Deep Learning Framework for Optical and Microwave Image Matching

*

***Abstract*—Remote sensing imagery captured purely in the optical spectrum may have environmental disturbances such as clouds. In order to get a full understanding of a scene, it is important to capture multi-spectral imaging data so that we get complementary information about a particular scene. This however poses another challenge which is the matching and registration of multi-spectral images taken of the same scene. In this paper, we present a multi-modal image matching framework that performs image matching on optical and microwave remote sensing images of the same scene. Our framework uses an image-to-image translation network that generates a microwave image from an optical image and then uses the generated microwave image for matching. This approach results in improved matching accuracy. A common issue with deep-learning-based image matching approaches is the need for labelled image datasets. We overcome this hurdle by training our framework on unlabelled remote sensing image data which is of abundance. We also employ a method involving angles and z-scores to identify and discard potential false matches.**

***Index Terms*—Image Matching, Image Registration, Multi-spectral Imaging, Convolutional Neural Networks, Generative Adversarial Networks, Remote Sensing.**

I. Introduction

REMOTE sensing is the process of detecting and monitoring the physical characteristics of an area by measuring the reflected and emitted radiation at a distance (typically from a satellite or an aircraft). Special cameras collect remotely sensed images, which help researchers "sense" things about the Earth.

Several technological advances in this area have taken place over the past few decades such as sensor system improvements, cost-effective computing systems, efficient algorithms for interpreting Geo-spatial data, and so on.

Consequently, due to increased efficiency, reliability, and robustness of data obtained from remote sensing, the applications have grown tremendously, including but not limited to large-scale ride-sharing services, high-resolution maps for autonomous driving, the monitoring and modelling of changes over time for precision farming, and urban planning [1].

The advancements in remote sensing technologies have brought with it an abundance of data. To make use of this vast remote sensing data, there is a need to develop advanced and efficient algorithms. However, the amount of information made available using only optical imaging may be limited due to several environmental disturbances. Hence, collecting complementary information from multi-modal imaging data gives a better understanding of an image scene or a specific object. Images of the same scene contain a complement of information. An optical image may contain information that a microwave image lacks and vice versa. As a result, data fusion has become a key topic within the field of remote sensing [1].

To make the most out of multi-modal image data we must have Geo-referenced and precisely co-registered multi-spectral images. So far, data fusion has largely been constrained to applications across similar modalities mainly because we need to be able to determine corresponding points, and perform alignment of data sources before data fusion. For image-based data, these correspondences are obtained through image matching, whereby common points are co-located across a set of images.

Our work aims to provide a solution to data fusion between optical and microwave images. In this paper, we present a deep learning-based framework for image matching between optical and microwave images. In this work, we propose a multi-modal image matching method that employs a GAN to generate microwave imagery corresponding to the input optical image, which we then feed into an image matching network. We also propose a method based on angles and z-scores that operates on the matches generated by the matching network to identify and eliminate potential false matches.

We train our GAN using a paired and unlabelled image dataset and as a part of the image matching network, we use a CNN that has been pre-trained for image classification tasks as the base of our matching network. This approach to training enables us to avoid a common drawback of deep learning-based approaches to image matching which require labelled image datasets.

In this paper, we first discuss previous approaches to image matching and image registration and point out some shortcomings in these approaches. We then discuss the methodology used in our work.

II. Related Work On Image Matching

A. SIFT

Scale-invariant feature transform, abbreviated as SIFT, is used for extracting invariant features from an image. It does so by transforming image data into scale-invariant coordinates with respect to local features [3].

This algorithm accepts an image as input and outputs a multi-dimensional feature vector consisting of feature descriptors. SIFT features are invariant to image scale and rotation and are prominent across various affine distortions, 3D viewpoint changes, noise, and alterations in illumination.

As shown in recent studies, the performance of SIFT for cross-spectral remote sensing is affected quite significantly by intensity differences among spectral images. Various approaches such as using scale-restricted SIFT [4], modifying the SIFT descriptor [5], and so on have attempted to tackle this problem. SIFT also uses hand-crafted feature descriptors and is largely constrained to specific scene geometry and sensor

resolutions and is computationally expensive. By using Deep Learning approaches to perform image matching, we bypass the need for expensive computation and get better and more accurate results.

B. Deep Learning Methods

In recent times, Convolutional Neural Networks have been used to improve image matching techniques.

Since there is a non-linear relationship between corresponding pixels in cross-spectral images, SIFT and other related algorithms such as SURF [6], AKAZE [7], etc. which traditionally have been used for comparing image patches in mono-spectral settings are limited in their matching performance. As stated in [8], pixel intensity variations in the Long Wavelength IR spectrum pertain to variation in the objects' temperature, whereas pixel intensity variations in the visible spectrum cause colour and texture variations. Moreover, their computational complexity is another cause of concern when testing on larger datasets.

[8] used a CNN for comparing the similarity of cross-spectral images using image patches. They evaluate different deep network architectures to measure similarity. They evaluated the 2-channel network, the Siamese network, and the Pseudo-Siamese network. They built a cross-spectral image patch dataset using the public Visible-Near IR scene dataset [9]. The patches were extracted around interest points detected using SIFT in the visible spectrum images.

Results from their experiment showed that the 2ch network performed significantly better than the Siamese and pseudo-Siamese networks in terms of accuracy, but was also slower. They attributed the performance of the 2ch network to the fact that the information is jointly processed right from the first layer. They were able to use the network trained on a VIS-NIR cross-spectral dataset in a VIS-LWIR dataset, which is significant as data available in the LWIR spectrum is very limited.

Another approach taken in [10] explains the use of feature matching. They implement a deep CNN structure to process the image patch which represents the image feature point and obtain the feature point description of the image. SIFT detectors are used to achieve feature point detection, but a deep CNN is used for feature descriptors instead. Image matching was done using the KNN algorithm. The KD tree is established using feature descriptors. Finally, RANSAC [11] is used to reject abnormal data. The UBC dataset [12] is used to train the model and the W1BS dataset [13] is used to test it. SIFT and SURF were compared with their method and were found to be inferior in terms of sensitivity to geometry and appearance.

In the approach taken in [14], a GAN is used for training data augmentation. They train a deep matching network that is capable of taking as input an optical image patch and a SAR image patch and determining if the two given image patches match. In order to find matching points between a SAR and an optical image, they extract SIFT key points and use these points as centers to obtain the SAR and optical image patches. They use the trained deep matching network to find candidate matching points among the points identified by SIFT. Finally, false matches are eliminated by the use of correlation constraints and geometric constraints.

A drawback of the approach taken in [14] is that SIFT key points are used as a basis for finding matches between images. As mentioned before, SIFT uses hand-crafted feature descriptors and is largely constrained to specific scene geometry and sensor resolutions [1]. SIFT based methods also have the issue of generating insufficient feature points. These issues limit the application of SIFT in image registration.

Overall, a drawback of image matching using deep learning methods is the scarcity of labelled microwave and optical remote sensing image datasets.

C. Our Contribution

To overcome the previously mentioned drawbacks, in this work, we use a modified network based on the network in [2]. In [2], the authors use a pre-trained VGG-16 architecture which is well established to be strong in finding features [15]. We, however, use the EfficientNet [16] for this purpose as it gives more accurate matches between optical and microwave images. Using a model that has been pre-trained on image classification tasks enables us to avoid the previously mentioned drawbacks of SIFT based approaches to image matching and registration and avoid the need for training datasets.

For image matching and we also employ a method involving angles of line segments of matches to determine false matches as well as a microwave image generating network (see Fig. 1.). We train a GAN and use the generator as the microwave image generator. Since the GAN can be trained on unlabelled microwave and optical images, we do not require labelled remote sensing images, thereby overcoming the common drawback of needing labelled image datasets.

III. Our Method

A. The Objective

The objective of our work was to create a deep learning-based framework capable of taking as input two images of the same scene, with one of the images in the optical spectrum and the other in the microwave spectrum and as output give a list of pairs of image coordinates where the image matching algorithm identified matching features between the 2 images.

More precisely, the algorithm gives as output a list of pairs of coordinates. Say, for example, that one of the pairs in the list is $[M : (120, 121), O : (500, 512)]$. This means that the algorithm detected a feature in the neighbourhood of coordinate $(120, 121)$ in the microwave image and that its corresponding feature in the optical image was detected in the neighbourhood of coordinate $(500, 512)$.

Having a list of such matches gives us matching features and key points between the optical and microwave images that can be used for co-registration of the images.

B. The Framework

We propose a deep learning approach that is capable of taking two images (optical and microwave images) and is

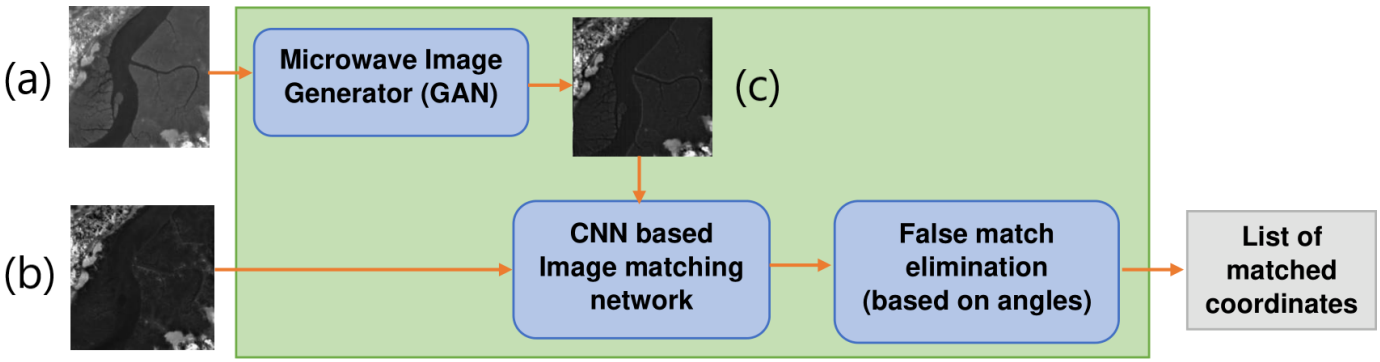

Fig. 1. Pipeline of our proposed algorithm. Image (a) is the input optical image. Image (b) is the input microwave image of the same scene. Image (c) is an intermediate image that is generated by the Microwave Image generator.

capable of giving as output a list of image coordinates where matching features were identified.

This approach makes combined use of an image-to-image translation network (referred to as the microwave image generator) and a CNN-based matching network. The image-to-image translation network, given an optical image as input, is capable of generating its corresponding microwave image.

1) Microwave Image Generator: The first step is to take the input optical image and use it to generate its corresponding microwave image. For this purpose, we train a GAN using a paired, unlabelled image dataset [17].

2) CNN Matching Network: We take the generated microwave image and compare it to the input microwave image to get the matching feature points between the true microwave image and the generated microwave image. This step is done with the help of a CNN-based matching framework based on the one presented in [2].

Since the generated microwave image and the input optical image have their corresponding features in the same coordinates, **by finding the matching points between the generated and the true optical image, we effectively get the matching points between the original optical and original microwave images given as input.** (The microwave generator network does not alter locations of any features in the optical image given as input). The entire framework is summarized in Fig. 1.

As previously mentioned, we have used the Sentinel-2 images from the SEN12MS dataset [17] as the unlabelled dataset to train the GAN. We discuss how this dataset was used in more detail in the following section.

IV. Dataset Creation

We use Sentinel-2 images from the SEN12MS dataset [17] as the unlabelled dataset to train the GAN. Shown in Table I are the spectral bands contained in an image taken by Sentinel-2 [18].

The images captured by Sentinel-2 have 13 spectral bands. We extract band 4 and use it as the microwave image. By extracting out bands 3,4 and 8 and concatenating them, we have the corresponding 3 channel optical image.

TABLE I
The 13 Bands of a Sentinel-2 Image

Sentinel-2 Bands	Central Wavelength (μm)	Resolution (m)
Band 1 - Coastal aerosol	0.443	60
Band 2 - Blue	0.490	10
Band 3 - Green	0.560	10
Band 4 - Red	0.665	10
Band 5 - Vegetation Red Edge	0.705	20
Band 6 - Vegetation Red Edge	0.740	20
Band 7 - Vegetation Red Edge	0.783	20
Band 8 - NIR	0.842	10
Band 8A - Vegetation Red Edge	0.865	20
Band 9 - Water Vapor	0.945	60
Band 10 - SWIR - Cirrus	1.375	60
Band 11 - SWIR	1.610	20
Band 12 - SWIR	2.190	20

TABLE II
Encoder Architecture

No.	Layer Type	Parameters	Input Size	Output Size
1	Convolutional Layer	Kernel Size = 2x2 No. of kernels = y Stride Length=2	[2n,2n,x]	[n,n,y]
2	Batch Normalization Layer	-	[n,n,y]	[n,n,y]
3	ReLU activation Layer	-	[n,n,y]	[n,n,y]

The images extracted from the SEN12MS dataset are 16-bit images. This means that the pixel value can vary from 0 to 2^{16}, with 0 corresponding to the lowest possible intensity and 2^{16} corresponding to the highest possible intensity. In order for the image data to work well during the training of the deep learning algorithms, we normalize these intensity values between $[0, 1]$.

We also convert the 3-band RGB optical image to its single-band gray-scale equivalent. By doing this, we reduce the amount of data that we have to handle.

TABLE III
DECODER ARCHITECTURE

No.	Layer Type	Parameters	Input Size	Output Size
1	Deconvolutional Layer	Kernel Size = 2x2 No. of kernels = y Stride Length = 2	[n,n,x]	[2n,2n,y]
2	Batch Normalization Layer	-	[2n,2n,y]	[2n,2n,y]
3	ReLU activation Layer	-	[2n,2n,y]	[2n,2n,y]

TABLE IV
RESNET BLOCK ARCHITECTURE

No.	Layer Type	Parameters
1	Convolutional Layer	Kernel Size = 3x3 No. of kernels = 256 Stride Length = 1
2	Batch Normalization Layer	-
3	ReLU activation Layer	-
4	Convolutional Layer	Kernel Size = 3x3 No. of kernels = 256 Stride Length = 1
5	Batch Normalization Layer	-
6	Concatenation	-

TABLE V
GENERATOR ARCHITECTURE

No.	Architectural Unit	Input Size	Output Size
1	Encoder1	[256,256,1]	[128,128,32]
2	Encoder2	[128,128,32]	[64,64,64]
3	Encoder3	[64,64,64]	[32,32,128]
4	Encoder4	[32,32,128]	[16,16,256]
5	Resnet1	[16,16,256	[16,16,256]
6	Resnet2	[16,16,256	[16,16,256]
7	Resnet3	[16,16,256	[16,16,256]
8	Decoder1	[16,16,256]	[32,32,128]
9	Decoder2	[32,32,128]	[64,64,64]
10	Decoder3	[64,64,64]	[128,128,32]
11	Decoder4	[128,128,32]	[256,256,1]

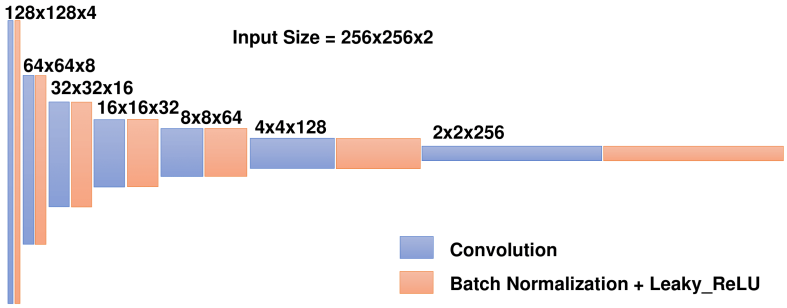

Fig. 2. The discriminator model of the GAN

V. DETAILS OF THE FRAMEWORK

A. The GAN

The GAN that we use learns a mapping from observed optical image o and random noise vector z to the corresponding microwave image m. That is, $G : \{o, z\} \rightarrow m$. We train the generator G to produce output microwave images that resemble the real microwave images. An adversarially trained discriminator D is trained to identify copies created by the generator.

1) The Generator Model: The generator model G is composed of 4 encoders, 3 resnet blocks and 4 decoders. Each encoder has the architecture shown in Table II.

When an image is given as input to the generator, it first goes through the four encoders and undergoes changes in dimensions as shown in Table V. The input optical image given to the first encoder is first resized to a 256 by 256 image if required. The $[16, 16, 256]$ tensor generated by the 4 encoders is then given as input to the 3 resnet blocks. The output of the third resnet block is then given to the first decoder, Decoder1. The output of Decoder4 (with dimensions $[256, 256, 1]$) is the generated microwave image.

Below is the loss function of the generator.

$$-\mathbb{E}_{o,z}[\log(D(o, G(o, z))+\epsilon)]+\lambda\mathbb{E}_{o,m,z}[\|m-G(o, z)\|_1] \quad (1)$$

Where we use an L1 distance loss and λ is set to 100. The training process of the GAN aims to minimize this loss.

2) The Discriminator Model: The discriminator of the GAN takes 2 images as input – an optical and a microwave image and it gives as output the probability that the microwave image is real.

The discriminator, given two images, each of dimensions $[256, 256, 1]$, concatenates them to form a $[256, 256, 2]$ tensor which we give as input to the network shown in Fig. 2. The $[2, 2, 256]$ tensor which we get as output is first flattened into a fully connected layer. This is followed by two more fully connected layers of size 512 and 128 with Leaky_ReLU activations, followed by the final output layer which is of size one and has a Sigmoid activation function to ensure that we get a value between 0 and 1 which represents the required probability.

Below is the objective of the discriminator.

$$\mathbb{E}_{o,m}[\log(D(o, m)+\epsilon)]+\mathbb{E}_{o,z}[\log(1-D(o, G(o, z))+\epsilon)] \quad (2)$$

The training process of the GAN attempts to maximize this objective, while the adversary, the generator attempts to minimize its loss function.

B. The CNN-based matching framework

After the microwave image generation, we use a CNN based network to find feature points and perform image matching between the generated and true microwave images. In order to do this, we perform multiple convolutions on the images and form feature descriptors of the two input images.

The approach we take to generate our feature descriptor is based on the approach taken in [2]. However, rather than using a pre-trained VGG-16 [15] network, we construct the feature descriptors of the image using the outputs of certain layers of a pre-trained EfficientNet b0 network [16]. EfficientNets are accurate and computationally efficient models which achieved top-1 accuracy on the Imagenet dataset and managed to achieve state-of-the-art accuracy on CIFAR-100.

We make this change as EfficientNet b0 often gives us more accurate matches. The pre-trained network that we use was trained on the Imagenet dataset [19]. The pre-trained network is highly capable when it comes to finding strong candidate feature points of an image.

To generate the feature descriptors of an image we use the outputs of the following layers of EfficientNet b0:

- block3b_activation.
- block4b_activation.
- block6a_activation.

The outputs of these layers have the dimensions $(28, 28, 240), (14, 14, 480)$ and $(7, 7, 672)$ respectively. The framework given in [2] includes a feature pre-matching step which involves finding Euclidean distances between the 3 feature descriptors and summing them up. To account for the differences in the output dimensions of the first 2 output layers compared to the third, we include the multipliers of $\sqrt{(672/240)}$ and $\sqrt{(672/480)}$ to the respective Euclidean distances of the block3b_activation and block4b_activation output feature descriptors. That is, we modify equation 4 of [2] as follows.

$$d(x, y) = \sqrt{\frac{672}{240}} d_1(x, y) + \sqrt{\frac{672}{480}} d_2(x, y) + d_3(x, y) \quad (3)$$

After the CNN generates the feature descriptors for the two input images, we compare them to find matches. In order to compare two features, we use the framework given in [2].

C. Identifying false matches using a method of comparison of neighbouring angles

Once The CNN has performed the initial matching between the input microwave image and the generated microwave image, we attempt to identify and eliminate potential false matches using a method of finding angles and z-scores.

As seen in Fig. 3, we draw line segments between the matched points of the two images to denote the matched features in the images. To identify whether a feature point in the reference image (which has been matched to another feature point in the secondary image) has been matched falsely or not, we consider the k-nearest neighbouring feature points of the point in consideration within the reference image (Fig. 4) and we calculate the angles of the lines formed by these k points and their corresponding matching points in the secondary image (Fig. 5). We then normalize these k angles to a normal distribution with $\mu = 0$ and $\sigma = 1$ using equation 4.

$$Z = \frac{X - \mu}{\sigma} \quad (4)$$

Finally, we perform the angle calculation for the point in consideration. If the angle of the point in consideration is not within the range of $\mu \pm (1.5)\sigma$, then we consider the point to be falsely matched.

This method of eliminating potential angles is summarized in algorithm 1. In algorithm 1, n is the number of matched points found between the two images and $X[i]$ and $Y[i]$ contain the coordinates of points in the true microwave image and generated microwave image that were given as input. $indices$ is an array of shape (n, k) where $indices[i]$ contains the k nearest neighbouring feature points from point i.

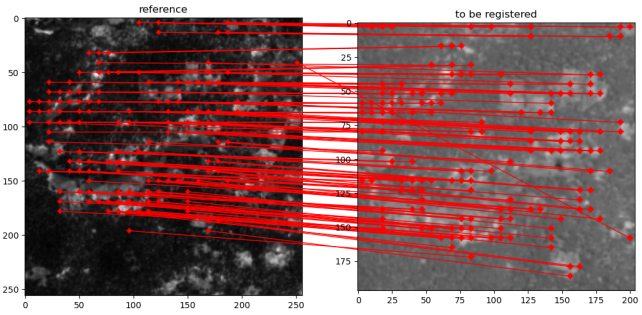

Fig. 3. Matching performed by the CNN based matching network

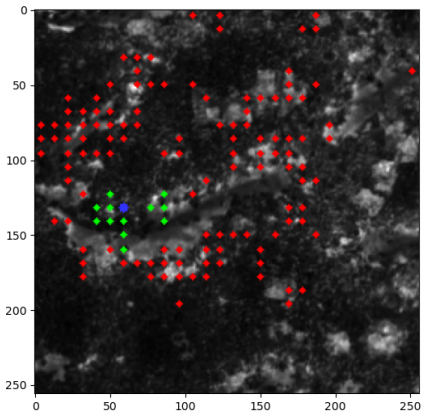

Fig. 4. Feature point in consideration (blue) and its k nearest neighbours (green).

In lines 7 and 9 of algorithm 1, the angles are calculated using the formula,

$$\tan \text{inverse} \left(\frac{X[p][0] - Y[p][0]}{(X[p][1]) - (Y[p][1] + W)} \right) \quad (5)$$

where W is the width of microwave image given as input and $X[i][0]$ is the y of point $X[i]$ and $X[i][1]$ is the x coordinate of $X[i]$.

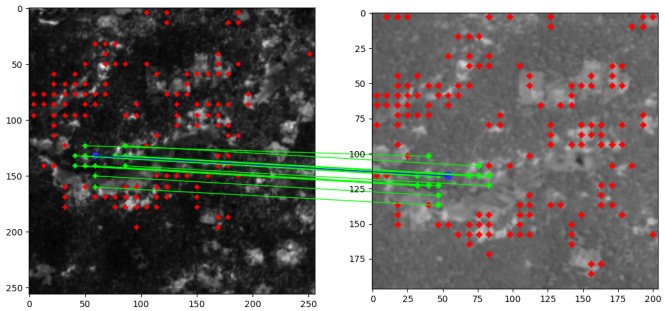

Fig. 5. Angle of the point in consideration (blue) and the angles of its k nearest neighbours (green).

Algorithm 1: Identification of false matches using comparison method involving k-nearest neighbouring angles.

input : An array X and Y of each of shape $(n, 2)$ and k

Initialize $indices$, $angles$, $angle$, ids and $false\_points$;
Use a k nearest neighbours algorithm to assign the indices (with respect to array X) of the k nearest neighbours of feature point $X[i]$ to $indices[i]$;
for $i \leftarrow 0$ **to** n **do**
    $ids \leftarrow indices[i]$;
    for $j \leftarrow 0$ **to** k **do**
        $p \leftarrow ids[j]$;
        $angles[j]$ = angle of the line segment between $X[p]$ and $Y[p]$;
    end
    $angle \leftarrow$ angle of the line segment between $X[i]$ and $Y[i]$;
    $mean \leftarrow$ Mean of the array $angles$;
    $std \leftarrow$ Standard deviation of the array $angles$;
    if $|\frac{(angle-mean)}{(std)}| \geq 1.5$ **then**
        Points $X[i]$ and $Y[i]$ are falsely matched;
        Append i to $false\_points$;
    end
end

output: Array $false\_points$

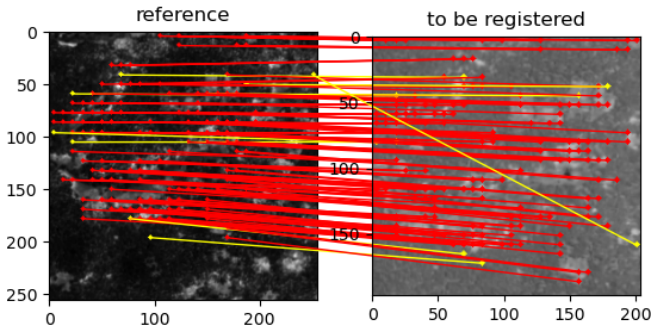

Fig. 6. Potential false matches identified through the angle method (yellow).

An example of the results of this false matching identification is shown in Fig. 6 with $k = 14$. The matches that have been identified to be false through this method are shown with yellow lines. As can be seen, this method is rather effective in detecting false matches.

VI. Experiments

In this section, we test our framework on unseen optical and microwave image pairs generated from the SEN12 dataset [17] and compare its performance with other image matching algorithms. We measure the performance using a precision metric. The precision metric is given in equation 6.

TABLE VI
Precision experiment results

Algorithm	Mean	Min.	Max.	Std Dev.
Feature matching using SIFT	76.36%	49.40%	91.10%	11.41
Feature matching using ORB	86.69%	71.16%	95.74%	5.99
Our proposed framework	94.08%	91.36%	96.55%	1.35

$$\frac{AM}{TM} \tag{6}$$

That is, we find the percentage of accurate matches among the total number of matches.

We perform image matching on a set of optical and microwave image pairs using the following 3 algorithms.

1) Feature matching using SIFT.
2) Feature matching using ORB [20].
3) Feature matching using The framework proposed in this paper.

We pass unseen optical and microwave image pairs which are extracted from the dataset [17] as explained earlier in section IV. We alter the optical images by rotation. Table VI shows the performance of the three algorithms on the test image pairs.

As can be seen from Table VI. Our proposed framework performs significantly better than the SIFT and the ORB algorithms on remote sensing images of the sentinel-2 image dataset [17]. Our network shows a significant increase in accuracy when it comes to finding matches and shows less variability in precision as well across the image pairs used in the experiment.

The reason why we compare the real microwave and generated microwave images instead of directly comparing the real microwave and optical images is to increase the number of matches and accuracy of the matches yielded by the CNN network.

In Fig. 7, we show the result of image-to-image translation performed by our microwave image generator network. The microwave image generator network removes features that are prominent in the optical image, (such as the river in this example) and generates features that would potentially be present in the corresponding microwave image of the same scene.

In Fig. 8, we show the results of the image matching performed by our framework and ORB. From this figure, it is clear that our framework performs significantly better than ORB at image matching.

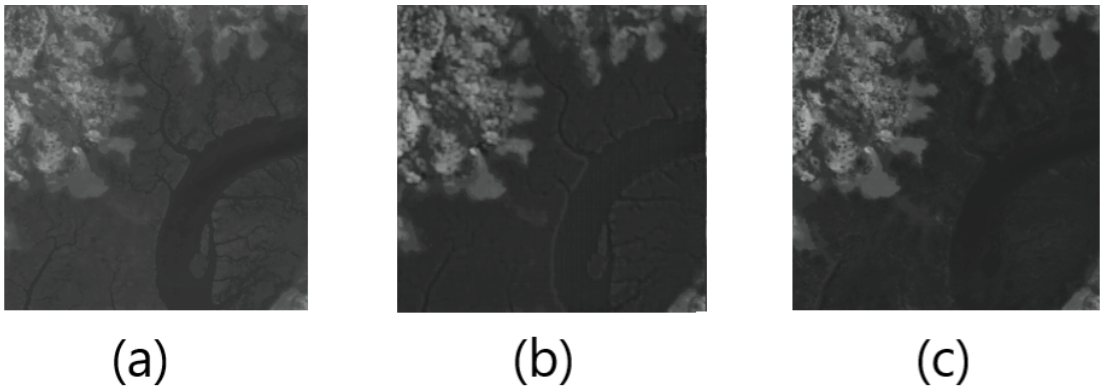

Fig. 7. Result of passing a gray-scale optical image through the microwave image generating network. Image (a) is the input optical image, image (b) is the microwave image generated by the GAN and image (c) is the ground truth microwave image corresponding to optical image (a).

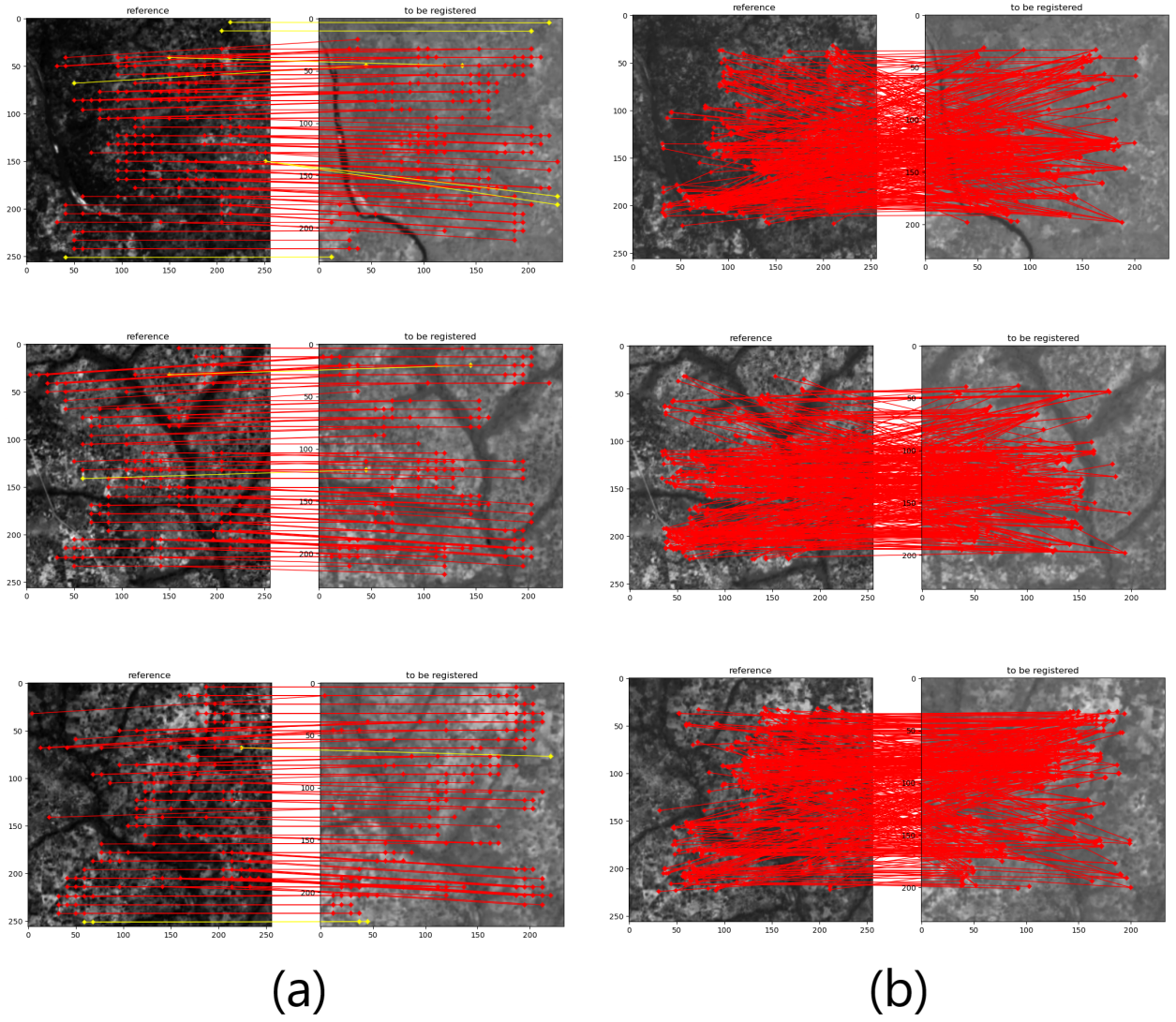

Fig. 8. Comparison between the matching performed by our framework (column (a)) and ORB (column (b)). The red lines indicate matching features in the two images. The yellow line segments indicate the false matches identified by the method of computing neighbouring angles as explained before.

VII. Conclusion

In this paper, we propose a deep learning framework capable of taking an input optical image and a microwave image and accurately finding matching features between the two images. We proposed a GAN which performed image-to-image translation by taking the input optical image and generating its corresponding microwave image. We then propose a CNN-based matching network, based on [2] which is capable of taking the generated and input microwave images and finding matching points/features between them. We also propose a method based on comparing slopes to weed out potential false matches, thereby making the matching more accurate.

By using an image-to-image translation network to generate a microwave image that corresponds to the input optical image and performing image matching using the generated image rather than the input optical image, we improve on the accuracy of the matches. The method of comparing slopes helps us in eliminating any false matches that may have crept in during the image matching phase. It is also worth reiterating that this framework only requires unlabelled remote sensing image datasets for training, which is of abundance.

References

[1] N. M. Merkle, "Geo-localization Refinement of Optical Satellite Images by Embedding Synthetic Aperture Radar Data in Novel Deep Learning Frameworks," Univ. of Osnabrück, Osnabrück, Lower Saxony, Germany, 2018.

[2] Z. Yang, T. Dan, and Y. Yang, "Multi-temporal remote sensing image registration using deep convolutional features," IEEE Access, vol. 6, pp. 38544–38555, 2018.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.

[4] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," in Computer Vision – ECCV 2016, Cham: Springer International Publishing, 2016, pp. 467–483.

[5] S. Saleem and R. Sablatnig, "A modified SIFT descriptor for image matching under spectral variations," in Image Analysis and Processing – ICIAP 2013, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 652–661.

[6] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in Computer Vision – ECCV 2006, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[7] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE Features," in Computer Vision – ECCV 2012, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 214–227.

[8] C. A. Aguilera, F. J. Aguilera, A. D. Sappa, C. Aguilera, and R. Toledo, "Learning cross-spectral similarity measures with deep convolutional neural networks," in 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016.

[9] "RGB-NIR Scene Dataset," Epfl.ch. [Online]. Available: https://www.epfl.ch/labs/ivrl/research/downloads/rgb-nir-scene-dataset/. [Accessed: 22-Apr-2021].

[10] Y. Liu, X. Xu, and F. Li, "Image feature matching based on deep learning," in 2018 IEEE 4th International Conference on Computer and Communications (ICCC), 2018.

[11] 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, 24(6), pp. 381-395, 1981.

[12] "Learning Local Image Descriptors Data," Washington.edu. [Online]. Available: http://phototour.cs.washington.edu/patches/default.htm. [Accessed: 22-Apr-2021].

[13] D. Mishkin, J. Matas, M. Perdoch, and K. Lenc, "WxBS: Wide baseline stereo generalizations," in Procedings of the British Machine Vision Conference 2015, 2015.

[14] D. Quan et al., "Deep generative matching network for optical and SAR image registration," in IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv [cs.CV], 2014.

[16] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional Neural Networks," arXiv [cs.LG], 2019.

[17] M. Schmitt, L. H. Hughes, C. Qiu, and X. X. Zhu, "SEN12MS – A curated dataset of georeferenced multi-spectral Sentinel-1/2 imagery for deep learning and data fusion," arXiv [cs.CV], 2019.

[18] "Sentinel-2A SatelliteSensor," Satimagingcorp.com. [Online]. Available: https://www.satimagingcorp.com/satellite-sensors/other-satellite-sensors/sentinel-2a/. [Accessed: 22-Apr-2021].

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in 2011 International Conference on Computer Vision, 2011.