsolved problem. At present, Gaussian noise in any direction is added to the depth map. However, after this operation, there is negative values existing. It obviously does not conform to the physical meaning of depth map and the depth value needs to ensure non negativity. In addition, there is a upper bound constraints on depth values. When adding Gauss noise, we need to ensure that the depth values do not exceed the upper bound constraints, shown in (2,3).

$$d_{noised} = d_{pool} + R_{scale} \bullet Noise(0, 1) \tag{1}$$

$$s.t.\ \max(d_{noised}) \leq \max(d_{gt}) \tag{2}$$

$$\min(d_{noised}) \geq 0 \tag{3}$$

where R_{scale} is used to constrain the scale of Gauss noise $Noise(0, 1)$. As a hyperparameter, the optimal parameters need to be determined in Sect. IV-D(2). The depth of d_{noised} should be limited by max depth of ground truth d_{gt}. And the Fig.3 shows the result of adding the Gauss noise.

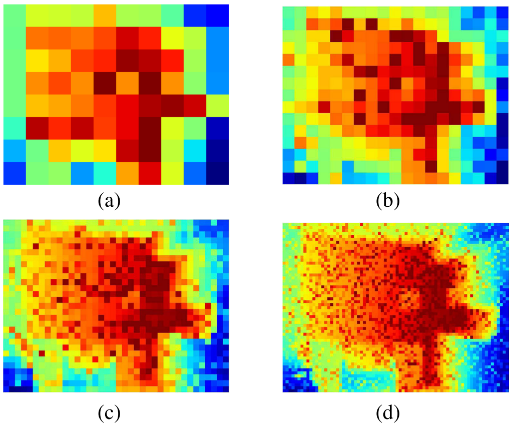

(a) (b) (c) (d)

FIGURE 3. Depth map in different scale based on max pooling with the Gauss noise and R_{scale} = 0.2.(a) 8 × 10 (b)15 × 19 (c)29 × 38 (d)57 × 76.

3) MULTI-SCALE INTERVAL THRESHOLD

Convolution neural network is composed of several convolution layers. For the same input, the features extracted by different convolution layers are different. With the deepening of network layers, the information structure of feature information is from high dimension to low dimension, and the expression ability is from shallow layer to deep layer. And the output of the last convolution layer have lost a lot of original information. Using this kind of feature has a good performance in classification tasks, but the effect is not ideal in information reconstruction tasks such as depth estimation. In order to improve the quality of reconstruction, we propose a multi-scale method according to the image pyramid technology, which combines the feature information of different scales with the depth topology graph. So that the features of different scales can retain the topology information.

Depending on multi-layer CNN, the multi-scale features can be obtained easily. But the depth topological graph with different scales still need to be determined. According to the perspective principle of camera, the depth information is featured with the characteristic that there are dense depth interval at near and sparse depth interval at far. Based on this principle, the depth relationship of all adjacent pixels can not be interpreted through a simple linear function. In addition, the non-linearity relationship of depth value is very obvious. In order to obtain multi-scale depth topological graph, a multi-scale interval threshold is proposed to determine the depth interval between nodes in different scales. And this parameter is also regarded as the radius of depth scope. If the node is one adjacent pixel of the central node and their depth value locates in the depth scope, they are considered to be connected. The threshold can be obtained by (4):

$$\theta_{(m,n)} = \frac{\max(\tilde{d}) - \min(\tilde{d})}{\min(m, n)} \tag{4}$$

where $\tilde{d}$ is the coarse-grained depth map. m is the rows number. n is the columns number.

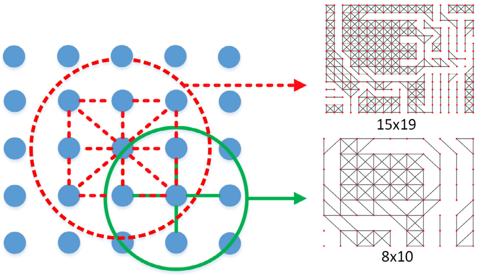


FIGURE 4. Depth graph based on prediction depth map with different thresholds.

As shown in Fig.4, the nodes are connected with similar depth value and position to construct the depth topological graph. The $\theta_{(m,n)}$ is depth interval threshold. The red circle indicates the bigger scale of depth topological graph. And the green circle indicates the smaller. Different values determine the sparsity of graph.

4) GRAPH DROPOUT

In order to avoid the over-fitting phenomenon of graph convolution module in training, dropout operation is used during constructing depth graphs at different scales. Different from its usual applications in the feature layer of the network [29], the dropout is applied in optimizing the value of adjacent matrix that represents depth topological graph. In training, we randomly drop edges from this matrix based on probability p using samples from a Bernoulli distribution, which is shown in (5,6):

$$A_{drop} = dropout(A) \tag{5}$$

$$dropout(k, p) = \begin{cases} p & if\ k = 1 \\ 1 - p & if\ k = 0 \end{cases} \tag{6}$$

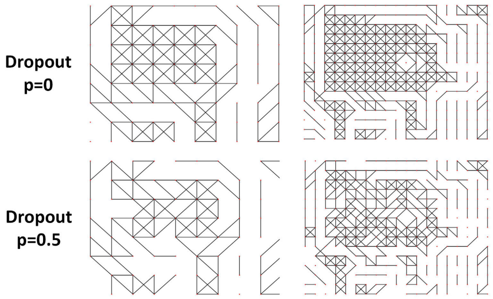


FIGURE 5. Multi-scale depth graph with dropout. The first and second rows show depth topological graphs with dropout probability $p = 0$ and $p = 0.5$.

where p is the value of graph edge probability, and k is the value of 0-1.

During testing, the dropout layer will be ignored. The depth topological graph with dropout rate is shown in Fig.5:

B. GRAPH NEURAL NETWORK

As shown in Fig.6, our model contains three sub-networks. There are two CNN networks and one GNN network. The upper CNN networks freeze parameters according to the pre-trained model which could infer a rough depth map. Depending on the depth topological graph strategy, the depth graph can be obtained. In our experiments, the parameters of CNN encoder networks are initialed by the ResNet or SENet based on ImageNet. The output of bottom CNN networks will be transformed to a new style processed by GNN.

1) GRAPH CONVOLUTION NETWORK

The goal of graph convolution network(GCN) is to learn a function of mapping the depth features to the low-dimensional vectors. GCN will preserve the topological architecture of depth graph $G = \{X, E\}$. Suppose that an depth graph is constructed by the method in Sect. III-A, and it is composed of X as x_i for every node feature and E as e_i for every link between nearest nodes. In our model, the feature X comes from the output of encoder layer and it is composed of N nodes with $(1 \times D)$ feature. The node relationship can be described as adjacency matrix A, which is a $N \times N$ sparse matrix.

The simplest neural network is Multi-Layer Perceptron(MLP) [30], which is composed of an input layer, a hidden layer and an output layer. This architecture is competent for fitting the non-linear data. The GCN layer has the similar structure of neural network, it can be written as a non-linear function and shown in (9):

$$H^{(l+1)} = f(H^{(l)}, A) \qquad (7)$$

where $H_{(0)} = X$, l is the number of layers, A is adjacency matrix, $f(\bullet)$ is the non-linear function shown in (10):

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)}) \qquad (8)$$

where A is an adjacency matrix with $N \times N$ and it represents the relationship of nodes. $H^{(l)}$ are node features of the $l - th$ layer with $N \times D$. When l is 0, $H^{(0)}$ is input X which is the output of encoder network. $W^{(l)}$ is a trainable weight matrix of the $l - th$ layer with $N \times D$(N is node number, D is the shape of node feature). And the result is a matrix with $N \times D$. $\sigma(\bullet)$ is a non-linear activation function like ReLU.

$AH^{(l)}$ means that summation of all the feature vector of neighboring nodes. The first problem is that the node itself is not considered in (10). Secondly, there is no available method to normalize the adjacent matrix completely change the scale of the feature vectors. The identity matrix is added to adjacency matrix for obtaining self-loop as $\tilde{A}$. And symmetric normalization is applied in adjacent matrix, which is shown in (11): $AH^{(l)}$ means that summation of feature vector of all nodes and remains two problems. The first problem is that the feature of node itself is not considered in $AH^{(l)}$, because the diagonal values of A matrix are all 0. Secondly, there is no available method to normalize the value of $AH^{(l)}$, because every feature summation of nodes has different degree. Too big degree or too small degree will lead to gradient vanishing or explosion. To address the first problem, the identity matrix is added to adjacency matrix for obtaining self-loop as $\tilde{A}$, which is shown in(7):

$$\tilde{A} = A + I \qquad (9)$$

where I is an unit diagonal matrix with $N \times N$.

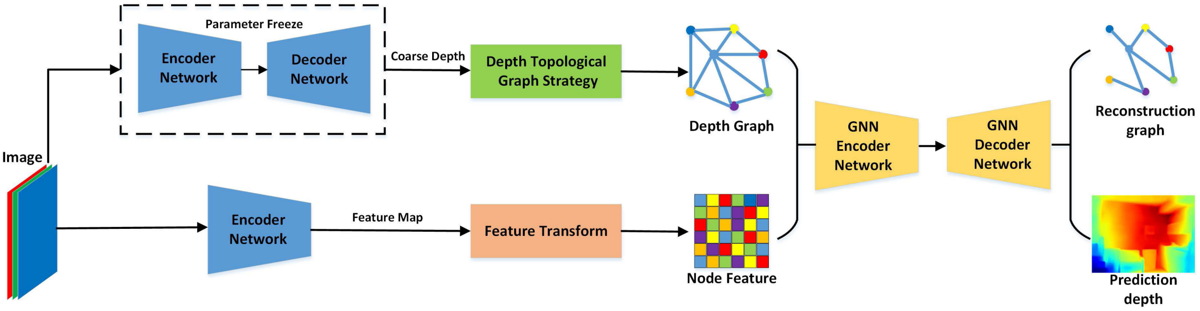


FIGURE 6. Network architecture. There are three sub-networks, depth graph generation network, feature network and depth graph convolution network.

And symmetric normalization is applied in $\tilde{A}H^{(l)}$ to avoid graident vanishing and explosion, which is shown in (8):

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{10}$$

where $\tilde{D}$ is a degree matrix of $\tilde{A}$, $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ means that the value of each row in the adjacency matrix is divided by the degree of the corresponding node.

The GCN layer is defined as following Algorithm.2:

Algorithm 2 Message Propagation of GCN

```
Input: $Feature\ H^{l}, Adjacent\ Matrix, A\ Layer\ l$
Output: $New\ feature: H^{l+1}$
function Propagation($X, A, l$)
    $\tilde{A} \leftarrow A + I$
    $H^{(l)} \leftarrow Linear(H^{(l)})$
    $H^{(l)} \leftarrow Normlization(H^{(l)})$
    $H_i^{(l+1)} \leftarrow \sum_{j \in N(i) \cup (i)} \frac{1}{\sqrt{\tilde{D}(i)}\sqrt{\tilde{D}(j)}}(W^{(l)}h_j^{(l)})$

    $H^{(l+1)} \leftarrow \sigma(H^{(l)})$
    return $H^{(l+1)}$
end function
```

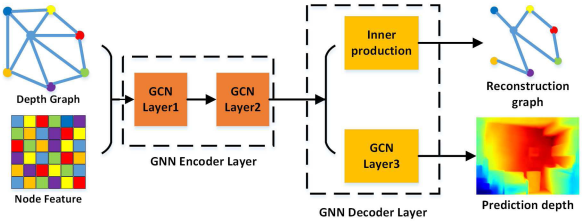


FIGURE 7. **The architecture of depth graph convolution network.**

2) NETWORK ARCHITECTURE

The graph model architecture is designed for keeping topological information within the latent variables. The model is composed of two parts, which are encoder network and decoder network.

As shown in Fig.7, the encoder network is composed of a two-layer GCN, and the methmatical expressions are shown in (11,12):

$$H^{(l)} = ReLU(GCN_1(X, A)) \tag{11}$$
$$H^{(l+1)} = ReLU(GCN_2(H^{(l)}, A)) \tag{12}$$

It has to be noted that ReLU activation function has ability of setting the output of some neurons as zero [31]. This operation reduces the interdependence of parameters and helps avoid over-fitting problem. Thus, it is employed in the first layer and second layer.

In the process of extracting depth relation feature in convolution module, only using depth map reconstruction error to constrain model parameters constraints on graph data also needs adding. Inspired by graph autoencoder(GAE) model [32], we add a depth graph reconstruction layer in the decoder network, and introduce a loss function to measure the error of depth topological graph. This decoder layer is composed of a simple inner production operation. The input of decoder layer is the lantent variable vector $H^{(l+1)}(N \times D)$ from encoder network. The reconstructed adjacency matrix and prediction depth can be calculated by (13,14):

$$\hat{A} = sigmoid(H^{(l+1)}H^{(l+1)^{T}}) \tag{13}$$
$$depth = GCN_3(H^{(l+1)}, A) \tag{14}$$

Since the adjacency matrix of depth topological graph is a sparse matrix, the difference is too sparse to be observed when using Euclidean distance to calculate similarity. The dice loss [33] is introduced to measure the similarity between reconstructed depth graph and ground truth depth graph. In addition, multi-scale reconstruction errors are obtained by averaging the dice losses at different scales, which is shown in (15,16):

$$Dice_s = \frac{2|\hat{A}_s \bigcap A_s|}{|\hat{A}_s| + |A_s|} \tag{15}$$
$$l_{dice} = 1 - \frac{1}{S}\sum_{s=1}^{S} Dice_s \tag{16}$$

where $|\hat{A}_s \bigcap A_s|$ represents the intersection of prediction adjacency matrix and ground truth. $|\hat{A}_s|$ and $|A_s|$ representing prediction adjacency matrix and ground truth. In (16), the coefficient 2 in the molecule is due to the existence of common elements between prediction adjacency matrix and ground truth in the denominator. When they become more similar, dice coefficients tend to be 1, and vice versa, to be 0.

Depth maps from the natural scene can be modeled roughly by a limited number of smooth surfaces and step edges according to [34]. Especially in the boundary of different objects, the depth values between show obvious step changes. Error depth value is often found on the discontinuous boundary of an object. Errors around such strong edges are well penalized by l_{grad}. The gradients of depth error is proposed in [35], as shown in (17):

$$l_{grad} = \frac{1}{n}\sum_{i=1}^{n}\left[(\nabla_x d_i)^2 + (\nabla_y d_i)^2\right] \tag{17}$$

where ∇_x is the gradient operation on x axis, and ∇_y is the gradient operation on y axis. d_i is the depth value of i-th pixel, n is the total number of pixels. Eigen's [36] proposed a scale-invariant mean squared error (SI-MSE) to measure the relationships between points in the scene, irrespective of the absolute global scale as shown in the (18):

$$l_{si} = \frac{1}{n}\sum_i d_i^2 - \frac{1}{n^2}(\sum_i d_i)^2 \tag{18}$$

where d_i is the depth value of i-th pixel, n is the total number of pixels.

Finally, we define the loss by (19) as follow:

$$L = l_{si} + l_{grad} + \lambda l_{dice} \tag{19}$$

In traditional image recognition tasks, using one single classifier is not good enough. The results of multiple classifiers are significantly improved by adding different weights which is called boosting [37]. Inspired by this method, we propose depth estimation structures at different scales. The depth results at different scales are normalized into one scale by bilinear interpolation method, and a set of weights are obtained by training. The structure of GNN module is shown in Fig.8.

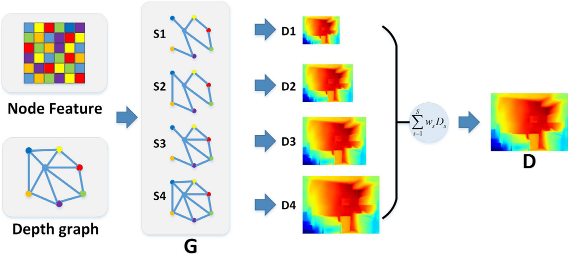


FIGURE 8. **Multi-scale GNN Module. S1 is 8 × 10, S2 is 15 × 19, S3 is 29 × 38, S4 is 57 × 76.**

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS

The NYU-v2 depth dataset [38] is composed of video sequences from a variety of indoor scenes as recorded by the RGB-D camera, Microsoft Kinect. The official splits for 464 scenes, 284 scenes for training and 180 scenes for testing. The training dataset contains 50688 pairs of RGB images and depths. Following [36], the original size (640 × 480) is downsampled to (320 × 240) by bilinear interpolation, and then crop their central parts to obtain images with (304 × 228) pixels. For testing, we use the same official subset of 654 samples.

To verified our method is capable in both indoor and outdoor scenes, the KITTI depth dateset [39] is introduced in our experiment. The ground truth depth information is sampled by Lidar sensor which is extensively different from samlped by rgbd camera sensor. The former provides us a sparse depth value and the later provides a dense depth value. In the surpvised training problem, dense depth value has more reliable than sparse one. Following eigen's spilt plan [36], the training set contains 22600 RGB images and sparse depth maps, then we crop the images into (1216 × 352) and resize into (512 × 256). The testing set contains 697 RGB images and sparse depth maps.

We implement our depth estimation networks based on the public deep learning platform pytorch. And two NVIDIA 1080Ti GPUs have been used for training. The feature encoder layer in the model is initialized by the pre-trained with the ImageNet dataset [40]. And the parameters of coarse depth networks are frozen according to [41]. The Adam optimizer with an initial learning rate of 0.01. The learning is reduced to one tenth of former value after 2000 iterations. The weight of l_{dice} are set as $\lambda = 1$. In all experiment, the batch size is set as 8.

B. EVALUATION METRICS

The following accuracy measurements are used to evaluated depth maps which are commonly employed in [6], [41]–[43], d is the ground truth depth, $\tilde{d}$ is the prediction of depth:

Root mean squared error(rms): $\sqrt{\frac{1}{n}\sum_i (d_i - \tilde{d}_i)^2}$

Average log10 error($log10$): $\frac{1}{n}\sum_i |\log_{10} d_i - \log_{10} \tilde{d}_i|$

Average abstruct relative error(rel): $\frac{1}{n}\sum_i \frac{|d_i - \tilde{d}_i|}{d_i}$

Accuracy with threshold(t): $\max(\frac{d_i}{\tilde{d}_i}, \frac{\tilde{d}_i}{d_i}) = \delta < t$.

C. PERFORMANCE COMPARISION

In this section, we compare our method with [2], [6], [36], [41]–[45], which represent the different typical methods in monocular depth estimation. Reference [42] proposed a multi-task model generating two predictions, depth and surface normal. [6], [43] introduces the continuous CRF into the CNN model. Eigen, *et al.* [36] is a multi-scale model composed of coarse network and refined network. Liana, *et al.* [41] is the encoder-decoder networks with up-sampling blocks. Godard's work [2] proposed a left-right consistency constrain in training to improve the quality of depth images.Atapour-Abarghouei, *et al.* [44] introduced the style transfer to train models on a large amount of generated data and enhance the robust in the different scenes. Fu, *et al.* [45] regards the depth regression problem as an ordinal regression which is based on a depth discretization strategy. This work achieved the first place in the Robust Vision Challenge 2018.

According to the Table.1, the results based on NYU-v2 depth dataset of our method are compared with those of existing methods in terms of the above-mentioned measurements. The results of our method are close to the Fu's work in $log10$ and abstruct relative error and outperform in $\delta < 1.25$, $\delta < 1.25^2$. Compared with the other four methods in rms, $log10$, rel, our model with GNN network can reduce the error in local depth estimation. The results in $\delta < 1.25$, $\delta < 1.25^2$ and $\delta < 1.25^3$ indecate that our model improve the accuracy of depth estimation.

Final predictions of five monocular depth estimation methods are shown in Fig.9, including the [6], [36], [41], [45], and our models based on SENet. The results of following algorithms have been listed in an ascending order. The predictions of [36] are blur in the boundary of objects. According to RMSE statistics of models in [36], [41], we can find that the prediction error of later one is much smaller. In [6], the global depth value achieves a desired accuracy, but the boundary of objects still suffer from distortion. Our method shows significant improvement, especially in terms of reconstructed edges of objects. We can find the obvious boundary of small structures, such as feet of chairs and bottles on the tables. Those details are easy to be ignored by other exsiting work.

According to the Table.1 and Fig.10, our model outperforms [2], [36], [43] in KITTI depth comparision. Because of introducing the synthetic depth training data, [44] is more

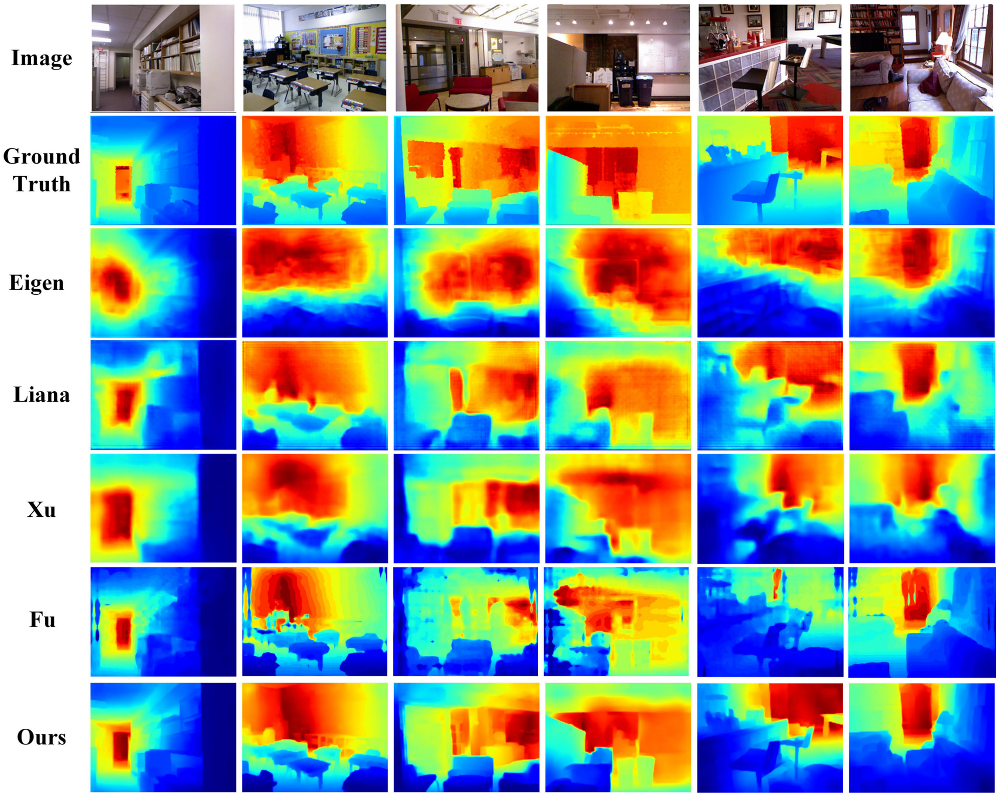


FIGURE 9. Results of NYU-v2 depth dataset. The first row shows the input images. The second row shows the ground truth depth map. From the third row to the last are the prediction of different algorithms [6], [36], [41], [45] and ours model based on SENet-154.

TABLE 1. Comparision on the depth datasets.

		Error(Lower is better)			Accuracy(higher is better)		
Method	Dataset	rms	log_{10}	rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Li [42]	NYU	0.821	0.094	0.232	0.621	0.886	0.968
Liu [43]	NYU	0.759	0.087	0.213	0.650	0.906	0.976
Eigen [36]	NYU	0.907	-	0.215	0.611	0.887	0.971
Laina [41]	NYU	0.573	0.055	0.127	0.811	0.953	0.988
Xu [6]	NYU	0.586	0.052	0.121	0.811	0.954	0.987
ours(ResNet)	NYU	0.568	0.056	0.122	0.855	0.970	0.989
ours(SENet)	NYU	0.554	0.053	0.118	0.861	0.973	0.992
Fu [45]	NYU	0.509	0.051	0.115	0.828	0.965	0.992
Liu [43]	KITTI	0.289	6.986	0.217	0.647	0.882	0.961
Eigen [36]	KITTI	0.270	7.156	0.190	0.692	0.899	0.967
Godard [2]	KITTI	0.206	4.935	0.114	0.861	0.949	0.976
Atapour-Abarghouei [44]	KITTI	0.194	4.726	0.110	0.923	0.967	0.984
ours(SENet)	KITTI	0.177	4.256	0.097	0.918	0.970	0.985
Fu [45]	KITTI	0.120	2.271	0.072	0.932	0.985	0.994

robust than those three models. However, it is sensitive with lighting. Once the scene changed suddenly, the result of depth estimation is not accurate. Our model avoids the unstability of the changing scenes by reconstruction of depth topology graph which provides one kind of efficient depth clues to improve the quality of depth images. There is still a small gap between Fu's [45] work, but the results show that our method adapt the indoor and outdoor scenes.

D. ABLATION STUDY

Due to the sparsity of the depth values from KITTI depth dataset, the color information can not match the depth one pixel by one pixel. Thus, the depth topological graph may produce large errors in different scales. In this part of experiments, we use the NYU-v2 depth dataset to train and validate.

1) POOLING METHOD

According to [46], the error of feature extraction mainly results from two aspects: (1) the increasing of neighborhood domain leads to the increasing of variance of eigenvalues; (2) errors in convolution layer parameters lead to deviations in estimating mean values. In Sect. III-A(1), three pooling methods are introduced to reduce this kind

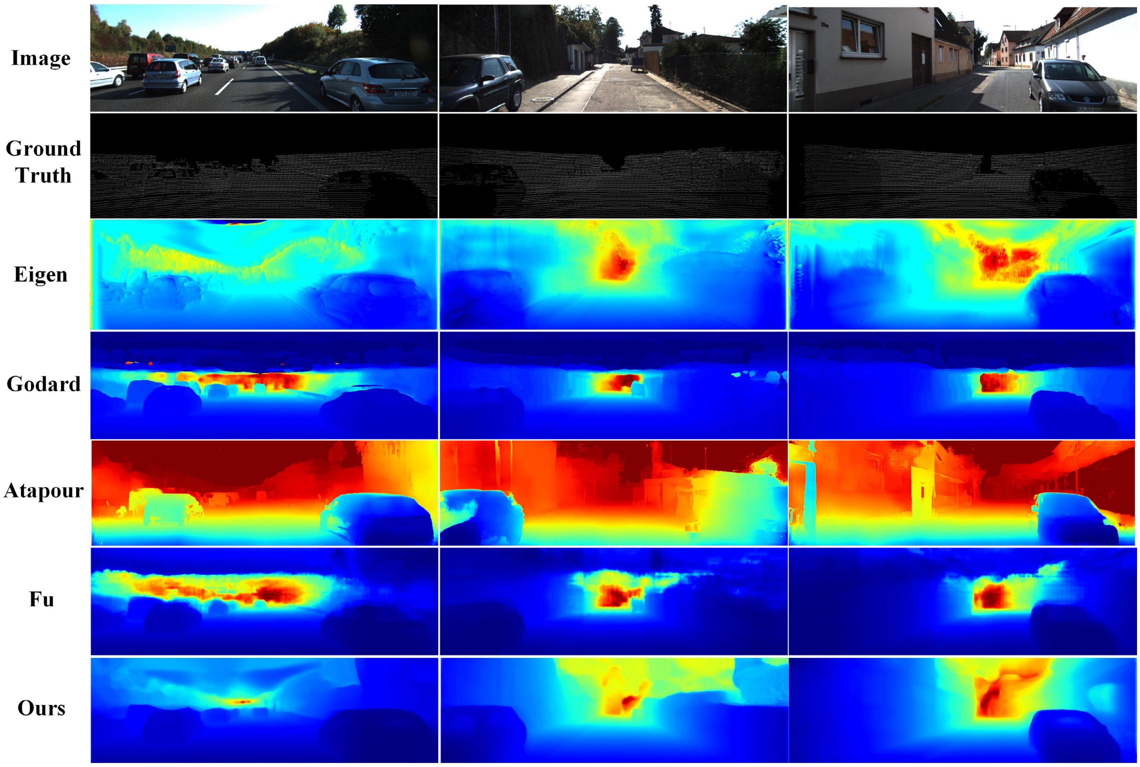


FIGURE 10. Results of KITTI depth dataset. The first row shows the input images. The second row shows the ground truth depth maps. From the third row to the last are the prediction of different algorithms [2], [36], [44], [45] and ours model based on SENet-154.

TABLE 2. Comparision of three pooling methods.

Pooling	8×10	15×19	29×38	57×76
	rel	rel	rel	rel
Average Pooling	0.432	0.371	0.347	0.337
Stochastic Pooling	0.443	0.378	0.350	0.338
Max Pooling	0.409	0.359	0.340	0.334

TABLE 3. Comparision of different R_{scale} of the Gauss noise.

R	Error(Lower is better)			Accuracy(higher is better)		
	rms	log_{10}	rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
0.0	0.7471	0.0805	0.1751	0.7542	0.9398	0.9863
0.2	0.7491	0.0774	0.1591	0.7491	0.9378	0.99861
0.4	0.6584	0.0676	0.1557	0.7716	0.9541	0.9891
0.6	0.6565	0.0723	0.1751	0.7494	0.9381	0.9857
0.8	0.7467	0.0747	0.1723	0.7737	0.9540	0.9883
1.0	0.8528	0.0813	0.1748	0.7512	0.9378	0.9851

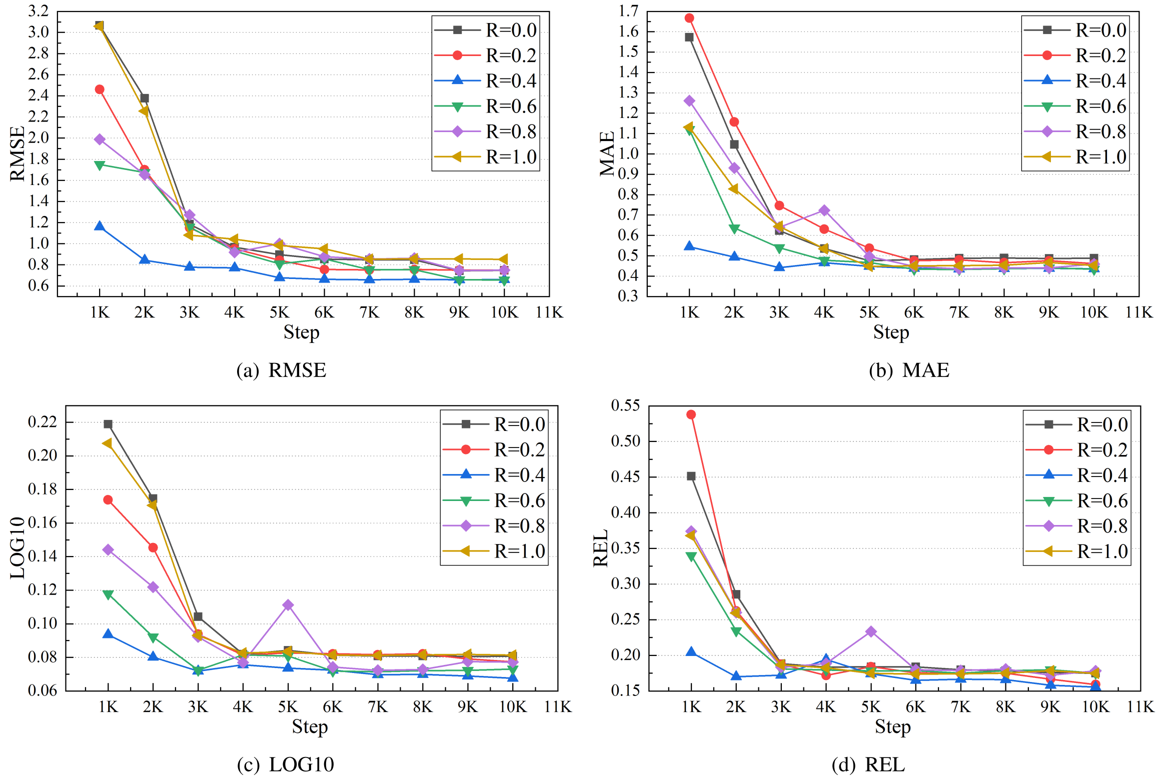
of error. The mean-pooling operation reduces the first error and background information of the image remaining. The Max-pooling operation improves the second error and retains texture information. As for the stochastic pooling operation, the probability is firstly assigned to the pixels according to their numerical size, then the depth map is down-sampled according to the probability. It is similar to a mean-pooling operation in global feature maps and the max-pooling operation in local feature maps.

According to the relative depth error from Table.2, the experimental results show that the max pooling operation achieves the best performance among other pooling methods on the relative error. Thus, the max pooling operation is chosen to improve the depth map error in the following experiments.

2) THE SCALE OF GAUSS NOISE

The stochastic pooling method introduces randomness to enhance the generalization ability of the network. But it is more complex because the occurrence of the current depth value needs to be selected according to the probability. To balance the generalization ability and the complexity, the Gauss noise directly introduced into the depth information. However, the scale of Gaussian noise needs to be determined through experiments. R_{scale} is defined as a factor reflecting the scale of the Gaussian noise. And the evaluation metrics of results is shown in Table.3:

When the scale of Gauss noise is too small, the predicted depth changes slightly, and the result is infinitely close to the situation without noise, which is not conducive to improve the generalization ability of the model. When the scale of Gaussian noise is too large, the predicted depth will be distorted, and the results of the model will deviate from the true value. According to the evaluation metrics graph in Fig.11, we find that when $R_{scale} = 0.4$, the performance of the model is outstanding. And the model has faster convergence speed.



(a) RMSE

(b) MAE

(c) LOG10

(d) REL

FIGURE 11. GNN model with different scale of the Gauss noise.

3) THE PROBABILITY OF GRAPH DROPOUT

When training the depth map based on small scale, the model is prone to over-fit. In traditional convolutional neural networks, dropout operation is introduced to avoid over-fitting. When residual values propagates forward, it can stop the activation of a neuron with a certain probability p, which makes the model more generalized. Probability p does not depend too much on some local features. But in the process of graph convolution, it is necessary to disconnect the edges between nodes in the graph properly to prevent to over-fitting. In this way, some residual can not be transmitted back, improving the diversity of features.

As shown in Fig.12 and Table.4, when the probability is close to 1.0, the converge speed is slow. When the probability is close to 0.5, the model display the best performance in first 1K step and achieve convergence quickly.

4) DICE LOSS

According to the GNN networks, two results will be obtained, one is the depth map, and the other is the reconstruction graph of the node topology. Previous studies use scale interval depth loss and grad loss to directly calculate the error of depth value, so as to improve the training of parameters. The purpose of reconstructing depth topological graph is to ensure that the output features of GNN encoder could remain this kind of information. In order to measure the reconstructed result of the topological graph, the Dice loss function is introduced as the criterion. The value range of Dice loss is between 0 and 1. When the value of loss closes to 0, the reconstruction result is more similar with ground truth. However, the parameter λ of Dice loss will be determined by experiments with single-scale. When λ is set as 1.0, the dice loss convergence faster and more stable. The results are shown in Fig.13.

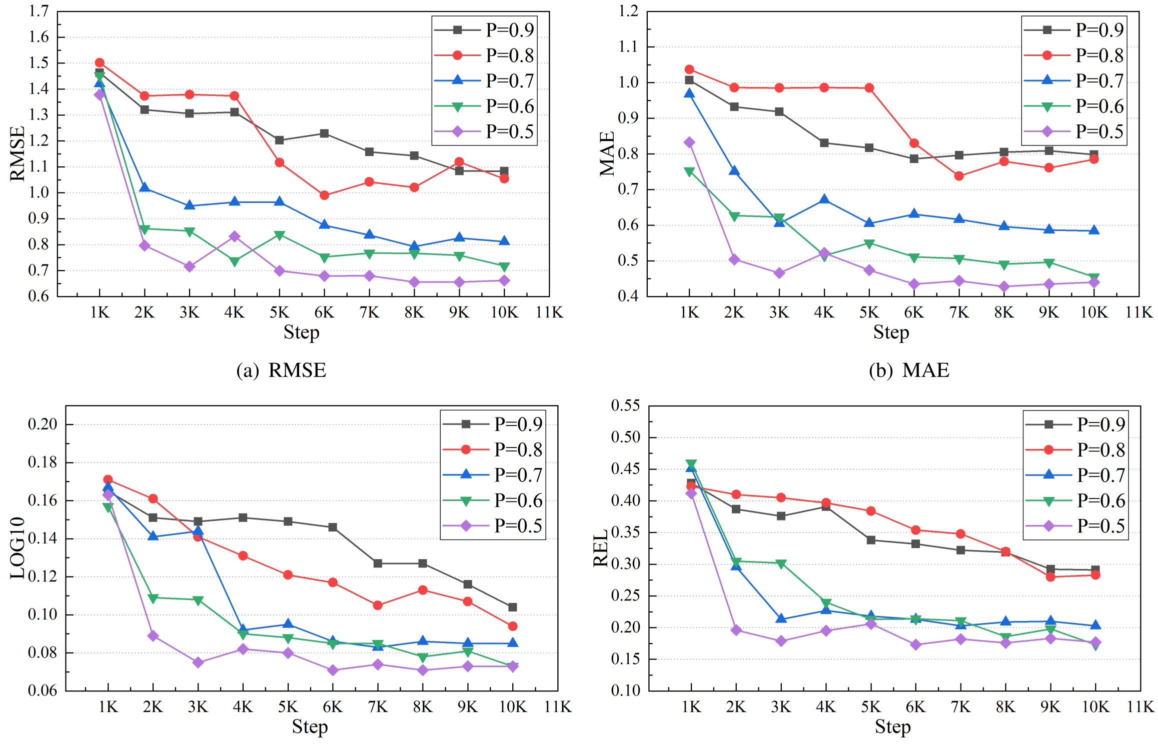
5) MULTI-SCALE GNN MODULE

In order to further improve the performance of the network, a multi-scale GNN model is proposed. At the beginning, the model is desgined to remain the depth topological information of feature maps in every layer, in order that the depth of the pixels can be predicted more robustly. The multi-scale model is composed of several single-scale models. Once the scale of the model increases, the complexity of the model increases. The number of parameters at different scales can be obtained by (20):

$$Num_{param} = \sum_{s=1}^{S}\left(Node_s \sum_{l=1}^{L} Cin_s^l \times Cout_s^l + B_s^l\right) \quad (20)$$

And the complexity of the different scales are shown in Table.5:

We use SENet as backbone network to extract the features of each layer. The size of the input image is 114 × 152, and the characteristic scales are 8 × 10, 15 × 19, 29 × 38, 57 × 76. In the comparison experiemnt, the evaluation metrics of the CNN model, the single-scale



(a) RMSE

(b) MAE

(c) LOG10

(d) REL

FIGURE 12. GNN model with different probability of graph dropout.

TABLE 4. Comparision of different probability of graph dropout.

P	Error(Lower is better)			Accuracy(higher is better)		
	rms	log_{10}	rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
0.9	1.083	0.104	0.291	0.440	0.717	0.875
0.8	1.021	0.094	0.283	0.493	0.786	0.911
0.7	0.812	0.085	0.203	0.690	0.913	0.976
0.6	0.718	0.073	0.173	0.721	0.931	0.981
0.5	0.656	0.071	0.176	0.750	0.935	0.983

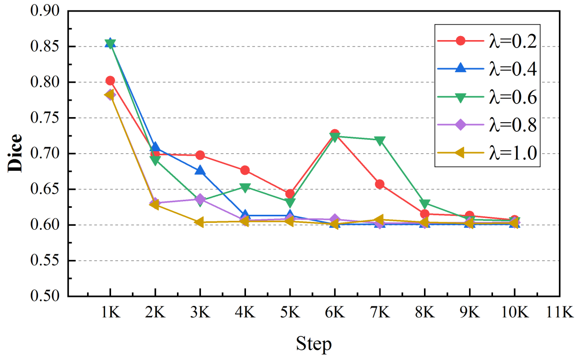


FIGURE 13. Dice loss with different λ.

TABLE 5. Parameter statistics of Multi-scale.

Scale	Channel	Param	FLOPs
8 × 10	2048	1,082,880	41M
15 × 19	1024	19,900,125	75M
29 × 38	512	40,836,814	156M
57 × 76	256	89,555,436	342M

TABLE 6. Comparision with different models.

Method	Error(lower is better)			Accuracy(higher is better)		
	rms	log_{10}	rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
CNN	0.598	0.072	0.141	0.786	0.943	0.962
GNN	0.571	0.060	0.122	0.803	0.950	0.979
ML-GNN	0.563	0.056	0.122	0.821	0.970	0.983
MS-GNN	0.554	0.053	0.118	0.861	0.973	0.992

GNN model, the single-scale GNN model with multi-loss and the Multi-scale GNN model with dice loss are shown in Table.6, the results and the detail of prediction depth are shown in Fig.14:

The difference between the CNN model and the single scale GNN model is that the latter has a graph convolution module layer to instead the depth regression layer of the former. And the comparison results show the GNN module outperform the depth regression layer in all indicators. Although a ResNet-based model is proposed by [41] which achieves similar performance as ML-GNN does, the decoder layer of former is composed of multi-layer fast-up convolution networks, which is different with the depth regression layer of the CNN model. This comparision shows that our GNN decoder layer can be regarded as a candidator decoder layer for the monocular depth estimation. Table.6 shows that the multi-scale GNN model achieves the best performance. Thus, the multi-scale GNN model really contributes to the performance. According to the Fig.14, most depth deviation

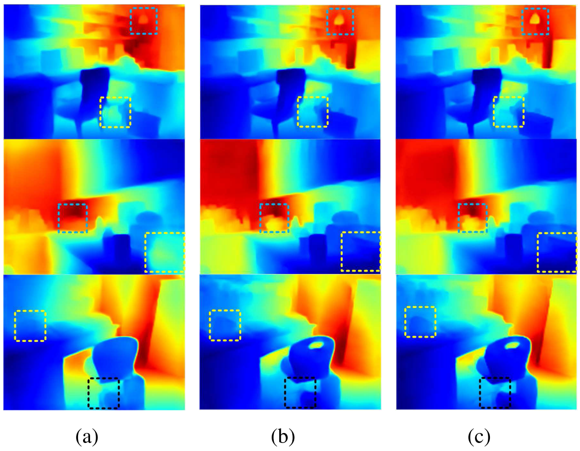


FIGURE 14. GNN model with different scales.(a)Single-scale GNN (b)Single-scale GNN with multi-loss(c)MS-GNN with multi-loss.

of single-scale model are rectified by the multi-scale model, especially there are several little objects coexisting in the scenes.

V. CONCLUSION

In this paper, we have proposed a monocular depth estimation based on the multi-scale graph convolution networks. Compared with the traditional convolution methods, it has three improvements. First, a new construction strategy of depth topological graph is proposed. It guarantees the quality of the extracted information through 3 steps, down-sampling, adding the Gauss noise, linking nodes with multi-scale interval threshold. Second, a new multi-scale graph convolution networks is proposed to remain the depth relationships according to the depth topological graph. Third, a multi-task loss function is proposed, which includes the depth loss and the reconstruction depth topological graph loss. The former constrains the prediction of depth directly. The latter is competent for the sparse adjacency matrix and accelerates convergency of GNN. Finally, our method is verified in the NYU-v2 depth dataset and KITTI depth dataset, and has the better performance in comparison with other existing methods, especially in estimating the depth of small objects and boundaries.

REFERENCES

[1] A. A. A. Mohammed, H. Tao, and M. A. Talab, "Review of deep convolution neural network in image classification," in *Proc. ICRAMET*, Jakarta Selatan, Indonesia, Jul. 2017, pp. 26–31.

[2] G. Clement, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 270–279.

[3] B. Li, D. Yuchao, and M. He, "Monocular depth estimation with hierarchical fusion of dilated CNNs and soft-weighted-sum inference," *Pattern Recognit.*, vol. 83, pp. 328–339, Nov. 2018.

[4] Y. Han, S. Zhang, Y. Zhang, and L. Zhang, "Monocular depth estimation with guidance of surface normal map," *Neurocomputing*, vol. 280, pp. 86–100, Mar. 2018.

[5] S. Liwicki, C. Zach, and O. Miksik, "Coarse-to-fine planar regularization for dense monocular depth estimation," in *Proc. ECCV*, Amsterdam, The Netherlands, Oct. 2016, pp. 458–474.

[6] D. Xu, R. Elisa, and W. Ouyang, "Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 5354–5362.

[7] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 3917–3925.

[8] S. Zhao, L. Zhang, Y. Shen, S. Zhao, and H. Zhang, "Super-resolution for monocular depth estimation with multi-scale sub-pixel convolutions and a smoothness constraint," *IEEE Access*, vol. 7, pp. 16323–16335, 2019.

[9] X.-L. Deng, X.-H. Jiang, Q.-G. Liu, and W.-X. Wang, "Automatic depth map estimation of monocular indoor environments," in *Proc. MMIT*, Three Gorges, China, Dec. 2008, pp. 646–649.

[10] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: https://arxiv.org/abs/1511.07122

[11] R. Li, K. Xian, C. Shen, Z. Cao, H. Lu, and L. Hang, "Deep attention-based classification network for robust depth prediction," in *Proc. ACCV*, Perth, WA, Australia, Jul. 2018, pp. 663–678.

[12] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 5831–5840.

[13] B. Dai, Y. Zhang, and D. Lin, "Detecting visual relationships with deep relational networks," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 3076–3086.

[14] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 5410–5419.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[16] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 4700–4708.

[17] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[18] A. E. Orhan and X. Pitkow, "Skip connections eliminate singularities," 2017, *arXiv:1701.09175*. [Online]. Available: https://arxiv.org/abs/1701.09175

[19] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.

[20] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IJCNN*, Montreal, QC, Canada, Jul. 2005, pp. 729–734.

[21] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*. [Online]. Available: https://arxiv.org/abs/1312.6203

[22] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. NIPS*, Barcelona, Spain, Dec. 2016, pp. 3844–3852.

[23] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: https://arxiv.org/abs/1609.02907

[24] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "DeepStereo: Learning to predict new views from the world's imagery," in *Proc. CVPR*, Boston, MA, USA, Jun. 2015, pp. 3668–3678.

[25] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 5308–5317.

[26] C.-Y. Lee, P. Gallagher, and Z. Tu, "Generalizing pooling functions in CNNs: Mixed, gated, and tree," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 863–875, Apr. 2018.

[27] Y. Huang, X. Sun, M. Lu, and M. Xu, "Channel-max, channel-drop and stochastic max-pooling," in *Proc. CVPR*, Boston, MA, USA, Jun. 2015, pp. 9–17.

[28] M. E. Torres, M. A. Colominas, and G. Schlotthauer, "A complete ensemble empirical mode decomposition with adaptive noise," in *Proc. ICASSP*, Prague, Czech Republic, May 2011, pp. 4144–4147.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, Lake Tahoe, CA, USA, Dec. 2012, pp. 1097–1105.

[30] D. Ruppert, "The elements of statistical learning: Data mining, inference, and prediction," *Publications Amer. Stat. Assoc.*, vol. 99, no. 466, pp. 557–567, 2010.

[31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines Vinod Nair," in *Proc. ICML*, Haifa, Israel, Jun. 2010, pp. 807–814.

[32] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*. [Online]. Available: https://arxiv.org/abs/1611.07308

[33] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Proc. MICCAI*, Montreal, QC, Canada, Sep. 2017, pp. 240–248.

[34] J. Huang, A. B. Lee, and D. Mumford, "Statistics of range images," in *Proc. CVPR*, Hilton Head, SC, USA, Jun. 2000, pp. 324–331.

[35] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. ICCV*, Santiago, Chile, Dec. 2015, pp. 2650–2658.

[36] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. NIPS*, Montreal, QC, Canada, Dec. 2014, pp. 2366–2374.

[37] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[38] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. ECCV*, Florence, Italy, Oct. 2012, pp. 746–760.

[39] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[41] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 3DV*, Stanford, CA, USA, Oct. 2016, pp. 239–248.

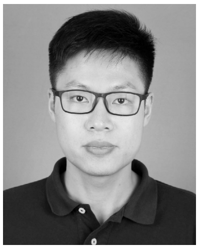
[42] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He, "Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs," in *Proc. CVPR*, Boston, MA, USA, Jun. 2015, pp. 1119–1127.

[43] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.

[44] A. Atapour-Abarghouei and T. P. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 2800–2810.

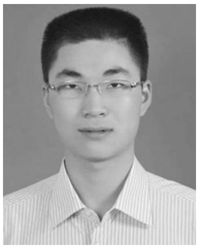
[45] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 2002–2011.

[46] C. Li, S. X. Yang, Y. Yang, H. Gao, J. Zhao, X. Qu, Y. Wang, D. Yao, and J. Gao, "Hyperspectral remote sensing image classification based on maximum overlap pooling convolutional neural network," *Sensors*, vol. 18, no. 10, p. 3587, 2018.



JUNWEI FU was born in Huzhou, Zhejiang, China, in 1990. He received the B.S. degree from the College of Information from Shanghai Finance University, in 2013, and the M.S. degree from the College of Information from Zhejiang Sci-Tech University, Hangzhou, Zhejiang, in 2016. He is currently pursuing the Ph.D. degree with the College of Control Science and Engineering, Zhejiang University.

JUN LIANG was born in May 1963. He received the B.S. degree with the Department of Automation, Tsinghua University, in 1984, and the M.S. and Ph.D. degrees in engineering with the College of Control Science and Engineering, Zhejiang University, in 1993. After graduating from the School, he has been a Professor and the Ph.D. Tutor with the College of Control Science and Engineering, Zhejiang University.



ZIYANG WANG was born in Taizhou, Zhejiang, China, in 1992. He received the bachelor's degree from the College of Information and Control Engineering, China University of Petroleum, in 2014, and the Ph.D. degree with the College of Control Science and Engineering from Zhejiang University, in 2019. He is undertaking equity investment and post-investment management with the Hangzhou Finance Investment Group.