Assisting UAV Localization Via Deep Contextual Image Matching

Muhammad Hamza Mughal, Muhammad Jawad Khokhar, and Muhammad Shahzad

***Abstract*—In this article, we aim to explore the potential of using onboard cameras and pre-stored geo-referenced imagery for Unmanned Aerial Vehicle (UAV) localization. Such a vision-based localization enhancing system is of vital importance, particularly in situations where the integrity of the global positioning system (GPS) is in question (i.e., in the occurrence of GPS outages, jamming, etc.). To this end, we propose a complete trainable pipeline to localize an aerial image in a pre-stored orthomosaic map in the context of UAV localization. The proposed deep architecture extracts the features from the aerial imagery and localizes it in a pre-ordained, larger, and geotagged image. The idea is to train a deep learning model to find neighborhood consensus patterns that encapsulate the local patterns in the neighborhood of the established dense feature correspondences by introducing semi-local constraints. We qualitatively and quantitatively evaluate the performance of our approach on real UAV imagery. The training and testing data is acquired via multiple flights over different regions. The source code along with the entire dataset, including the annotations of the collected images has been made public.[1] Up-to our knowledge, such a dataset is novel and first of its kind which consists of 2052 high-resolution aerial images acquired at different times over three different areas in Pakistan spanning a total area of around 2 km^2.**

***Index Terms*—Deep learning, neighborhood consensus networks, remote sensing, SIFT, template matching, UAV, vision-based localization.**

I. Introduction

FINDING a template (source) patch in a relatively larger (target) image is a task of fundamental importance in numerous computer vision applications including object detection, motion estimation and tracking, image based retrieval in large database systems, image registration/stitching, dense image matching for 3D reconstruction, and many others. It involves characterizing a way to measure the similarity between the template – a known reference pattern, i.e., representation of a patch or region of interest (ROI) in the source image – and the unknown test patterns/regions in the target image via matching operation.

One particular application of template matching lies within the domain of autonomous vision-based navigation where a terrestrial (e.g., robots) [1] or an aerial (e.g., unmanned aerial vehicles (UAVs) or drones) platform [2] [3] tries to localize itself using visual cues. Specifically for UAV navigation, it is of vital importance in situations where the integrity of the global positioning system (GPS) is in question (i.e., the GPS signals may get corrupted or become unavailable due to multipath reflections when operating close to obstacles, jamming, or any other unforeseen reason). Consequently, the navigation gets dependent solely on the inertial navigation system (INS) based state estimation which in turn is subject to drifting behavior in time i.e., the localization error accumulates over time which renders the estimation prone to errors and hence quickly becomes unusable after a few seconds. Thus, the UAVs whose localization estimation blindly relies on GPS signals are quite exposed to malevolent activities and therefore need an alternate autonomous navigation solution that is able to robustly cope with long- and short-term GPS signal losses.

Since every UAV is equipped with an onboard vision sensor, a viable solution is to extend the UAV localization capability by searching the current video frame (source template) within a pre-stored large map of orthomosaics (i.e., ortho-rectified photo mosaics). The problem can be easily framed as a template matching procedure where the naive implementations based on normalized cross-correlation [4][5] could be used to localize source image patches in the target orthomosaic map. However, such an (aerial) image matching procedure is not that simple as the template is often subject to complex deformations such as illumination and viewpoint changes, variations in the background, partial occlusions, and nonrigid deformations of the objects appearing in the scene. This makes the image matching procedure highly challenging and therefore simple representations based on intensity values alone have practical limitations.

More advanced representations based on mathematical transforms (e.g., wavelet [6], Fourier [7], [8], annulus projection transformations [9]), or scale and rotation invariant feature descriptors (e.g., histogram of dominant gradients [10], SIFT [11], SURF [12], ORB [13], BRISK [14], adaptive radial ring code [15] etc.) have been proposed. These invariant descriptions encompassing the local and spatial context are in turn used for feature matching by establishing correspondences between them. The matching is often performed using approximate nearest neighbor search based algorithms followed

Manuscript received July 21, 2020; revised October 5, 2020 and December 5, 2020; accepted January 16, 2021. Date of publication January 26, 2021; date of current version February 18, 2021. This work is financially supported by NUST, Islamabad, Pakistan. *(Corresponding author: Muhammad Shahzad.)*

Muhammad Hamza Mughal is with the School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan (e-mail: mmughal.bee15seecs@seecs.edu.pk).

Muhammad Jawad Khokhar is with the Teradata Global Delivery Center (GDC), Islamabad, Pakistan (e-mail: mjawadak@hotmail.com).

Muhammad Shahzad is with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan, and also with the Deep Learning Laboratory, National Center of Artificial Intelligence, Islamabad, Pakistan (e-mail: muhammad.shehzad@seecs.edu.pk).

Digital Object Identifier 10.1109/JSTARS.2021.3054832

[1]https://github.com/m-hamza-mughal/Aerial-Template-Matching

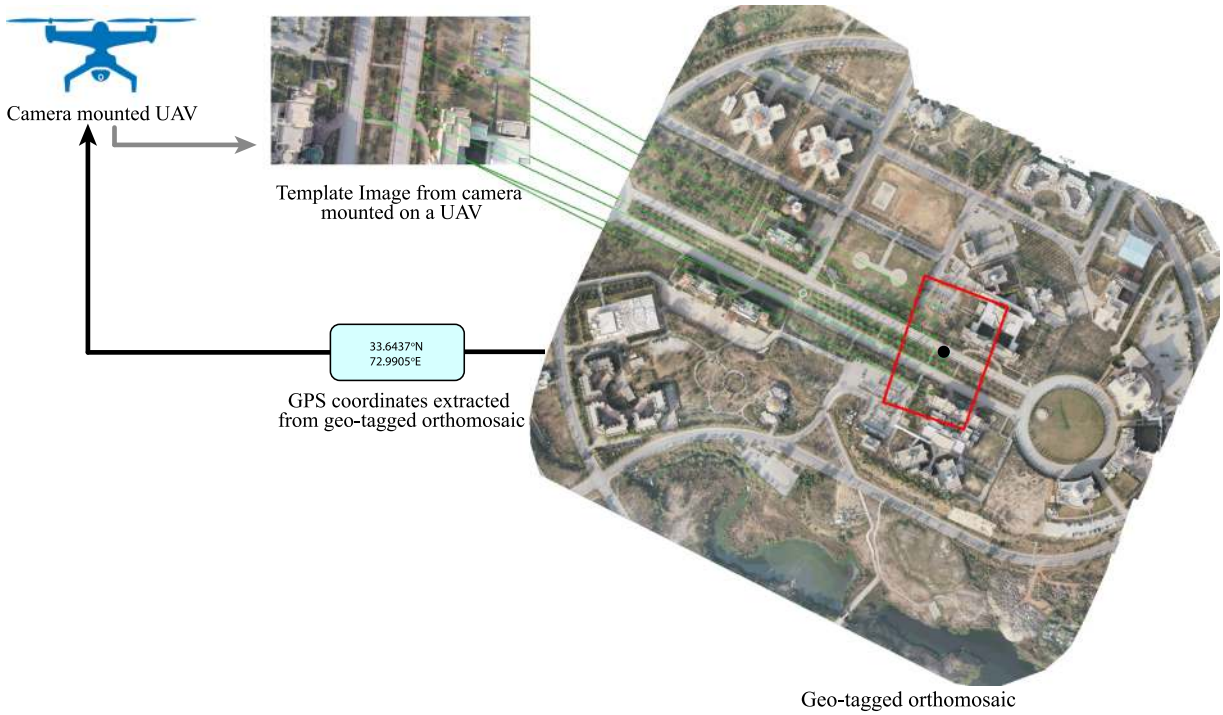


Fig. 1. Overview of UAV localization pipeline: The imagery from air-bound drone acts as an input to the system which is then matched through the presented algorithm and projected to the pre-ordained orthomosaic. The projected pixel coordinates are then used to extract GPS coordinates from the geo-tagged orthomosaic and these GPS coordinates in turn, aid in localization of UAV over the geographical map.

by post-processing strategies to enforce certain geometric constraints [16], [17]. However, these nearest neighbor approaches may result in inaccurate matching in the absence of local texture. To cope with such issues, few researchers have recently proposed deep trainable architectures to extract robust features and establish correspondences for matching by analyzing relatively larger context of the illuminated scene [18]–[23]. Such deep learning based architectures improve matching accuracy in comparison to handcrafted based approaches by directly learning meaningful features for localization of the template image as its training objective.

Inspired by the idea of end-to-end feature learning and matching, this article presents a complete trainable pipeline to localize an aerial image in a pre-stored given orthomosaic map in the context of UAV localization as illustrated by Fig. 1. The proposed deep architecture extracts features from the aerial imagery and localizes it in a pre-ordained, larger and geo-tagged image. The idea is essentially based on the concept of trainable neighborhood consensus [20], [24], [25] which analyzes the local patterns in the neighborhood of the established dense feature correspondences by introducing semi-local constraints. Such methods, however, have the following three drawbacks: 1) they utilize local features at same scale which renders matching of image pairs with large scale variances ineffective; 2) Some of them (e.g., [20]) require (interactive) provision of pre-selected key points in the source image to be matched in the target image; 3) Lastly, particularly to UAV localization use-case, they have limitation of not dealing with varying source and target image dimensions. The proposed approach deals with all these limitations and provides a complete framework for extracting point to point correspondences between the two images. Further to enable localization, the proposed method incorporates certain stochastic constraints to find the best matches and uses fully-connected layers to extract correspondences via regression between the source template and the target orthomosaics of varying dimensions. In the context to the proposed approach, the following are the significant contributions.

- A complete end-to-end trainable architecture has been proposed which simultaneously performs the feature learning and template localization by imposing the probabilistic constraints on the densely correlated feature maps of varying dimensions.
- The performance of the developed approach has been quantitatively and qualitatively demonstrated on real UAV imagery where the training and the testing data is acquired via multiple flights over different regions.
- All the dataset including the annotations of the collected imagery has been made public. Up-to our knowledge, such a dataset is novel and first of its kind and consists of 2052 high resolution aerial images acquired at different times over three different areas in Pakistan spanning a total area of around 2 km^2.

II. Related Work

Several researchers have addressed the problem of template matching in aerial images using scale invariant feature descriptors based techniques. For instance, Shan *et al.* [2] studied the

problem of UAV localization in GPS denied environments and utilized optical flow to determine the UAV position. They used inter-frame translation for pose-tracking and histogram of oriented gradient features for registration on the Google maps and later employed particle filtering for more refined localization. Moreover, Canhoto *et al.* [26] designed a template matching system to automate image sequence processing for autonomous aerial navigation by employing the extracted SIFT features from aerial imagery to estimate the UAV displacement for navigation. Koch *et al.* [27] also explored SIFT features as a solution to the problem of image matching for UAV. However, they concluded that the scale-invariant features are not an efficient approach to match aerial scenes. From image registration perspective, Lin and Medioni [28] employed iterative frame-to-frame and frame-to-map registration by exploiting mutual information to find correspondences to match a frame in a UAV acquired image sequence onto a reference high resolution map image. Similarly, Solbrig *et al.* [29] also performed video frames registration onto the prestored orthophoto. They first registered the first video frame to the orthophoto and later used image mosiacking to register subsequent frames using SIFT based key point descriptors. Usually, these descriptive features are combined with approximate nearest neighbor based approach [16] [9] to draw matches between high-dimensional feature representations of the images and obtain point-to-point correspondences between images. However, all of these feature representations are not descriptive enough to localize and match aerial imagery onto the orthomosaics due to larger variance in local context and illumination conditions [30]. Hence, employing deep feature representations learned using deep neural network based algorithms, which contain rich features encompassing larger context of aerial scene via layers of representational learning may be beneficial in developing robust solution towards the task of template localization over the orthomosaic.

A few deep learning based template aerial matching approaches have been recently proposed in the literature. For instance, Marcu *et al.* [31] proposed a cascaded deep learning based semantic segmentation and regression framework for vision-based location map prediction. Ahmad Nassar *et al.* [32] used a deep convolution neural network based method for aerial image matching and utilized it for localization of UAV. They exploited semantic shape matching and U-Net based segmentation to localize the current UAV frame onto a satellite map. Zhuo *et al.* [33] presented a way of registering UAV imagery by extracting dense and uniformly distributed features. They proposed a one-to-many matching scheme with pixel distances used as a global geometric constraint to verify whether the matching is correct or not. Chen *et al.* [34] presented an approach that uses local deep hashing based matching of aerial images. Altwaijry *et al.* [35] utilized attention mechanism in deep networks and framed the task of aerial image matching as a classification task. Noh *et al.* [36] have also proposed deep attentive local feature (DELF) descriptors for image verification. Tian *et al.* [37] proposed a siamese network based model to match cross-view image pairs for localization in urban environments using image pixels/patches belonging to buildings. Buniatyan *et al.* [38] have improved template matching algorithm based on normalized cross-correlation by using siamese convolutional neural networks to maximize the difference between true and false matches.

In the context of trainable template matching, Cheng *et al.* [19] have applied deep neural network based bottom-up pattern matching (BUPM) for spatial image localization. They have also introduced a quality-aware template matching (QATM) algorithm [18] which can also serve as trainable layer in deep neural networks. They have used a soft-ranking among all matching pairs between the deep features of an image to deal with various template localization scenarios. Rocco *et al.* [20] also proposed a trainable deep learning based method of extracting point to point correspondences between the images using neighborhood consensus network which takes two images i.e. template and main image and extracts dense features from using a deep learning-based feature extractor. The extracted features are then refined using soft mutual nearest neighbor filtering to build a correlation tensor which in turn is used to compute the matches between the source and target images of similar dimensions.

Most of the aforementioned algorithms, specifically the ones which rely on deep learning, focus on aerial image matching but do not include template localization over the target image which needs a different approach. To this end, we have proposed a matching and localization scheme by extracting dense features between the source template and the larger pre-stored geo-coded orthomosaic. For extraction of dense feature correspondences, we have adapted the neighborhood consensus network [20] trained on PASCAL-VOC [39] and InLoc [40] datasets to work over the problem of aerial image matching. The adaptations are necessary to enable scale invariant feature extraction as well as to cope with varying sizes of the two images (i.e, the source template and the large target orthomosaic). This is achieved by incorporating the softmax based constraints over the correlation matrix. This enables scale invariance as well as better point-to-point correspondences by employing the strongest feature matches between the differently sized images for better localization. The details of the developed approach are presented next in the following sections.

III. Methodology

Fig. 2 illustrates the proposed pipeline of the aerial image localization network which is essentially based on feature point learning using neighborhood consensus strategy that refines the matches in the template image and the pre-stored orthomosaic. This is achieved by exploiting the correlation information between the convolutional features of two images and subsequently imposing probabilistic constraints to obtain the point-to-point correspondences between them. To elaborate, the convolutional features are extracted from the images by processing them through a deep feature extractor. Subsequently, these feature maps, which embody the local and global information, are used to build a correlation matrix to encapsulate the feature matches for each extracted feature point. The correlation matrix is then processed through a trainable network that learns to establish more reliable correspondences. Furthermore, the probabilistic

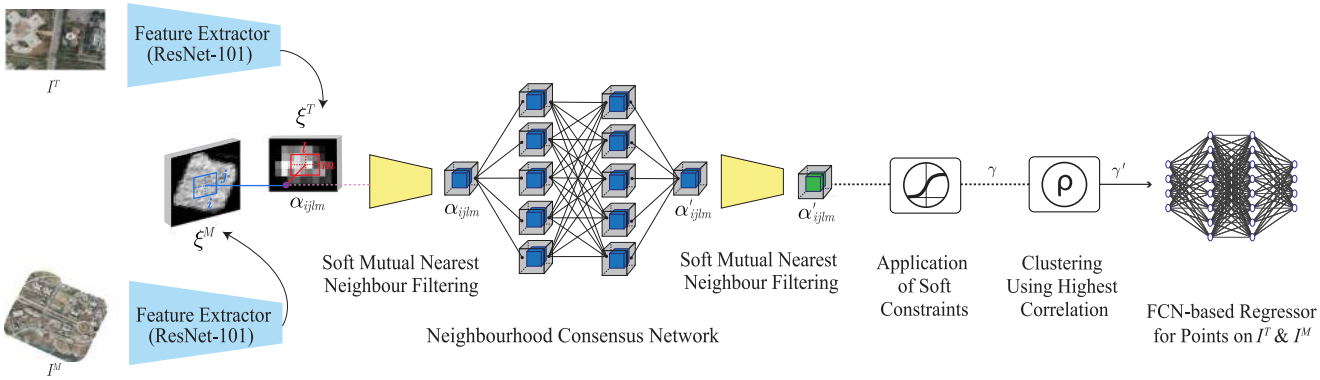


Fig. 2. Proposed Algorithm: Template Image I^T and orthomosaic I^M act as two inputs to the pipeline as they are processed by the convolutional feature extractor. The resulting feature maps ξ^M and ξ^T are then used in calculation of correlation tensor α. Subsequently, α is processed by 4D convolutional network and by applying probablistic constraints over the processed correlation matrix α', points of strongest feature matches are calculated. These points are then used to predict point-to-point correspondance between I^T and I^M, which are further utilized for projection of template over orthomosaic. The blocks, i.e., "Application of Soft Constraints" and "Clustering Using Highest Correlation" are detailed in Figs. 3 and 4.

constraints are added to these established correspondences to associate each feature point in the source image with the feature points of the orthomosaic. Similarly, the same is repeated for the orthomosaic feature points. Later, using the generated probability maps, the soft-argmax layer is incorporated to extract the indices of the best matches which are in turn passed through a fully-connected network (acting as a regressor) to estimate the point-to-point correspondences between the source and the target images. All components of the pipeline are differentiable and hence, the network is trained in an end-to-end fashion. In the following subsections, we present further details of the proposed pipeline.

A. Generating Refined Correlation Maps

Let us assume that the source (template) image I^T and the target image I^M having dimensions 224 $\times$ 336 and 896 $\times$ 896 respectively are processed by a fully convolutional neural network (e.g., ResNet [41] in our case) to extract features maps. The resulting feature maps ξ^T and ξ^M have downsized dimensions of 1/32 of the respective image dimensions. These dense feature maps are then used to compute the exhaustive pairwise cosine similarities to get a 4D tensor α which represents the correlation between each feature of ξ^T and ξ^M. The tensor α has dimensions $h^T \times w^T \times h^M \times w^M$, where h^T and w^T are height and width of ξ^T while h^M and w^M are the height and width of ξ^M. The pairwise cosine similarities are computed as follows:

$$\alpha_{ijlm} = \frac{(\xi^T_{ij}, \xi^M_{lm})}{||\xi^T_{ij}||_2 ||\xi^M_{lm}||_2} \tag{1}$$

where $ijlm$ is the index of correlation between the features that are in the local neighborhoods of ξ^T_{ij} and ξ^M_{lm}. The generated correlation tensor is subsequently processed by a 4D convolutional neural network [20] that learns to exploit the correlation patterns to ensure the extraction of spatially consistent feature point matches. This three-layered neural network learns to identify matches in a 4-D space by using 4D convolutions with a kernel size of $3 \times 3 \times 3 \times 3$. The learned network thus provides a single channel 4D tensor α' that represents *filtered* matches between the two feature maps. We further apply soft mutual nearest neighbor filtering to the correlation tensor before and after processing through the 4D convolutional neural network as presented in [20] to act as a global filtering mechanism which reduces the weight of the matching scores which are not mutual nearest neighbors to obtain a correlation tensor α'.

B. Extracting Pair-Wise Correspondences

Once the refined correlation maps are obtained, the next step is to extract the pairwise feature correspondences between the two feature maps. This is achieved by computing the probability distributions using the soft-max function over the dimensions corresponding to ξ^T and ξ^M as depicted in the following equations:

$$\beta^T_{ijlm} = \frac{e^{\alpha'_{ijlm}}}{\sum_{ab} e^{\alpha'_{ablm}}} \tag{2}$$

$$\beta^M_{ijlm} = \frac{e^{\alpha'_{ijlm}}}{\sum_{cd} e^{\alpha'_{ijcd}}} \tag{3}$$

In 2 and 3, β is a function of α' and represents the scores expressed as probability distributions showing the amount of similarity between every pair of points in the two feature maps.

From the implementation perspective, the correlation tensor is reshaped into a single dimension along the dimensions of the correlation tensor which correspond to the dimensions of feature map ξ^T to get probability distribution over the feature map ξ^M by applying the soft-max function on this reshaped dimension. Therefore, for each point in the flattened dimension which corresponds to that point in ξ^T, we get a 2-D probability distribution of dimensions equal to dimensions of ξ^M. These are the scores of similarity for each point in feature ξ^M given a point in feature ξ^T. A similar process is done to get probability distributions for points over ξ^T by reshaping and applying soft-max along the dimensions of the correlation tensor which corresponds

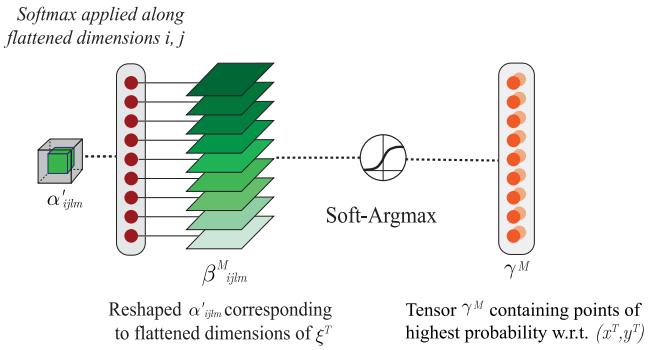


Fig. 3. Application of probablistic constraints on the processed correlation tensor α': β^M and β^T represent the probability distribution maps (shown in green) corresponding to the flattened dimension. Here x and y refer to the dimensions of the correlation tensor along which softmax function is applied to get probability maps and to calculate the corresponding points of highest probabilities. These probability maps are then associated with each point (x and y) which belong to the dimensions which are flattened for softmax. To extract the points of highest probability in the 2D probability maps, soft-argmax is employed to give out γ^M and γ^T as its output (dimension: $IR^{k\times 2}$).

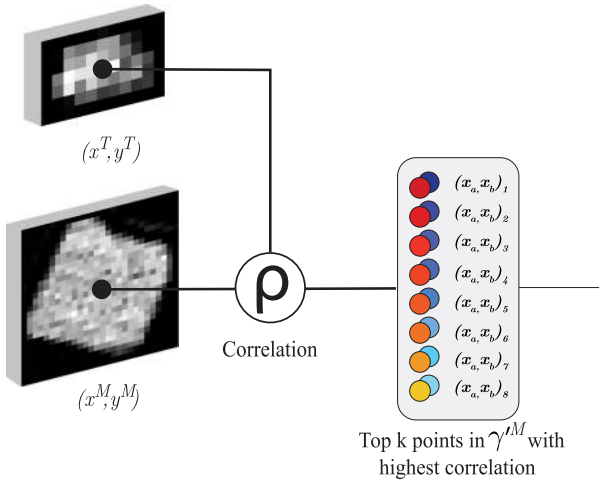


Fig. 4. Clustering using Highest Correlation: After calculation γ^M and γ^T, they are sorted in order of the highest correlation value of feature points represented by indices in the last dimension of the tensor. After sorting, top k points are selected which changes the shape of tensor. In the fig., the darker circles show the point with the highest correlation value and lighter circles represent decreased correlation.

to the dimensions feature ξ^M. β^T and β^M are interpreted as a collection of probability distributions for respective feature maps, as shown in the following two equations:

$$P(I = i, J = j | L = l, M = m) = \beta^T_{ijlm} \tag{4}$$

$$P(L = l, M = m | I = i, J = j) = \beta^M_{ijlm} \tag{5}$$

In the above equations, note that i and j corresponds to the feature map ξ^T while l and m corresponds to the feature map ξ^M.

C. Incorporating Soft Constraints

After the computation of the scores, we find the 2-D indices of the point of highest probability in each distribution using the soft-argmax function. The choice of soft-argmax function, represented by (6) and (7), instead of argmax is because the former is differentiable and thus ensures the smooth gradient transfer for backpropagation. It returns indices for the highest point in each probability distribution corresponding to the flattened dimension and these are stored as tensors γ^T and γ^M for respective feature maps ξ^T and ξ^M. Fig. 3 illustrates process.

$$\gamma^T = \left(\sum_{ij} \frac{e^{\beta_{ijlm}}}{\sum_{ab} e^{\beta_{ablm}}} \cdot i, \sum_{ij} \frac{e^{\beta_{ijlm}}}{\sum_{ab} e^{\beta_{ablm}}} \cdot j \right) \tag{6}$$

$$\gamma^M = \left(\sum_{lm} \frac{e^{\beta_{ijlm}}}{\sum_{cd} e^{\beta_{ijcd}}} \cdot l, \sum_{lm} \frac{e^{\beta_{ijlm}}}{\sum_{cd} e^{\beta_{ijcd}}} \cdot m \right) \tag{7}$$

The extracted indices belong to the highest probability points in one feature map associated with every point in the second feature map. Then, we use indices given by soft-argmax and the unraveled 2-D indices of each point in the flattened dimension to correlate the corresponding feature points in both the feature maps. The resulting correlation is subsequently used to sort the tensors γ^T and γ^M according to the descending correlation values and take the top-k points in the flattened dimension of both the tensors with highest correlation values in order to use them to predict corresponding points in Image I^T and Image I^M (see Fig. 4). This results in tensors γ'^T and γ'^M which represent matching points between feature maps. Here, k represents the number of points to select after correlation based sorting. Subsequently, the k 2-D points, represented by γ'^T (for feature A) and γ'^M (for feature B), are flattened and concatenated, as shown in 8 and 9, to be fed to a fully connected network. This network predicts the point-to-point correspondences between the two images. The output of the network consists of 32 elements which consist of eight 2-D (16) points for image I^T, represented by y^T_p and for image I^M, represented by y^M_p. This results in an output tensor of length 32.

$$\gamma'^T : \mathbb{R}^{k\times 2} \rightarrow \mathbb{R}^{2k} \qquad \gamma'^M : \mathbb{R}^{k\times 2} \rightarrow \mathbb{R}^{2k} \tag{8}$$

$$\gamma' = \gamma'^T \| \gamma'^M \tag{9}$$

The point-to-point correspondences, y^T_p and y^M_p, between images are then used to compute the Homography matrix. However, prior to this, RANSAC [17] has also been employed to minimize the effects of any outliers which may affect the computation of the Homography matrix. The four corners of the template image are then projected over to the main image and the center point of those projected pixel locations of the four corners is calculated. The GPS coordinates of the pixel location on the orthomosaic (main image) are extracted from the geo-tagged image and hence the UAV is localized on the map through its camera feed. The overview of the whole proposed approach is presented in form of Algorithm 1.

Algorithm 1: Template Matching-Based GPS Localization.

Input: Labeled template images $\{I_n^T\}_{n=1}^N$;
Geo-tagged Orthomosaics $\{I_o^M\}_{o=1}^O$;
Number of corresponding points k;
Output: GPS Coordinates: Latitude (ϕ), Longitude (λ);

1: Initialize $\mathbf{w}_\xi$ with weights for ImageNet.
2: Feature Extraction: $\xi^M = \theta(I^M, \mathbf{w}_\xi)$, $\xi^T = \theta(I^T, \mathbf{w}_\xi)$;
3: Correlation Matrix: $\alpha_{ijlm} = \frac{(\xi_{ij}^T, \xi_{lm}^M)}{\|\xi_{ij}^T\|_2 \|\xi_{lm}^M\|_2}$;
4: Soft Mutual Nearest Neighbor Filtering: $\alpha = \mathbf{s}^T \mathbf{s}^M \alpha$;
where $\mathbf{s}_{ijlm}^T = \frac{\alpha_{ijlm}}{\max_{ab} \alpha_{ablm}}$, $\mathbf{s}_{ijlm}^M = \frac{\alpha_{ijlm}}{\max_{cd} \alpha_{ijcd}}$
5: Neighborhood Consensus Network [20]:
$\alpha' = \varphi(\alpha, \mathbf{w}_n)$;
6: Soft Mutual Nearest Neighbor Filtering: $\alpha' = \mathbf{s}^T \mathbf{s}^M \alpha'$;
where $\mathbf{s}_{ijlm}^T = \frac{\alpha'_{ijlm}}{\max_{ab} \alpha'_{ablm}}$, $\mathbf{s}_{ijlm}^M = \frac{\alpha'_{ijlm}}{\max_{cd} \alpha'_{ijcd}}$
7: Pairwise Correspondences: $\beta_{ijlm}^T = \frac{e^{\alpha'_{ijlm}}}{\sum_{ab} e^{\alpha'_{ablm}}}$,
$\beta_{ijlm}^M = \frac{e^{\alpha'_{ijlm}}}{\sum_{cd} e^{\alpha'_{ijcd}}}$;
8: Soft-Argmax:
$\gamma^T = (\sum_{ij} \frac{e^{\beta_{ijlm}}}{\sum_{ab} e^{\beta_{ablm}}} \cdot i, \sum_{ij} \frac{e^{\beta_{ijlm}}}{\sum_{ab} e^{\beta_{ablm}}} \cdot j)$,
$\gamma^M = (\sum_{lm} \frac{e^{\beta_{ijlm}}}{\sum_{cd} e^{\beta_{ijcd}}} \cdot l, \sum_{lm} \frac{e^{\beta_{ijlm}}}{\sum_{cd} e^{\beta_{ijcd}}} \cdot m)$;
9: Selection of highly correlated top k points from γ^T and $\gamma^M \longrightarrow \gamma'^T, \gamma'^M$
10: Vectorization: γ'^T: $\mathbb{R}^{k\times 2} \rightarrow \mathbb{R}^{2k}$, γ'^M: $\mathbb{R}^{k\times 2} \rightarrow \mathbb{R}^{2k}$
11: Concatenation: $\gamma' = \gamma'^T \| \gamma'^M$;
12: Fully Connected Network: $y_p = \zeta(\gamma,' \mathbf{w}_z)$
where $y_p = y_p^T \| y_p^M$
y_p^T and y_p^M are regressed points on the image belonging to template and orthomosaic respectively.
13: Homography Estimation Using Direct Linear Transformation $\longrightarrow \mathbf{H}$;
14: Projection of Template onto Orthomosaic: $y_M^T = \mathbf{H} \cdot y^T$
15: Extraction of 2D Spatial Coordinates Using Geo-coded Orthomosaic: $y_M^T \longrightarrow (\phi, \lambda)$

D. Model Training and Optimization

Data for training and optimizing the proposed model comprises of a set of template images I^T and their labels y_a which represent the pairs of labeled point-to-point correspondences between template images and the relevant orthomosaic. Moreover, orthomosaics I^M, belonging to three different areas, act as the second input to the model and their correspondence with the template image is taken into account by the training procedure. Furthermore, training data has also been augmented using image transformations in order to increase variance and the number of images in the dataset. This also eliminates the dependence of point-to-point correspondence prediction on orientation and size of the template image. Lastly, the training data is also mean-subtracted and normalized with the standard deviation.

The feature extraction model, based on convolutional layers of ResNet-101 [41], is used to process template images and orthomosaics by mapping them into a convolutional feature space. The network weights $\mathbf{w}_\xi$ are initialized using ImageNet weights [42]. Fully connected network, which is present in the later part of the proposed pipeline, is initialized with random weights using He-normal initialization [43]. Moreover, as this network acts as a regressor, ReLU activations have been used at the output layer of this network and the proposed approach has been trained and optimized in an end-to-end supervised manner which aims to minimize the L_2 regression loss. The optimization routine is presented by (10) where y_p represents the predicted correspondences between template and orthomosaic. Minimization of cost function and optimization of trainable parameters of the network are done using stochastic gradient descent with momentum.

$$\min_{\mathbf{w}_\xi \mathbf{w}_n \mathbf{w}_z} \sum_{i=1}^{n} L(y_p, y_a) \tag{10}$$

The model is trained for 185 epochs using early-stopping paradigm with an initial learning rate of 0.0008. The complete architecture contains a total of 27.733 million parameters. The learning rate is also decayed with a factor of 0.5 in the case of plateauing of validation loss and the optimization algorithm uses momentum with a value of 0.9. Training process has been executed on NVIDIA Titan X GPU and it has been deployed on NVIDIA Jetson TX2 for inference to emulate real-life setting by providing on-board computing capability for UAVs.

After the prediction of point-to-point correspondences, the spatial 2D coordinates (in standard WGS-84 format) are extracted out via GDAL [44] library from pre-ordained geotagged orthomosaic by utilizing the center pixel coordinates of the template projection over the orthomosaic.

IV. Experimental Evaluation

Various experiments have been performed in order to measure the performance of the proposed approach by emulating real conditions of UAV imagery in our test dataset. Extensive study of algorithmic details and hyperparameters has also been carried out to determine the best possible configuration of the model in terms of efficiency and accuracy and are presented in form of ablation study in this section.

A. Dataset

1) Data Acquisition: To demonstrate the correctness and efficiency of the proposed solution, we trained and tested the performance of the proposed approach on custom built dataset whose detail is provided next.

To our knowledge, there does not exist any dataset for aerial imagery with respect to aerial image localization perspective. Thus, to train and test the proposed approach, we collected our own dataset of the aerial imagery. For this task, we selected three different cities/regions for UAV flights and collected overlapping images over them using the DJI Phantom 4 Pro drone. Specifically, the images were acquired in nadir looking orientation via multiple flights over the partial regions of NUST, Islamabad, DHA, Rawalpindi and Gujar Khan district at regular intervals along the flight path. Table I shows statistics and information about the area covered and the number of images in the raw data collected.

Fig. 5. Geo-tagged Orthomosaics of three different areas. (a) A geographical patch from NUST Islamabad which covers about 0.522 km^2 area. (b) Area in DHA Rawalpindi having sparsely populated terrain and covering upto 0.64 km^2 area. (c) Densely populated urban area of Gujar Khan District in Pakistan which represents 0.664 km^2 land. (a) NUST. (b) DHA. (c) Gujar Khan.

Fig. 6. Comparison of images of the same landmark collected from UAV at different times of the day. The inclusion of images in the training data that are variant in illumination conditions helps deep networks to become robust to changes in lighting, hence, making image matching more accurate.

TABLE I
DATASET STATISTICS SHOWING THE AMOUNT OF AREA COVERED BY THE NUMBER OF IMAGES COLLECTED OVER THE COURSE OF NINE FLIGHTS IN THREE DIFFERENT AREAS

Area Name	**No. of Images**	**Area Covered** (m^2)
NUST, Islamabad	1200	522,000
DHA, Rawalpindi	480	640,000
Gujar Khan District	372	664,000

After the images had been taken and compiled, they were stitched together to form orthomosaics in an efficient manner as images were overlapped during collection. Since the captured images had GPS coordinates associated with them, every pixel of the corresponding orthomosaic was geo-tagged automatically. Images from one of the flights for each area (the one with better lighting conditions) are used for orthomosaic generation process. Generated orthomosaics are shown in Fig. 5.

The collected images, which act as template images in our proposed approach, are taken during three different periods of the day in order to maximize variance in illumination conditions as shown in Fig. 6. This makes our approach more robust and invariant to change in lighting conditions as possible in several flight settings. Moreover, the difference in terrain of three different areas also helps to combat overfitting of approach to one specific terrain. We have three different terrains in all of three areas as Gujar Khan district appears to be a densely populated urban area, DHA Rawalpindi presents a sparsely populated residential area with water bodies and greenery and NUST Islamabad lies in the middle of this spectrum as it has complex terrain pattern with buildings, water bodies without the density of an urban area. This is best illustrated in Fig. 5.

The collected dataset (images) have also been expanded by applying standard augmentation techniques in order to increase the size of the dataset to prevent overfitting. Specifically, we performed the image flipping over the vertical axis, horizontal axis and around the image center. Furthermore, the respective transformations are also reflected in point-to-point correspondence labels to correctly match the points of the source images and the target orthomosaic. These are the only augmentations which we employed. Usually, there exist other (affine) transformations as well such as random rotations, but we have not employed them as they may cause differences in the tagged GPS coordinates.

2) Annotations: Each image in the dataset is annotated with at maximum 16 associated point-to-point correspondences on both the orthomosaic and the image itself. During model training, we have used eight out of them to better optimize the

TABLE II
COMPARISON OF MATCHING METHODS VIA CONVENTIONAL KEYPOINT DESCRIPTORS AND DEEP-LEARNING-BASED FRAMEWORKS

Matching Method	Pipeline	Correct Matches	Accuracy (%)
Classical Keypoint Descriptors	SURF [12]	167/300	55.66
	BRISK [14]	196/300	65.33
	SIFT [12]	210/300	70.00
Deep Learning Based	DELF [36]	109/300	36.33
	BUPM [19]	164/300	54.66
	BUPM (fine-tuned)	228/300	76.00
	QATM [18]	172/300	57.33
	QATM (fine-tuned)	236/300	78.66
	Proposed	278/300	92.66

performance. Each annotated correspondence has two points and every point is two dimensional. The output of the model also predicts the corresponding points and then the mappings of the template image over the orthomosaic using labeled and predicted points are compared. The correspondences for all the training and test data are labeled and verified by human annotators. The dataset including the annotations has been made public on GitHub.[2]

B. Results Evaluation

The presented approach has been tested in terms of correctness of point-to-point correspondences and error in GPS coordinate localization of predicted position over the geographical map.

1) Matching Accuracy: The model essentially provides us with point-to-point correspondences between the two images. Subsequently, the Homography and RANSAC [17] are employed after extraction of corresponding points to localize the source template in the target patch. Table II provides the comparison of the correct matches obtained by the proposed architecture as well as with the classical key-point descriptors and recent deep learning based approaches. For this comparison, the match is considered to be correct when the predicted bounding box overlaps with the 90% of the bounding box obtained from the labeled correspondences. As expected, the deep learning based matching accuracies are much higher than the conventional key-points descriptor based techniques due to their ability of contextual feature learning. Among them, the proposed approach evidently provides the best matching accuracy.

It is worth mentioning that the overlapping imagery used for the orthomosaic generation is from only one of the three flights over each of the selected regions. The achieved matching accuracy when the source image is taken either from the same flight or different flights, whose images are used to generate the target orthomosaic, are 92.95% and 92.35% respectively. It is evident that there is indeed a slight reduction when the images from the different flights owing to variations in illumination. However, the reduction is quite small (less than 1%) which highlights the robustness of the proposed approach.

2) GPS Localization Error: To estimate the positioning accuracy, we have computed the center pixel deviation of the predicted localized region in the orthomosaic from its actual (ground-truth) GPS coordinates. Table III shows the comparison

[2][Online]. Available: github.com/m-hamza-mughal/aerial-template-matching-dataset

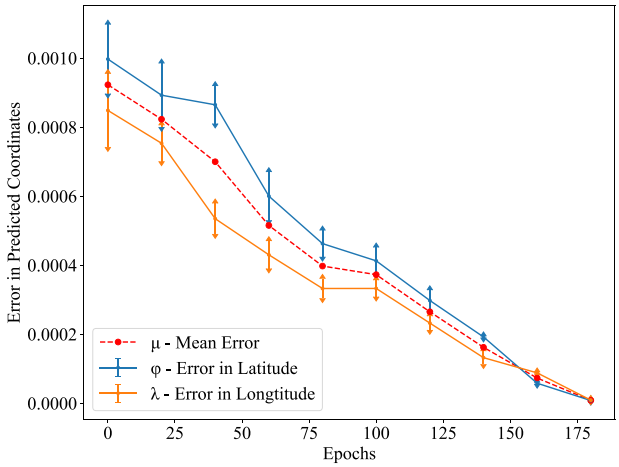


Fig. 7. Average mean error (depicted in red) and its standard deviation in predicted values for the latitude/longitude and their corresponding average errors shown with respect to the increasing number of epochs. The mean error in coordinates starts at 9.2255×10^{-04} and steadily decreases to 7.325×10^{-06} at 185^{th} epoch as the training procedure is stopped at the point where validation loss is the least. Here, the error in latitude and longitude are represented by ϕ and λ, respectively while, the μ represents the mean error in both the latitude and dimensions.

of the average positioning error (AE) and the maximum positioning error (ME) in squared meters of the proposed method along with classical and recent approaches. Fig. 7 graphically shows the decline in the average mean error in the predicted latitudinal and longitudinal coordinates as the number of epochs increases. It also shows the standard deviation at each point and it can be seen that the standard deviation in error is also decreased as the network learns to localize the template better.

3) Intersensorial Registration: The intersensorial registration problem deals with matching the onboard UAV image sequences (or the source templates) with the pre-installed larger reference target orthomosaic generated using the satellite imagery (instead the orthomosaic generated by the same sensor which is the drone in our case). Although the article does not specifically target this problem and instead solves the intrasensorial registration, still we have conducted an experiment where we generated the orthomosaic using the Bing maps satellite imagery of the same regions for which we have conducted the UAV flights. The generated orthomosaics are then taken as the reference target images in which the source template images acquired from UAV are matched/searched. The exact four corner coordinates of the generated orthomosaics for the three different areas where we have flown the drone including NUST, DHA and Gujar Khan respectively are provided in the GitHub link of the Dataset (mentioned above) along with the generated orthomosaics and the drone flight imagery. The overall matching accuracy over the three areas which has been achieved with the correct matching pairs is 91.04% with the average and maximum GPS localization error of the center pixel of around 3.708 m^2 and 33.164 m^2 respectively. It can be seen that these results are quite close to the obtained intrasensorial results (mentioned in Table II) which demonstrates the robustness of the proposed approach.

TABLE III
COMPARISON OF GPS LOCALIZATION ERROR FOR CONVENTIONAL KEYPOINT DESCRIPTORS AND DEEP-LEARNING BASED FRAMEWORKS. HERE **AVERAGE ERROR** (AE) IS CALCULATED AS THE AVERAGE POSITIONING ERROR IN THE LOCALIZED COORDINATES FOR ALL TEMPLATE IMAGES IN THE TESTING SET WHILE THE **MAXIMUM ERROR** (ME) IS THE MAXIMUM OF ALL THE POSITIONING ERRORS CORRESPONDING TO THE TESTING SET. BOTH **AVERAGE ERROR** (AE) AND **MAXIMUM ERROR** (ME) ARE REPRESENTED IN METERS FOR DISTANCE AND DEGREES FOR COORDINATES

Pipeline	**Average Error** (m^2)	**Maximum Error** (m^2)	**Average Error** (deg)	**Maximum Error** (deg)
SIFT [11]	12.537	75.615	$8.81 \times 10^{-6o}N$ $1.880 \times 10^{-5o}E$	$3.49 \times 10^{-5o}N$ $3.24 \times 10^{-5o}E$
SURF [12]	17.958	108.918	$1.007 \times 10^{-5o}N$ $2.270 \times 10^{-5o}E$	$4.09 \times 10^{-5o}N$ $4.04 \times 10^{-5o}E$
BRISK [14]	15.881	98.352	$9.07 \times 10^{-6o}N$ $2.170 \times 10^{-5o}E$	$3.99 \times 10^{-5o}N$ $3.68 \times 10^{-5o}E$
DELF [36]	28.291	147.766	$1.221 \times 10^{-5o}N$ $2.89 \times 10^{-5o}E$	$4.139 \times 10^{-5o}N$ $5.49 \times 10^{-5o}E$
BUPM [19]	15.442	104.459	$6.34 \times 10^{-5o}N$ $2.27 \times 10^{-5o}E$	$4.12 \times 10^{-5o}N$ $3.78 \times 10^{-5o}E$
BUPM (fine-tuned)	4.723	53.118	$1.097 \times 10^{-5o}N$ $1.30 \times 10^{-6o}E$	$3.455 \times 10^{-5o}N$ $1.58 \times 10^{-5o}E$
QATM [18]	17.561	128.961	$1.197 \times 10^{-5o}N$ $2.11 \times 10^{-5o}E$	$5.072 \times 10^{-5o}N$ $3.28 \times 10^{-5o}E$
QATM (fine-tuned)	4.129	47.12	$9.81 \times 10^{-6o}N$ $3.80 \times 10^{-6o}E$	$2.64 \times 10^{-5o}N$ $2.77 \times 10^{-5o}E$
Proposed	3.594	31.281	$7.65 \times 10^{-6o}N$ $7.00 \times 10^{-6o}E$	$1.49 \times 10^{-5o}N$ $2.90 \times 10^{-5o}E$

TABLE IV
COMPARISON OF THE COMPUTATIONAL COMPLEXITY OF FEATURE EXTRACTOR IN TERMS OF NUMBER OF PARAMETERS WITH THE PERCENTAGE ACCURACY. AS DEMONSTRATED BY THIS COMPARISON, RESNET-101 GIVES MOST ACCURATE RESULTS BUT REQUIRES HIGH COMPUTATIONAL POWER

Feature Extractor	**Number of Parameters (million)**	**Accuracy (%)**
VGG-16 [45]	7.63M	79.3
ResNet-50 [41]	8.54M	86.9
DenseNet-201 [46]	1.42M	89.1
ResNet-101 [41]	27.53M	92.6

TABLE V
HIDDEN LAYER (HL) CONFIGURATIONS FOR 4D CONVOLUTIONAL NETWORK

Configuration	**HL 1**	**HL 2**	**Output**
Proposed-5	5	5	1
Proposed-10	10	10	1
Proposed-15	15	15	1

C. Ablation Study

1) Comparison of Feature Extractors: The convolutional feature extractors are used for deep feature extraction from the images which in turn enable the neighborhood consensus network to learn matching schemes based on consistent features in the images. The correlation patterns in the feature maps are highly dependent on the architecture of the feature extractor. To see the effect, we have compared different feature extractors including VGG-16 [45], ResNet-50, ResNet-101 [41] and DenseNet-201 [46] and chose the one giving the best performance.

As presented by Table IV, VGG-16 proves to be the least robust in our case as it is the least accurate. DenseNet-201 proves to be highly efficient in terms of parameters but is less accurate as compared to the most accurate ResNet-101. ResNet-50 has also been tested but it turns out to be in the less accurate feature extractors despite being more efficient. ResNet-101 provides the most accurate results which are shown in Table II. Moreover, it can also be inferred that the residual connections tend to help in meaningful feature extraction which aids subsequent 4D convolutional network to learn better filtration of matches. DenseNet-201 or ResNet-50 can be used in those UAV localization settings which have lesser reliance on accurate matches and can bear error in GPS coordinate estimation.

2) Exploration of Probabilistic Constraints: Introducing probabilistic constraints enables the network to filter out the strongest matches after the nearest neighbor filtering of the correlations between features. These softmax-based constraints help determine consistent matches between all the feature points of both images. The corresponding probability scores are represented by (4) and (5) and are visualized in form of heatmaps in Fig. 8. These probability distributions show high values in the vicinity of pixel locations in the main image which are similar to the template image thus, showing the model's ability to correctly identify matching regions in the main image and localize the template image accordingly.

3) Variation in Hyperparameters: The 4D convolutional network processes the correlation tensor and finds patterns according to neighborhood consensus matching scheme which it has learned during the training process. The network contains certain number of hidden layer units which can be tweaked in order to achieve an efficient balance between accuracy and latency. Table V shows different configurations of neighborhood consensus network [20] which have been tried on the deployment platform i.e. NVIDIA Jetson TX2 for comparison in terms of performance and inference time. Fig. 9 clearly illustrates that the network with

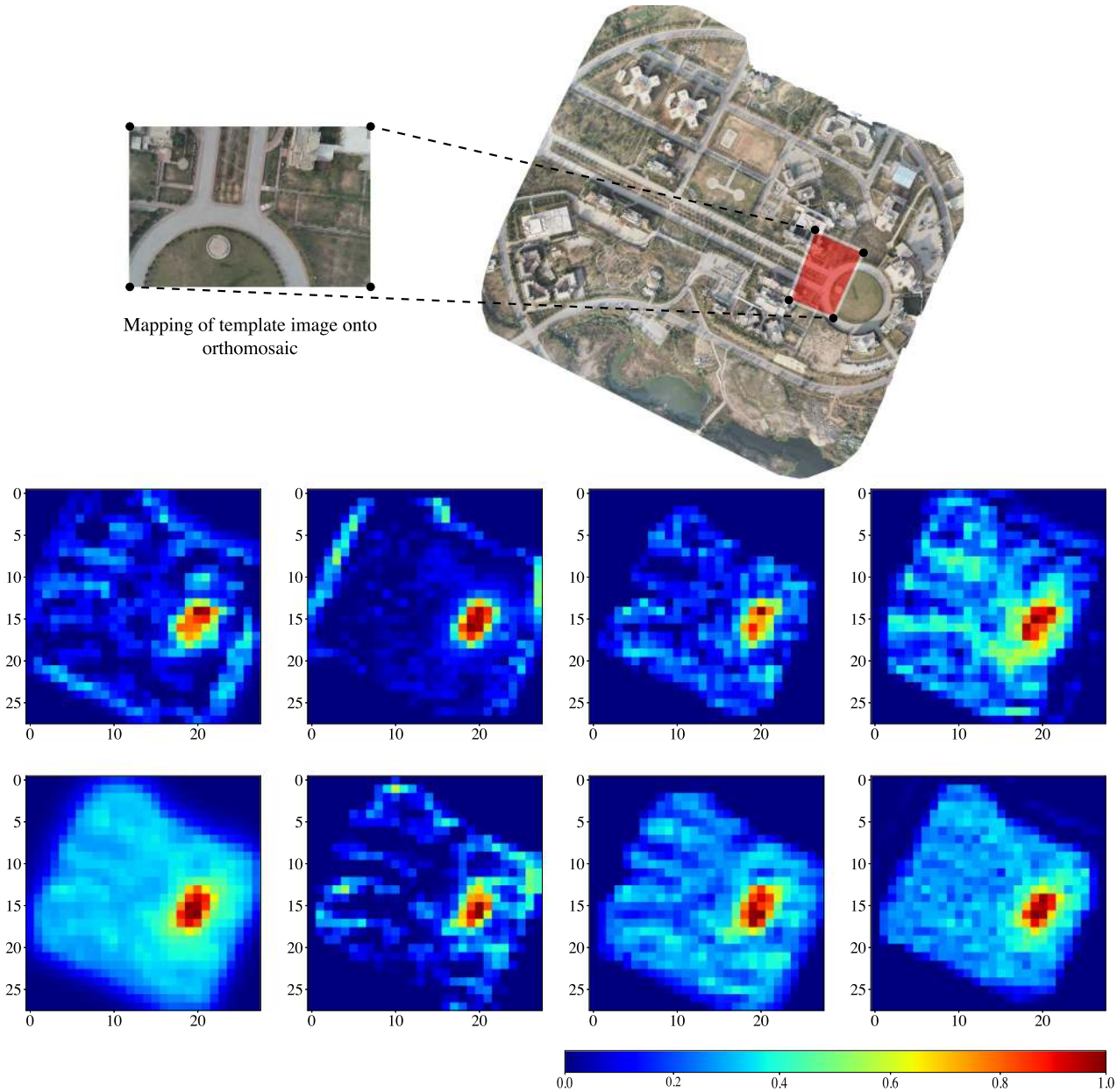


Fig. 8. Softmax-based constraints visualized: This image shows 2-D probability distributions (28 x 28) of ξ^M (orthomosaic) for ξ^T (template) as an input. These eight images show probability distributions over ξ^M for top eight points which have highest correlation with points in ξ^T. Matching capability of probabilistic constraints can be visualized using this fig. as they highlight the areas in the orthomosaic feature map which match the template.

10 units provides an optimal balance of latency and accuracy by giving 92.66% accuracy at 3.77 seconds of inference time.

4) Comparison of Number of Point-to-Point Correspondences: The output of the whole pipeline outputs pairs of point-to-point correspondences between the template image and the main image which in turn is used to extract GPS coordinates in the geographical map. These correspondences are made dependent on the number of the feature point indices passed to the fully-connected network (FCN) which regresses the point-to-point correspondences. Changing the number of feature point indices in order to improve performance or to achieve required latency can provide valuable insights for the selection of the configuration which minimizes positioning error without much computational burden. The network has been trained to generate 4, 8 and 16 pairs of correspondences between the images, detailed in Table VI, and they have been compared on basis of complexity and average positioning error as illustrated by Fig. 10. As evident, the eight number of point-to-point correspondences provides the best results and have total number of parameters equals to 16 672. The calculation of homography matrices between the template and main image is done from these correspondences which eventually contribute to localization of the template over the main image.

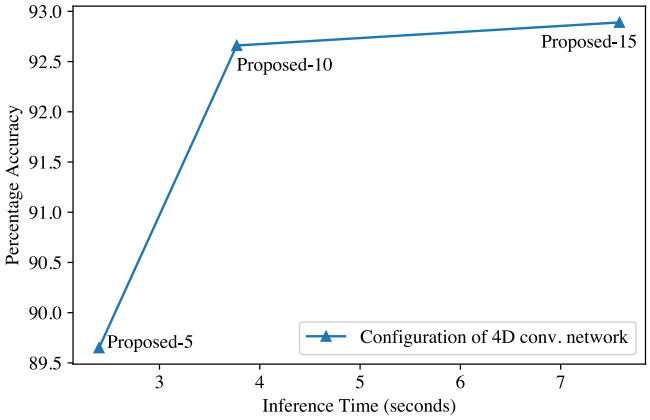


Fig. 9. Configurations of 4D convolutional network shown w.r.t to their percentage accuracy at the output and their inference times on NVIDIA Jetson TX2 platform. This clearly shows that using five hidden units in the network does not give enough capacity to learn better feature matches and using 15 hidden units proves to be very computationally expensive. Therefore, ten hidden layers come out as an efficient and accurate option.

TABLE VI
FCN SPECIFICATIONS FOR 4, 8, 16 POINT-TO-POINT CORRESPONDENCES

Layer	Number of Units		
Correspondences:	**4**	**8**	**16**
Input	16	32	64
Layer 1	64	64	64
Layer 2	64	64	64
Layer 3	64	64	64
Layer 4	64	64	64
Output	16	32	64

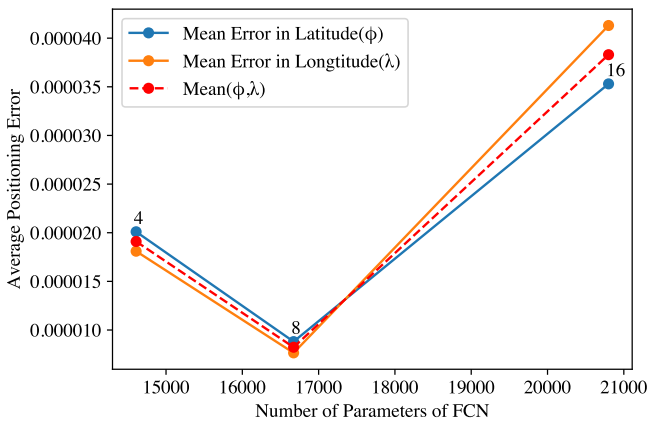


Fig. 10. Configurations of Fully Connected Network (FCN) based Regressor compared with the average positioning error. This comparison shows eight point-to-point correspondance configuration is very efficient in terms of computational burden as it also gives the least average positioning error.

V. DISCUSSION AND CONCLUSION

Vision-based autonomous localization of UAVs is of vital importance in situations where the GPS signals may suffer from outages or jamming problems. Such a task of vision based localization estimation of an airborne platform essentially boils down to the task of robust template matching. In this context, we have presented a robust deep learning based feature points extraction approach which is used for template matching by establishing point-to-point correspondences between the source template and the target pre-stored orthomosaic. The results on real UAV imagery show a high matching accuracy and much superior performance compared to already existing methods in classical computer vision. Despite this superior performance, there are a few worth mentioning points in relation to the proposed approach which are as follows:

- *Novel Dataset:* We have collected and annotated novel dataset consisting of 2052 high resolution aerial images acquired at different times over three different areas in Pakistan spanning a total area of around 2 km^2. We shall make the dataset public which we believe would ignite and open up new possibilities of further advancement in vision based GPS localization.
- *Nadir View Imagery:* Although we have tried to cater for changes in topography by acquiring all the overlapping imagery in the "nadir" view over different areas of varying topographical nature including densely populated urban areas and sparsely populated residential areas with water bodies and vegetation, still the effect of terrain consisting of a mix of mountain and a flat area may impact the matching accuracy which in turn could lead to errors in localization. Generalizing this to other viewing angles and varying heights would make the problem of finding the source image patch in the target orthomosaic highly challenging. Having said this, the "nadir" view makes the matching somewhat easier as it alleviates the problem of learning complex viewpoint changes.
- *Height Estimation:* Firstly, instead of 2-D, we need the actual 3-D position to aid the navigation system. Assuming the terrain is flat and the UAV is equipped with the barometer pressure sensor that provides the estimated height information, the "nadir" viewing geometry in our case somewhat resolves the 3-D positioning issue. However, if there is no sensor that provides the UAV height information, the height estimation using a single monocular RGB onboard camera is still possible using odometric and/or visual simultaneous localization and mapping based techniques but has not been addressed in this work.
- *Latency & Onboard Computing:* To reduce latency in the matching scheme, the whole processing should be done on the onboard flight computer. To this end, we have deployed the model on NVIDIA Jetson TX2 and inferred latency of around 3.7 seconds. With this inference latency time, the proposed localization system can aid the GPS based navigation system in scenarios where the integrity of the GPS is in question by reducing the drifting effect in INS based state-estimation. Nevertheless, the inference latency can be further improved by incorporating feature extractors such as MobileNet [47] to achieve better efficiency. Moreover, the substitution of convolution operations with depthwise separable convolutions [48] and implementation of 4D convolution with lesser number of calculations seems a promising future direction that can reduce the computational complexity of the neighborhood consensus

network to enable models to efficiently run on edge devices.

- *System Integration:* From product perspective, the integration of the estimated 3-D UAV localization (i.e., the 2-D position estimation with the proposed scheme plus the barometric height estimate) with the INS based state estimation is indeed vital to develop a complete and autonomous navigation solution.
- *Practical Solution (Inter-Sensorial Matching):* Although the orthomosaic has been pre-built and has to be stored for template matching, this may have practical limitations. A more viable solution would be to use large-scale satellite orthomosaics as e.g., in Wan *et al.* [49] or available in Google Earth software. Preliminary results in this direction have been included in this work. However, large-scale matching would pose additional challenges in the context of domain adaptation since the drone imagery would have to be matched with the rectified satellite imagery. In the future, we would extend the presented approach in this direction.

REFERENCES

[1] C. Wang, T. Wang, J. Liang, Y. Chen, Y. Zhang, and C. Wang, "Monocular visual SLAM for small UAVs in GPS-denied environments," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2012, pp. 896–901.

[2] M. Shan, F. Wang, F. Lin, Z. Gao, Y. Tang, and B. Chen, "Google map aided visual navigation for UAVs in GPS-denied environment," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2015, pp. 114–119.

[3] J. Li *et al.*, "Real-time simultaneous localization and mapping for UAV: A survey," in *Proc. Int. Micro Air Veh. Conf. Competition*, 2016, pp. 237–242.

[4] E. J. Bekkers, M. Loog, B. M. Romeny, and R. Duits, "Template matching via densities on the roto-translation group," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 452–466, Feb. 2018.

[5] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," in *Proc. Opt. Pattern Recognit. XII*, 2001, vol. 4387, pp. 95–102.

[6] J.-F. Dou and J.-X. Li, "Robust image matching based on wavelet transform and SIFT," in *Proc. 3rd Int. Conf. Digit. Image Process.*, 2011, vol. 8009, pp. 437–444.

[7] D. Nair, R. Rajagopal, and L. Wenzel, "Pattern matching based on a generalized fourier transform," in *Proc. Int. Symp. Opt. Sci. Technol. Int. Soc. Opt. Photon., Bellingham*, 2000, pp. 472–480.

[8] Z. Zhang, J. Chen, X. Li, W. Li, and W. Yuan, "An image matching method based on fourier and LOG-Polar transform," *Sensors Transducers*, vol. 169, pp. 61–66, 2014.

[9] J. Lai, L. Lei, K. Deng, R. Yan, Y. Ruan, and Z. Jinyun, "Fast and robust template matching with majority neighbor similarity and annulus projection transformation," *Pattern Recognit.*, vol. 98, 2020, Art. no. 107029.

[10] J. Yoo, S. S. Hwang, S. D. Kim, M. S. Ki, and J. Cha, "Scale-invariant template matching using histogram of dominant gradients," *Pattern Recognit.*, vol. 47, no. 9, pp. 3006–3018, 2014.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Understanding*, vol. 110, pp. 346–359, 2008.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 Int. Conf. Comput. Vis.*, Barcelona, 2011, pp. 2564–2571, doi: 10.1109/ICCV.2011.6126544.

[14] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2548–2555.

[15] H. Yang, C. Huang, F. Wang, K. Song, S. Zheng, and Z. Yin, "Large-scale and rotation-invariant template matching using adaptive radial ring code histograms," *Pattern Recognit.*, vol. 91, pp. 345–356, 2019.

[16] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. 4th Int. Conf. Comput. Vis. Theory Appl.*, 2009, pp. 331–340.

[17] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, Jun. 1981.

[18] J. Cheng, Y. Wu, W. AbdAlmageed, and P. Natarajan, "QATM: Quality-aware template matching for deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11545–11554.

[19] J. Cheng, Y. Wu, W. AbdAlmageed, and P. Natarajan, "Image-to-GPS verification through a bottom-up pattern matching network," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 546–561.

[20] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," in *Adv. Neural Inf. Process. Syst.*, vol. 31, US: Curran Associates, Inc., 2018, pp., pp. 151–1662.

[21] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: A comparison to SIFT," 2015, [Online]. Available: *arXiv:1405.5769*.

[22] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2016, pp. 1191–1191.

[23] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk, "PN-Net: Conjoined triple deep network for learning local image descriptors," 2016, [Online]. Available: *arXiv:1601.05030*.

[24] J. Bian, W. Lin, Y. Matsushita, S. Yeung, T. Nguyen, and M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2828–2837.

[25] F. Schaffalitzky and A. Zisserman, "Automated scene matching in movies," in *Proc. Int. Conf. Image Video Retrieval*, 2002, pp. 186–197.

[26] A. Canhoto, E. H. Shiguemori, and M. A. P. Domiciano, "Image sequence processing applied to autonomous aerial navigation," in *Proc. IEEE Int. Conf. Signal Image Process. Appl.*, 2009, pp. 496–499.

[27] T. Koch, X. Zhuo, P. Reinartz, and F. Fraundorfer, "A new paradigm for matching UAV and aerial images," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 83–90, 2016.

[28] Y. Lin and G. Medioni, "Map-enhanced uav image sequence registration and synchronization of multiple image sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–7.

[29] P. Solbrig, D. Bulatov, J. Meidow, P. Wernerus, and U. Thonnessen, "Online annotation of airborne surveillance and reconnaissance videos," in *Proc. Int. Conf. Inf. Fusion*, 2008, pp. 1–8.

[30] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6959–6968.

[31] A. Marcu, D. Costea, E. Slusanschi, and M. Leordeanu, "A multi-stage multi-task neural network for aerial scene interpretation and geolocalization," 2018. [Online]. Available: *arXiv:1804.01322*.

[32] A. Nassar, K. Amer, R. ElHakim, and M. ElHelw, "A deep CNN-Based framework for enhanced aerial imagery registration with applications to UAV geolocalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 1594–159410.

[33] X. Zhuo, T. Koch, F. Kurz, F. Fraundorfer, and P. Reinartz, "Automatic UAV image geo-registration by matching UAV images to georeferenced image data," *Remote Sens.*, vol. 9, no. 4, p. 376, 2017, doi: 10.3390/rs9040376.

[34] S. Chen, X. Li, Y. Zhang, R. Feng, and C. Zhang, "Local deep hashing matching of aerial images based on relative distance and absolute distance constraints," *Remote Sens.*, vol. 9, 2017, Art. no. 1244.

[35] H. Altwaijry, E. Trulls, J. Hays, P. Fua, and S. Belongie, "Learning to match aerial images with deep attentive architectures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3539–3547.

[36] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3476–3485.

[37] Y. Tian, C. Chen, and M. Shah, "Cross-view image matching for geo-localization in urban environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1998–2006.

[38] D. Buniatyan, T. Macrina, D. Ih, J. Zung, and H. Seung, "Deep learning improves template matching by normalized cross correlation," 2017, [Online]. Available: *arXiv:1705.08593*.

[39] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, Jun. 2010.

[40] H. Taira *et al.*, "InLoc: Indoor visual localization with dense matching and view synthesis," in *2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, 2018, pp. 7199–7209, doi: 10.1109/CVPR.2018.00752.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

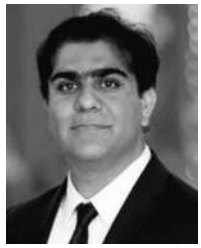
[42] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A. large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.
[44] GDAL – GDAL documentation, Accessed date: Feb. 7, 2021. [Online]. Available: https://gdal.org/
[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, May 7–9, 2015, , pp. 1–14.
[46] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
[47] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, [Online]. Available: *arXiv:1704.04861*.
[48] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1800–1807.
[49] X. Wan, J. Liu, H. Yan, and G. L. K. Morgan, "Illumination-invariant image matching for autonomous UAV localisation based on optical sensing," *ISPRS J. Photogrammetry Remote Sens.*, vol. 119, pp. 198–213, 2016.

Muhammad Hamza Mughal received the bachelor's in electrical engineering degree from the National University of Sciences and Technology, Islamabad, Pakistan.

He is currently working as a Deep Learning Engineer where he is responsible for building deep learning and computer vision capabilities to provide solutions for various use-cases in the field of artificial intelligence. He is doing research in the field of computer vision and deep learning. Moreover, his work on human attributes classification on body images has been filed for a patent. He is avidly working toward research in fields of applications of deep learning in remote sensing and 3D computer vision particularly in image matching, object detection, person re-identification and tracking, and multi-object tracking.

Muhammad Jawad Khokhar received the Ph.D. degree in computer science from Inria Sophia Antipolis, France.

He is currently working as a Data Scientist with Teradata Global Delivery Center, Islamabad, Pakistan. At Teradata, he has worked on Data Science and machine learning projects for several industries including telecom and Healthcare. He also has extensive experience in the telecom sector mainly in operations, analysis, and optimization of mobile core networks. Overall, his expertise include data science, machine/deep learning, and internet measurements.



Muhammad Shahzad received the B.E. degree in electrical engineering from the National University of Sciences and Technology, Islamabad, Pakistan, the M.Sc. degree in autonomous systems (robotics) from the Bonn Rhein Sieg University of Applied Sciences, Sankt Augustin, Germany, and the Ph.D. degree in radar remote sensing and image analysis awith the Department of Signal Processing in Earth Observation (SiPEO), Technische Universitt München (TUM), Munich, Germany in 2004, 2011, and 2016, respectively. His Ph.D. topic was the automatic 3-D reconstruction of objects from point clouds retrieved from spaceborne synthetic-aperture-radar (SAR) image stacks. Besides, he has also attended twice two weeks of a professional thermography training course with Infrared Training Center (ITC), North Billerica, Massachusetts, USA in 2005 and 2007.

He worked as a Guest Scientist with the Institute for Computer Graphics and Vision (ICG), Technical University of Graz, Austria from November 2015 to January 2016. Since October 2016, he has been working as an Assistant Professor with the School of Electrical Engineering & Computer Science (SEECS), National University of Sciences & Technology (NUST), Islamabad, Pakistan. His research interests include application of deep learning to solve remote sensing and computer vision problems especially processing both unstructured/structured 3D point clouds, optical RGBD images, and very high-resolution radar data.