Research Article

SAR Image Matching Based on Local Feature Detection and Description Using Convolutional Neural Network

Mohammed Elwan,a Ahmed S. Amein,a Aiman Mousa,b Abdelmoty M. Ahmed,a,b,c Belgacem Bouallegue,d and Abdelhameed S. Eltananyc

1Egyptian Academy of Engineering and Advanced Technology (EAEAT), Cairo, Egypt
2Faculty of Information Systems and Computer Science, October 6 University, Giza, Egypt
3Electrical Engineering Department, Military Technical College (MTC), Cairo, Egypt
4Department of Computer Engineering, College of Computer Science, King Khalid University, Abha, Saudi Arabia

Correspondence should be addressed to Abdelmoty M. Ahmed; abd2005moty@yahoo.com

Received 1 November 2021; Revised 7 April 2022; Accepted 20 April 2022; Published 24 May 2022

Academic Editor: Bruno Carpentieri

Copyright © 2022 Mohammed Elwan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Feature detection is a vital step for the image registration process whose target is the misalignment correction among images to increase the convergency level. Deep learning (DL) in remote sensing has become a worldwide sensation. Despite its huge potential, DL has not reached its intended target concerning the applications of Synthetic Aperture Radar (SAR) images. In this study, we focus on matching SAR images using a Convolutional Neural Network. The big challenge in this study is how to modify a pretrained Visual Geometry Group model based on the multispectral dataset to act as a SAR image feature detector where it does not require any prior knowledge about the nature of the SAR feature. Since SAR images have different characteristics from optical images such as SAR dynamic range and imaging geometry, some problems arise and should be considered during the matching process. Despite all these difficulties, results demonstrate the robustness of the registration process where it can provide descriptors that preserve the localization data of features. Also, the proposed approach provides reasonable results compared to the state-of-the-art methods and outperforms the correlation approach and ORB descriptor under scaling. In addition, it may be considered an end-to-end image matching tool for SAR images, although the calculations of fine matching parameters are included.

1. Introduction

The process of image registration aims to find the deformations among the used images to be corrected for further processing. These deformations resulted from capturing the images by different sensors, at different times, or from different viewpoints. The registration process can be defined as an optimization issue to get the correct values of a transformation matrix required for mapping one image to another one for providing the dedicated similarity factor. The accuracy of this process is mainly dependent on the correct detection and extraction of the same features demonstrating the same objects inside images. Generally, image registration techniques are classified into two main categories, traditional and automated techniques. Traditional techniques suffer from consuming a high processing time since they depend on a manual feature selection that is strongly dependent on the user’s experience. Automated techniques are proposed to overcome the limitations of traditional techniques, where they are classified into three groups: Area-Based Matching (ABM) methods, Feature-Based Matching (FBM) methods, and hybrid methods. FBM methods include both feature detectors and feature descriptors, where descriptors may act as detectors and/or descriptors. Many approaches are performed from detection and/or description processes, and each approach has its advantages and limitations. Generally, two types of features can be detected: global and local features. Image content is
wholly depicted using global features, while key points 
detection inside the image requires the usage of local features 
as shown in Figure 1 [1–4].

Many registration problems arise during the capturing 
process of images as viewpoint, multimodal, temporal, and 
template registration probe ability to use these images to-
gether requires recovery from these problems. Viewpoint 
registration problem concerns the variations of viewing 
angles and acts as a shape reconstructing process. Multi-
modal registration problem concerns the usage of different 
payloads and acts as an information integration process. On 
the other hand, the template registration problem concerns 
the effect of imagined geometry and acts as a recognition 
process. Finally, the temporal registration problem is the 
most famous one that concerns the effects of imaging and 
environmental variations, including mainly the effect of 
oise, and acts as a detection process [5–8].

Nowadays, many remote sensing image applications 
allow the usage of Artificial Intelligence (AI) for multi-
spectral images in many tasks such as object detection and 
recognition, image registration, image fusion, and image 
segmentation. But, this is not the case for SAR image ap-
lications. AI can be described as Machine Learning (ML) 
which can be classified as Neural Networks (NN), Con-
volutional Neural Networks (CNN), Deep Learning (DL), 
and Spiking Neural Networks (SNN). Since the character-
stics of SAR images differ from that of multispectral images, 
some problems should be considered while applying AI 
approaches to SAR images, such as SAR image dynamic 
range, SAR signal statistics, SAR imaging geometry, complex 
nature of SAR data, and shortage of SAR images library. 
Finally, the registration process should resist any imaging 
and environmental variations to maximize the similarity 
among captured scenes. So, the wide scope of registration 
approaches makes it difficult to compose these approaches 
since each one concerns a certain problem for a certain 
application [5, 9–17].

The proposed approach demonstrates the usage of a 
Visual Geometry Group (VGG-16) model [18] as a SAR 
image feature detector. Since the VGG-16 model is mainly 
used as a classifier for multispectral remote sensing images, 
modifications related to its structure and operational con-
cept are required to adapt to opt new task, SAR image 
registration. First, a Gaussian kernel filter is used to smooth 
the intensities variations acting as a normalization process to 
overcome the problem of SAR image dynamic range. Using 
the image intensity form overcomes the problem of SAR image 
signal statistics. The intensity image is then applied to 
the model after being resized to a multiple of 32 using the 
cubic spline interpolation technique that provides a fine 
output having a smaller error. Detected feature locations are 
depicted using two parameters representing both x and y 
coordinates; then, descriptors corresponding to these fea-
tures are generated. These descriptors may be good or not; 
good descriptors are required, while others should be re-
duced since they affect the output accuracy. Two matching 
methods are used for this purpose to detect the most 
powerful descriptors. The first matching method is a pre-
matching step using Euclidean distance [19], and the second 
one is a fine matching step using Random Sample Consensus 
(RANSAC) algorithm [20]. Then, a transformation matrix 
based on the most powerful key points is generated for 
mapping one image to the other image. Mean Squared Error 
(MSE), Peak-Signal-to-Noise Ratio (PSNR), and correlation 
coefficient (CC) are used to evaluate the overall performance 
of the registration process.

This paper is organized as follows. Some relevant works 
concerning this research field are presented in Section 2. 
CNN is briefly introduced in Section 3. Section 4 discusses 
the methodology, including the proposed structure, dataset, 
and assessment criteria. Results and discussion are discussed 
in Section 5. Finally, the conclusion is given in Section 6.

2. Related Works

Recently, AI approaches have been considered for many 
applications of image registration. The proper usage of AI 
requires the availability of datasets where common training 
features exist [14–17, 21–23]. A survey concerning all image 
matching techniques is introduced starting with traditional 
approaches, including Area-Based Matching (ABM) and 
Feature-Based Matching (ABM) methods, followed by a 
description of matching methods from handcrafted to deep 
approaches. Its target is analyzing several tasks such as 
mapping and registration using the introduced approach 
[14, 15, 23]. In [12], it demonstrates a VGG-16 model to 
register multitemporal RGB remote sensing images. It 
provides a powerful multiscale feature descriptor and in-
creases the number of inliers matched points to enhance the 
registration process depending on the Thin Plate Spline 
(TPS) interpolation technique. It claims that it outperformed 
Scale Invariant Feature Transform (SIFT) descriptor [24]. In 
[16], an effective method known as Local Preserving 
Matching (LPM) is presented. Its target is how to keep the 
structure of local adjacent correctly matched. It mainly 
overcomes the process of mismatch removal keeping the 
number of true matched points high for the matching 
process.

In [17], an efficient method known as Neighborhood 
Manifold Representation Consensus (NMRC) is developed 
aiming to remove the incorrect matches based on de-
scription similarity. It is mainly dependent on the stability 
of the adjacent topologies around the real correspondences 
through iterative filter blocks, that is, keeping the structure 
of local adjacent correctly matched. It enhances the per-
formance under the situation of dead earnest deteriorated 
information. It can be considered as an optimization for-
maul dependent on NMR and iterative filtering. In [18], a 
framework for the registration process was proposed based 
on learning of mapping function among patch pairs and 
their labels using forward and backward operations. It en-
hances the overall accuracy compared with the state-of-the-
art methods and minimizes the training cost. But it con-
sumes a high processing time to estimate the required 
mapping functions as an initial step. In [25], a combination 
of features between CNN and SIFT was proposed. It is based 
on a manual dataset generation for fine-tuning the VGG-16 
model to detect the features. Then, combining CNN features
with those of SIFT and feeding the output to PSO-SIFT to perform the registration process, it improved the accuracy and the number of correct correspondences. Hence, [25] contradicts what was presented in [12] as it indicates the importance of SIFT descriptor.

Although the availability of datasets is a big challenge related to SAR images, many researchers have succeeded in applying AI to SAR image applications. The earlier attempts concerning SAR images were based on the utilization of a model for optical ones with few modifications to recover the raised problems due to SAR image unique specifications and the shortage of common SAR datasets. These attempts were designed for the detection of the military vehicle based on a custom dataset [9–17, 21–23]. In [26] as a first attempt, the author used an unsupervised sparse autoencoder and exhibited good efforts acting as a target recognition process. More attempts such as [27–29] were performed based on [26] using the same dataset with different conceptual and operational concepts. More researchers tried to perform different tasks such as ship detection and building detection acting as not only SAR target recognition but also a classifier where SAR signal was used instead of the processed SAR images [2].

All these attempts were mainly dependent on a customized dataset and search for a specific object to be recognized using CNN models. Finally, these attempts addressed the main problem concerning the application of CNN to SAR images; this problem is the lack of SAR datasets for training purposes. Although these attempts were designed to perform specific tasks as SAR target recognition and classification, they did not achieve SAR image registration. So, this paper demonstrates the utilization of the VGG-16 model with little operational concept modifications to perform SAR image registration taking into account SAR image characteristics.

### 3. Convolutional Neural Network

Artificial Neural Network (ANN) can be defined as adaptive fully automated data driven based on mathematical assumptions. Depending on the flowing data, its structure can be modified. It depicts a group of interconnected neurons performing the information processing. Generally, NN is nonlinear statistical data modeling between the input and the output where the interconnections between different neurons have dedicated weights values. NN has two design forms: forward propagation form and backward (recurrent) propagation form. CNN model, as a subfield of AI, was developed for processing of multiple arrays data forms. It contains 3 layers: input layer, hidden (CNN) layer, and output layer. CNN model may be simple having only one CNN layer or complex having more than one CNN layer.

Image registration using CNN models is an interesting research area based on two steps: feature extraction and the estimation of similarity metric among these extracted features. The most important constraint concerning the utilization of the CNN approach is the shortage of common training dataset which covers all aspects of remote sensing images. A single CNN layer contains 3 different layers: convolutional layer, pooling layer, and fully connected (FC) layer as shown in Figure 2 [9–17, 30].

Convolutional layer consists of many filters searching for a specified pattern. Each convolutional layer provides what is called a feature map. Then, an activation function (linear or nonlinear) is applied to produce an activated feature map. The most commonly used activation functions are sigmoid, tanh (hyperbolic transient), identity, and rectified linear unit (ReLU). Pooling layers (maximum or average) target the aggregation of the pixels values within a predefined window size. The number of provided and outcome feature maps is equal but changes in dimension due to the subsampling process. This means that the output dimension equals 1/x of the input activated feature map dimension where (x) is the subsampling mask size or patch size where $x^j = down(x^{j-1})$ such that down(.) represents a pooling function [9–11, 15, 17, 25, 30].

As being a fully automated model, CNN has achieved a great success where it can automatically learn the nature of features using input images. Performance of CNN models has a direct proportional relationship to both processing capabilities and dataset availability. These two factors permit CNN model to be trained with millions of parameters, and hence the performance increases. All CNN frameworks focus on the estimation of dedicated metrics provided by using large images without the need for location information [10–13, 18, 25]. On the other hand, CNN suffers from vanishing the gradients and may not work properly for severe distortions. In addition, the data overfitting is one of CNN’s limitations. So, researchers try to solve these limitations via different directions as the modifications of both activation function and training phase strategies, adding normalization layers, and updating the model architecture. Finally, the CNN structure depth is a critical parameter for getting better performance. As being one of the most commonly used tools for feature matching, SIFT descriptor may be combined with CNN’s models to improve both performance and accuracy [10–13, 18, 25]. Figure 3 demonstrates a single CNN model.
where a kernel filter is applied to the input, and then an activation function is performed followed by the pooling process.

4. Methodology

Visual Geometry Group (VGG) model [18] consists of several convolutional blocks followed by many FC layers. Each convolutional block has two convolutional layers followed by a maximum pooling layer. All convolutional layers and FC layers use the ReLU activation function to overcome the problem of vanishing gradient. Figure 4 represents the basic structure of the VGG network. The operational concept of the VGG network depends on a fixed input size (224 × 224) delivered to the convolutional layer, where the spatial resolution is kept after each convolution process. The convolution process is carried out over a window of size equal to 3 × 3 with a stride equal to 1, whereas the maximum pooling process is carried out over a window of size equal to 2 × 2 with a stride equal to 2.

There are several variants of VGG-Net such as VGG-11, VGG-16, and VGG-19, where they differ in the number of convolutional layers. A VGG network model is mainly used for image classification, where it has many advantages such as being trained on tremendously and different datasets of optical images and has a high performance. The most important advantage of the VGG model is its simple architecture that does not require any shorter alternative routes to strengthen the flow of gradient and hence can be adapted for many image processing applications [31].

One of the most important terminologies is the pretrained model. It is a model which is already trained in an environment similar to the required one. Shortage of the common training dataset covering all aspects of remote sensing images is an important factor related to the model structure. So, pretrained models are very useful in case of no datasets shortage. There are several pretrained models for many remote sensing applications such as feature detection, image recognition, and image classification. One of the most famous pretrained models is VGG-Net which is trained on the ImageNet dataset such that it searches for general patches and is more useful for many applications. Figure 5 shows the general workflow diagram.

The operational target of the registration process is how to transform sensed image to coincide with reference image. The general workflow of the registration process has four steps where the reference image is kept without any changes, and only the sensed image will be changed. First, a preprocessing step is performed due to the different characteristics between optical and SAR to overcome the raised problems such as SAR dynamic range, SAR signal statistics, SAR imaging geometry, and complex nature of SAR data. Second, a set of key points will be detected from sensed image, and another key point set will be detected from the reference image. Then, a description process is carried out. Third, a matching step is performed to determine which points are classified as inliers that are important and outliers which are needed to be removed. Finally, the required transformation matrix is generated using the inliers set of key points and applied to the sensed image to produce the registered image.

4.1. Preprocessing of SAR Images. As mentioned before, some problems should be considered while applying AI approaches to SAR images such as SAR dynamic range, SAR signal statistics, imaging geometry, and SAR image specifications. These problems should be recovered to get high performance. SAR images have a wide dynamic range which may reach 90 dB where pixels are within low or high values of dynamic range. Since CNN networks cannot control this range, preprocessing steps should be performed as a
normalization process to recover this dynamic range safely. SAR signal statistics plays an important role where it takes a complex form (real and imaginary). Hence, the speckle effect must be recovered to recapture the features. Although attempts to recover the speckle effect, such as the usage of complex reduction filters, will reduce the dynamic range, CNN can approximately imitate the features inside SAR image. Since the SAR imaging geometry is not the same as that of multispectral, any attempts to generate a SAR dataset will provide meaningless images. In addition, SAR image specifications are very important where the phase information plays an important role. According to the above mentioned, the used CNN network should be able to perform all operations related to a distorted phase concerning different degrees [2, 5, 11].

4.2. Feature Detection and Description. To recover the raised problems due to the different characteristics between SAR and multispectral images, two considerations should be taken into account for SAR feature detection. The first one is the characteristics of the SAR image, and this is recovered by performing a preprocessing step before being applied to the model by using the Gaussian kernel filter. The second is the shortage of common SAR datasets, and this will be recovered using a pretrained model based on multispectral dataset with minor or major modifications. The VGG-16 model is utilized for key point detection and extraction in many neural network techniques where there are many filters of small sizes embedded in the convolutional layers searching for a certain pattern inside the input image. The depth of the CNN model can be considered as a measurement of how many layers are included, especially the number of convolutional blocks (convolutional and pooling layers). As the CNN depth increases, the search scaling increases, and the model will be more complex as the number of patterns becomes larger. The searching strategy of the VGG model is mainly dependent on universal features and patterns. Also, since the VGG model is a pretrained model based on optical images not SAR, the detection of features requires a slight modification for the output of the model. The detection process is strongly depending on the created patches of fixed sizes from the SAR image. Then, the identification process, whether patches are related to objects or not, is performed depending on the model’s parameters.

Firstly, the input images are resized to a size which is a multiple of 32. 224 × 224 single layer (amplitude) of SAR image will be provided to the network after applying a Gaussian kernel filter through a bank of convolutional blocks. Each block uses filters whose receptive field is small and equals 3 × 3. The stride for the convolution process is one pixel where its padding is selected to preserve the spatial resolution after each convolution process. Each 3 × 3 convolution requires one-pixel padding. After each convolution process, a spatial maximum pooling process is carried out concerning a window of size 2 × 2 with a stride equal to 2. ReLU is used as an activation function after each convolutional block. The model output is delivered from certain layers covering various sizes as shown in Figure 6 where h and w present the height and width of the input image. Since the feature description process is mainly dependent on the detected feature, performing proper description requires the output to be delivered from certain layers covering different sizes.

There are many handling data managers for the utilization of CNN models such as TensorFlow. These handling tools collect the data from various layers and feed others. These data differ in size. Starting from this point, Kronecker product (⊗) is enrolled due to the utilization of outputs from different convolutional blocks 3, 4, and 5. Kronecker product is one of the most widely used tools for matrices operations depicting the operations on two different sizes matrices to provide what is known as a block matrix or tensor product matrix. A tensor object is a special type of multilinear map where its input is a certain number of matrices of different sizes generated from specific nodes within a manifold. Its
output is the tensor metric, where this metric is a scalar representing the local dot product of these matrices concerning this node. This operation is an encoding process for these matrices data such as lengths and angles among them. This operation is mainly dependent on the gradient values concerning neighborhood key points. From the conceptual point of view, each node on the manifold has only one tensor metric, and any changes in the metric tensor reflect how the data encoding varies [32–34].

Let \( A \) be \( m \times n \) matrix with \( ij^{th} \) element \( a_{ij} \) for \( i = 1, \ldots, m \) and \( j = 1, \ldots, n \) and \( (B) \) be any \( r \times q \) matrix. \( A \otimes B \) is a block (partitioned) matrix \( (I) \) of size \( mr \times nq \) formed by multiplying each \( a_{ij} \) element by the entire matrix \( B \) as follows:

\[
A \otimes B = \begin{bmatrix}
a_{11}B & a_{12}B & \cdots & a_{1n}B \\
a_{21}B & a_{22}B & \cdots & a_{2n}B \\
\vdots & \vdots & & \vdots \\
a_{m1}B & a_{m2}B & \cdots & a_{mn}B
\end{bmatrix}
= I_{mr \times nq},
\]

where \( I \) is depicted by blocks or submatrices as \( a_{11}B, a_{1n}B, \ldots, a_{m1}B, \) and \( a_{mn}B \). A block matrix \( (I) \) can be generated depending on the required division for its rows and columns. Tensor product of vector spaces \( A_{mxn} \) and \( B_{nxq} \) is itself a vector space \( I_{mxnxq} \) and has a dimension equal to the product of the two vectors’ dimensions. So, the tensor product is distinguished from direct summation vector spaces whose output has a dimension equal to the summation of these two vector spaces dimensions. So, the tensor product approach is considered to be the most common approach in case of handling fewer constraints [32, 33, 35].

Since the detection operation of the proposed approach is based on different size layers, the activated feature map from convolutional blocks 3, 4, and 5 is provided for the description process. These activated feature maps include common patterns. First, the output of convolutional block number 3 is an image of size \( 32 \times 32 \). A descriptor of 16 bins will be generated for each window of size \( 8 \times 8 \), considering its center as a key point inside the pooling layer number 3.

The output feature map, \( M_1 \), from convolutional block number 3 has a length of 256, and its key points are centers with blue color as shown in Figure 7(a).

The detection process concerning convolutional block number 4 is obtained in a slightly different way based on the tensor object operational point of view and the Kronecker product. Detected key points will be generated for each window of size \( 16 \times 16 \) inside the pooling layer number 4 considering its center as a key point with green color as shown in Figure 7(b). Equation (2) represents the generation of the output feature map, \( M_2 \), from pooling layer number 4 using the Kronecker product as follows:

\[
M_2 = P_4 \otimes T_{2 \times 2 \times 1},
\]

where \( P_4 \) is the output of pooling layer number 4 and \( T \) presents the tensor product with 1 stride.

Detected key points are generated for each window of size \( 32 \times 32 \) inside the pooling layer number 5, considering its center as a key point with red color as shown in Figure 7(c). Equation (3) represents the generation of the output feature map, \( M_3 \), from pooling layer number 5 using the Kronecker product as follows:

\[
M_3 = P_5 \otimes T_{4 \times 4 \times 1},
\]

where \( P_5 \) is the output of pooling layer number 5 and \( T \) presents the tensor product with 1 stride. Then, features maps \( M_1, M_2, \) and \( M_3 \) are normalized to the unit variance as follows:

\[
M_i \leftarrow \frac{M_i}{\sigma(M_i)} \quad i = 1, 2, 3,
\]

where \( \sigma \) is the standard deviation of matrix elements [5, 12].

4.3. Feature Matching. The feature matching process is carried out via two steps: the first one is the prematching step, while the second is the fine matching step. For the
prematching step, the Euclidean distance metric [19] of the detected feature is defined and estimated between the respective feature descriptors. Distance metric of two features \((x, y)\) is composed of the summation of three distances, each corresponding to the selected layers’ output descriptor concerning this point as follows:

\[ d(x, y) = \sqrt{d_1(x, y)^2 + d_2(x, y)^2 + d_3(x, y)^2}, \]

where \(d_1(x, y)\) value is Euclidean distance value. Since the output of convolutional block 3 is 256-D and outputs of blocks 4 and 5 are 512-D each, the value \((\sqrt{2})\) is used to compensate for this dimensional difference. Euclidean distances \(d_1(x, y), d_2(x, y),\) and \(d_3(x, y)\) are concerning the outputs of the third, fourth, and fifth convolutional blocks, respectively, and can be expressed as \((6)\). The sensed features are matched to its congruent inside the reference image if its distance \(d(x, y)\) is the smallest distance measurement for all distance measurements \(d(., y)\) according to a predefined threshold value \(\theta\); that is, \(d(x, y) > \theta\).

\[ d_i(x, y) = \text{Euclidean distance} \left( D_i(x), D_i(y) \right). \]  

A patch whose center is a matched point is generated after satisfying these two conditions. Since these generated patches may be partially or completely overlapped, a fine matching step is used to overcome this issue. RANSAC approach [20] is used to coherently remove the outliers. The remaining inliers set of matched points are then utilized to generate the required transformation matrix to align the sensed features to their correspondence in the reference set. The outlier removal process enhances the transformation matrix generation and subsequently the accuracy of the registration process.

5. Dataset, Result, and Discussion

5.1. Data and Computational Cost. Two aspects of SAR image sets are used where their specifications are listed in Table 1. The first image is an intensity image captured by Space Shuttle Imaging Radar (SIR-C) for Kilauea, Hawaii, dated May 1999 as shown in Figure 8. The second image is an intensity image captured by Terra-SAR-X (strip map) for Aswan High Dam, Egypt, as shown in Figure 9.

Computational cost is performed on a machine with Intel(R) Core (TM) i7-4790 CPU @ 3.60 GHz and 16 GB installed RAM.

5.2. Evaluation of the Proposed Approach. There are many metrics to evaluate the performance of the registration process. MSE, PSNR, and CC are used to measure the similarity between images before and after the registration process. Although being one of the most commonly used tools for feature matching, SIFT descriptor [24] is not good for SAR images because the speckle noise affects the image quality producing many false interest points, and this impedes Laplacian of Gaussian (LoG) from working properly. Also, multiplicative noise will affect the estimations of orientation’s histogram. So, there are many modifications based on SIFT state of the art which were introduced to enhance the performance of SIFT in general for multisensor imaging. In addition, SIFT is combined with the CNN to improve the performance accuracy.

The proposed approach is compared to correlation approach as an ABM method, SIFT [24] as a floating descriptor, Oriented FAST and Rotated BRIEF (ORB) [36] as a binary descriptor, and the CNN model proposed in [12]. Since SIFT and ORB descriptors are not related to the same category of the proposed approach, so they are used as monitoring tools to visually evaluate the performance of the proposed approach. Subset, translation, scaling, and rotation cases are included in the evaluation process. Due to the shortage of a common training dataset covering all aspects of SAR images, each sensed image related to each case of study is generated automatically.

5.3. Experimental Results. The experimental workflow to perform an evaluation of the proposed approach performance is shown in Figure 10. There is no doubt that making a dependable comparison related to a specific topic requires the existence of all the metrics used for this area during a certain period of time. But, due to the shortage of these measurement results, these results may be regenerated again by the recreation process of its source code. Performance results of the first and second data sets are shown in Tables 2 and 3, respectively, concerning the four cases of study and
including the most famous metrics measurements. Figures 11 and 12 depict the output registered image using the proposed approach compared to different approaches concerning the four cases of study.

5.4. Discussion. Figures 13 and 14 represent the performance evaluation, including MSE, PSNR, and CC related to the first and second datasets, respectively. Detection of features based on the CNN model is an advantage where there is no prior knowledge about the nature of features inside the image. Results demonstrate that the proposed method can successfully recover the deformation error related to translation and scaling cases of studies for the used image pairs. However, it cannot recover the rotation issue as the correlation approach and the approach in [12] did. Also, it performs more effectively than the approach in [12]. The proposed approach uses a spline interpolation method, while the approach in [12] uses the Thin Plate Spline (TPS) interpolation technique which is more complex as it requires ground points to calibrate the results. Concerning scaling, the proposed approach outperforms the ORB descriptor, correlation approach, and approach in [12] considering MSE, PSNR, and CC.

Despite having the ability to overcome the scaling problem, the correlation method requires the sensed image to be resized first in order to have the same dimensions as the master image. This of course affects the values of pixels and cannot indicate the correct results. So, the correlation approach should not be considered for the scaling problem. Also, no doubt that a relatively large reduction in matched points is not recommended since it will strongly affect the result if these points are not powerful enough. But, this is not the case for the proposed approach as it has the ability to detect and extract the most powerful key points. So, using Random Sample Consensus (RANSAC) approach as a fine-tuning enhances the removal of outliers and hence improves the accuracy of the registration process. On the other hand, the proposed method cannot recover the deformation error related to rotation. This is a common problem for all CNN models applied to SAR data. In addition, as mentioned before, any attempts to generate a SAR dataset may provide meaningless images because the imaging geometry is not real to reflect real aspects.
Figure 10: Experimental workflow for performance evaluation.

Table 2: Performance results of first dataset.

| Study case | Metric     | Before  | CC     | SIFT   | ORB    | Proposed | [12]   |
|------------|------------|---------|--------|--------|--------|----------|--------|
| Subset     | Shift      | −4, −6  | 0, −1  | 0, 0   | 0, 0   | 0, 0     | 0, 0   |
|            | MSE        | 103.24  | 102.06 | 39.67  | 53.19  | 67.89    | 90.37  |
|            | PSNR       | 13.979  | 14.021 | 16.073 | 15.436 | 14.906   | 14.285 |
|            | CC         | 0.059   | 0.278  | 0.621  | 0.572  | 0.552    | 0.344  |
| Translation| Shift      | 8, −1   | 3, 0   | 0, 0   | 0, 0   | 0, 0     | 0, 0   |
|            | MSE        | 103.97  | 104.16 | 70.39  | 87.15  | 81.65    | 101.15 |
|            | PSNR       | 13.981  | 13.976 | 14.827 | 14.364 | 14.505   | 14.041 |
|            | CC         | −0.079  | −0.062 | 0.429  | 0.333  | 0.376    | 0.031  |
| Scaling    | Shift      | 0, 0    | 0, 0   | 0, 0   | 0, 0   | 0, 0     | 0, 0   |
|            | MSE        | 4.09    | 81.27  | 58.88  | 92.51  | 84.17    | 94.62  |
|            | PSNR       | 21.001  | 14.515 | 15.215 | 14.234 | 14.439   | 14.185 |
|            | CC         | −0.093  | 0.901  | 0.938  | 0.618  | 0.853    | 0.428  |
| Rotation   | Shift      | 10, −2  | 10, 8  | 0, 0   | 0, 0   | 0, 0     | 0, 0   |
|            | MSE        | 103.91  | 103.59 | 88.32  | 91.68  | 103.98   | 103.62 |
|            | PSNR       | 13.982  | 13.988 | 14.335 | 14.254 | 13.981   | 13.988 |
|            | CC         | −0.142  | −0.153 | 0.475  | 0.444  | −0.018   | −0.119 |

Table 3: Performance results of second dataset.

| Study case | Metric     | Before  | CC     | SIFT   | ORB    | Proposed | [12]   |
|------------|------------|---------|--------|--------|--------|----------|--------|
| Subset     | Shift      | −20, 21 | −2, 7  | 0, 0   | 0, 0   | 0, 0     | 0, 0   |
|            | MSE        | 95.99   | 96.13  | 31.48  | 48.58  | 83.03    | 85.34  |
|            | PSNR       | 14.154  | 14.151 | 16.575 | 15.633 | 14.445   | 14.409 |
|            | CC         | −0.012  | 0.122  | 0.721  | 0.708  | 0.433    | 0.398  |
| Translation| Shift      | 5, −23  | 0, −20 | 0, 0   | 0, 0   | 0, 0     | 0, 0   |
|            | MSE        | 95.35   | 95.55  | 58.42  | 63.27  | 82.52    | 86.45  |
|            | PSNR       | 14.168  | 14.164 | 15.233 | 15.059 | 14.483   | 14.381 |
|            | CC         | −0.023  | −0.036 | 0.716  | 0.704  | 0.534    | 0.319  |
Table 3: Continued.

| Study case | Metric | Before | CC | SIFT | ORB | Proposed | [12] |
|------------|--------|--------|----|------|-----|----------|-----|
| Scaling    | Shift  | 0, 0   | 0, 0 | 0, 0 | 0, 0| 0, 0     | 0, 0 |
|            | MSE    | 3.60   | 58.13| 45.16| 69.12| 60.88    | 70.81|
|            | PSNR   | 21.282 | 15.243| 15.791| 14.867| 15.143   | 14.814|
|            | CC     | −0.023 | 0.963| 0.981| 0.841| 0.939    | 0.881|
| Rotation   | Shift  | 8, −21 | −24, 0| 0, 0 | 0, 0| 0, 0     | 0, 0 |
|            | MSE    | 92.68  | 90.51| 63.95| 66.34| 91.36    | 87.83|
|            | PSNR   | 14.230 | 14.281| 15.036| 14.957| 14.261   | 14.347|
|            | CC     | −0.101 | −0.008| 0.949| 0.926| −0.141   | 0.171|

Figure 11: Output registered images of first dataset: (a) subset, (b) translation, (c) scaling, and (d) rotation.

Figure 12: Continued.
**Figure 12:** Output registered images of second dataset: (a) subset, (b) translation, (c) scaling, and (d) rotation.

**Figure 13:** Performance evaluation of the first set of images: (a) MSE, (b) PSNR, and (c) CC.
Concerning the consumed processing time, the results show that the proposed approach overcomes that presented in [12] as shown in Table 4.

### 6. Conclusion

SAR image registration process is performed based on the utilization of the Visual Geometry Group (VGG-16) model. Due to the unique characteristics of SAR images such as dynamic range, signal statistics, imaging geometry, and complex nature of data, some problems arise and should be considered. Feature detection is based on a slight modification to the VGG-16 model. First, a Gaussian kernel filter is used to smooth the intensities variations, followed by input image resizing to a multiple of 32, and then applied to the proposed approach for feature detection such that the outputs from predefined layers are picked up. SQ_he location of the detected feature is extracted, followed by the description process. Two matching methods are used to generate a more accurate transformation matrix: Euclidean distance and RANSAC algorithm. Finally, a warping process is carried out to align the sensed image to the reference image.

Results show that the proposed approach has high performance and the ability to provide high-level features and reasonable outputs. In addition, it provides descriptors that preserve the localization data. SQ_he proposed approach has the ability to detect and extract the most powerful key points, so using the RANSAC approach as a fine-tuning enhances the removal of outliers and hence improves the performance.
accuracy of the registration process. Although the calculations of fine matching parameters are included, it may be considered an end-to-end SAR image matching approach. As a future direction, the proposed approach is planned to be applied to real SAR datasets covering all aspects of images and combined with other CNN models to improve performance.

Data Availability
The data are available any time if needed from the authors Abdelmoty M.Ahmed (e-mail: abd2005moty@yahoo.com) and Mohammed Elwan (e-mail: msafy@eaeat.edu.eg).

Conflicts of Interest
The authors declare no potential conflicts of interest.

Acknowledgments
The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through the Research Group Project (Grant no. RGP.1/157/42).

References
[1] S. Paul and U. C. Pati, “A comprehensive review on remote sensing image registration,” International Journal of Remote Sensing, vol. 42, no. 14, pp. 5396–5432, 2021.
[2] X. Zhu, S. Montazeri, M. Ali et al., “Deep learning meets SAR: concepts, models, pitfalls, and perspectives,” IEEE Geoscience and Remote Sensing Magazine (GRSM), vol. 9, 2021.
[3] A. S. Eltanany, A. S. Amein, and M. S. Elwan, “A modified corner detector for SAR images registration,” International Journal of Engineering Research in Africa, vol. 53, pp. 123–156, 2021.
[4] A. A. Goshtasby, Theory and Applications of Image Registration, John Wiley & Sons, Hoboken New Jersey, 2017.
[5] M. P. S. Tonewad and M. M. P. Dale, “Remote sensing image registration methodology: review and discussion,” Procedia Computer Science, vol. 171, pp. 2390–2399, 2020.
[6] V. T. Wang, Registration of Synthetic Aperture Imagery Using Feature Matching, University of Canterbury, Christchurch, 2018.
[7] A. Amein and A. H. El-Tanany, “Advanced selective key point detection techniques,” Journal of Engineering Science and Military Technologies, vol. 3, no. 2, pp. 61–69, 2019.
[8] M. Hassaballah, H. A. Alhazly, and A. A. Ali, “Analysis and evaluation of keypoint descriptors for image matching,” in Recent Advances in Computer Vision, pp. 113–140, Springer, Cham, 2019.
[9] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, “Deep learning in remote sensing applications: a meta-analysis and review,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 152, pp. 166–177, 2019.
[10] M. Z. Alom, T. M. Taha, C. Yakopcic et al., “A state-of-the-art survey on deep learning theory and architectures,” Electronics, vol. 8, no. 3, p. 292, 2019.
[11] M. Vakalopoulou, S. Christodoulidis, M. Sahasrabudhe, and N. Paragios, “Deep learning for image matching and Co-registration,” in Deep Learning for the Earth Sciences: With Applications, pp. 120–135, Wiley, Hoboken, New Jersey, 2021.
[12] 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.
[13] T. Georgiou, Y. Liu, W. Chen, and M. Lew, “A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision,” International Journal of Multimedia Information Retrieval, vol. 9, no. 3, pp. 135–170, 2019.
[14] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, “Image matching from handcrafted to deep features: a survey,” International Journal of Computer Vision, vol. 129, no. 1, pp. 23–79, 2021.
[15] X. Jiang, J. Ma, G. Xiao, Z. Shao, and X. Guo, “A review of multimodal image matching: methods and applications,” Information Fusion, vol. 73, pp. 22–71, 2021.
[16] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, “Locality preserving matching,” International Journal of Computer Vision, vol. 127, no. 5, pp. 512–531, 2019.
[17] J. Ma, Z. Li, K. Zhang, Z. Shao, G. Xiao, and R. Sensing, “Robust feature matching via neighborhood manifold representation consensus,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 183, pp. 196–209, 2022.
[18] S. Wang, D. Quan, X. Liang, M. Ning, Y. Guo, and L. Jiao, “A deep learning framework for remote sensing image registration,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 145, pp. 148–164, 2018.
[19] M. Greenacre and R. Primicerio, “Measures of distance between samples: Euclidean,” in Multivar. Anal. Ecol. Data, pp. 47–59, FBBVA, 2013.
[20] J. Janicka and J. Rapinski, “Outliers detection by RANSAC algorithm in the transformation of 2D coordinate frames,” Boletim de Ciências Geodésicas, vol. 20, no. 3, pp. 610–625, 2014.
[21] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, “DeepGMR: learning latent Gaussian mixture models for registration,” in Proceedings of the European Conference on Computer Vision, pp. 733–750, Springer, Cham, Switzerland, August 2020.
[22] X. Zhang, W. Jian, Y. Chen, and S. Yang, “Deform-GAN: an unsupervised learning model for deformable registration,” 2020, https://arxiv.org/abs/2002.11430.
[23] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: a review,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212–3232, 2019.
[24] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.
[25] F. Ye, Y. Su, H. Xiao, X. Zhao, and W. Min, “Remote sensing image registration using convolutional neural network features,” IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 2, pp. 232–236, 2018.
[26] S. Chen and H. Wang, “SAR target recognition based on deep learning,” in Proceedings of the 2014 International Conference on Data Science and Advanced Analytics (DSAA), pp. 541–547, Shanghai, China, November 2014.
[27] D. A. E. Morgan, “Deep convolutional neural networks for ATR from SAR imagery,” SPIE Proceedings, vol. XXII, Article ID 94750F, MD, USA, May 2015.
[28] J. Ding, B. Chen, H. Liu, and M. Huang, “Convolutional neural network with data augmentation for SAR target recognition,” IEEE Geoscience and Remote Sensing Letters, vol. 13, pp. 1–5, 2016.
[29] K. Du, Y. Deng, R. Wang, T. Zhao, and N. Li, “SAR ATR based on displacement- and rotation-insensitive CNN,” Remote Sensing Letters, vol. 7, no. 9, pp. 895–904, 2016.

[30] C. C. Aggarwal, “Convolutional neural networks,” in Neural Networks and Deep Learning, pp. 315–371, Springer, Cham, 2018.

[31] K. Kuppala, S. Banda, and T. R. Barige, “An overview of deep learning methods for image registration with focus on feature-based approaches,” International Journal of Image and Data Fusion, vol. 11, no. 2, pp. 113–135, 2020.

[32] B. K. Moser, “Linear algebra and related introductory topics,” in Linear Models: A Mean Model Approach, pp. 1–22, Elsevier, Amsterdam, Netherlands, 1996.

[33] E. o. Mathematics, “Tensor product,” 2020, https://encyclopediaofmath.org/index.php?title=Tensor_product.

[34] W. M. World, “Vector space tensor product,” 2022, https://mathworld.wolfram.com/VectorSpaceTensorProduct.html.

[35] K. K. Wu, Y. Yam, H. Meng, and M. Mesbahi, “Kronecker product approximation with multiple factor matrices via the tensor product algorithm,” in Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 004277–004282, Budapest, Hungary, October 2016.

[36] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” in Proceedings of the 2011 International conference on computer vision, pp. 2564–2571, Barcelona, Spain, November 2011.