Dense-Cascade Neural Network for Thermal and Visible Image Registration

*JiahaoXu and XiufenYe**

College of Intelligent System Science and Engineering Harbin Engineering University
No.145, NanTong Street, Harbin, Heilongjiang, 150001, China

{*xujiahao* & *yexiufen*}@hrbeu.edu.cn

***Abstract* – The cross-modality image registration task is becoming more and more important in the field of image fusion, thermal and visible image fusion can reduce the impact of environmental factors while maintaining the image's texture and detail. However, the common registration methods always have the problem of overfitting and poor generalization when facing large modality span task. To register the thermal and visible images, a simple yet effective registration model called Dense-Cascade network is presented in this paper. Our network uses two independent branches to extract the cross-modality features respectively. In order to further reduce the regression error, we use a cascade structure by stacking the networks and using STN layer to achieve gradient backpropagation. Finally, to validate the robustness of our model, we add gaussian noise severity from 0-4 on thermal images to test the performance of our models, and the results show that the registration performances of our models are better than others.**

Index Terms –Thermal and visible image; Image registration; Cascade structure.

I. INTRODUCTION

Image registration is always a key step in image preprocessing, and registration based on homography matrix is the most widely used and common method. Traditional algorithms such as SIFT[1], ORB[2], LPM[3], need to extract the key points manually and match the key points through the feature descriptors, then use algorithms like RANSAC[4], MAGSAC[5] to filter the outliers. However, it is difficult to find a proper and general expression to describe the distance between feature descriptors when dealing with cross-modality images, and the registration effect depends heavily on the pre-set parameters, which means the poor generalization and poor robustness.

In recent years, deep learning models have achieved a great success in image registration. The main idea of these method is: using a deep convolutional network to extract features from the unregistered image pairs, then the offset of four corners is fitted based on the extracted features, finally the homography matrix is obtained by using DLT (Direct Linear Transformation). The first supervised end-to-end homography matrix estimation method between two mono-modality images was proposed by DeTon et al[6]. On this basis, Nowruzi et al.[7] proposed a cascaded network to further reduce the regression error. PFNet[8] further improves the robustness of regression by using the offset of each point from the feature map. Zhang et al.[9] proposed a triplet loss and use STN (Spatial Transformer Networks) layer and mask feature predictor to further improve the robustness of homography matrix obtained by regression. The above networks have made great breakthroughs in mono-modality image registration, however there is still a general problem when applying them directly to cross-modality image registration (such as thermal and visible images): The feature extraction layers of the above networks are all Siamese structure, which means the cross-modality image pair share the same weights, this structure can easily fall into overfitting when facing large modality span tasks, this problem is more prominent when the amount of data is insufficient.

Meanwhile, attention has been widely used in visual tasks in recent years, RepVGG[10] with CBAM[16] makes a great difference in detection task. Song et al.[11] adopted a non-local neural network to recalibrate voxel weights of image features for rigid registration of TRUS and MR images. Chen et al.[12] proposed the cross-modality attention in the feature extraction stage to fuse the cross-modality images.

The remain of this article is arranged as follows. Related work is introduced in section II. The mechanism of the proposed network is described in section III. The experimental results are reported in section IV. At last, some conclusions are drawn in section V.

II. RELATED WORKS

Since our network structure uses STN(Spatial Transformer Networks) layer and cascade structure, we will introduce DLT (Direct Linear Transformation) and STN layer in part *A,* and cascade structure will be introduced in part *B*.

A. DLT And STN Layer

Direct Linear Transformation (DLT) can be used to solve the problem: $H = f: (x, y, 1) \rightarrow (x', y', z')$, where $H \in \mathbb{R}^{3\times3}$ is the transformation, usually we use $I_{reg} = warp(I_{src}, H)$ to register the image I_{src} to image $I_{reg}(x, y, 1)$ is the coordinates from I_{src}, $(\frac{x'}{z'}, \frac{y'}{z'}, 1)$ is the coordinates of the registered image I_{reg}.

Let's consider such a problem:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

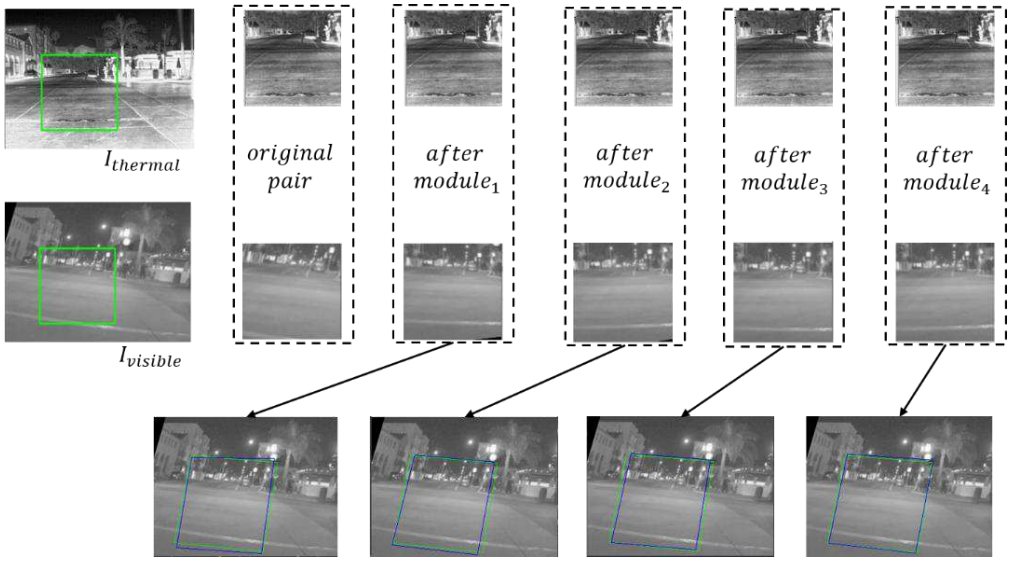


Fig.1. Visualization of cascade registration process

Since for a homologous transformation multiplying the transformation matrix H with a constant $\frac{1}{h_{33}}$ does not affect the final result, let $h'_{ij} = \frac{h_{ij}}{h_{33}}$, for a pair of coordinates, we can get:

$$\begin{bmatrix} x\ y\ 1\ 0\ 0\ 0\ -xx'-yx' \\ 0\ 0\ 0\ x\ y\ 1\ -xy'-yy' \end{bmatrix} \begin{bmatrix} h'_{11} \\ h'_{12} \\ h'_{13} \\ h'_{21} \\ h'_{22} \\ h'_{23} \\ h'_{31} \\ h'_{32} \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{2}$$

Four pairs of coordinates are needed to solve this problem, and this process is called DLT. And STN is the addition of gradient recording on the basic of DLT to make this process learnable, it's not hard to realize this process by common deep learning framework like Pytorch, we mark this process as:

$$H = DLT(P, O) \tag{3}$$

where P is the original coordinates and O is the offset of coordinates.

B. Cascade Structure.

The cascade structure can further reduce the registration error on the basis of single model, and improve the robustness and generalization performance.

Let $Modules_i$, $i = 1 \ldots m$ represents the ith model, where m represents the number of cascades, $\hat{O}_i = Module_i(pair_i)$ represents each module's output, the final homography matrix can be got by

$$\hat{H} = DLT\left(P, \sum_{i=1}^{m} \hat{O}_i\right) \tag{4}$$

The process is shown in Fig.1. With the number of modules increase, the blue frame is getting more similar to the green target frame, which means the registration error is getting down as well.

III. Method

In this section, we will introduce the dataset we used in Part *A*, the common network structures are introduced in Part *B*, the structures we proposed is in Part *C*. All models mentioned below are constructed by Pytorch and trained on a computer with Intel Xeon CPU E5-2620 v4 @ 2.10GHz 2.10GHz and GTX1080Ti. The learning rate is 0.0001, Adam optimizer, batch size is 16.

A. Dataset

For cross-modality registration task, we need pre-aligned image pairs to build the training set. FLIR[13] is a challenge cross-modality dataset that includes day and night scenes, an aligned version is recently released that manually removes unaligned thermal-visible image pairs. This new dataset contains 5,142 well-aligned image pairs, we random choose 4,129 pairs as the training set, and 1,013 pairs are used for testing.

Because the images of FLIR are in various resolutions, we need to adjust all of them to 320 × 240 at first. The production process of a pair is shown in the figure below:

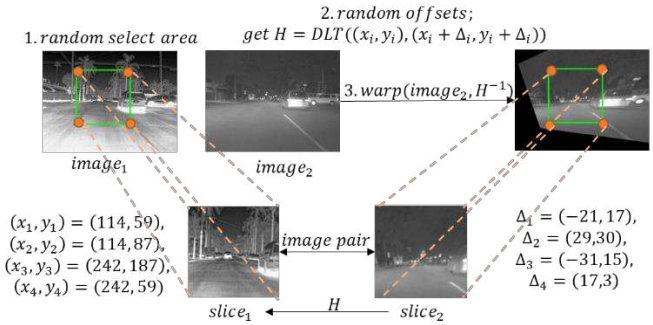


Fig.2. The process of making a pair.

The process can be summarized in the following steps:

*1)*Random crop a 128 × 128 slice from $image_1$ as $slice_1$, which can be thermal or visible image.

*2)*Implement a random offset in $[-32,\ 32]$ to four corners, the transformation H can be obtained through DLT.

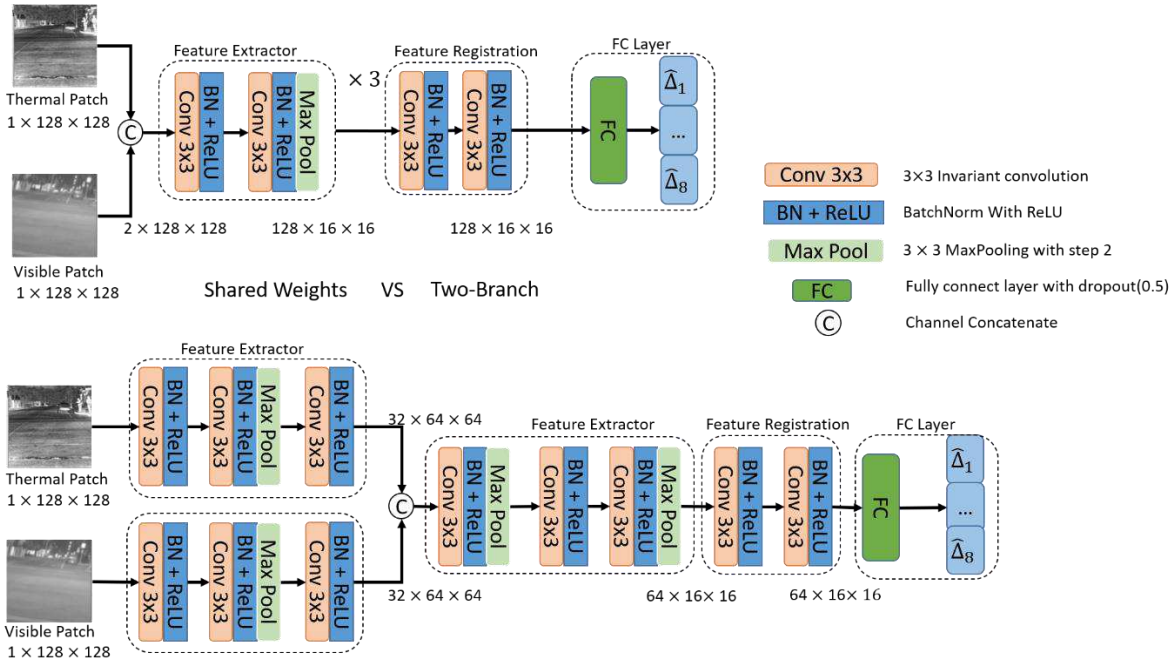


Fig.3. Shared weight VS Two Branch

3) To make sure the corresponding area in warped image is also square, we need to implement H^{-1} transformation to $image_2$, then slice the same area as $slice_2$.

4) Thus, an image pair $= [slice_1,\ slice_2]$ is formed, and the corresponding homography transformation is H.

B. Common Networks

The registration network based on deep learning can be divided into three parts:

1) Feature Extractor: It's used to extract the common or difference feature between image pairs.

2) Feature Regression: It's used to pre-align the extracted features, to make sure the subsequent offset regression more accurate.

3) FC Layer: The pre-aligned features are flattened start from the channel dimension, and the offset of each vertex can be obtained by a fully connected layer inside.

In order to verify that independent branch structure's performance is better than shared weight structure, inspired by DeTon[6] we built two networks as shown in Fig.3.

The shared weight network is composed of three-layer feature extractor, one-layer feature registration, and one FC layer. The feature extractor layer takes a $N \times 2 \times 128 \times 128$ tensor as an input, after each layer, the height and width of tensor will shrink by a factor of two, the number of channels will raise. The feature registration layer takes a $N \times 128 \times 16 \times 16$ tensor as input, which will not change the shape of tensor, finally the fc layer takes a $N \times 128 \times 16 \times 16$ tensor as input after going through liner and dropout(0.5) blocks, an $\hat{O} \in \mathbb{R}^{N\times 8}$ tensor will be exported as four corners' offsets. The loss function is the MSE Loss:

$$loss = \frac{1}{2}\left\|O - \hat{O}\right\|^2 \tag{5}$$

The only difference of Two-Branch structure is on feature extractor, each branch takes an $N \times 1 \times 128 \times 128$ tensor as input, the channel concatenate will happen when each branch's feature is extracted. To ensure that the number of parameters in the two models is as consistent as possible, in the second feature extractor layer, the number of convolutions will be reduced, and the rest of the structures remain the same.

In order to determine which network structure is more suitable for cross-modal image registration work, we ran the two networks with a small dataset of 144 images randomly selected from the FLIR dataset. The registration effect of the two different networks is detailed in section IV, as can be seen from Fig.6 and Fig.7, the network with two-branch independent structure is more suitable for processing cross-modality images.

C. Improvement for Common Network

The network shown in Fig.3 is a typical vgg-style network, with the development of deep learning technology, many excellent feature extraction structures have emerged.

RepVGG[10] with CBAM[16] is widely used in CV task, and DenseBlock is widely used in the fusion application of visible and thermal images like DenseFuse[14]. Inspired by these, we proposed the following structure on the basic of Fig.3, which is shown in Fig.4.

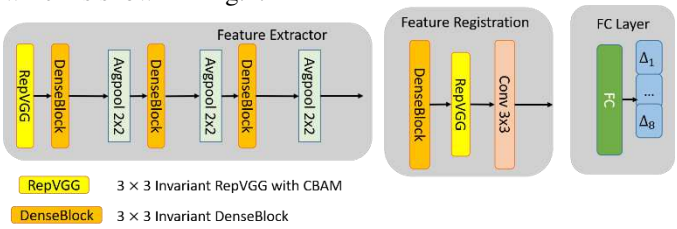


Fig.4. Improved structures.

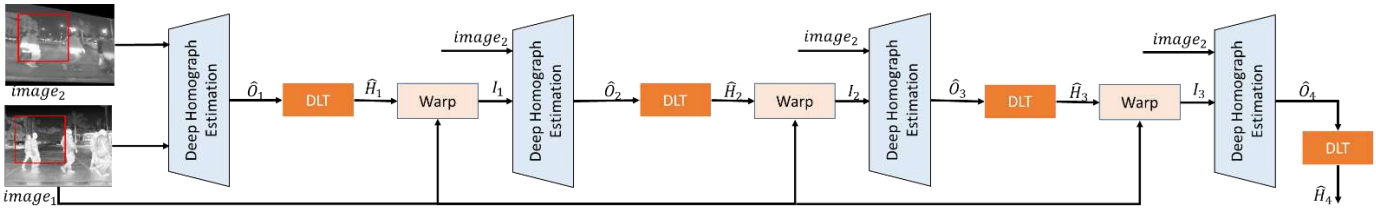


Fig.5. Cascade structures.

Compared with the feature extractor layer of the previous model, the first pair of 3×3 invariant Convolution is replaced by RepVGG[10] with CBAM[16], and the last two pairs are replaced with the combination of DenseBlock and Avgpool2d. For the feature registration layer, the first pair of 3 × 3 invariant convolution is changed to DenseBlock, and a 3 × 3 invariant convolution is added at the end, the FC layer and loss function remain the same.

With this improved structure, the MACE (Mean Average Corner Error) index on the validation dataset can be reduced from 9 pixels (7.03%) to only 7 pixels (5.47%), but this is still intolerable when it comes to large resolution images. Inspired by Nowruzi[7], we use cascade structure to register the thermal and visible images. The structure is shown in Fig.5.

For $Module_i$, the regression residual is $R_i = R - \sum_{j=1}^{i} R_j$, the expected result of $Module_{i+1}$ is $R_{exp} = R_i$, so the loss function should be updated to:

$$loss = \frac{1}{2}\sum_{i=1}^{m} \|R_i\|^2 \tag{6}$$

Where m is the number of cascades, for Fig5. the following formula holds:

$$\begin{cases} \hat{H}_1 = DLT(P, \hat{O}_1) \\ \hat{H}_2 = DLT(P, \hat{O}_1 + \hat{O}_2) \\ \hat{H}_3 = DLT(P, \hat{O}_1 + \hat{O}_2 + \hat{O}_3) \\ \hat{H}_4 = DLT(P, \hat{O}_1 + \hat{O}_2 + \hat{O}_3 + \hat{O}_4) \end{cases} \tag{7}$$

Where $\hat{H}_i, i = 1 \ldots 4$ is the predicted homography matrix of each module, $\hat{H}_4$ is the final result.

$$\begin{cases} I_1 = warp(image_1, \hat{H}_1), R_1 = O - \hat{O}_1 \\ I_2 = warp(image_1, \hat{H}_2), R_2 = O - \hat{O}_1 - \hat{O}_2 \\ I_3 = warp(image_1, \hat{H}_3), R_3 = O - \hat{O}_1 - \hat{O}_2 - \hat{O}_3 \\ R_4 = O - \hat{O}_1 - \hat{O}_2 - \hat{O}_4 \end{cases} \tag{8}$$

where $\hat{O}_i$ is the output of each module, O is the target, R_i is the residual of each module, and the final loss can be got as:

$$loss = \frac{1}{2}\sum_{i=1}^{4} \|R_i\|^2 \tag{9}$$

IV. Experiments

In this section, we will introduce the experiment of shared weight and two-branch structures in part *A*, the comparison of various models will be introduced in part *B*, the robustness experiment will be introduced in part *C*.

A. Shared Weight or Two-Branch

As we have introduced in section III, we trained 512 epochs for each model with a small dataset of 144 images randomly selected from the FLIR dataset. The MACE index is used to evaluate the models.

$$MACE = \frac{1}{4}\sum_{i=1}^{4} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \tag{10}$$

The result of training and validation phases is shown in Fig.6.

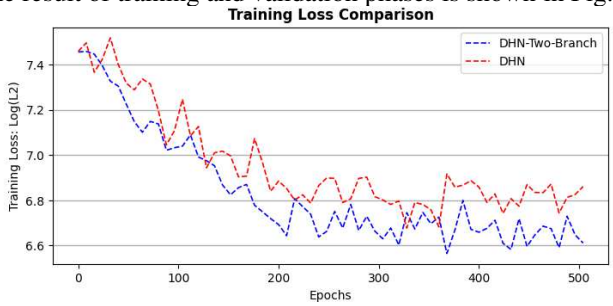


Fig.6. Training loss comparison for DHN and DHN-Two-Brach

During training process, the convergence speed of these two models is nearly the same, but the loss value of the two-branch structure is much lower than the shared weight structure.

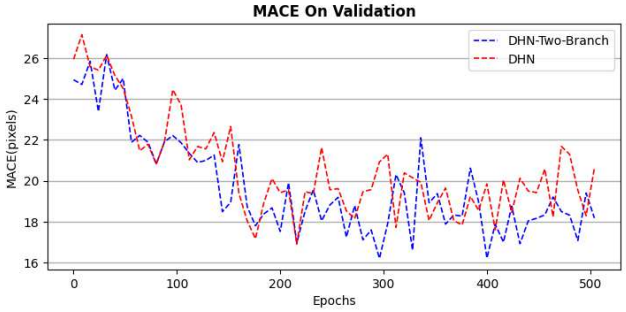


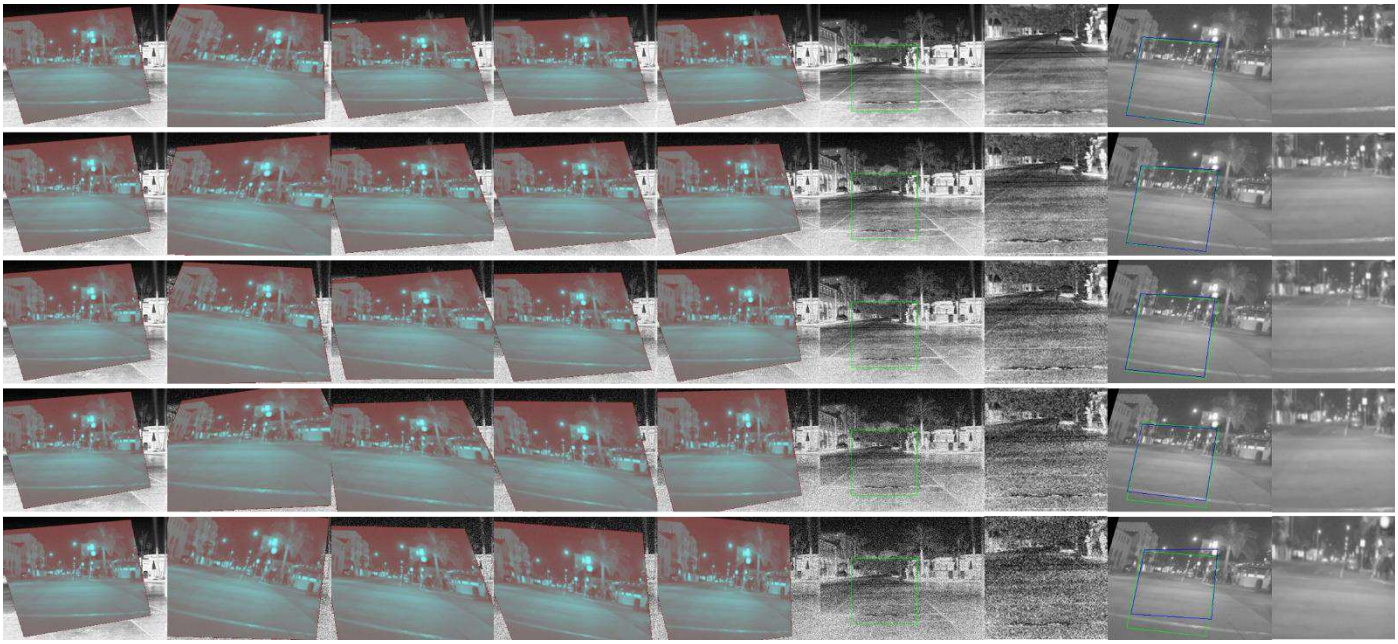
Fig.7. MACE on validation for DHN and DHN-Two-Brach

As for the MACE evaluation index on validation two-branch is also lower than shared weight. The overall training process was relatively volatile due to the small size of the data set and the small batch size setting.

We think shared weight structure is good at extracting the common feature between two kinds of images, but two-branch structure is just on the contrary, and for the cross-modality task, the greater difference of the features can lead to better pre-registration for cross-modality task.

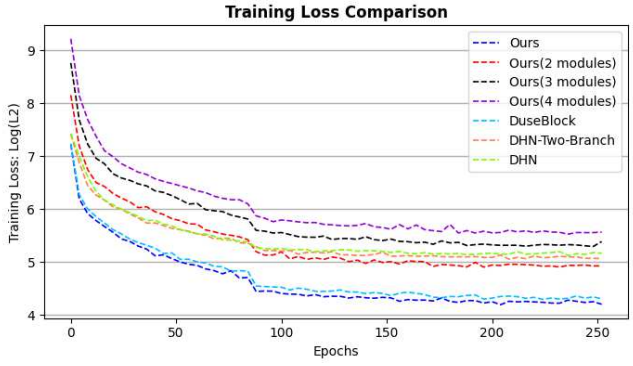
B. Performance Comparison

In order to evaluate the performance of our proposed Dense-Cascade structure on cross-modality image registration task. We choose DuseBlock[12], DHN[6], DHN-Two-Branch as the experimental control group. The results of the training loss and MACE index are shown in Fig.9 and Fig.10.



(a)Target (b)PFNet[8] (c)DHN[6] (d)DHN-Two-Branch (e)Ours(4) (f)Thermal Image (g)Thermal Patch (h)Visibale Image (i)Visible Patch

Fig.8. Visualization of various kinds of registration



.

Fig.9. Training loss comparison for various structures.

It can be seen from the training process, since the loss function of the cascaded multi-module is composed of multiple parts (shown in equation (9)), the loss function of the cascaded multi-models is higher than that of other models in the training process, but increasing the number of cascades does not affect the convergence speed, almost all models converge around 80 epochs.

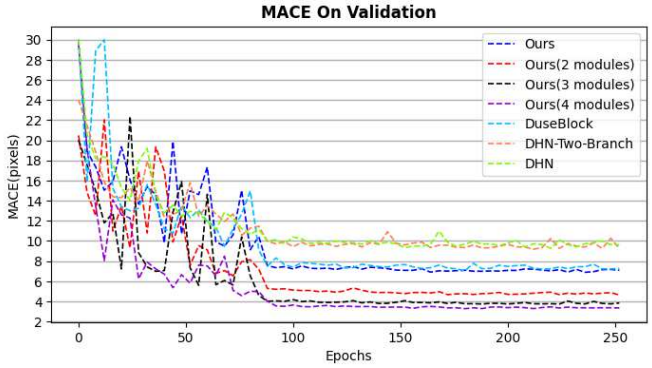


Fig.10. MACE on validation comparison for various structures.

As analyzed in part *A*, the performance of DHN-two-branch is significantly better than that of the shared weight DHN. The improved structure is also significantly better than DuseBlock[12]. The model with the cascading structure performs much better than the other models. However, the performance improvement from 3 modules to 4 modules is not obvious, so the additional number of cascades should not be considered any more.

C. Robustness Experiment

In this part, we compare the proposed method with PF-Net[8], DHN[6], and DHN-Two-Branch to test the robustness of the algorithm in a noise pollution environment with Gaussian noise. The noise severity levels are ranging from 0 to 4 realized by Imgaug toolbox

In order to further test the robustness of our proposed model, all tests were carried out on the networks with no noisy image training, and the experimental results are shown in TABLE I.

TABLE I
MACE INDEX COMPARISON

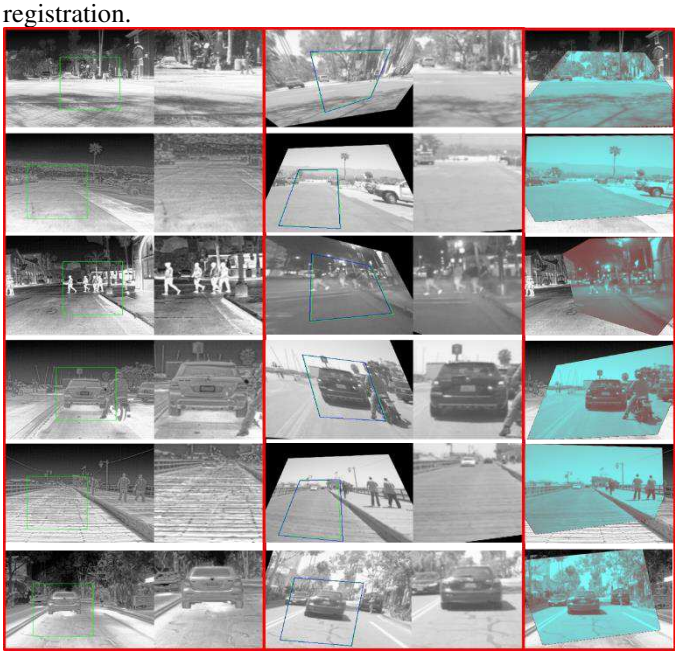
	MACE (pixels)				
Method	No noise	Severity1	Severity2	Severity3	Severity4
PFNet[8]	9.81	11.82	14.6	21.10	24.15
DHN[6]	10.07	10.42	11.41	15.47	19.91
DHN-TB	9.23	10.02	10.95	14.88	19.11
Ours(1)	7.18	8.17	10.60	14.70	18.38
Ours(2)	4.73	5.67	7.46	11.45	17.41
Ours(3)	3.83	4.71	6.34	10.76	16.67
Ours(4)	**3.48**	**4.23**	**6.28**	**10.56**	**15.89**

As can be seen from the TABLE I (Ours(i) means Dense-i-Cascade model), although PF-Net[8] performs well in the registration of mono-modality images, its performance will be greatly affected when it comes to cross-modality images. The offset error rate can reach 16.49% (21 / 128) in the case of severity 3. And Our 4-cascades model's registration error rates are all below 10% for all cases.

Fig.8 shows a registration case: columns (a)-(i) show the tagret, registration effect of PF-Net[8], DHN[6], DHN-Twin-Branch, Ours (4 Modules), thermal image, the thermal patch, visible image, the registrated visible patch by Ours (4 Modules). The rows 1-5 represent the noise severity ranging from 0-4. The quality of registration can be evaluated by comparing the red areas shape similarity of columns (b)-(e) to target column (a) or compare the similarity of blue and green frames in column (g).

It's not hard to see the red areas in column (e) are most similar to those in column (a), which also represents our method is better than others. Column (g) and column (i) show the registration effects more intuitively.

Here are some other examples of registration with our method shown in Fig.11. The similarity of blue ane grenn target in the column (b) can be used to evaluated the quality of registration.



(a)Thermal Image and Patch (b)Visible Image and Patch (c) Effect

Fig.11. Registration effect examples

V. CONCLUSIONS

In this paper, in order to solve the problem of cross-modality registration between thermal and visible images, we proposed a Dense-Cascade neural network. Start from the traditional registration network, we use independent branches to replace the shared weight structure. To further improve the registration precision, we introduced a cascade structure to compensate the residual error. Moreover, we applied 1-4 levels of gaussian noise on the thermal images to verify the robustness of our model, the experiments demonstrate that our model is less affected by noise environment with respect to other methods. Although our work suggests a promising solution, there is still a large room for improvement. For example, there is no general method for determining the number of cascades, which could be further studied in detail, in the future we can also consider our work to other tasks like object-detection, and automatic driving.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 42276187) and the Fundamental Research Funds for the Central Universities, China (Grant No.3072022FSC0401).

REFERENCES

[1] Low D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60: 91-110.

[2] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]//2011 International conference on computer vision. Ieee, 2011: 2564-2571.

[3] Ma J, Zhao J, Jiang J, et al. Locality preserving matching[J]. International Journal of Computer Vision, 2019, 127: 512-531.

[4] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6): 381-395.

[5] Barath D, Matas J, Noskova J. MAGSAC: marginalizing sample consensus[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 10197-10205.

[6] DeTone D, Malisiewicz T, Rabinovich A. Deep image homography estimation[J]. arXiv preprint arXiv:1606.03798, 2016.

[7] Erlik Nowruzi F, Laganiere R, Japkowicz N. Homography estimation from image pairs with hierarchical convolutional networks[C]//Proceedings of the IEEE international conference on computer vision workshops. 2017: 913-920.

[8] Zeng R, Denman S, Sridharan S, et al. Rethinking planar homography estimation using perspective fields[C]//Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part VI 14. Springer International Publishing, 2019: 571-586.

[9] Zhang J, Wang C, Liu S, et al. Content-aware unsupervised deep homography estimation[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. Springer International Publishing, 2020: 653-669.

[10] Ding X, Zhang X, Ma N, et al. Repvgg: Making vgg-style convnets great again[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13733-13742.

[11] Song X, Guo H, Xu X, et al. Cross-modal attention for MRI and ultrasound volume registration[C]//Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part IV 24. Springer International Publishing, 2021: 66-75.

[12] Excitation Module for Cross-Modality Registration of Cardiac SPECT and CT[C]//Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VI. Cham: Springer Nature Switzerland, 2022: 46-55.

[13] F. Team, et al., Free flir thermal dataset for algorithm training. URL https://www.flir.com/oem/adas/adas-dataset-form/

[14] Li H, Wu X J. DenseFuse: A fusion approach to infrared and visible images[J]. IEEE Transactions on Image Processing, 2018, 28(5): 2614-2623.

[15] Tang H, Yuan C, Li Z, et al. Learning attention-guided pyramidal features for few-shot fine-grained recognition[J]. Pattern Recognition, 2022, 130: 108792.

[16] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.