IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.1120000

MGHE-Net: A Transformer-based Multi-Grid Homography Estimation Network for Image Stitching

YUN TANG[1], SIYUAN TIAN[1],PENGFEI SHUAI[1],YU DUAN[2]

[1]College of Computer Science and Cyber Security, Chengdu University of Technology , Chengdu 610059, China
[2]7th Floor, Building A, Caizhi Center, No.300 Tianfu 4th Street, High-tech Zone, Chengdu, Sichuan (second office area), 610000

Corresponding author: SiYuan Tian (1253863446@qq.com) and Yu Duan (duanyu20231017@163.com).

ABSTRACT Image stitching is one of the research hotspots in the fields of computer vision and image processing. Existing methods typically use traditional algorithms or deep learning-based algorithms to achieve this task. However, traditional image stitching algorithms perform poorly in images with weak textures, dark light and multiple noises. And the convolutional neural network (CNN) used by deep learning image stitching algorithms is difficult to capture the global contextual information of an image, resulting in limited accuracy. To address this issue, we designed a Multi-Grid Homography Estimation Network (MGHE-Net) based on Transformers. This network consists of cross-image integration feature extraction module, image matching module, and offset refinement module. The powerful global modeling capability of the Transformer is used to achieve multi-grid homography estimation from coarse to fine, improving the accuracy of image stitching. Experimental results demonstrate that our network not only achieves better stitching results in images with weak textures and dark light, but also reduces errors by 75.3% and 65.1%, respectively, compared to traditional algorithms and CNN-based algorithms on datasets with large parallax. Furthermore, our network improves the efficiency of image stitching.

INDEX TERMS Deep learning, Homography Estimation, Image Stitching, MGHE-Net, Transformer.

I. INTRODUCTION

Image stitching is one of the oldest and most widely used topics in computer vision and graphics [1]. It is a method of creating seamless panoramas or high-resolution images by stitching together multiple images with overlapping regions [2]. This process involves two primary steps: image registration and image fusion. Image registration aligns the overlapping parts of multiple images, while image fusion adjusts the colors, brightness, etc. of the images to ensure they blend together seamlessly. The application of image stitching technology has advanced significantly in recent years and can be used in a variety of fields such as panorama image synthesis, virtual reality [3], geographical mapping and remote sensing imagery [4]. Also its demand is increasing day by day [5].

As a challenging task, the effectiveness of image stitching is not only highly dependent on the stitching algorithm used, but is also affected by the quality of the image itself. In general, the following types of images pose a challenge to image stitching:

1) Weak texture image: texture refers to the grooves presented by the unevenness of the surface of the object, the texture in the image is usually manifested in more obvious shapes, colours, structures, etc., which contains the detailed feature information of the image. Weak texture image is the image that the texture is not obvious, there is no obvious edge or texture features in this type of image, more gradient or solid colour uniform distribution, such as the sky, snow and so on.

2) Low-light images: Due to their low brightness and a higher proportion of black background, low-light images often contain feature points concentrated in specific local areas. This can lead to a larger error in estimating the homography transformation, thereby affecting the image stitching effect.

3) High-noise images: Due to the distinct difference between noise and the original content of the image, noise is easily identified as feature points by feature extraction algorithms, thereby interfering with subsequent feature matching and leading to failed image stitching.

4) Large parallax images: In images with large parallax, there is a significant relative displacement between the foreground and background, making image stitching significantly more challenging. Since homography transformations can

only align scenes on the same plane, and scenes in images with large parallax clearly exist on different planes, this often results in ghosting artifacts in the stitched image.

Image stitching algorithms can be divided into traditional algorithms and deep learning-based algorithms. Traditional algorithms [6], [7], [8] generally follow five steps in image stitching: feature extraction, feature matching, computation of homography matrix, image registration, and image fusion. Feature extraction is a crucial step in the traditional image stitching process, and many scholars have been devoted to improving feature extraction algorithms, proposing various improvement algorithms. Traditional image stitching algorithms can achieve good results in ideal images with rich textures; however, they often struggle when dealing with images containing weak textures, low-light conditions, and high noise levels. This is because existing traditional feature extraction and matching algorithms are not stable in handling such images, leading to poor or failed image stitching results.

In comparison, research on algorithms related to homography estimation and image stitching in the field of deep learning began in 2016 [9]. Although the current research time is significantly shorter than traditional methods, deep learning-based algorithms utilize deep neural networks to extract image features and estimate stitching parameters, thus possessing stronger generalization ability and robustness. This has somewhat improved the stitching effect of images with weak textures, low light, and high noise levels. However, previous deep learning image stitching algorithms [9], [10], [11], [12], [13] are all based on CNN. CNN use convolutional operations to extract features, where the receptive field depends on the kernel size, which is not conducive to modeling long-distance dependencies and leads to poor stitching performance in images with significant parallax.

Compared to CNN, existing research has shown that the Transformer architecture using a multi-head self-attention mechanism is more advantageous for long-range dependency modelling [14], which can improve the stitching of images with large parallax. Therefore, we propose the following innovations for the previous deep-learning based image stitching algorithms:

1) We propose a cross-image integration feature extraction module that effectively combines the information from the reference image with the features extracted from the target image. Additionally, we enhance the fused features using the CSWin Transformer to capture a broader range of global correlations. By doing so, the network gains a better understanding of the interplay and coherence between distinct regions, resulting in improved accuracy and consistency throughout the stitching process.

2) We have enhanced the image matching process by utilizing efficient matrix multiplication to enable parallel comparison of dense feature similarities between two images. This accelerates the computation of feature similarity, thereby significantly improving the efficiency and accuracy of the image stitching process.

3) We propose an offset refinement module aimed at refining and extrapolating the initial normalized relative grid offset obtained. It improves the accuracy of grid offset between overlapping and non-overlapping regions in image alignment, resulting in more accurate image stitching.

II. RELATED WORK

A. TRADITIONAL IMAGE STITCHING ALGORITHMS

Over the past few decades, numerous scholars have conducted extensive research on traditional image stitching algorithms, proposing different algorithms for each step of the image stitching process. Richard et al. proposed a method in 1997 that uses a rotational motion model for image stitching but requires estimation of the camera focal length.In 2003, Brown et al. proposed an automatic panorama image stitching algorithm based on invariant features and further improved the algorithm in 2007. Brown et al. [6] first introduced a more comprehensive multi-image stitching process, treating image stitching as a multi-image matching problem. They used the Scale Invariant Feature Transform (SIFT) algorithm [7] to extract features from the images to determine the matching relationships between all images and employed the Multi-Band Blending algorithm for image fusion. In practice, features extracted by algorithms like SIFT often result in many mismatches during matching. Therefore, Brown et al. used the Random Sample Consensus (RANSAC) algorithm [15] to eliminate mismatches. Subsequently, they solved for a relatively robust global homography matrix based on the matching relationships.

Zaragoza et al. [8] proposed the "As Projective As Possible" (APAP) algorithm to eliminate artifacts created by using global homography transformation for image stitching. They utilized the Moving DLT algorithm to calculate local homography transformations and divided the images into multiple grids for deformation. This algorithm enables images to be locally aligned as much as possible based on matching features, thereby enhancing the precision of image registration and effectively improving the image stitching results. Due to the favorable performance of the APAP algorithm, subsequent traditional image stitching algorithms have largely been improved based on this approach.

Although traditional image stitching algorithms have achieved certain results through years of development, they still face challenges in processing images with weak textures, low lighting, and high noise levels. This is because traditional image stitching algorithms are based on explicit feature extraction and feature matching. Existing feature extraction algorithms struggle to extract a sufficient number of features in images with weak textures or low lighting. Additionally, feature matching algorithms are prone to interference from noise in images with repetitive textures and high levels of noise. As a result, it becomes difficult to establish stable and effective transformation relationships between two images, significantly impacting the image stitching results.

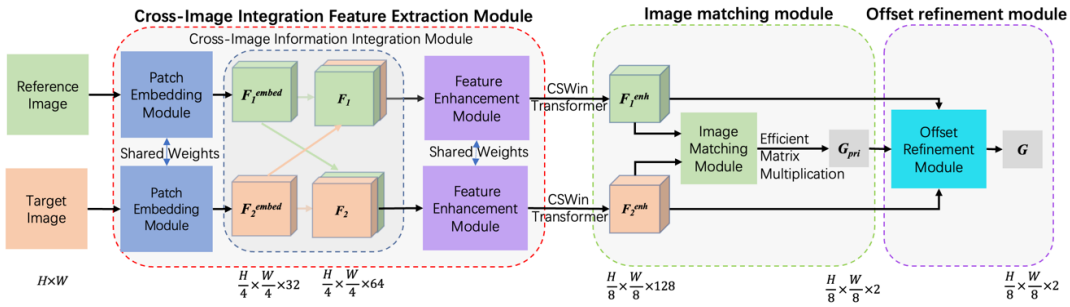


FIGURE 1. MGHE-Net Network Model.

B. DEEP LEARNING-BASED IMAGE STITCHING ALGORITHMS

With the widespread application of deep learning in computer vision, many scholars have begun to explore how to apply deep learning to homography estimation and image stitching to improve their robustness in images with weak textures. Detone et al. [9] first proposed a deep learning-based global homography estimation algorithm in 2016, which utilizes global homography transformation to describe the transformation relationship between two images. Only after obtaining the transformation relationship can the mapping algorithm be used to map the target image to the reference image in order to achieve image registration and complete image stitching. Nowruzi et al. [10] used a hierarchical stacked Siamese network to estimate global homography transformations, refining the estimates gradually by stacking multiple structurally identical network modules. Nguyen et al. [13] further developed the Tensor Direct Linear Transformation and Spatial Transformation Layer based on the work of Detone et al. [9] to achieve unsupervised learning.

Zhang et al. [11] proposed an algorithm that uses a residual network as the backbone and content masks to select reliable regions for global homography estimation, aiming to mitigate the impact of dynamic objects in images on homography estimation. Nie et al. [2] introduced an image stitching method based on global homography transformation. They first estimated the global homography transformation using the method proposed by Detone et al. [9], then conducted initial image stitching and fusion using the estimated global homography, and finally employed a content correction network to eliminate artifacts that may appear in overlapping regions. Subsequently, Nie et al. [12] presented a multi-grid homography estimation network, incorporating a context-aware layer to capture matching relationships between image feature maps. They also utilized depth information from images as additional supervision to compute depth-aware shape-preserving loss, thereby enhancing the image stitching results to a certain extent. Furthermore, Nie et al. [16] introduced an unsupervised deep image stitching method, developing an unsupervised deep image stitching framework that eliminates artifacts in overlapping regions in an unsupervised manner from features to pixels.

The above-mentioned deep learning-based image stitching algorithms have to some extent improved the stitching effect of images with weak textures, low light, and high noise. However, existing deep learning algorithms based on CNN architecture use convolutional operations to extract features, and their receptive fields depend on the size of the convolution kernel. Current CNN networks typically use small convolution kernels to improve computational efficiency, often requiring the stacking of multiple convolutional layers to expand the receptive field globally. This makes it difficult for the algorithms to capture the global contextual information of images. Even if they can capture it, it may lead to overly large network models that are challenging to train. As a result, these algorithms often perform poorly in image stitching for images with large disparities. In contrast, our method leverages the advantages of Transformers in modeling long-range dependencies, thereby improving the image stitching performance for images with large parallax.

III. OUR METHOD

A. NETWORK MODEL

We have designed MGHE-Net, a multi-grid homography estimation network based on the Transformer architecture. This network directly outputs parameterized multi-grid homography transformations (normalized grid offsets). The multi-grid homography transformation divides the image into a series of grid planes and employs a homography transformation in each grid plane to describe the mapping relationship between the reference image and the target image, all with a fixed size. After estimating the multi-grid homography transformations through the network, the target image is mapped to the reference image using a mapping algorithm and then fused to obtain the stitched image.

The network model for MGHE-Net is shown in Fig.1. In order to integrate the information from the two images across the image in order to perceive the image information in a wider field of view and to make the features contain more global relevance in the feature extraction, we integrated the information from the features of both images across the image and further enhanced them using the Transformer. In the image matching module, we improved the efficiency of similarity computation between the two images by employing

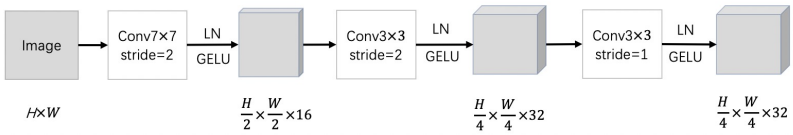


FIGURE 2. **Patch Embedding Module Structure Diagram.**

efficient matrix multiplication to parallelly compare the dense feature similarities. Additionally, to enhance the accuracy of grid offsets, we incorporated an offset refinement module that continuously refines and extrapolates the primary normalized grid offsets. From the ablation experiments detailed in SectionIV-D, these improvements have demonstrated positive effects in enhancing the network accuracy and improving the accuracy of the image stitching results.

B. CROSS-IMAGE INTEGRATION FEATURE EXTRACTION MODULE

The cross-image integration feature extraction module comprises the patch embedding module, cross-image integration module, and feature enhancement module. In this module, the features of both the reference image and the target image are initially extracted. These two sets of features are then connected in different orders along the channel dimension to obtain two new sets of features. Finally, through feature enhancement, these two sets of features encapsulate a greater amount of global correlation information.

1) Patch embedding module

We have adopted the Transformer as the backbone in our proposed network structure. To input the image into the Transformer, it undergoes a process called patch embedding, which involves splitting the image into a series of patches and preliminarily extracting the image features.

To perform the preliminary feature extraction, we utilized existing generic methods and leveraged the high efficiency of CNN. Specifically, we employed a CNN-based patch embedding module to downsample the image in spatial dimensions while increasing the number of channels.

The patch embedding module consists of three different stacked convolutional layers, as shown in Fig.2. Specifically, the first convolutional layer has a kernel size of 7×7 and a stride of 2, which reduces the size of the input image by half and expands the number of channels to 16. The second convolutional layer has a kernel size of 3×3 and a stride of 2, further reducing the size of the feature map while doubling the number of channels to 32. The third convolutional layer has a kernel size of 3×3 and a stride of 1, which further transforms the features while preserving their original shape. Following each convolutional layer, a Layer Normalization layer [17] is used to normalize the features for accelerated training. Subsequently, a GELU activation function [18] is applied to enhance the network's nonlinearity and improve its learning capacity.

2) Cross-image integration module

The patch embedding module transforms the two input images into two sets of embedded features with a shape of $\frac{H}{4}\times\frac{W}{4}$. At this stage, each set of features only contains information corresponding to its respective image. Research by Xu [19] has demonstrated that integrating knowledge from another image into the features of the current image effectively enhances the quality of the extracted features. This improvement cannot be achieved by operating solely on the features of each image separately.Considering that in image stitching, the reference image and the target image do not completely overlap, they each contain unique content. By integrating this unique content into the other image, it becomes possible to obtain a complete representation of the content, thus facilitating a wider perception of the image information. As a result, the accuracy of normalized grid offset estimation can be improved.

To integrate the information from another image, we used a simple channel concatenation operation. Specifically, by concatenating the embedded feature of the reference image $F_1^{embed}\in\frac{H}{4}\times\frac{W}{4}\times32$ with the embedded feature of the target image $F_2^{embed}\in\frac{H}{4}\times\frac{W}{4}\times32$ along the channel dimension, we obtain a new reference image feature $F_1\in\frac{H}{4}\times\frac{W}{4}\times32$; similarly, we can obtain a new target image feature $F_2\in\frac{H}{4}\times\frac{W}{4}\times32$:

$$\begin{aligned} F_1 &= Concat(F_1^{embed}, F_2^{embed}) \\ F_2 &= Concat(F_2^{embed}, F_1^{embed}) \end{aligned} \tag{1}$$

By connecting the features, we can effectively integrate the information from both images without incurring any additional computational resources, thereby avoiding an increase in computational cost. Moreover, when interchanging the order of the two sets of input features, the order of the integrated features will also be interchanged accordingly. This correlation between the order of the integrated features and the original features ensures that the features are not confused or mixed up. Consequently, this approach proves advantageous in enhancing the robustness of the network.

3) Feature enhancement module

After the two aforementioned modules, the image features are initially extracted and integrated. However, since the patch embedding module only utilizes CNN to extract features, the features at this stage only contain local contextual information. If these features are directly applied to the subsequent correlation calculation in the network, it will result in significant training loss and difficulty in convergence. Therefore, we employ a feature enhancement module to enhance the

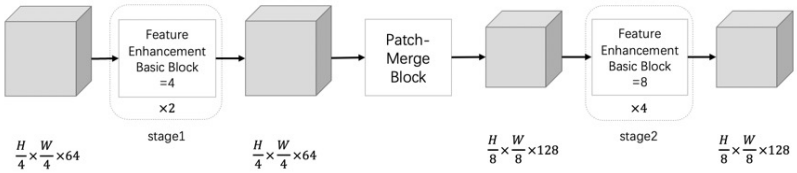


FIGURE 3. Feature Enhancement Module Structure Diagram.

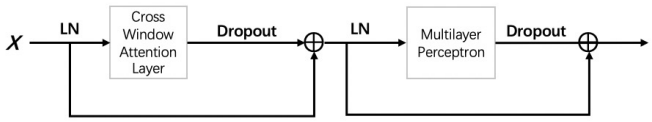


FIGURE 4. Feature Enhancement Basic Block Diagram.

embedded features and improve the learning capacity of the network.

The feature enhancement module is based on the Transformer, and its structure is shown in Fig.3. It consists of two stages (stage 1 to 2) connected by one patch-merge block. In order to improve the module's modeling capacity while balancing memory consumption, we have set the number of feature enhancement basic blocks to 2 and 4 for stage 1 and stage 2, respectively. Additionally, the number of multi-head attentions are 4 and 8, with a window size of cross-attention at 2 and 7, respectively. The patch-merge block reduces the feature map size and expands the channel numbers by merging adjacent feature vectors, thereby generating a multi-scale feature representation between stage 1 and stage 2.

Furthermore, in order to effectively improve the computational efficiency and modeling capacity of the network, we have adopted the CSWin Transformer as the backbone of the feature enhancement module, as compared to the original Transformer [20] and SWin Transformer [21]. The CSWin Transformer incorporates two key designs: the cross-window attention and the local enhanced position encoding. The cross-window attention mechanism balances global attention and computational complexity more effectively, while the local enhanced position encoding, proposed by Dong [14], differs from the encoding method proposed by Vaswani [22] and Shaw [23]. It introduces positional information on different channels of the features to enhance their representation.

Finally, the basic component of the feature enhancement module is the feature enhancement basic block, which has a structure shown in Fig.4. It includes a cross-window attention layer and a multi-layer perceptron module, where the cross-window attention layer integrates the aforementioned cross-window attention and local enhanced position encoding. In addition, both components use residual structures, and LayerNorm (LN) and Dropout layers [24] are used before and after them.

C. IMAGE MATCHING MODULE

The goal of image stitching is to align the target image onto the reference image. In the MGHE-Net network, normalized grid offsets are used to represent the coordinate correspondence, and efficient matrix multiplication is employed to parallelly compare the dense feature similarity between the two images, thus obtaining the correlation in the overlapping region. Specifically, the computation process is as follows:

$$C = \frac{F_2 F_1^T}{\sqrt{128}} \tag{2}$$

In equation(2), F_1 and F_2 respectively represent the features of the reference image and the target image, F_1, $F_2 \in (H_G \times W_G) \times 128$ (For the sake of convenience, H_G and W_G are used to represent the grid offset dimensions of the network output, with H_G=$H/8$ and W_G=$W/8$); C represents the feature correlation matrix between the two images; and $1/\sqrt{128}$ serves as a normalization factor to prevent numerical amplification after matrix multiplication.

To convert feature correlation into coordinate matching relationships, it is necessary to normalize the feature correlation. We use the SoftMax function to normalize the correlation matrix C, thus obtaining the normalized correlation weights W for the same coordinates on the target image and all coordinates on the reference image:

$$W = SoftMax(C, dim = 1) \tag{3}$$

Here, $W \in (H_G \times W_G) \times (H_G \times W_G)$; dim=1 (indexing starts from 0) means applying the SoftMax function on the second dimension. Expanding the equation(3), we can write:

$$W[(i,j),(x,y)] = \frac{e^{C[(i,j),(x,y)]}}{\sum_{a=0}^{a=H_G-1}\sum_{b=0}^{b=W_G-1} e^{C[(i,j),(a,b)]}} \tag{4}$$

Here, (i, j) and (x, y) are indices on different dimensions. Then, the coordinate matching relationship can be obtained:

$$G_{abs} = WG_{base} \tag{5}$$

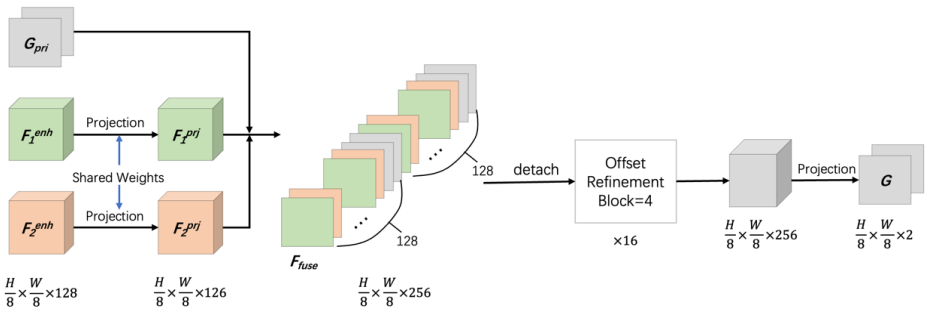


FIGURE 5. Offset Refinement Module Structure Diagram.

$G_{abs}\in(H_G\times W_G)\times2$ represents the absolute pixel grid coordinates of the target image on the reference image; and $G_{base}\in(H_G\times W_G)\times2$ represents the grid coordinates on the reference image.

Finally, by subtracting the base grid G_{base} from the absolute pixel grid G_{abs} and normalizing it using broadcast multiplication, we can obtain the primary normalized relative grid offset G_{pri} :

$$G_{pri}=(G_{abs}-G_{base})\begin{bmatrix}\frac{1}{H} & 0\\ 0 & \frac{1}{W}\end{bmatrix} \tag{6}$$

It should be noted that the primary normalized grid offset G_{pri} obtained based on the above operations can only map the target image to the interior of the reference image. Therefore, it is only valid in the overlapping area between the two images, and invalid in the non-overlapping area. For the grid offset in the non-overlapping area, the offset refinement module in SectionIII-D needs to be used for further estimation.

D. OFFSET REFINEMENT MODULE

To further enhance the accuracy of the grid offset in the overlapping area and extend it to the non-overlapping area, we have designed the offset refinement module. This module refines and extrapolates the initial grid offset obtained from the image matching module to achieve a high-precision grid offset across the entire image region.

In the offset refinement module, ensuring a sufficiently large attention region is crucial in order to capture long-range correlations between images. This plays a vital role in improving the estimation accuracy of the grid offset. Therefore, the offset refinement module also employs Transformer as its backbone, with a structure depicted in Fig.5.

The specific approach is as follows: First, a set of linear layers with shared weights are used to perform channel-wise linear transformations on the enhanced features of the reference image F_1^{enh} and the enhanced features of the target image F_2^{enh} obtained in SectionIII-B, compressing their channel numbers from 128 to 126, resulting in F_1^{pri} and F_2^{pri}. Next, the fused feature $F_{fuse}\in\frac{H}{8}\times\frac{W}{8}\times256$ is obtained by stacking F_1^{pri}, F_2^{pri}, and the estimated primary grid offset G_{pri} from SectionIII-C according to the method shown in Fig.5. Subsequently, a separation operation is used to detach F_{fuse} from the computation graph before inputting it into the subsequent network structure of the module, ensuring that the parameters of the previous network modules are not affected by the offset refinement module. When the separated F_{fuse} is input into a series of offset refinement basic blocks, these blocks refine and extrapolate the primary grid offset from the fused features. After a series of offset refinement basic blocks, the features are further compressed to 2 channels using linear layers, ultimately outputting the high-precision grid offset $G\in\frac{H}{8}\times\frac{W}{8}\times2$ after refinement and extrapolation. Subsequent ablation experiments in SectionIV-D demonstrate that this coarse-to-fine design effectively improves the accuracy of grid offset estimation.

IV. EXPERIMENT AND ANALYSIS

A. RAW DATASET

In order to fully utilize the extensive image datasets available, we conducted experiments using two prominent datasets: the MS-COCO dataset and the Places dataset. The MS-COCO dataset [25], which was released by Microsoft in 2014, is a publicly accessible dataset that encompasses 80 diverse object categories, 1.5 million instances of objects, and includes training, validation, and test sets comprising 118K, 41K, and 41K images respectively. On the other hand, the Places dataset, introduced by Zhou et al. [26] in 2017, is specifically designed based on principles of human visual perception and aims to provide a comprehensive dataset for training visual understanding tasks at a higher level. This dataset consists of an impressive collection of over 10 million images, encompassing more than 400 distinct scene categories.

B. EXPERIMENTAL SETUP

We selected original images from the MS-COCO and Places datasets, and used the image stitching dataset generation algorithm [2] to generate the test samples required for the experiments. In order to comprehensively evaluate the image stitching performance of different algorithms under various conditions, we set up four groups of datasets with varying degrees of disparity for algorithm testing based on experience. Each dataset consists of 20,000 image pairs gen-

TABLE 1. Experiment Dataset List.

Dataset	Maximum Vertex Offset ρ	Overall Maximum Displacement μ	Maximum Grid Deformation Ratio φ
A	28 pixels	32 pixels	0
B	56 pixels	64 pixels	0
C	56 pixels	64 pixels	0.25
D	56 pixels	64 pixels	0.5

erated from test samples from the MS-COCO dataset and Places dataset. Datasets A, B, and C are created by randomly selecting images from the MS-COCO dataset and Places dataset, while dataset D is primarily comprised of images with weak textures, low light, and high noise. In dataset D, images with weak textures account for 40%, low light images for 30%, and high noise images for 30%. The datasets and corresponding hyperparameter configurations are shown in Table1. The complexity of the transformation relationships between image pairs increases sequentially across the four datasets, making the image stitching difficulty progressively more challenging.

In order to evaluate the effectiveness of image stitching, in the absence of recognized evaluation standards, we adopted the method proposed by Detone [9] and used the Average Corner Error (ACE) mentioned in SectionIV-C1 to measure the stitching error of individual samples. In fact, during the testing process, some samples may occasionally exhibit extremely large ACE errors (especially with traditional algorithms), leading to higher mean ACE (Mean Average Corner Error, Mean-ACE) values across the entire test set. Additionally, traditional algorithms may fail due to a limited number of feature points. Taking into account these two scenarios and aiming to ensure fairer test results, this study imposes restrictions on the ACE error: we chose 32 pixels as the threshold for ACE error. That is, when ACE is greater than 32 pixels or when the traditional method fails, we consider ACE as 32 pixels. Since the Mean-ACE can only reflect the average error on the test set and cannot reflect the error distribution, we introduced the Median-ACE as one of the evaluation metrics. For cases where ACE is greater than 32 pixels or the traditional method fails, indicating an ineffective stitching, we also introduced the Invalid Rate (IR) as one of the evaluation metrics, which represents the proportion of ineffective cases in the entire test set. All methods in the experiment were tested multiple times and the average values were taken to avoid accidental occurrences.

C. NETWORK TRAINING

1) Loss Function

The network we proposed, MGHE-Net, employs a coarse-to-fine design. It uses the image matching module to estimate the primary normalized grid offset $G_{pri} in\ G_{base} \in H_G \times W_G \times 2$, and then feeds it into the offset refinement module to obtain the refined normalized grid offset, denoted as G. To achieve this, during training, both the primary normalized grid offset G_{pri} and the refined grid offset G need to be used to calculate the loss.

Since G_{pri} is only effective in the overlapping region of images and invalid in non-overlapping regions, it is necessary to first use Equation (6) and the label G_{label} to obtain the absolute grid offset of the target image on the reference image $G_{abs} \in H_G \times W_G \times 2$. Then, by comparing whether each grid point in G_{abs} is within the coordinate range of the reference image, we can obtain a mask matrix for the overlapping region $M \in H_G \times W_G$. After obtaining the mask M, it is applied to G_{pri} and G_{label} respectively, resulting in masked G_{pri}^M and masked G_{label}^M.

$$\begin{aligned} G_{pri}^M &= G_{pri} \circ M \\ G_{label}^M &= G_{label} \circ M \end{aligned} \tag{7}$$

The "$\circ$" represents the Hadamard product, which means element-wise multiplication.

Finally, use the average corner error as the basic loss function, and calculate the sum of the average corner errors between G_{pri}^M and G_{label}^M, and between G and G_{label}, as the final loss.

$$\|X\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}, X = [x_1, \cdots, x_n] \tag{8}$$

$$\begin{aligned} &ACE(G_{pri}^M, G_{label}^M) = \frac{1}{H_G W_G} \cdot \\ &\sum_{i=0}^{H_G-1} \sum_{j=0}^{W_G-1} \left\| (G_{pri}^M(i,j) - G_{label}^M(i,j)) \begin{bmatrix} H & 0 \\ 0 & W \end{bmatrix} \right\|_2 \\ &ACE(G, G_{label}) = \frac{1}{H_G W_G} \cdot \\ &\sum_{i=0}^{H_G-1} \sum_{j=0}^{W_G-1} \left\| (G(i,j) - G_{label}(i,j)) \begin{bmatrix} H & 0 \\ 0 & W \end{bmatrix} \right\|_2 \end{aligned} \tag{9}$$

$$Loss_{train} = ACE(G_{pri}^M, G_{label}^M) + ACE(G, G_{label}) \tag{10}$$

Here, $\|X\|_2$ denotes the L2 norm, which is used to calculate the Euclidean distance of the error. Since the network output is normalized grid offset, in order to make it more intuitive, it needs to be converted into pixel distance by multiplying it with the corresponding scaling coefficient.

2) Training method

We used the Pytorch deep learning framework to complete the experiments, and optimized the network using the AdamW algorithm [27]. The L2 regularization weight decay coefficient was set to 0.05, and the initial learning rate (LR) was set to 1.25×10^{-4}. We trained 16 pairs of images at each iteration and used 60,000 pairs of images for training and 5,000 pairs of images for validation in each epoch. We trained the network for a total of 120 epochs. In addition, we used a learning rate scheduling strategy that combines multiple algorithms to improve the performance of the network model [28], [29].

TABLE 2. Results of Network Structure Ablation Experiments.

Cross-Image Information Integration module	Feature Enhancement module	Image Matching module	Offset Refinement module	Mean-ACE	Median-ACE	IR
	✓	✓	✓	2.215	1.928	0.075%
✓		✓	✓	3.201	2.678	0.275%
✓	✓		✓	2.994	2.384	0.155%
✓	✓	✓		9.728	8.367	0.690%
✓	✓	✓	✓	1.776	1.646	0.020%

D. ABLATION EXPERIMENT

To examine the influence of distinct network modules on MGHE-Net, we executed a sequence of network structure ablation experiments. These tests encompassed the Cross-Image Information Integration module, Feature Enhancement module, Image Matching module, and Offset Refinement module. As the Patch Embedding module is fundamental, it was not considered in the ablation experiments. All experiments were trained using the techniques outlined in SectionIV-C and were assessed utilizing dataset D, which was mentioned in SectionIV-B. The outcomes of the ablation experiments are presented in Table2.

From Table2, it can be observed that the four modules in the table have different roles and all contribute to improving the accuracy of the network to some extent. These experiments demonstrate that by combining these network modules, MGHE-Net achieves multi-grid homography estimation from coarse to fine, resulting in higher accuracy. This further confirms the rationality and effectiveness of the MGHE-Net structure.

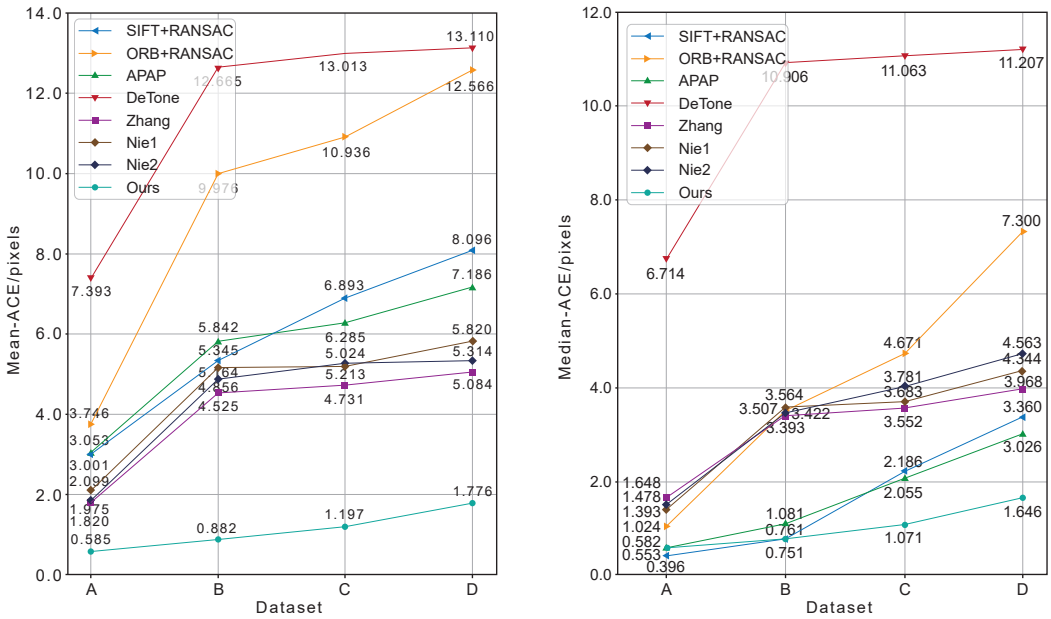
E. COMPARATIVE EXPERIMENT

1) Quantitative error comparison

After conducting the ablation experiments, in order to further demonstrate the effectiveness of our algorithm, we conducted comparative experiments between our algorithm and representative image stitching algorithms proposed in the past. The algorithms involved in the comparative experiments include traditional image stitching algorithms such as SIFT+RANSAC algorithm, ORB+RANSAC algorithm, and APAP algorithm, as well as deep learning-based image stitching algorithms such as the algorithms proposed by Detone [9], Zhang [11] and Nie [12]. Considering that the algorithms proposed by Detone, Zhang are only suitable for scenes with relatively large overlapping areas, we retrained their networks on datasets with smaller overlap ratios to ensure a fair comparison. Table3 shows the experimental results of the above algorithms on different datasets, where the units of Mean-ACE error and Median-ACE error are in pixels. The units are omitted in Table3 for clarity. Here, "Nie1" represents Nie's algorithm using a multi-grid homography estimation network [12], while "Nie2" represents Nie's algorithm using

TABLE 3. Experimental Results of Different Algorithms on Four Datasets.

Algorithm	Dataset	Mean-ACE	Median-ACE	IR
SIFT+RANSAC	A	3.001	0.396	6.165%
	B	5.345	0.751	12.195%
	C	6.893	2.186	13.060%
	D	8.096	3.360	13.690%
ORB+RANSAC	A	3.746	1.024	6.525%
	B	9.976	3.507	20.215%
	C	10.936	4.671	20.730%
	D	12.566	7.300	23.190%
APAP	A	3.053	0.582	5.815%
	B	5.842	1.081	9.250%
	C	6.285	2.055	9.895%
	D	7.186	3.026	10.395%
Detone	A	7.393	6.714	0.095%
	B	12.665	10.906	3.425%
	C	13.013	11.063	4.285%
	D	13.110	11.207	4.540%
Zhang	A	1.820	1.648	0.015%
	B	4.525	3.393	0.205%
	C	4.731	3.552	0.295%
	D	5.084	3.968	0.385%
Nie1	A	2.099	1.393	0.200%
	B	5.164	3.564	0.360%
	C	5.213	3.683	0.480%
	D	5.820	4.344	0.500%
Nie2	A	1.975	1.478	0.120%
	B	4.856	3.422	0.300%
	C	5.024	3.781	0.510%
	D	5.314	4.563	0.550%
Ours	A	0.585	0.553	0.000%
	B	0.882	0.761	0.010%
	C	1.197	1.071	0.020%
	D	1.776	1.646	0.020%



(a) Mean-ACE error

(b) Median-ACE error

FIGURE 6. Variation of Errors for Different Algorithms on Four Datasets.

unsupervised depth image [16].

To have a more intuitive understanding of the variation of errors for different algorithms on the four datasets, Fig.6 presents the line graphs depicting the changes in Mean-ACE error and Median-ACE error for each algorithm on the four datasets. From Table3 and Fig.6, it can be observed that all algorithms exhibit an increasing trend in terms of Mean-ACE error, Median-ACE error, and IR on the four datasets. Among the traditional algorithms, the APAP algorithm performs well. In terms of deep learning-based image stitching algorithms, Detone's algorithm performs poorly due to limitations in the network structure at that time, while the other algorithms achieve lower Mean-ACE error compared to traditional methods. Due to the utilization of our carefully designed multi-grid homography estimation network, our algorithm achieves the lowest Mean-ACE error among all algorithms and consistently maintains a smaller Median-ACE error on datasets with parallax and dataset D. Specifically, our algorithm achieves a decrease of 75.3% and 65.1% in Mean-ACE error compared to the APAP algorithm and Zhang's algorithm, respectively, indicating a significant improvement in accuracy.

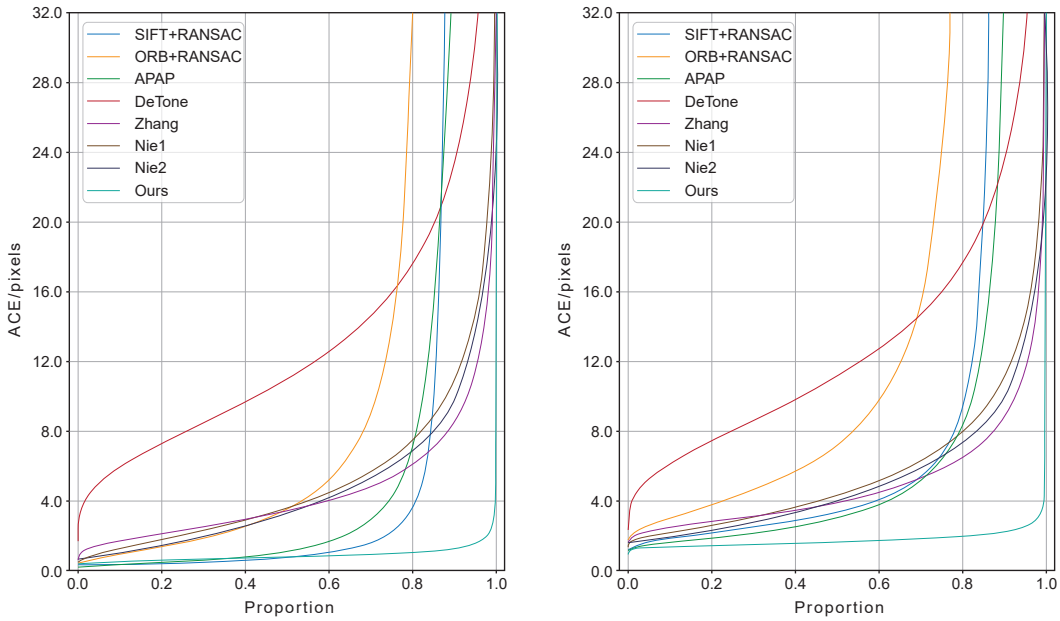
Fig.7 displays the cumulative distribution curves of errors for different algorithms on Dataset B (disparity-free) and Dataset D (with disparity). Among them, traditional image stitching algorithms maintain lower errors in some disparity-free images. However, in other images with weak textures, their errors increase sharply, indicating their weaker robustness. In deep learning-based image stitching algorithms, Detone's algorithm performs the worst, but its IR is still lower than that of traditional algorithms. On the other hand, the remaining deep learning-based algorithms can function properly in over 99% of cases, demonstrating the significantly superior robustness of deep learning-based image stitching algorithms over traditional methods. Among them, our algorithm exhibits the strongest robustness and is the only one capable of maintaining high accuracy (ACE < 4 pixels) in over 95% of cases.

TABLE 4. Efficiency Comparison of Different Algorithms.

Algorithm	Model Size	Running Speed
SIFT+RANSAC	-	35 FPS
ORB+RANSAC	-	50 FPS
APAP	-	0.2 FPS
Detone	32.61M	980 FPS
Zhang	20.31M	1000 FPS
Nie1	32.77M	30 FPS
Nie2	-	15 FPS
Ours	12.56M	200 FPS

2) Runtime efficiency comparison

In addition to comparing the accuracy of different image stitching algorithms, we also compared their efficiency. Table4 illustrates the comparison of model size and running speed for different algorithms. Here, "Nie1" represents Nie's algorithm using a multi-grid homography estimation network [12], while "Nie2" represents Nie's algorithm using unsupervised depth image [16].



(a) Cumulative Distribution Curve of Errors on Dataset B (b) Cumulative Distribution Curve of Errors on Dataset D

FIGURE 7. Cumulative Distribution Curve of ACE Errors for Different Algorithms.

Among them, the image size is $224\times224\times3$. SIFT+RANSAC, ORB+RANSAC, and APAP algorithms run on CPU (AMD Ryzen 5 5600X 6-core 3.7GHz), while the remaining deep learning-based image stitching algorithms run on GPU (NVIDIA RTX 3080Ti 12GB) utilizing deep learning frameworks. Our algorithm has significantly lower parameter count compared to previous deep learning-based image stitching algorithms and achieves faster running speeds compared to traditional image stitching algorithms and Nie's [12], [16] algorithm.

3) Visual effect comparison

Fig.8 and Fig.9 display the image stitching results on Nie's [12] dataset and our self-collected dataset, respectively. In the overlapping regions of the images, a simple average fusion method is used to mitigate artifacts for better visualization. Due to significant errors observed in the ORB+RANSAC method and Detone's algorithm in SectionIV-E1, these two algorithms are excluded from the result images. It is evident from the figures that our algorithm achieves superior stitching results in images with weak textures, large disparities, and low-light conditions, demonstrating the practical value of our image stitching algorithm. Here, "Nie1" represents Nie's algorithm using a multi-grid homography estimation network [12], while "Nie2" represents Nie's algorithm using unsupervised depth image [16].

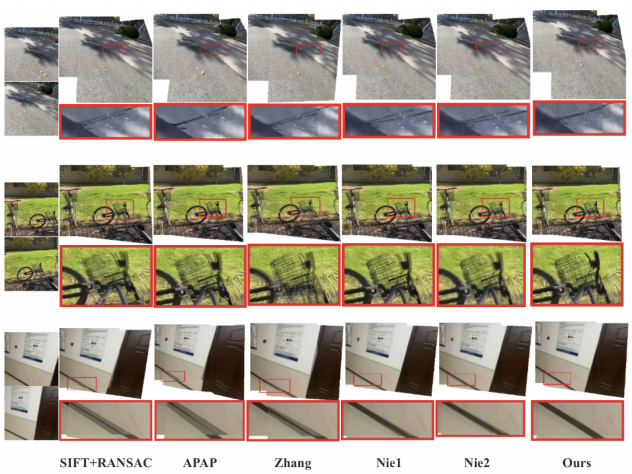


FIGURE 8. Image Stitching Results Comparison on Nie's Dataset.

V. CONCLUSION

We have designed a Transformer-based multi-grid homography estimation network called MGHE-Net to better align image pairs with disparities. MGHE-Net consists of several key network modules, including a cross-image integrated feature extraction module, an image matching module, and an offset refinement module.In the cross-image integrated feature extraction module, features are initially extracted from the reference image and the target image. These two sets of features are then concatenated along the channel dimension in different orders to obtain two new sets of features. Finally, feature enhancement is conducted to incorporate more global contextual information into these two sets of features.Subsequently, in the image matching module, efficient matrix multiplication is employed to perform initial feature matching between the two images, obtaining normalized grid

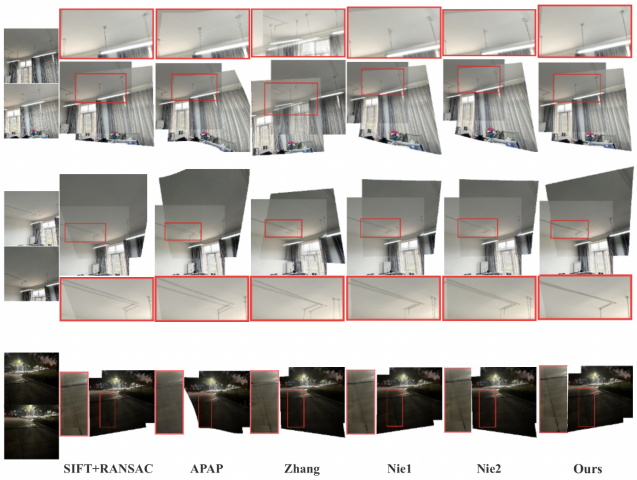


FIGURE 9. Image Stitching Results Comparison on Our Self-collected Dataset.

offsets in the overlapping regions.Lastly, in the offset refinement module, the primary normalized grid offsets are refined, and the normalized grid offsets in the overlapping region are extrapolated to the non-overlapping regions. The output is a high-precision refined normalized grid offsets.Experimental results demonstrate that on datasets with significant disparities, our algorithm achieves a reduction of 75.3% and 65.1% in errors compared to traditional algorithms and CNN-based algorithms, respectively. Our algorithm also outperforms traditional algorithms and Nie's [12], [16] algorithm in terms of running speed and exhibits stronger generalization capability in real-world scenarios.

REFERENCES

[1] W. Lyu, Z. Zhou, L. Chen, and Y. Zhou, "A survey on image and video stitching," *Virtual Real. Intell. Hardw.*, vol. 1, pp. 55–83, 2019.

[2] L. Nie, C. Lin, K. Liao, M. Liu, and Y. Zhao, "A view-free image stitching network based on global homography," *J. Vis. Commun. Image Represent.*, vol. 73, p. 102950, 2020.

[3] R. Anderson, D. Gallup, J. T. Barron, J. Kontkanen, N. Snavely, C. Hernández, S. Agarwal, and S. M. Seitz, "Jump," *ACM Transactions on Graphics (TOG)*, vol. 35, pp. 1 – 13, 2016.

[4] S. Bang, H. Kim, and H. Kim, "Uav-based automatic generation of high-resolution panorama at a construction site with a focus on preprocessing for image stitching," *Automation in Construction*, vol. 84, pp. 70–80, 2017.

[5] D.-Y. Song, G.-M. Um, H. K. Lee, and D. Cho, "End-to-end image stitching network via multi-homography estimation," *IEEE Signal Processing Letters*, vol. 28, pp. 763–767, 2021.

[6] M. A. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, pp. 59–73, 2007.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[8] J. Zaragoza, T.-J. Chin, Q.-H. Tran, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving dlt," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 1285–1298, 2013.

[9] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," *ArXiv*, vol. abs/1606.03798, 2016.

[10] N. Japkowicz, F. E. Nowruzi, and R. Laganière, "Homography estimation from image pairs with hierarchical convolutional networks," *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 904–911, 2017.

[11] J. Zhang, C. Wang, S. Liu, L. Jia, J. Wang, and J. Zhou, "Content-aware unsupervised deep homography estimation," in *European Conference on Computer Vision*, 2019.

[12] L. Nie, C. Lin, K. Liao, S. Liu, and Y. Zhao, "Depth-aware multi-grid deep homography estimation with contextual correlation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 4460–4472, 2021.

[13] T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar, "Unsupervised deep homography: A fast and robust homography estimation model," *IEEE Robotics and Automation Letters*, vol. 3, pp. 2346–2353, 2017.

[14] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "Cswin transformer: A general vision transformer backbone with cross-shaped windows," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12114–12124, 2021.

[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, 1981.

[16] L. Nie, C. Lin, K. Liao, S. Liu, and Y. Zhao, "Unsupervised deep image stitching: Reconstructing stitched features to images," *IEEE Transactions on Image Processing*, vol. 30, pp. 6184–6197, 2021.

[17] J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *ArXiv*, vol. abs/1607.06450, 2016.

[18] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv: Learning*, 2016.

[19] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger, "Unifying flow, stereo and depth estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 13941–13958, 2022.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv*, vol. abs/2010.11929, 2020.

[21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, 2021.

[22] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017.

[23] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *North American Chapter of the Association for Computational Linguistics*, 2018.

[24] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[25] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014.

[26] B. Zhou, À. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1452–1464, 2018.

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2017.

[28] L. N. Smith, "Cyclical learning rates for training neural networks," *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472, 2015.

[29] B. Y. Hsueh, W. Li, and I.-C. Wu, "Stochastic gradient descent with hyperbolic-tangent decay on classification," *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 435–442, 2018.

YUN TANG received his master's degree from Chengdu University of Technology. Currently, he is mainly engaged in research in the fields of nonlinear computing, visualization, numerical computing, artificial neural network, information security and other scientific research. He has conducted several research projects, including national projects such as the Land Survey Project of the Ministry of Land and Resources, provincial and ministerial projects of the Department of Land and Resources, and several local horizontal projects. He has published more than 10 scientific papers as the first author, including one EI, one ISTP and six Chinese core journals.

SIYUAN TIAN completed his undergraduate studies at Chengdu University of Technology in 2023, earning a Bachelor's degree. Throughout his undergraduate years, he focused on software engineering and actively participated in nearly ten projects. His dedication and hard work were recognized through several scholarships awarded at both the faculty and university levels. Subsequently, in 2023, he decided to continue his academic journey by pursuing a Master's degree in Computer Science and Technology at Chengdu University of Technology. Currently, his main focus lies in conducting research related to deep learning and computer vision.

PENGFEI SHUAI graduated from Chengdu University of Technology with a bachelor's degree in 2020 and received a master's degree in software engineering from Chengdu University of Technology in 2023. During his undergraduate and graduate years, he mainly studied computer vision and deep learning, and participated in multiple related projects. During this period, he published multiple papers and won many scholarships. He is currently mainly engaged in the development of industrial software and algorithm research.

YU DUAN received a bachelor's degree in landscape architecture from China West Normal University in 2005 and a master's degree in landscape plants and ornamental horticulture from Southwest University in 2008. From 2008 to 2023, he worked in Chengdu Municipal Engineering Design and Research Institute, was awarded as a professor-level senior engineer, once served as the president of Landscape Architecture Institute, and is now the deputy director of the production and operation management department of Chengdu Design Consulting Group, engaged in the research, planning, planning and design of scenic spots, park cities, cultural tourism development, rural revitalization, urban public green spaces and urban landscapes. The author's awards and honors include the second and third prizes of excellent urban planning and design in China, the first and second prizes of excellent survey and design in Sichuan, the excellent prize of space creative design category (professional group) of the 9th Chengdu Creative Design Week Golden Panda Tianfu Creative Design Award, and the first prize of the collection of ecological sculpture and environmental art design scheme of the 31st Summer Universiade.