Panoramic Image Generation using Deep Learning

Dias Issa, Magzhan Gabidolla

***Abstract*—A traditional approach for panoramic image generation consists of a random sample consensus (RANSAC) algorithm on scale-invariant feature transform (SIFT) feature correspondences to generate a homography matrix between two images. Although producing adequate results for some type of images, hand-crafted SIFT features are not robust enough for highly varying natural images and the iterative RANSAC algorithm with its randomness does not always find the desired homography matrix. Recently, deep neural networks have been producing significant results in many challenging computer vision problems by learning features from large amounts of data. However, to the best of our knowledge, no previous work applied deep learning to the problem of panoramic image generation, and very few works addressed the problem of feature correspondences and homography matrix estimation with deep learning features. This paper attempts to generate panoramic images by extensively experimenting with various approaches of using deep learning.**

I. Introduction

Panoramic image generation consists of the process of stitching two or more images with the aim of producing a larger seamless image. Apart from the straightforward application of generating images with a larger field of view, image stitching methods are widely used in real-time applications with video processing, video summarization, and in medical imaging domains[1], [2]. For the most part, the techniques used for image stitching can be categorized into two broad classes: the direct approach, which work directly with pixel intensities across images, and feature-based techniques, which extract and match hand-crafted features between images. The effectiveness of direct techniques is rather limited, due to high computational costs and limited accuracy. The widely accepted method for panoramic image generation is the combination of SIFT feature correspondences with the RANSAC algorithm to generate a homography matrix. One may speculate that the main issue of combining two images could lie in the detection of similar features, which belong to both images. However, this assumption is incorrect. Indeed, the advent of deep learning motivated researchers to develop models for automatic feature detection in order to get better results for point-matching between two images and replace traditional approaches, which utilize hand-crafted features. Nevertheless, the results indicate that the enhancement in point-matching does not always contribute to the improvement of the overall process of image stitching [3]. This could happen because of the distinction between the two images in terms of illumination or various differences generated by the change in space performed by a photographer. Therefore, it could be said that the selection of best-fit features among all features, which are recognized as similar, is the main task during the process of image stitching [4].

In this work, we propose a new approach for image stitching utilizing deep learning techniques. Our model receives two input images with some overlap and outputs the transformation matrix that could be applied to combine the images into one panoramic image. We claim that our work can potentially outperform other traditional and deep learning approaches by its generalization, applicability, and speed.

II. Related works

There are a considerable amount of works in the field of image stitching utilizing a traditional approach with hand-crafted features, and very few works, which implement deep learning techniques for this task. The most popular traditional approaches currently are SIFT[5] and RANSAC [6]. SIFT [5] is a satisfactory technique for identification of initial correspondences, however, alone this technique does not produce adequate results for image stitching. At the same time, RANSAC [6], which was developed in 1981, still remains the state-of-art technique in this field. However, its basic concept of randomness, potentially long estimation time and presence of the space for further improvement motivates researchers to investigate better techniques.

Moo et al. [4] tried to solve this issue through the application of a deep learning approach. The authors utilized a deep neural network with multiple layers in order to find good correspondences between two images, which further could be applied for image stitching. Moo et al. [4] utilized two small sets of images taken outdoor and indoor, respectively. Though, authors state that they outperformed the state-of-art results, their approach requires supervision and still utilizes RANSAC for post-processing [4]. Additionally, their model performs adequately only on the restricted set of images and cannot be generalized further [4].

Perhaps the first ever work that uses only deep learning features to estimate the homography matrix was presented in [7]. By cropping and randomly perturbing the corners of the cropped part, the authors of the paper generate pairs of images with the true homography matrix estimated from the corners. According to the authors, it would be ineffective to provide the homography matrix as the final output for deep neural networks, because of the high variability in the values of the individual entries. Instead, the differences in the horizontal and vertical directions of the corresponding four pixels of the two images were used as the output, as there is one-to-one correspondence between this representation and the homography matrix. The architecture of their deep learning model, HomographyNet is similar to the standard image classification networks, with the input being two grayscale images stacked channel-wise. As an evaluation of their network, mean average corner error is reported with the comparison to ORB + RANSAC combination, where the result of the HomographyNet were better better by 21%.

III. Dataset

To our best knowledge, there is no available, free, and large enough datasets designed for image stitching task. Therefore, we decided to construct our own image series database. We chose Oxford Buildings dataset as a basis to generate the appropriate inputs and outputs. By taking overlapping small portions from one larger image, and trying to stitch those overlapping images, we generated 7000 training and 1000 testing samples. Our generated database has four different forms and consists only from scenes taken outside. Depending on the used architecture, the form of the dataset differs.

IV. Proposed techniques

We propose two different ways for the application of deep neural networks to perform image stitching. The first is a hybrid approach that utilizes handcrafted features as input data for the deep learning part that produces output values utilized during image stitching process. The second is an end-to-end deep learning methodology. Therefore, in this case, the deep neural network is responsible for the whole process starting from the feature extraction and ending with the panoramic image generation. Each technique has its own architecture and input, output values. The approaches are described in details in subsections below.

A. LSTM architecture

The first approach is based on the combination of Long-short Term Memory (LSTM) neural networks with SIFT (Figure 1). The input layer of our model receives features extracted from two images using SIFT algorithm and outputs the homography matrix, which in turn will be used to generate the panoramic image. In order to fix the input size for the model, k-means algorithm is applied to have only 64 representative features. As for the values of the descriptors, the average value is taken. In the future, we will consider using the weighted average. With SIFT features extracted from two images, best correspondences between pairs of feature descriptors are found. By ordering the best matching points in pairs, the coordinates of the points are fed as input to the neural network. For the training, the homography matrices generated from the RANSAC algorithm are used as targets.

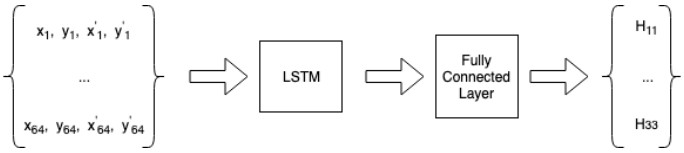


Fig. 1: LSTM architecture

The first experimented architecture is the stack of two LSTM layers with a hidden size of 512. The network is trained for 100 epochs with Adam optimizer and a learning rate of 0.01. The loss function used is the MSE loss. The input images consisted of 80% overlapping left and right images. For training, 7000 images are used, and for testing 1000 images are used. Figure 2 shows the loss during training for specific batches. The loss does not show the expected decrease, with being quite large for some inputs.

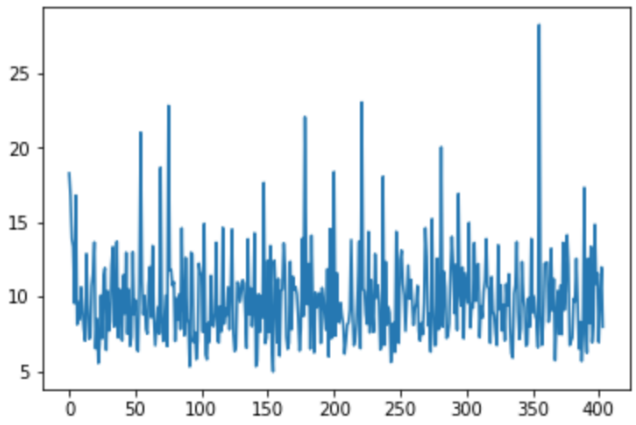


Fig. 2: Loss during training

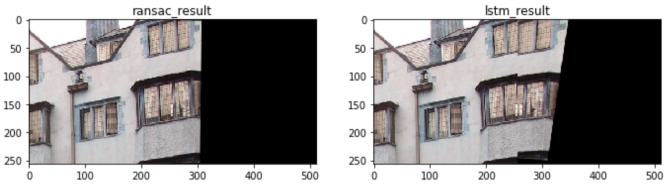


Fig. 3: Example image of 80% overlap by RANSAC and LSTM

Although the loss in the LSTM is not decreasing, visually, the resulting images generated using the homography matrices of the model are comparable to the RANSAC (Figure 3).

The same procedure, however, with a deeper model, was applied to images with 70% and 60% overlaps. The results are also comparable with the RANSAC (Figure 4, 5).

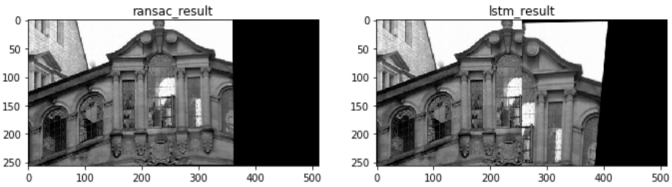


Fig. 4: Example image of 70% overlap by RANSAC and LSTM

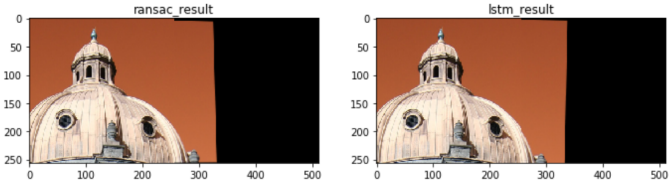


Fig. 5: Example image of 60% overlap by RANSAC and LSTM

B. CNN architectures

The second approach utilizes 2D Convolutional Neural Network (CNN) for the end-to-end panoramic image generation process. The model receives two input images with 80% overlap represented by a 2D array of numbers, where values of the right image are stacked after the left image values, and the target image is the panoramic image that is generated using RANSAC. For convenience, the panoramic image has the same dimensions as the combination array of two input pictures. This is achieved by padding it with 0 from the right side.

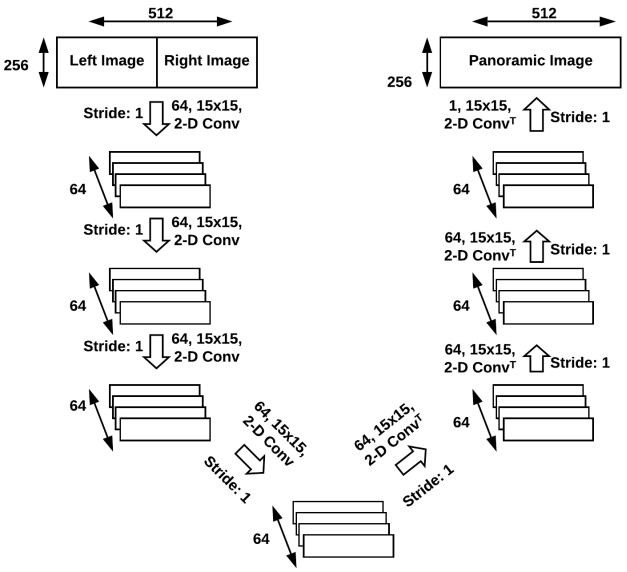


Fig. 6: The first CNN

The first CNN architecture is depicted in Figure 6. It consists of 4 convolution layers with 64 kernels of size 15x15, followed by 3 transpose convolution layers with the same parameters. The final layer has only 1 kernel with size 15x15. The architecture allows input and output images to have the same dimensions. This model was trained with 7000 training image set and 2000 testing image set. The batch size was 64, while the number of epochs was 500. Adam optimizer with the learning rate of 0.001, mean squared error loss function and RELU activation functions after each convolution layer were utilized. Though, the observed continuous increase in accuracy at the beginning, it reached its peak at the value of 40% and did not go further. It was found that the model had not learned to stitch the images but learned to make some of the pixels of the resulted image have a value of 0. Subsequently, these pixels coincided with the padding pixels of a panoramic image and contributed to the value of accuracy. It was decided to change the topology of the network in order to get better results.

Furthermore, the CNN architecture with a gradually increasing number of filters in each successive layer was adopted. The topology of this CNN is illustrated in Figure 7. Its architecture is similar to the architecture of the model described above, the only differences are that the number of filters increments in

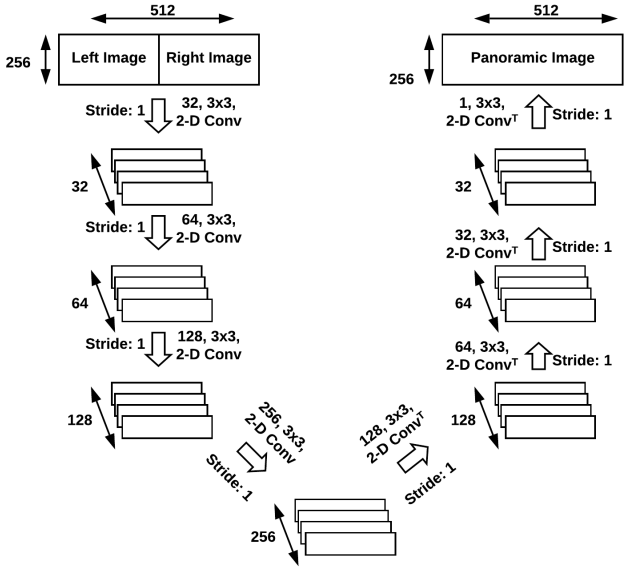


Fig. 7: The second CNN

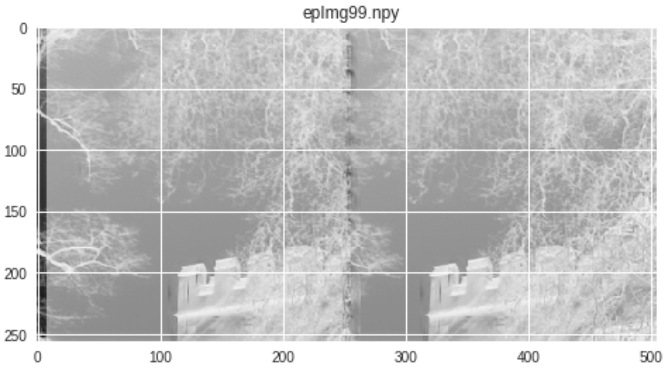


Fig. 8: Example of stitched image, performed by the second CNN

size from 32 to 256 for each following layer, and that the kernel size is 3x3. The network was trained and tested with the same sets of the images as before, however, the number of epochs was reduced to 100 because it was found that after that loss and accuracy values do not show any enhancements. The resulting image of the last epoch is depicted in Figure 8. It could be clearly seen that the CNN learned to stitch images one after another, nevertheless, the goal of generating a panoramic image was not achieved. Additionally, further experiments with the topology of CNN did not give better outcomes. Thus, it could be stated that the application of pure CNN to raw pixels of the input images in order to generate a panoramic image as the output is a dead end direction for research. However, the application of time distributed CNN in combination with LSTM for image stitching could provide better results.

Finally, we utilized the pre-trained VGG16 neural network [8] for the process of extracting visual features from the input images. After that, 4 fully connected layers were

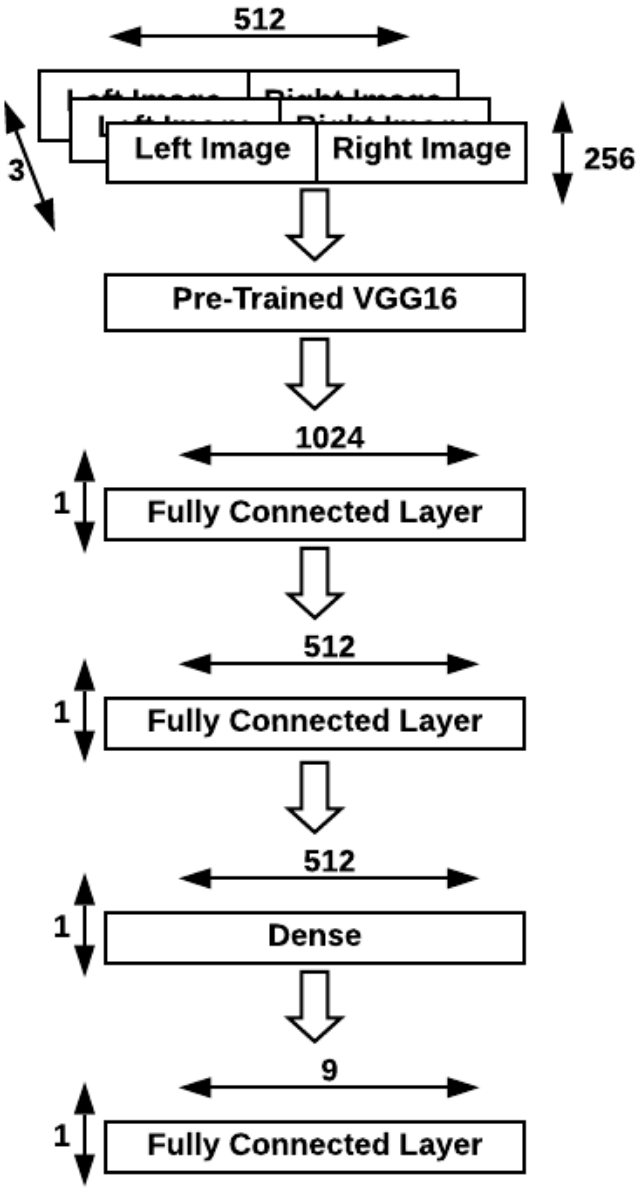


Fig. 9: CNN with VGG16 as a feature extractor.

added with 1024, 512, 512 and 9 units respectively. Therefore, the last layer outputs the corresponding homography matrix in order to stitch properly input images. All weights of the VGG16 were frozen, therefore, only the weights in the fully connected layers were trained. The architecture of the model is depicted in the Figure 9. The model was trained for 100 epochs using 1700 input images with 80% overlap and tested with 300 samples. Then the predicted homography matrices were utilized in order to produce panoramic images. The input images and the panoramic image generated out of them using these homography matrices are illustrated in Figures 10 and 11 respectively. From this we could see that the network has significantly better results than in the previous case, however, the image stitching procedure is still not appropriate. We propose that adding more layers and training for a larger number of epochs could enhance the results.

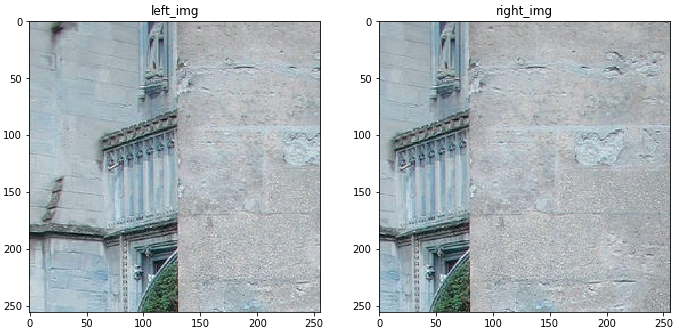


Fig. 10: Input images for the last CNN model

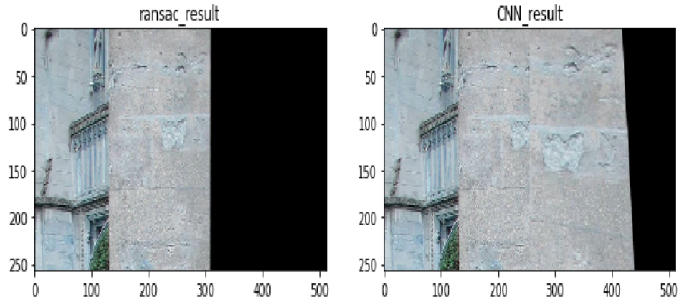
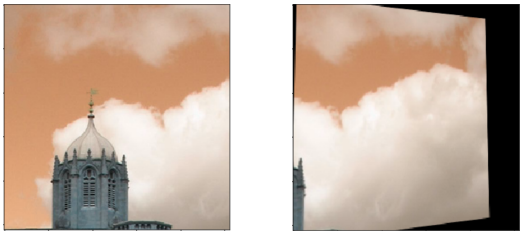


Fig. 11: Panoramic image generated by the last CNN model

C. Dataset with Random Perturbation

Because of the way we are generating the dataset with only horizontal overlap changes, the values of the individual entries of the homography matrices were the same except for the horizontal translation. And since we only experimented with 4 different overlap ratios, in total, there were only 4 different types of homography matrices. In order to create real and more complex data with high variability, we perform random perturbations similar to the work in [7]. We still choose 50%, 60%, 70% and 80% overlap images, but, in this case, the second image are subjected to random transformations. Given two overlapping images, four common points are chosen. The coordinates of those four points from the second (right) image are perturbed in the range $(\rho, -\rho)$. The homography matrix, estimated from the original coordinates and the perturbed coordinates, are applied to the whole second (right) image. Now, two images have some random overlap because of the changes, and the homography matrix responsible for stitching are completely different from sample to sample. Instead of using SIFT + RANSAC combination to find that homography matrix, as we previously did, we can find the exact homography matrix from the four corresponding points which are known by construction. Similar to the idea in [7], the corner differences of input and output images are more preferable than the exact homography matrix as outputs for deep learning. For this reason, corner coordinates of the second image are transformed using the homography matrix to find where those pixels map in the panoramic image. Having those pixel coordinates in the stitched image, x and y coordinate



(a) Left image (b) Right image

Fig. 12: Input images

differences are found and saved as outputs for training deep neural networks. Figure 12 shows two pair of images to be stitched and in Figure 13 one can observe the ground truth stitched image.

This data generation procedure was applied to the Oxford Buildings dataset with the crop size of 256 and with overlaps of 50%, 60%, 70% and 80% mixed. In total we have 47000 input-output pairs.

Fig. 13: Stitched ground truth image

D. Deep Homography Estimation

Inspired by the work in [7], an architecture similar to AlexNet was used with two grayscale images stacked channel-wise. In [7] the output is corner differences in x and y directions, which consists of 8 numbers. Unlike that work, we did not use all of the numbers as the output of the single deep neural network, because the distributions in the x and y directions are different. For this reason, two separate neural networks are created with the same architecture for x and y coordinate differences. The architecture of the network consists of the convolutional layers with doubly increasing filter sizes till 512 followed by two fully connected layers to output 4 numbers. 43000 images were used for training, 2000 images were used for validation, and the remaining 2000 images were reserved for testing. The outputs were normalized between 0 and 1. For loss function, L1 absolute value loss was chosen, because MSE loss produces very small numbers for numbers in the range (0, 1). Figure 14 shows how the loss for x coordinate difference changed during training, while Figure 15 illustrates the training loss for y coordinate difference. For test images, the average absolute value loss for x corner difference was 7.31 pixels, and for y corner difference it was 1.07 pixels.

When the images and outputs were visualized, in most of the cases the results of the deep learning were comparable with the ground truth. SIFT + RANSAC combination is also used for test images to compare the performance with the baseline. Figure 16 shows comparable result with the RANSAC, and Figure 17 shows where deep learning outperforms bad RANSAC example. In Figure 18, we see the case where RANSAC could not find the homography matrix, and in Figure 19 there are even no SIFT features. These examples illustrate the superiority of this deep learning approach over traditional SIFT + RANSAC combination.

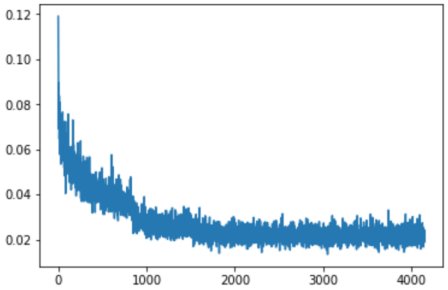


Fig. 14: Training loss for x coordinate differences

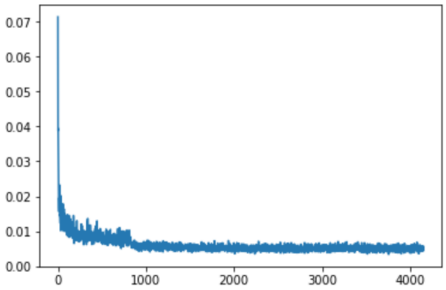


Fig. 15: Training loss for y coordinate differences

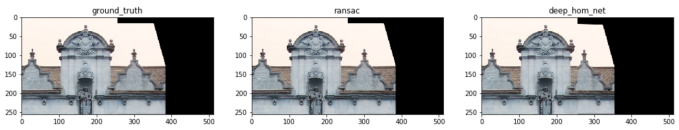


Fig. 16: Comparable results

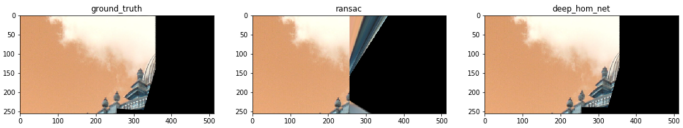


Fig. 17: Bad RANSAC example

V. Conclusion

To conclude, based on the evidence given above, we could say that the application of deep neural networks for image stitching is a promising direction of research. The ability of deep neural networks to produce better results by learning features and utilization of large datasets could further improve the procedure of image stitching in comparison with the state-of-art techniques, which utilize handcrafted features. Nevertheless, this area remains challenging, therefore, the process of generating panoramic images should be investigated in details

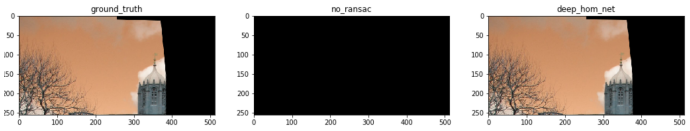


Fig. 18: No homography by RANSAC

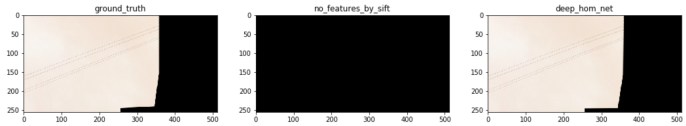


Fig. 19: No SIFT features

in order to theoretically find out the best configuration for deep neural networks. Additionally, more experiments in this field are also needed.

REFERENCES

[1] E. Adel and M. Elmogy, "Image stitching based on feature extraction techniques: A survey," 2014.

[2] M. V. Wyawahare, P. Patil, and H. K. Abhyankar, "Image registration techniques: An overview," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 2, 09 2009.

[3] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1482–1491.

[4] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2666–2674.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

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

[7] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," *CoRR*, vol. abs/1606.03798, 2016. [Online]. Available: http://arxiv.org/abs/1606.03798

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.