electronics

MDPI

Communication

Unsupervised Multi-Scale-Stage Content-Aware Homography Estimation

Bin Hou [1], Jinlai Ren [2,*] and Weiqing Yan [1]

[1] School of Computer and Control Engineering, Yantai University, Yantai 264005, China
[2] School of Civil Engineering, Yantai University, Yantai 264005, China
* Correspondence: qfrenjinlai@126.com

Abstract: Homography estimation is a critical component in many computer-vision tasks. However, most deep homography methods focus on extracting local features and ignore global features or the corresponding relationship between features from two images or video frames. These methods are effective for alignment of image pairs with small displacement. In this paper, we propose an unsupervised Multi-Scale-Stage Content-Aware Homography Estimation Network (MS2CA-HENet). In the framework, we use multi-scale input images for different stages to cope with different scales of transformations. In each stage, we consider local and global features via our Self-Attention-augmented ConvNet (SAC). Furthermore, feature matching is explicitly enhanced using feature-matching modules. By shrinking the error residual of each stage, our network achieves coarse-to-fine results. Experiments show that our MS2CA-HENet achieves better results than other methods.

Keywords: unsupervised; multi-scale; multi-stage; self-attention-augmented ConvNet; feature matching

1. Introduction

Image/video homography estimation is the process of finding corresponding relationships by estimating a projective transformation. It is a basic task in a variety of applications, including visual SLAM [1,2], image/video stitching [3,4] and augmented reality [5,6]. Most of the traditional methods for homography estimation [7,8] employ matched features, such as SIFT [9], SURF [10] and ORB [11], to establish the corresponding relationship. These methods are highly dependent on the extracted features and can typically provide good results in scenes with rich features and a uniform distribution of features. In addition, these steps (feature detection, feature matching and homography estimation) in traditional methods are performed independently; the total performance of alignment can easily be limited by the influence of any one step.

Deep homography estimation methods have drawn more attention from researchers due to their excellent performance in feature representation. These methods usually are divided into two categories: supervised estimation methods [12–16] and unsupervised methods [17–20]. These learning-based methods can often outperform traditional methods in some difficult scenarios, such as images/videos with few features or lacking texture. However, these methods focus on local features, ignoring long-range relationships and the corresponding relationship between features from two images or video frames. In addition, these approaches are effective for image pairs or video frames with small displacements.

Previous research [13,15] has shown that using a multi-stage process to progressively predict and refine homography can cope with large global displacement between two images/video frames. In this paper, we extend these methods to an unsupervised method and propose an unsupervised Multi-Scale-Stage Content-Aware Homography Estimation Network (MS2CA-HENet). In this framework, images with different resolutions are used as input at different stages, starting with low-resolution input images and gradually increasing

check for updates

Citation: Hou, B.; Ren, J.; Yan, W. Unsupervised Multi-Scale-Stage Content-Aware Homography Estimation. *Electronics* **2023**, *12*, 1976. https://doi.org/10.3390/electronics12091976

Academic Editor: Byung-Gyu Kim

Received: 21 February 2023
Revised: 22 March 2023
Accepted: 29 March 2023
Published: 24 April 2023

the size of input images. Large-scale and global transformations are estimated on low-resolution input images; small-scale and local transformations are estimated on high-resolution input images. The homography estimation network of each stage includes a feature-extraction module, feature-matching module and homography-estimation module. The feature-extraction module introduces a self-attention mechanism, which can cover a larger scope in the process of feature extraction to collect feature information, and considers local and global information for extracted features. The feature-matching module enhances the matching relationship between features. By shrinking the error residual of each stage, our network achieves coarse-to-fine results and promotes the performance of alignment. Compared with previous work, our contributions are listed as follows:

(1) We design a novel unsupervised Multi-Scale-Stage Content-Aware Homography Estimation Network (MS2CA-HENet), which effectively addresses homography estimation for a pair of images with large displacement.
(2) We propose a Self-Attention-augmented ConvNet (SAC) to capture local and global features. Moreover, a feature-matching module is introduced into the homography-estimation network to enhance the long-distance dependencies between two image feature maps.
(3) We estimate the residual offsets of the displacement instead of the complete offsets, which estimates the homography from coarse-to-fine via minimizing the error residual at each stage. Experiments show that our method achieves superior performance compared to other methods.

2. Related Work

2.1. Supervised Deep Homography Methods

DeTone et al. [12] made the first attempt to propose a deep homography estimation method, which used a deep convolution neural network to estimate homography. The authors of [13,14,16] utilize a hierarchical architecture that extract features from two image patches to perform homography estimation. Hierarchical approaches can gradually reduce estimation error from coarse-to-fine. Le et al. [15] extend this approach to estimate the motion mask in order to address potentially large dynamic motion. However, these methods are supervised approaches; they need a large number of ground truth annotations, which are costly to gather from real-world data.

2.2. Unsupervised Deep Homography Methods

Nguyen et al. [17] propose an unsupervised method via a Spatial Transformation Layer (STL) [21] to calculate pixel loss between two images/video frames. Their unsupervised method achieves comparable performance to the HomographyNet [12] method. Wang et al. [18] eliminate the need for ground-truth annotations and use invertibility constraints to improve previous unsupervised approaches. Ye et al. [22] use a homography flow rather than the typically used four-point parameterization to estimate homography. Koguciuk et al. [19] extend this approach by calculating the perceived loss [23], which considerably increases the robustness of the model to variations in light. Liu et al. [20] propose a content-aware homography estimation method that learns a mask to eliminate the outliers in a manner similar to the RANSAC [24] function.

2.3. Self-Attention

In computer vision, attention mechanisms [25,26] highlight key elements of an image or feature map while ignoring the rest. Attention is a crucial component of deep convolutional networks owing to its ability to concentrate on important regions within a given context. Self-attention is described as paying attention to a single context rather than to several contexts. The advantage of self-attention is the ability to interact remotely; it has produced cutting-edge models for a variety of tasks [27,28], e.g., image generation [29] and object detection [30]. It has recently shown benefits in a variety of vision tasks to complement convolution models with self-attention. Wang et al. [31] demonstrate that

self-attention is an instance of non-local [32,33] methods and that it can be used to improve video categorization and object recognition. Using a variation of non-local methods, Chen et al. [34] attain favorable outcomes in image classification and video action identification tasks. At the same time, Bello et al. [35] also see big improvements in object detection and image classification by adding global self-attention features to convolutional features.

3. Our Method

3.1. Overall Architecture

Figure 1 illustrates our overall framework. Our network takes the pyramid pairs generated by one initial pair of images or video frames as input, and outputs the homography transformation between the initial pair of images. Pyramid images are built by the down-sampling of 2^k from original input images. The resolution of the three-layer pyramid images is 128×128, 64×64 and 32×32 successively.

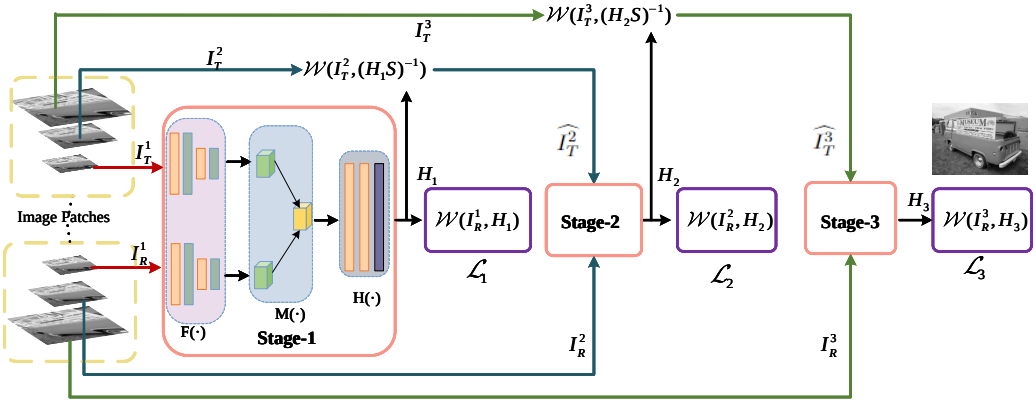

Figure 1. The proposed MS2CA-HENet architecture. The whole network consists of three parts: Stage-1, Stage-2 and Stage-3, respectively, for homography estimation. $\mathcal{W}()$ is an operation that performs a homography transformation on the input images.

The overall network model consists of three stages. In the first stage, we input the smallest-resolution images I_R^1, I_T^1, and output the displacement D_1 of the four image corner points from I_R^1 to I_T^1. Moreover, a Tensor Direct Linear Transform (Tensor DLT) [36] layer is applied to compute the differentiable mapping from the four-point parameterization D_1 to the homography matrix of 3×3 parameterization H_1. In the second stage, the reference image I_R^2 and the warped target image $\widehat{I}_T^2$ are input to the module similar to the first stage. The warped target image $\widehat{I}_T^2$ is obtained as follows:

$$\widehat{I}_T^2 = \mathcal{W}(I_T^2, (H_1 S)^{-1}) \tag{1}$$

where $\mathcal{W}()$ warps the target image using the homography transformation in the Spatial Transformation Layer; S is a scaling matrix at the two scales of the warped target image I_T^1 and the warped target image I_T^2. More specifically, the relationship of homography offsets between two adjacent scale images is calculated to scale the homography, and small-scale offsets are expanded by two times to make them equivalent to the changes on the large-scale images.

For the output of residual displacement ΔD_2 from the reference image I_R^2 to the warped target image $\widehat{I}_T^2$ in the second stage, the total displacement D_2 between the reference image I_R^2 and the warped target image I_T^2 is obtained by the displacement D_1 and the residual displacement ΔD_2:

$$D_2 = D_1 \times 2 + \Delta D_2 \tag{2}$$

Depending on the displacement D_2 from the reference image I_R^2 to the warped target image I_T^2, the homography transformation H_2 can be obtained by the DLT method. The

homography transformation calculated in the second stage is scaled and applied to the target image I_T^3.

Similar to the second stage, the reference image I_R^3 and the warped target image $\widehat{I_T^3}$ are inputs to the third stage, and the output is the residual displacement ΔD_3. The displacement D_3 from the reference image I_R^3 to the warped target image I_T^3 is obtained by the displacement D_2 and ΔD_3:

$$D_3 = D_2 \times 2 + \Delta D_3 \tag{3}$$

Based on the displacement D_3, the homography transformation H_3 can be obtained by the DLT method.

3.2. Network Modules In Stage 1

In this section, we introduce the modules of our network in Stage 1 (see Figure 2) in detail.

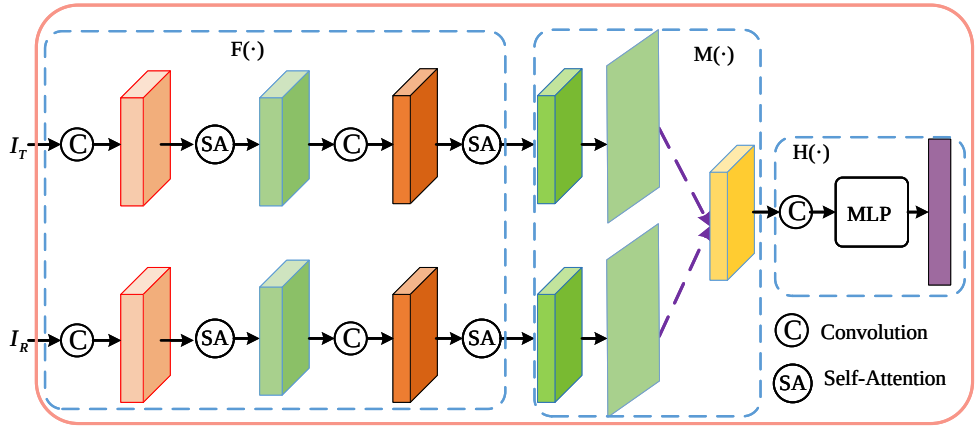

Figure 2. The proposed architecture in Stage 1. In the feature-extraction module F(.), the local and global features are extracted by combining the convolution and self-attention operation. Moreover, the feature-matching module M(.) is designed to enhance feature-matching explicitly from feature maps. At last, we estimate the homography matrix by the homography estimation module H(.).

3.2.1. Feature Extractor

In the feature-extraction module F(.), we combine the convolution and self-attention operation to obtain local and global features. The process is described as :

$$F_R = \mathbf{F}(I_R), F_T = \mathbf{F}(I_T) \tag{4}$$

We firstly employ ResNet34 [37] to extract the image features. Due to the limitation of the receptive field of the convolution kernel, the convolution processes data locally, which makes it computationally inefficient to predict long-range relationships in images. We embed a self-attention module into the feature extractor, enabling it to efficiently model long-distance interactions from the image features, as shown in Figure 3. To be specific, we use the output of *Layer2* in ResNet34 and embed a self-attention module after each layer of *Layer2*. For a pair of input images I_R and I_T of size $1 \times H \times W$, the size of the feature maps is $C \times H/8 \times W/8$. The specific network structure of the feature-extraction module F(.) is shown in Table 1.

Specifically, assume the features of an image extracted by ResNet34 are denoted as x_i. The feature maps x_i can be transformed into different feature spaces by different 1×1 convolutions:

$$K = W_k x_i, Q = W_q x_i, V = W_v x_i \tag{5}$$

where W_k, W_q, W_v are three different 1×1 convolutions. The spacial relationship is calculated by:

$$\beta_{i,j} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N} \exp(s_{ij})}, \text{ where } s_{ij} = K_{2D}(:,i)^T Q_{2D}(:,j) \tag{6}$$

where K_{2D} and Q_{2D} denote the flattened results of the tensors K and Q, respectively, in a sample. The size of K_{2D} (Q_{2D}) is $C \times N$, where $N = H \times W$. $\beta_{i,j}$ indicates the correlation between the ith location and the jth position. The output is $z = (z_1, z_2, \ldots, z_i, \ldots, z_N) \in \mathbb{R}^{C \times N}$:

$$z_i = \sum_{j=1}^{N} \beta_{i,j} V_{2D}(:,j) \tag{7}$$

where $V_{2D} \in \mathbb{R}^{C \times N}$ denotes the flattened matrix of the tensor V in a sample. In addition, the output result is multiplied by a learnable parameter and the feature maps are added. Consequently, the ultimate result is determined by:

$$f_i = \gamma z_i + x_i \tag{8}$$

where γ is a learnable scalable factor with an initial value of 0. By a learnable γ, the module first depends on neighborhood cues, and then progressively can be trained to give non-local evidence more weight.

Table 1. The architecture of feature-extraction module F(.), where 'Conv1', 'Maxpool', 'Conv2_x' and 'Conv3_x' are the components of ResNet34, 'SA1' and 'SA2', respectively, and represent self-attention modules.

Input Image Size	Layer Name	Conv1	Maxpool	Conv2_x	SA1	Conv3_x	SA2
128×128	Kernel Size	7×7	3×3	3×3	1×1	3×3	1×1
	Channels	64	64	64	64	128	128
	Output Size	64×64	32×32	32×32	32×32	16×16	16×16
64×64	Kernel Size	7×7	3×3	3×3	1×1	3×3	1×1
	Channels	64	64	64	64	128	128
	Output Size	32×32	16×16	16×16	16×16	8×8	8×8
32×32	Kernel Size	7×7	3×3	3×3	1×1	3×3	1×1
	Channels	64	64	64	64	128	128
	Output Size	16×16	8×8	8×8	8×8	4×4	4×4

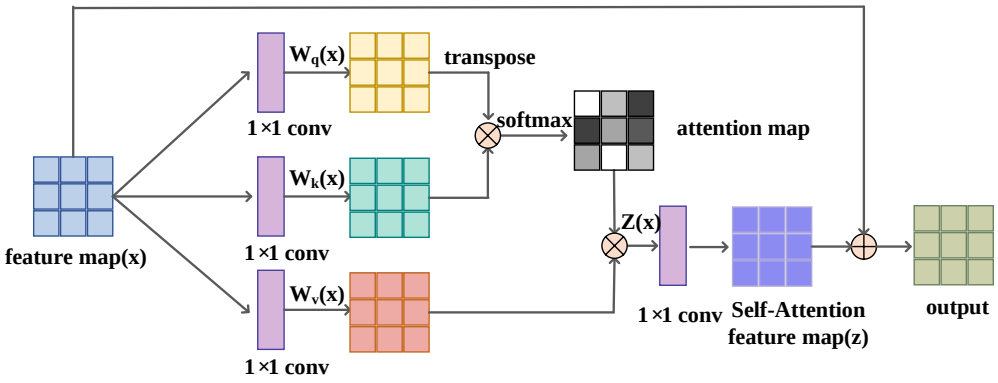

Figure 3. The architecture of the self-attention module.

3.2.2. Feature-Matching Module

Feature matching is an important step in traditional homography estimation methods. By comparing the distances between the feature-point descriptors on each pair of images, the feature points with the minimum distance between them are selected as the matching points. In deep-learning methods, the convolution layer is inefficient at learning the matching relation between features, especially when the displacement between corresponding points is large—the position of the matched feature is much larger than the receptive field of the convolution kernel. In the proposed feature-matching module M(.), feature maps F_R and F_T are inputs, the output is a cost volume S_{3D} to store the correlation values between features of the reference image and the target image in spacial position. The process is presented as follows:

$$S_{3D} = \mathbf{M}(F_R, F_T) \tag{9}$$

Specifically, we first reshape the extracted feature maps F_R and F_T that output from the feature extractor into corresponding 2D matrices $F_{R_{2D}}$ and $F_{T_{2D}}$, respectively. Then, the matching cost $S_{2D}(i,j)$ between the ith feature vector in $F_{R_{2D}}$ and the jth feature vector in $F_{T_{2D}}$ is implemented as the correlation between the feature vectors:

$$S_{2D}(i,j) = \frac{1}{C}\left(F_{R_{2D}}(:,i)\right)^T \odot F_{T_{2D}}(:,j) \tag{10}$$

where $S_{2D} \in \mathbb{R}^{B\times N\times N}$. N denotes the size of spatial resolution, C represents the dimension of the feature vectors, T stands for the transpose operator, and '$\odot$' stands for the dot product. Therefore, the full cost-volume calculation between two different feature maps F_R and F_T can be expressed as:

$$S_{2D} = \frac{1}{C}\left(F_{R_{2D}}\right)^T \otimes F_{T_{2D}} \tag{11}$$

where $\otimes$ means matrix multiplication. As a result, the total cost volume S_{3D} is converted by the 2D cost volume S_{2D}. The specific calculation process is shown in Figure 4.

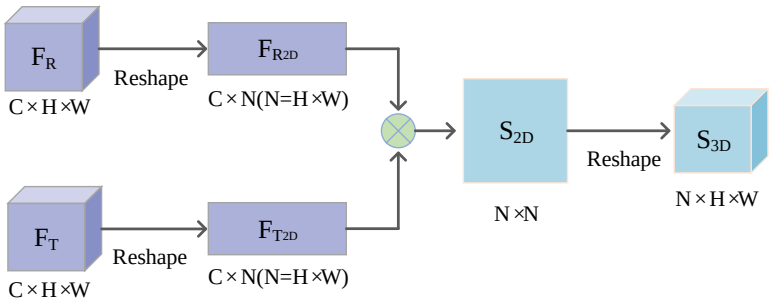

Figure 4. The feature-matching module computes a cost volume between two feature maps, where C, H and W, respectively, represent the number of channels and the height and width of the feature maps.

Compared with other parts of the model, the feature-matching module in our model does not have any trainable parameters. The cost volume may be conceptualized as a 3D form of a similarity matrix. It keeps track of how much it costs to match two sets of dense feature vectors.

3.2.3. Homography Estimator

In the homography estimation module H(.), we employ three successive convolutional layers and two fully connected layers to obtain the displacement D of the four image corners from reference images to target images. To prevent over-fitting, we use the dropout method [38] between the last convolutional layer and the first fully connected layer with a drop probability of 0.5. Our homography estimator function between the cost volume S_{3D} and the displacement D is described as follow:

$$D = \mathbf{H}(S_{3D}) \tag{12}$$

By applying the directly linear transform to D, we can obtain the homography matrix of 3×3 parameterization between a pair of images.

3.3. Loss Function

Utilizing the estimated H_k in the kth stage, we warp the patch images I_R^k to $\mathcal{W}(I_R^k)$ via the Spatial Transformation Layer [21] and compute the L1 loss between the warped target images and the reference images. The network as a whole is differentiable and may be trained through back propagation.

At each stage, we minimize the average L1 pixel-wise photometric loss during the training process. According to previous studies [17,39], an L1-type loss function is more suitable for image alignment problems and the network is more easily trained with an L1-type loss function, so we choose an L1-type loss function instead of an L2-type loss function. In addition, the images may contain some artifacts due to the injection of random illumination offset and distortion, and the L1-type loss function is more robust to outliers [40]. The total loss function can be expressed as followed:

$$\mathcal{L} = \alpha_1 \left\| \mathcal{W}(I_R^1, H_1) - I_T^1 \right\|_1 + \alpha_2 \left\| \mathcal{W}(I_R^2, H_2) - \widehat{I}_T^2 \right\|_1 + \alpha_3 \left\| \mathcal{W}(I_R^3, H_3) - \widehat{I}_T^3 \right\|_1 \tag{13}$$

where the balancing weights are set to $\alpha_1 = 0.5$, $\alpha_2 = 0.3$ and $\alpha_3 = 0.2$. Our loss function consists of three parts, which represent the homography estimation network of three stages and set different weights. $\mathcal{W}()$ is an operation that performs the predicted homography of each stage on the input images using a Spatial Transformation Layer. In Stage 1 of the loss function, we warp patch images I_R^1 to $\mathcal{W}(I_R^1, H_1)$ by the predicted homography transformation H_1. The average L1 pixel-wise photometric loss function is used to minimize the difference in pixel values between the corresponding pixel points $\mathcal{W}(I_R^1, H_1)$ and I_T^1. In Stage 2, we minimize the difference between $\mathcal{W}(I_R^2, H_2)$ and $\widehat{I}_T^2 = \mathcal{W}(I_T^2, (H_1 S)^{-1})$ instead of the difference between $\mathcal{W}(I_R^2, H_2)$ and the original input I_T^2. Since the warped $\widehat{I}_T^2$ is closer to ground truth than I_T^2, the loss shrinks the error residual of each stage. The third stage is similar to the second stage.

4. Experiments

4.1. Dataset and Evaluation Metric

We utilize the method given by Detong et al. [12] for generating datasets on the MS-COCO [41] dataset due to the lack of publicly available datasets for homography estimation. We select 82,783 images from MS-COCO train2014 for the training set and 5000 images from test2014 for the testing set. For each image, a patch with the size of 128×128 is arbitrarily cropped, and each corner point then obtains a random disturbance in the range of 45 pixels, which provides the ground truth four-point corner values to evaluate the proposed method. Then, the image is warped using the inverse of the homography matrix that is defined by the four correspondences. We crop out a second patch from the same position in the warped image. Considering the multi-scale input images of our network, we downsample the patch pairs of 128×128 to different resolution sizes of 64×64 and 32×32. We use the Mean Average Corner Error (MACE) [12] as a metric, which computes the L2 distance between the ground-truth corners and the predicted corners. A lower MACE means better performance.

4.2. Implementation Details

Our network is implemented in PyTorch. The network is trained using an Adam Optimizer with the stochastic gradient descent. The initial value of the learning rate is $l_r = 5.0 \times 10^{-5}$. We train our homography network for 60 epochs. All of our training and testing procedures are carried out on a single NVIDIA Titan XP GPU.

4.3. Comparison

To evaluate the effectiveness of the proposed MS2CA-HENet, we compare the proposed method with different homography estimation methods, including one traditional method—ORB+RANSAC—and eight deep homography estimation methods—HomographyNet [12], HierarchicalNet [13], STN-HomographyNet [14], Self-SupervisedNet [17], SSR-Net [18], SRHEN [16], biHomE [19] and Content-AwareNet [20].

Figure 5 shows the comparative results for the MACE on the MS-COCO dataset. Specifically, we obtain the following observations: the result of the traditional ORB+RANSAC method is higher than those learning-based homography estimation methods. The main reason is that deep learning methods can extract more-robust features than traditional methods.

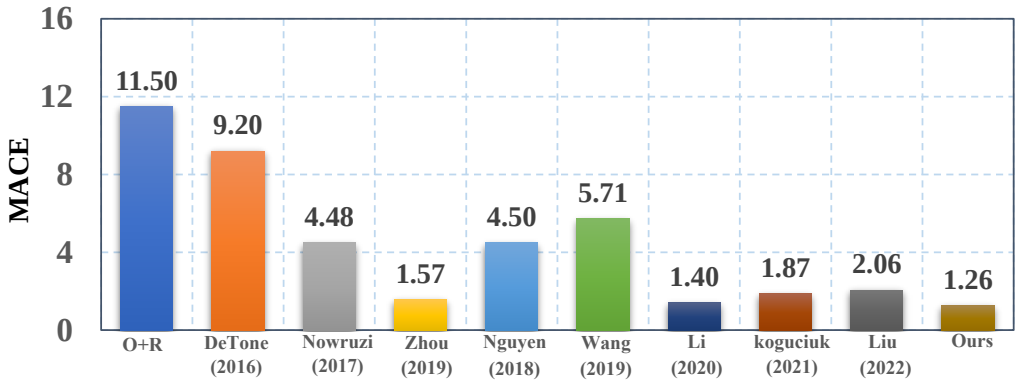

Figure 5. The MACE comparison of different methods. We compare our method with ORB+RANSAC method, HomographyNet [12], HierarchicalNet [13], STN-HomographyNet [14], Self-SupervisedNet [17], SSR-Net [18], SRHEN [16], biHomE [19] and Content-AwareNet [20].

Compared with those deep learning models [12–14,17–20] without the feature-matching module, both SRHEN [16] and our model have the feature-matching module, leading to better results. This demonstrates the necessity of the feature-matching module in deep homography estimation models. Compared with other method (i.e., SRHEN [16]) without the self-attention mechanism, our model adopts a Self-Attention-augmented ConvNet to extract local and global features and enhance the long-distance reliance of the features. Moreover, our model adopts a feature-matching module to strengthen the long-distance reliance of the different feature maps, which can better capture the spatial correspondence between the reference and target images. Our method reduces the MACE by 10.0% compared to SRHEN. As shown in Figure 5, the proposed MS2CA-HENet achieves the best performance.

The visual comparative results of different homography estimation methods are illustrated in Figure 6. As can be seen from the figure, compared with some related homography estimation methods [12,16,17,19], the proposed method obtains better alignment results, which is consistent with the MACE in Figure 5.

In the process of generating the synthetic images, we set different values of point-perturbation parameter ρ to control the displacement of the four corner points in the image patches. The positions of the four corner points are disturbed by taking random values in the range $[-\rho, \rho]$. As the value of point-perturbation parameter ρ increases, so does the displacement of the corresponding corner points. The greater the degree of image distortion transformation, the lower the overlap rate between the input of image patches intercepted at the same position. The quantitative comparison results and the visual results are shown in Table 2 and Figure 7. As shown in this table, all methods perform well when the displacement is small. However, the performance of all methods degrades when the displacement increases. The approaches detailed in [12,17] take the convolution operator to obtain features, which can only capture short-range features due to the limit of the receptive field. Establishing correspondences between features only used by the convolution layer cannot bridge the gap between feature maps and homography. Hence, the values of MACE

are higher than those of our proposed method. In contrast, our method still can keep relatively low values of MACE as the displacement increases. The visual results show the effectiveness of our method for a pair of images with large displacement.

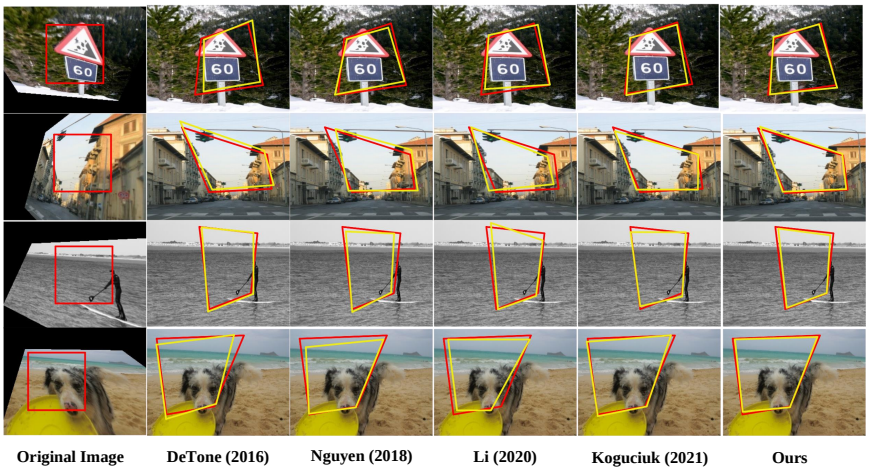

Figure 6. Visualization of the test samples. We compare our method with HomographyNet [12], Self-SupervisedNet [17], SRHEN [16] and biHomE [19]. The red boxes are the ground-truth boxes, and the yellow boxes are the prediction results.

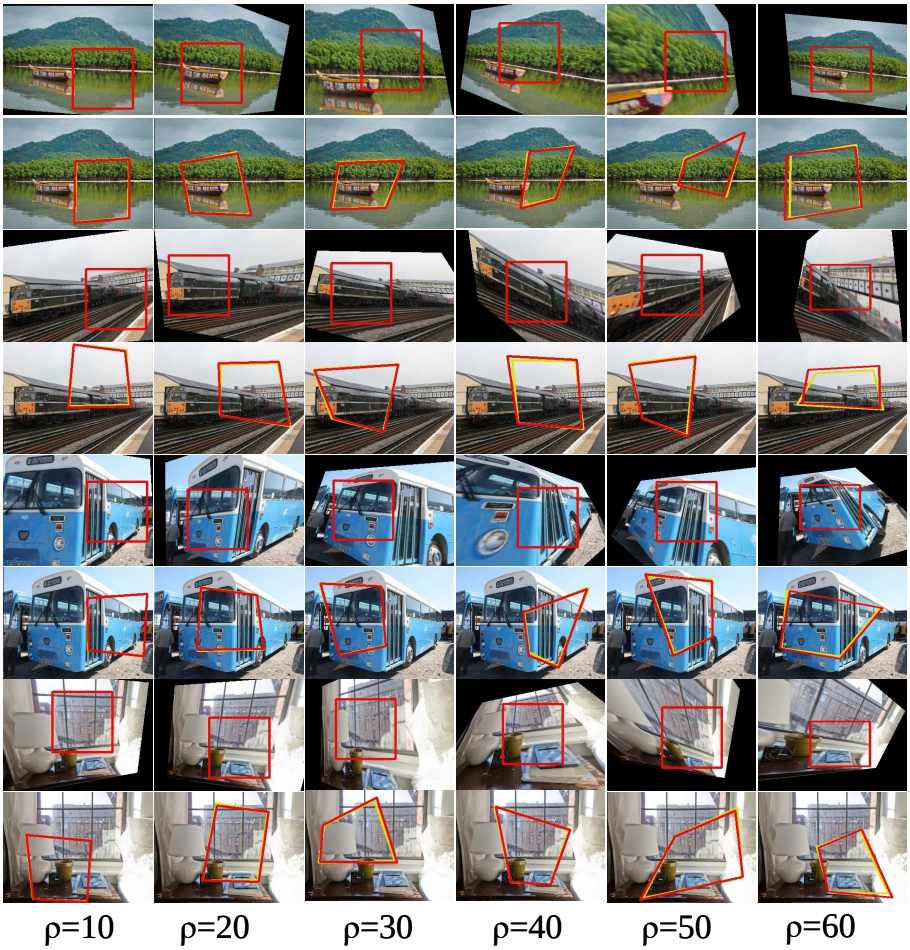

Figure 7. Visualization results for different displacements from 10 to 60. In each example, the first row represents the target image, and the second row represents the warped target image. The red boxes are the ground-truth boxes, and the yellow boxes are the prediction results.

Table 2. The MACEs of different displacements.

Method	[12]	[17]	MS2CA-HENet		
			1st Stage	2nd Stage	3rd Stage
$\rho = 10$	3.19	3.89	1.35	0.70	0.30
$\rho = 20$	3.71	4.16	1.61	0.77	0.35
$\rho = 30$	5.88	4.91	2.12	1.01	0.48
$\rho = 40$	9.58	5.86	2.84	1.48	0.69
$\rho = 50$	16.18	6.68	4.93	2.92	1.72
$\rho = 60$	20.78	7.94	7.15	4.88	3.54

Since HierarchicalNet [13] takes a multi-stage network to estimate the homography, it is compared with the proposed method. As shown in Figure 8, the values of MACE gradually reduce as the number of stacked models increases, which shows a multi-stage network can gradually estimate and refine a homography. Because of the Self-Attention-augmented ConvNet and the global-feature-matching module between two images/video frames, the result of our MS2CA-HENet is lower than that of HierarchicalNet in each stage. From this figure, it can also be observed that the value of MACE in our method is higher when the hierarchy size is 4. Due to the use of multi-scale input, the homography estimation network in the first stage deals with very small images and the training becomes unstable. Hence, we take three stages to train our network.

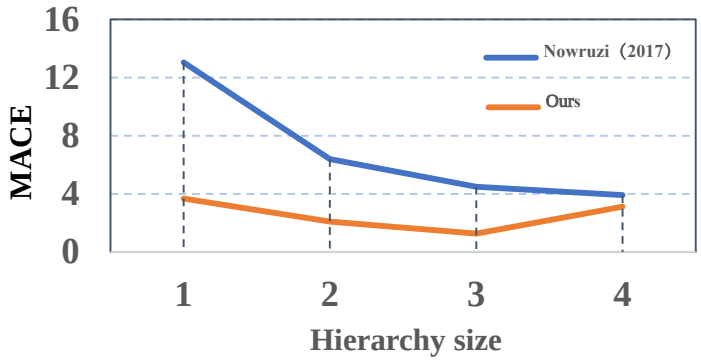

Figure 8. Hierarchy size evaluation. We compare our method with HierarchicalNet [13].

4.4. Ablation Study

Module Selection: We conduct an ablation study in Table 3 to show the effectiveness of the local–global feature-extraction module F(.) and feature-matching module M(.). In the first row of the table, we use ResNet34 instead of the local–global feature-extraction module and feature-matching module. From the first row, we can see the Mean Average Corner Error gradually decreases as the scale increases. However, the MACEs in the first row (only multi-scaled images) are higher than the results of other rows (our designed module F(.) and M(.)). Especially, it can be observed that the error rate without our F(.) and M(.) modules (the first row in the table) is higher than our method by 6.28, 3.72 and 2.87, respectively. This demonstrates the importance of using the local–global feature-extraction module F(.) and feature-matching module M(.) for homography estimation in our model.

Scale Selection: Our model adopts different scale images as the input of each stage. To verify this effectiveness, we compare the same-scale images as input with our multi-scale images. The quantitative comparison and the visual results are shown in Table 4 and Figure 9, respectively. It can be observed that the MACEs of networks with same-scale images is higher than that of our multi-scale network. It seems obvious that for homography estimation models with different input sizes, the models can capture the homography transformation of different input sizes by dividing the transformation space

into different stages. High resolution images contain more details of the image; low resolution images focus on the overall information. The visual results (Figure 9) also show our multi-scale method obtains better results.

Table 3. Ablation study of the module selection.

F(.)	M(.)	1st Stage (32 × 32)	2nd Stage (64 × 64)	3rd Stage (128 × 128)
		10.01	5.89	4.13
	✓	3.91	2.46	1.41
✓		9.96	6.38	3.02
✓	✓	3.73	2.17	1.26

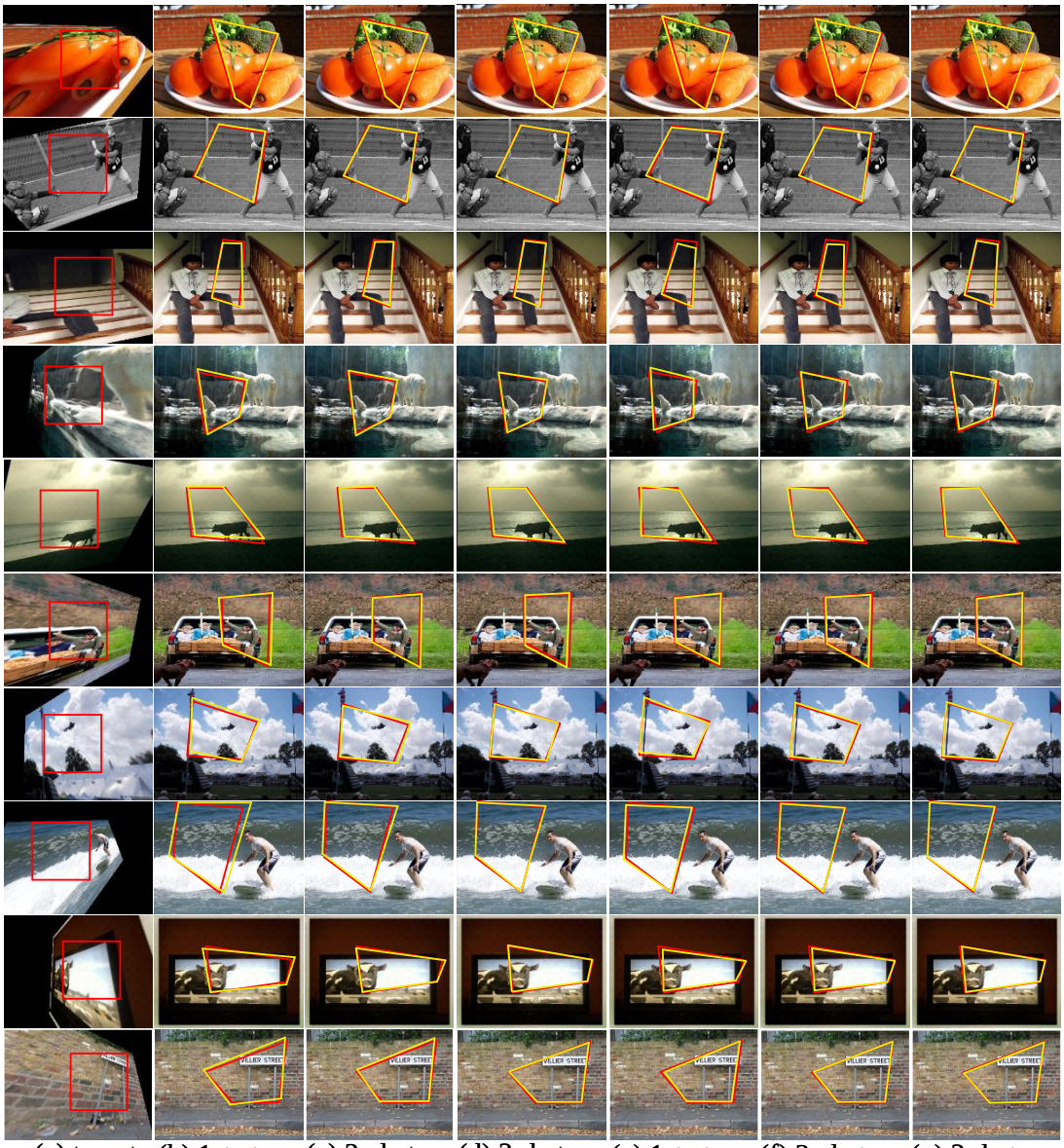

Figure 9. Visualization results with differently scaled images in different stages: (**a**) represents the target images; (**b**–**d**) represent the warped target images by same-scale images with 128 × 128 in different stages; (**e**–**g**) represent the results of different scale image inputs with 32 × 32, 64 × 64 and 128 × 128.

Table 4. Ablation study of the image scale selection.

Image Scale	1st Stage	2nd Stage	3rd Stage
The same 128×128	3.95	3.02	1.67
$32 \times 32, 64 \times 64, 128 \times 128$	3.73	2.17	1.26

5. Conclusions

In this paper, we design a novel unsupervised Multi-Scale-Stage Content-Aware Homography Estimation Network (MS2CA-HENet), which effectively copes with homography estimation for a pair of images with large displacement. In each stage, we consider local and global features via our Self-Attention-augmented ConvNet (SAC) and strengthen feature correspondences explicitly by a feature-matching module. The output of the homography estimation network in each stage is the residual value of the displacement for a pair of images. By shrinking the error residual of each stage, our network achieves coarse-to-fine results and promotes alignment performance. Extensive experiments demonstrate our method achieves favorable performance compared with other methods.

Author Contributions: Conceptualization, B.H. and W.Y.; methodology, W.Y.; software, B.H.; validation, B.H. and W.Y.; formal analysis, W.Y.; data curation, B.H.; writing—original draft preparation, B.H.; writing—review and editing, W.Y.; visualization, B.H.; supervision, J.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China (Grant No. 61801414, 62072391, 62066013 and 61801415) and Shandong Provincial Natural Science (Grant No. ZR2019MF060).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shao, X.; Zhang, L.; Zhang, T.; Shen, Y.; Zhou, Y. MOFIS SLAM: A Multi-Object Semantic SLAM System With Front-View, Inertial, and Surround-View Sensors for Indoor Parking. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 4788–4803. [CrossRef]
2. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
3. Xue, W.; Xie, W.; Zhang, Y.; Chen, S. Stable linear structures and seam measurements for parallax image stitching. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 253–261. [CrossRef]
4. Nie, L.; Lin, C.; Liao, K.; Liu, S.; Zhao, Y. Depth-aware multi-grid deep homography estimation with contextual correlation. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 4460–4472. [CrossRef]
5. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
6. Tang, F.; Wu, Y.; Hou, X.; Ling, H. 3D mapping and 6D pose computation for real time augmented reality on cylindrical objects. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 2887–2899. [CrossRef]
7. Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615–1630. [CrossRef]
8. Szeliski, R. Image alignment and stitching: A tutorial. *Found. Trends Comput. Graph. Vis.* **2007**, *2*, 1–104. [CrossRef]
9. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
10. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. *Lect. Notes Comput. Sci.* **2006**, *3951*, 404–417.
11. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
12. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Deep image homography estimation. *arXiv* **2016**, arXiv:1606.03798.
13. Erlik Nowruzi, F.; Laganiere, R.; Japkowicz, N. Homography estimation from image pairs with hierarchical convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 913–920.
14. Zhou, Q.; Li, X. STN-homography: Direct estimation of homography parameters for image pairs. *Appl. Sci.* **2019**, *9*, 5187. [CrossRef]

15. Le, H.; Liu, F.; Zhang, S.; Agarwala, A. Deep homography estimation for dynamic scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7652–7661.
16. Li, Y.; Pei, W.; He, Z. SRHEN: Stepwise-refining homography estimation network via parsing geometric correspondences in deep latent space. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 3063–3071.
17. Nguyen, T.; Chen, S.W.; Shivakumar, S.S.; Taylor, C.J.; Kumar, V. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2346–2353. [CrossRef]
18. Wang, C.; Wang, X.; Bai, X.; Liu, Y.; Zhou, J. Self-supervised deep homography estimation with invertibility constraints. *Pattern Recognit. Lett.* **2019**, *128*, 355–360. [CrossRef]
19. Koguciuk, D.; Arani, E.; Zonooz, B. Perceptual loss for robust unsupervised homography estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4274–4283.
20. Liu, S.; Ye, N.; Wang, C.; Luo, K.; Wang, J.; Sun, J. Content-Aware Unsupervised Deep Homography Estimation and Beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 2849–2863.
21. Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. *arXiv* **2015**, arXiv:1506.02025.
22. Ye, N.; Wang, C.; Fan, H.; Liu, S. Motion basis learning for unsupervised deep homography estimation with subspace projection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2021; pp. 13117–13125.
23. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; Part II 14, pp. 694–711.
24. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
25. Guo, M.H.; Xu, T.X.; Liu, J.J.; Liu, Z.N.; Jiang, P.T.; Mu, T.J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
26. Zhou, W.; Lv, Y.; Lei, J.; Yu, L. Embedded control gate fusion and attention residual learning for RGB–thermal urban scene parsing. *IEEE Trans. Intell. Transp. Syst.* 2023, *Early Access*.
27. Zhou, W.; Guo, Q.; Lei, J.; Yu, L.; Hwang, J.N. ECFFNet: Effective and consistent feature fusion network for RGB-T salient object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 1224–1235. [CrossRef]
28. Zhou, W.; Zhu, Y.; Lei, J.; Yang, R.; Yu, L. LSNet: Lightweight spatial boosting network for detecting salient objects in RGB-thermal images. *IEEE Trans. Image Process.* **2023**, *32*, 1329–1340. [CrossRef]
29. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the International Conference on Machine Learning (PMLR), Long Beach, CA, USA, 9–15 June 2019; pp. 7354–7363.
30. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; Part I 16, pp. 213–229.
31. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.
32. Zhou, W.; Yu, L.; Zhou, Y.; Qiu, W.; Wu, M.W.; Luo, T. Local and global feature learning for blind quality evaluation of screen content and natural scene images. *IEEE Trans. Image Process.* **2018**, *27*, 2086–2095. [CrossRef] [PubMed]
33. Ma, J.; Zhou, W.; Lei, J.; Yu, L. Adjacent bi-hierarchical network for scene parsing of remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [CrossRef]
34. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J. A^2-nets: Double attention networks. *arXiv* **2018**, arXiv:1810.11579.
35. Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; Le, Q.V. Attention augmented convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 3286–3295.
36. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
39. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss functions for neural networks for image processing. *arXiv* **2015**, arXiv:1511.08861.
40. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
41. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; Part V 13, pp. 740–755.