A Residual-Dyad Encoder Discriminator Network for Remote Sensing Image Matching

Numan Khurshid†, *Student Member, IEEE*, Mohbat†, Murtaza Taj, *Member, IEEE,* and Faisal Z. Qureshi, *Senior Member, IEEE*

***Abstract*—We propose a new method for remote sensing image matching. The proposed method uses encoder subnetwork of an autoencoder pre-trained on GTCrossView data to construct image features. A discriminator network trained on University of California Merced Land Use/Land Cover dataset (LandUse) and High-resolution Satellite Scene dataset (SatScene) computes a match score between a pair of computed image features. We also propose a new network unit, called *residual-dyad*, and empirically demonstrate that networks that use residual-dyad units outperform those that do not. We compare our approach with both traditional and more recent learning-based schemes on LandUse and SatScene datasets, and the proposed method achieves state-of-the-art result in terms of mean average precision and ANMRR metrics. Specifically, our method achieves an overall improvement in performance of 11.26% and 22.41%, respectively, for LandUse and SatScene benchmark datasets.**

***Index Terms*—Remote Sensing Image Search, Convolutional Neural Network (CNN), Residual Encoder Decoder, Deep Learning, Content Based Remote Sensing Image Retrieval (CBRSIR).**

I. Introduction

REMOTE Sensing Imaging (RSI) technologies promise to revolutionize how we study Earth's surface, atmosphere and ionosphere. Imagery collected via special hardware mounted on satellites and aircrafts is now routinely used to examine weather patterns, plant habitats, urban infrastructure, road networks, archaeological sites, forest fires, flood planes and mineral resources [1]–[6]. Advances in airborne imaging and sensor technologies especially source-type approaches, have made it possible to capture large volumes of imagery covering extended areas [7]. Our ability to collect large quantities of RS imagery for a variety of domains have engendered a need for "intelligent" tools. Ideally, these tools should be able to perform useful analysis with little or no human intervention.

Remote sensing data consists of either panchromatic images photographed using high definition cameras [8], [9] or hyper-spectral imagery acquired through specialized imaging devices [10]. A first step towards designing automated systems that support a variety of sophisticated task-driven analysis [4], [11] is to develop basic image analysis techniques, such as image matching, classification, or retrieval. Developing such low-level image analysis techniques have been the primary focus of much of the work done by computer vision researchers in this domain [12]–[14]. The focus of this work is in this vain as well.

Broadly speaking, existing image matching techniques developed for remote sensing image retrieval (RSIR) can be divided into two classes: 1) classical hand crafted approaches that do not require any training data [15]–[18], and 2) more recent learning-based approaches that often need access to labelled data [19]–[22]. In this paper, we propose a deep learning approach for remote sensing image matching (RSIM) for the purpose of image retrieval. Specifically, our method uses a convolutional autoencoder to construct deep features of a given image. These features are subsequently used for image matching. Generator network in combination with discriminator network have been employed to classify remote sensing images before; however, previous approaches [23][24] require large labelled datasets since these perform feature extraction and discrimination simultaneously within a supervised learning setting.

In this work we exploit the fact that 1) autoencoders can be trained to minimize the reconstruction loss in an unsupervised manner and 2) the trained encoder sub-network can be used to extract deep features (of a given image) that can be useful for tasks other than reconstruction. The proposed discriminator uses these deep features to decide whether or not an image pair contains matching images. The discriminator is trained in a supervised fashion using labelled data. The labelled data consists of image pairs along with a *true* or *false* value indicating if the image pair consists of matching images. Discriminator has far fewer parameters than those in the overall autoencoder+discriminator network. This suggests that unlike end-to-end learning, it is possible to train the discriminator from scratch using a smaller set of labelled data.

Many deep learning architectures benefit from *transfer learning* and often use features computed by networks pre-trained on ImageNet dataset [19], [25], such as features from AlexNet [26] or GoogleNet [27]. RS images exhibit different visual characteristics than those seen in images present in ImageNet. We found that networks trained on ImageNet images often did not yield the desired performance on RS images. We propose to address this issue by pre-training the autoencoder on images from GTCrossView [28] dataset. We evaluate the proposed image matching framework on two standard benchmarks: LandUse [29] and SatScene [12].

N. Khurshid, Mohbat, and M. Taj are with the Department of Computer Science, Syed Babar Ali School of Science and Engineering, Lahore University of Management Sciences, Lahore 54792, Pakistan (emails: 15060051@lums.edu.pk, 16060073@lums.edu.pk, and murtaza.taj@lums.edu.pk). F. Z. Qureshi is with the Faculty of Science, Ontario Tech University, Oshawa ON L1G 0C5, Canada (email: faisal.qureshi@ontariotechu.ca).

†Authors contributed equally. Corresponding author: Numan Khurshid.

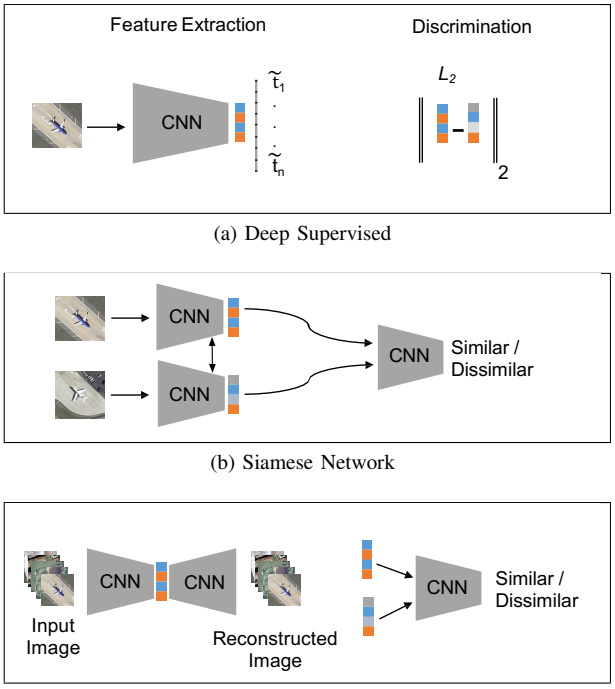
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Manuscript received March 27, 2019; revised October 24, 2019.

Following the principle of transfer learning, we fine-tune the pre-trained autoencoder using LandUse and SatScene datasets. We also train the discriminator on LandUse and SatScene dataset. Later in the paper we show the effect of fine-tuning the encoder sub-network when training the discriminator. Figure 1 provides an overview of our approach and how it differs from existing schemes.

Learning in Convolutional Neural Network (CNN) based architectures is often improved by using *residual* blocks, which address the problems related to vanishing and exploding gradients [30]. These blocks, however, suffer from diminishing feature reuse [31]. We propose a new residual block, called *residual-dyad*, that attempts to solve the problem of diminishing feature reuse. Experiments suggest that networks using residual-dyad outperform those that either use traditional residual blocks or do not use residual blocks at all.

We study the *performance* of our system using image retrieval (Figure 1(c)). For a given query image, image retrieval attempts to find the top k most similar images in a given collection [32], [33]. The relevance of the retrieved images is a proxy to the quality of the underlying image matching scheme. We use the University of California Merced LandUse/LandCover Dataset (LandUse) and High-Resolution Satellite Scene Dataset (SatScene) to compare the approach proposed in this paper with existing schemes. In the interest of completeness, we compare our method against both classical and more recent machine learning based approaches. The results suggest that our method outperforms existing techniques. We also include an ablative study that illustrates the role of the proposed residual connections in the autoencoder and discriminator stages of the image matching network proposed in this paper.



(a) Deep Supervised

(b) Siamese Network

(c) Proposed Framework for Unsupervised Image Matching

Fig. 1. Deep learning approaches for RS image matching. (a) Embeddings of deep supervised networks are extracted as image features and matched through Euclidean distance measure (L_2-Norm) [19]. (b) Siamese network computes the probability of matching image pair by simultaneously training the feature extraction and discrimination networks [28]. (c) A two-stage framework where the autoencoder is trained to learn unsupervised features while a discriminator is trained to discriminate between similar and dissimilar feature pair. Unlike, the previous approaches the autoencoder and discriminator networks are trained independently, requiring less labelled data.

A. Contributions

We propose a new architecture comprising autoencoder and discriminator for remote sensing image matching for the purposes of image retrieval. We evaluate this architecture on two standard benchmarks datasets. The proposed architecture outperforms existing techniques by a large margin (11.26% and 22.41%, respectively, for LandUse and SatScene datasets). Inspired by the success of the residual connections, we propose a residual-dyad unit, which minimizes the effects of both vanishing gradient and diminishing feature reuse. We use residual-dyad units in both autoencoder and discriminator sub-networks. Our ablative study supports the assertion that networks that incorporate residual-dyad outperform those that do not.

The remainder of this paper is organized as follows. Section II provides a short introduction of the relevant work on feature extraction and matching. We discuss the technical preliminaries and formulate the problem in Section III. Section IV describes the proposed method for unsupervised feature extraction and deep metric learning. Experimental setup and results are presented in Section V. We conclude the paper with a brief discussion and direction for future research.

II. Background

Features refer to image properties that can be derived from information present in an image [15]–[18]. Image features play a central role in image matching, and we begin our discussion with methods for computing image features.

A. Hand Crafted Features

Classical image matching techniques relied upon *hand crafted* features. These include low-level features that capture colors, edges and blobs information seen in the image [16]. Other techniques, such as those that compute spatial histograms or morphological features attempt to capture spatial structure present in the image [18]. Local features that capture surrounding structure at interest points have found wide-spread use for image matching [34]. Bag-of-words type features, which are frequently used in text document analysis, have also been adapted for the purposes of computing useful properties given in the image [18], [21]. These features are often combined with traditional machine learning approaches, such as K nearest neighbour, support vector machines, Bayes classifier, etc., for the purposes of image matching [35]. Hand-crafted features may not incorporate statistical information present in the dataset and often fail to achieve the desired performance on many computer vision tasks.

B. Feature Learning

More recent techniques use CNNs to "learn" image features [11], [19], [20], [22], [36], [37]. Broadly speaking, these approaches can be divided into two categories: 1) supervised and 2) unsupervised.

Supervised schemes require labeled data for training, and such datasets are often expensive and tedious to acquire [38]. In many cases, including remote sensing these pre-trained networks are fine-tuned and are used for feature extraction. Recent work by Famao *et al.* [11] proposed a weighted distance scheme to include class information while extracting features from a pre-trained AlexNet. Similarly, Napoletano [19] proposed to use ResNet [30] fine-tuned with satellite imagery to identify visual descriptors of the images. In cases where there is a scarcity of annotated data, researchers introduced unsupervised techniques for feature learning [39].

Unsupervised methods are able to learn image representations without access to labelled datasets, e.g. autoencoders [40]. Autoencoders attempt to construct low-dimensional encoding of an image with the aim to recover the original image from this encoding. These low-dimensional encodings, it turns out, can serve as image features for many computer vision tasks, such as image matching and classification. Chao *et a1*. [39], for example, also proposed autoencoder based unsupervised methods for deep feature extraction for hyper-spectral image matching. Deep convolutional autoencoders suffer from vanishing gradients, which complicates training [41], [42]. Lichao *et al.* [42] use skip connections [43], which introduce shorter paths with fewer non-linearities to the deeper layers of the network, to achieve faster convergence, avoid vanishing gradients, and capture fine-detailed image structures [42], [43].

C. Feature Matching

Given image features, the next step is to use these features to decide whether or not images match. Distance measures, such as Euclidean, Manhattan, Dominance, and Chi-square distance, or similarity measures, such as Cosine or Jaccard similarity, are widely used in the literature to compare image features [44]. These metrics rely mostly on the numerical values of the features, completely ignoring the hidden patterns of the features inherited from the images. Metric learning is an ideal alternate, capable of learning distance function for a specific task i.e. image retrieval.

1) Distance Metric Learning: Distance Metric Learning is often used for the task of measuring similarity. The goal is to learns a distance function over objects, which can help decide if two objects are similar [45]. Global supervised [46] approaches for distance metric learning attempt to satisfy all global constraints such as contextual information; whereas, local approaches [47] learn a metric that satisfies local constraints between image pairs. Chechik *et al.* [48] proposed an Online Algorithm for Scalable Image Similarity (OASIS), which learns a *linear* similarity measure. OASIS is not suitable for applications that benefit from non-linear similarity measures.

2) Deep Metric Learning: Deep learning provides an effective mechanism for learning highly non-linear similarity measures [49]. Within this context, contrastive and triplet losses have been employed for training a deep network that computes similarity measures [50]–[54]. The Siamese network proposed in [28], [55], for example, uses contrastive loss between image pairs; whereas, the triplet network [56] computes a loss for an image triplet (two similar and one dissimilar image). Such architectures have been used in cross-view matching applications. Vo *et al.* [28] uses deep metric learning to match street images to satellite images of the same region. Similarly, Wang *et al.* [57] uses an auto-encoder to match images captured by two different cameras. A convolutional discriminative network has also been used for face matching [58] and interest point matching [52].

Deep Metric Learning (DML) is sometimes also used for optimizing the sorting operation involved in ranking [59]. This is generally accomplished by converting the non-differentiable sorting step in to a differentiable operation by introducing customized loss surrogates [60]. The idea is to avoid zero or undefined derivatives of sorting and apply gradient-based optimization [61]. The whole phenomenon is called learning to rank paradigm.

D. Matching for RSI Retrieval

Many of the recent remote sensing image retrieval techniques rely on CNN based supervised features and linear distance measures for matching [11], [19], [22], [25]. Zhao *et al.* [20] used a combination of CNN and balanced linear discriminant analysis for feature extraction from remote sensing imagery. They then used Euclidean distance to compute the similarity between image features. Similarly, Gui-Song *et al.* [25] extracted features through fine-tuned GoogleNet trained with multi-patch pooling operation. Multiple metrics such as Euclidean, Cosine, Manhattan, and Chi-square distance were then used for matching. Instead of using a pre-trained classification network, a Fully Convolutional Network (FCN) was used in [22] to generate segmentation maps of the satellite images. The region convolution features from this pre-trained FCN was then used for region based retrieval using L_2 distance. Center loss-based multi-task learning has been proposed by Xiong *et al.* to match features acquired through a CNN network that uses attention mechanism [62]. All these approaches perform simultaneous feature learning and discrimination in an end-to-end supervised manner.

Unsupervised graph-theoretic approach has been employed by Chaudhuri *et al.* with graph-based similarity measure between the image features for RSIR [63]. Tang *et al.* [64], on the other hand, used a hybrid approach that computes bag-of-words on clustered autoencoder features. These features are then compared using linear similarity metrics including Euclidean, Cosine, Chi-Square and Bhattacharyya distance.

We also propose to learn features in an unsupervised fashion; however, similar to Han *et al.* [52] we decoupled feature learning and matching process. Instead of using hand-crafted features, we minimize reconstruction error and use activations from the last layer of the trained encoder network as our

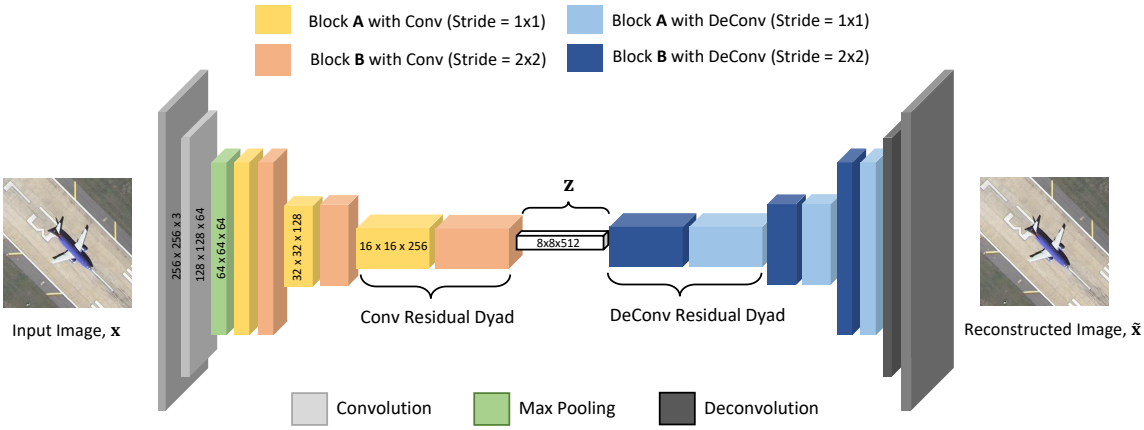


Fig. 2. Unsupervised Autoencoder Features: Image input from left (to encoder sub-network) and outputs to the right of (decoder) network. $\mathbf{z}$ is taken as the feature vector of the given image (best viewed in colour).

feature vector. We replace linear distance measures with a discriminator network to match image features.

III. Problem Formulation

The problem of RSIM can be split into two steps i) feature extraction, and ii) feature matching. Let $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^N\}$ is a set of images, where $\mathbf{x} \in \mathbb{R}^D$. In case of supervised learning, the problem is to estimate a function f_e that transforms the image into a M-dimensional latent space (where $M < D$),

$$\mathbf{z} = f_e(\mathbf{x}, \Theta_e, \beta_e). \tag{1}$$

Here $\mathbf{z} \in \mathbb{R}^M$ is the extracted feature vector, and Θ and β are the weights and biases of the estimation function f_e, respectively. The class labels $\tilde{t}$ from the feature vector $\mathbf{z}$ are extracted using a softmax activation $\sigma(.)$ defined as:

$$\tilde{t} = \sigma(\mathbf{z}). \tag{2}$$

This function f_e is learned by minimizing the error between actual label t and predicted class label $\tilde{t}$ using:

$$J_c = -\sum t \log \tilde{t}. \tag{3}$$

However, due to scarcity of labelled data our first problem is to extract features without using any annotations. We formulated this problem as *unsupervised feature learning* and adapted an autoencoder to resolve it. For this purpose, the feature vector $\mathbf{z}$ computed from the latent space in (1) must be decoded back to the image space by estimating a decoding function f_d,

$$\tilde{\mathbf{x}} = f_d(\mathbf{z}, \Theta_d, \beta_d). \tag{4}$$

The feature learning is then performed by minimizing the reconstruction error J_r between input image $\mathbf{x}$ and the decoded image $\tilde{\mathbf{x}}$ which is defined as:

$$J_r = \|\mathbf{x} - \tilde{\mathbf{x}}\|_2. \tag{5}$$

For our second problem—feature matching—Euclidean distance between image features is often used [19],

$$J_{ij} = \|\mathbf{z}^i - \mathbf{z}^j\|_2. \tag{6}$$

However, we observed that unlike supervised learning, these feature vectors $\mathbf{z}$ obtained via unsupervised learning are not discriminative in the linear Euclidean space (see Table II). One of the reasons for this is lack of any discrimination constraint in learning process of unsupervised features. We attempt to learn a non-linear metric space that projects the pairs of features $\{\mathbf{z}^i, \mathbf{z}^j\}$ into another latent space to discriminate between similar and dissimilar pairs. To this end the problem is addressed by estimating another transformation function f_m defined as:

$$\mathbf{v}_{ij} = f_m(\mathbf{z}^i, \mathbf{z}^j, \Theta_m, \beta_m). \tag{7}$$

The match/mismatch label from the feature vector $\mathbf{v}_{ij}$ of the discriminator is then extracted using softmax activation $\sigma(.)$ applied to the resultant feature vector $\mathbf{v}_{ij}$,

$$y_{ij} = \sigma(\mathbf{v}_{ij}). \tag{8}$$

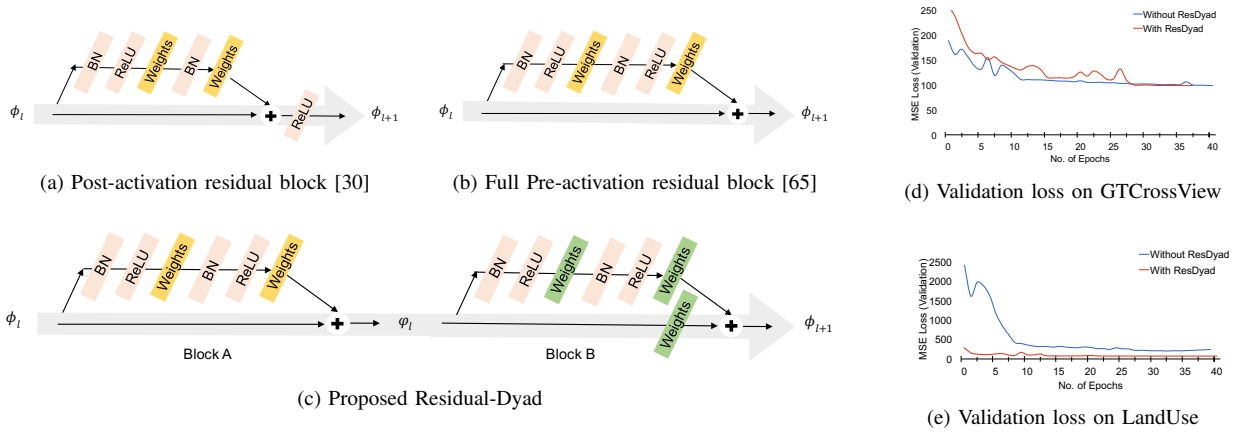
This function f_m is learned by minimizing the binary cross entropy loss between actual y_{ij} and predicted label $\tilde{y}_{ij}$ using:

$$J_m = -y_{ij} \log \tilde{y}_{ij}, \tag{9}$$

where y is ground truth. It is 1 if the two images are similar and 0 otherwise. $\tilde{y}$ is the estimated probability that the two images in a pair are the same.

IV. Proposed Method

We propose a modular approach for RSIM consisting of an autoencoder and a discriminator module. Unlike [23], the modular approach allows us to train the autoencoder independently in an unsupervised manner while the discriminator is trained on small set of labelled examples. The discriminator function f_m is determined by a DML network instead of standard linear distance metrics. To estimate the encoding and decoding functions f_e and f_d of an autoencoder, respectively, and the matching function f_m, we introduce a residual-dyad block in deep residual network architecture. Our deep residual architecture is explained next.



(a) Post-activation residual block [30]

(b) Full Pre-activation residual block [65]

(c) Proposed Residual-Dyad

(d) Validation loss on GTCrossView

(e) Validation loss on LandUse

Fig. 3. Residual blocks and the proposed residual-dyad block. (a) In standard residual block [30] the activation is performed after addition, while in (b) full pre-activation block [65] the activation is performed before addition. (c) Proposed ResDyad block combines a full pre-activation block with a convolutional shortcut block. This dyad block has been used in both the proposed autoencoder and DML networks. (d) Validation loss analyzed during training ResAE (without residual-dyad) and ResDyadAE networks (with residual-dyad) on GTCrossView. (e) Analyzing validation loss during fine-tuning the trained autoencoder networks on LandUse dataset.

A. Residual Autoencoder

Autoncoders have been previously used for unsupervised feature extraction in many applications which include remote sensing [42], [64]. However, these applications of autoencoders do not perform well due to increased complexity of RS imagery (complex spatial arrangement, variety of surface objects, limited resolution, and increased noise). We propose following three modifications to address these issues 1) we introduce a new type of residual block called residual-dyad 2) instead of 1D features in latent space, we introduce 2D latent features 3) and we also replace pooling with convolutional blocks.

Our residual autoencoder contains two symmetric subnetworks: an encoder f_e and a decoder f_d, each of which contains residual-dyad blocks. Instead of generating 1D features [39], our encoder generates n 2D features of 8×8 spatial resolution. The key advantage of these 2D features is to retain the structural information of the image that is usually lost in single dimensional feature vector [42].

In existing literature [64], maxpooling layers have been used to reduce the spatial resolution of image during encoding. This results in losing crucial information which could not be recovered efficiently through ordinary unpooling operation during decoding [42]. Therefore, our proposed autoencoder does not employ any pooling layer (in the deeper part of the network) instead we exploit convolutional layers to reduce spatial resolution. A filter of size 3×3 in combination with stride 2×2 is used in convolution layers of encoder. A similar operation with transpose convolution has been applied during upsampling in decoder. In addition, we also replace the standard full pre-activation residual block explained in [65] with the proposed residual-dyad block, which we discuss next.

B. Residual-Dyad

The key idea in residual networks is the introduction of identity skip connections to address the problem of vanishing gradients, improving the accuracy of deeper networks [65]. However, each fraction of a percent of improved accuracy requires nearly doubling the number of layers resulting in longer training time. These skip connections introduce another problem, called diminishing feature reuse, which further increases the training time. In this problem the network sometimes avoids learning intermediate weights because of the input being fed to each of the middle layers of the network through skip connections. This is sometimes referred to as vanishing gradient during the forward pass [66]. Such networks consist of several residual blocks and hence the features computed by early layers of the network are washed out by the time they reach the deeper layers due to multiple weight multiplications in between. Consequently, either a few blocks learn useful representations or blocks share very little information with small contributions to the final goal [67]. Zagoruyko *et al.* proposed Wide-ResNet [31] to address these problems. They emphasized on increasing the width of the network instead of its depth by introducing multiple convolutional shortcut skip connections in the residual block. Our approach, however, does not restrict the choice of using identity skip connection.

For both ResNet [65] and Wide-ResNet [31], the cascaded residual layers can result in chain of skip connections resulting in a cascade of identity mappings. This introduces diminishing feature reuse. Thus a solution is needed that can benefit from skip connections (to solve vanishing gradient problem) without introducing cascade of identity mappings (to avoid diminishing feature reuse). To solve this problem we propose to use two types of skip connection in cascade (identity skip connection followed by a convolutional-shortcut skip connection) and propose a new residual unit called residual-dyad (see Figure 3). Figure 4 suggests that residual-dyad (in ResDyadAE) has significantly reduced the number of (diminished) features that are unable to learn information necessary for image reconstruction. This can be seen in Figure 4, where dark patches represent such diminished features. This figure also shows that ResDyadAE achieved sharper image reconstructions.

Residual-dyad is the combination of two full pre-activation

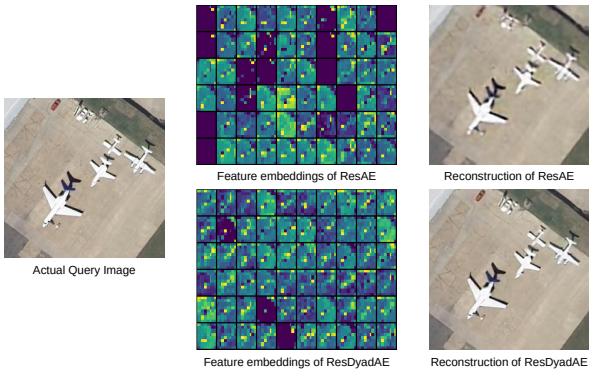


Fig. 4. Visualization of the activations and reconstructed images for ResAE (*top row*) and ResDyadAE (*bottom row*) for the given input image. It is obvious from these images that ResAE generated a blurred reconstructed image as well its features suffer from the problem of diminishing feature reuse (black feature components though its features).

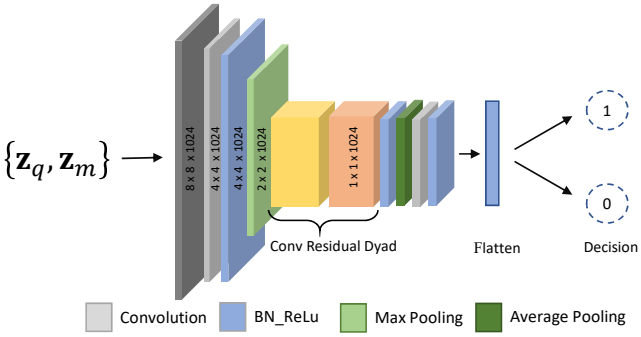


Fig. 5. Network architecture of the proposed ResDyadDML that takes features from ResDyadAE for an image pair and predicts the matching score. Residual-dyad block has been integrated to boost the performance of the network (best viewed in colour).

residual blocks: 1) the first block is the same as ResNets [30]; and 2) the second block includes a convolution layer through its residual skip connection. This combination forces the gradients to update the weights even if a residual path is adopted for learning thus avoiding diminishing feature reuse. While the size of the filters for the convolution layers of both these blocks is 3×3, the stride for the first residual block is 1×1, and the stride for second block is 2×2. This configuration decreases the spatial resolution in deeper layers. Mathematically, a conventional full pre-activation residual block [30] is defined as:

$$\varphi_l = \phi_l + \mathcal{F}(\phi_l; \Theta_{\phi_l}), \tag{10}$$

where $\mathcal{F}(.)$ consists of convolutional, batch normalization and activation layers. The proposed residual-dyad also includes *conv-shortcut* block [65]. This integrates another convolution layer $\mathcal{H}(.)$ in the shortcut path of residual block and can be expressed mathematically as

$$\phi_{l+1} = \mathcal{H}(\varphi_l; \Theta^h_{\varphi_l}) + \mathcal{F}(\varphi_l; \Theta^f_{\varphi_l}). \tag{11}$$

The overall residual-dyad can be mathematically expressed as:

$$\phi_{l+1} = \mathcal{H}(\phi_l; \Theta^h_{\varphi_l}) + \mathcal{F}(\phi_l; \Theta^f_{\varphi_l}) + \mathcal{H}(\mathcal{F}(\phi_l; \Theta_{\phi_l}); \Theta^h_{\varphi_l}) + \mathcal{F}(\mathcal{F}(\phi_l; \Theta_{\phi_l}); \Theta^f_{\varphi_l}). \tag{12}$$

Here, (10) and (11) are block A and block B, respectively and (12) shows the overall effect of dyad (see Figure 3(c)). ϕ_l indicate the feature maps fed into lth residual-dyad block, Θ^h are the parameters of convolution layer $\mathcal{H}$ while Θ^f are the parameters of residual function $\mathcal{F}$. These equations illustrate that the network is capable of learning new features between any starting block l and terminating block L.

1) Residual-Dyad Autoencoder: We propose a residual-dyad based autoencoder network (ResDyadAE) for unsupervised feature learning that is robust to both vanishing gradients and diminishing feature reuse problems. The autoencoder network is illustrated in Figure 2. We refer to this network as ResDyadAE and the network with traditional residual connection as ResAE in the remainder of this paper. ResDyadAE outperforms the ResAE. Image reconstructed using ResDyadAE exhibit fine-grained structures that are not visible in images reconstructed using ResAE (Figure 4). Figure 3(d) plots validation loss for ResDyadAE and ResAE for GTCrossView dataset. Notice that ResDyadAE achieves lower validation error faster. This trend continues when these networks are fine-tuned on LandUse dataset (Figure 3(e)).

C. Residual Deep Metric Learning

Tang *et al.* [64] use autoencoders for extracting features from image patches. These features are subsequently matched using eight different distance metrics to evaluate the overall feature extraction and matching performance. The metrics used are linear, and they ignore the non-linear discriminative patterns of the feature pair resulting in unsatisfactory retrieval score. We propose a new metric learning approach that relies upon deep residual network (ResDML). This network is trained in a supervised setting by minimizing the binary cross-entropy loss. Essentially, the network is trained to predict whether or not both input images belong to the same class. We also incorporated the proposed residual-dyad block in this network, and we refer to this variant as ResDyadDML (Figure 5).

The encoder sub-network computes features for both images in the pair. Recall that the encoder sub-network is trained in an unsupervised manner. this suggests that our feature extraction approach eschews class labels and can be easily extended to large unlabelled datasets. A downside of this approach is that it constructs higher dimensional features. We found that this is necessary to preserve the image details needed for image reconstruction during training.

We also studied the effect of feature size on the proposed approach. We varied autoencoder's depth and filter sizes (in the middle layers) to construct $1 \times 1 \times 1024$ dimensional features, $\mathbf{z}$. Figure 6 (b) shows reconstruction results using these features, and it is evident that these features result in poor reconstructions. Similarly, we constructed $8 \times 8 \times 20$ dimensional features. Figure 6 (c) shows reconstruction results for these features. Note that while reconstruction is better

when compared to that achieved by $1 \times 1 \times 1024$ dimensional features, it still exhibits blocky artifacts. In our experiments $8 \times 8 \times 512$ dimensional features achieved the best results. We have only experimented with MSE loss to train the autoencoder network. In the future, we plan to study feature size reduction by exploring techniques, such as model pruning and sparsity inducing loss functions [68].

These features are concatenated and the resulting 8×8×1024 tensor is sent to the ResDML and ResDyadML networks. These features are subsequently passed through a convolution layer and a batch normalization layer. ReLU activation is applied to it before feeding it to a residual-dyad block (simple residual block in case ResDML). The ReLU activations are passed through another convolutional layer followed by a fully connected layer with softmax activation as shown in Figure 5. The discriminator network has far fewer parameters than the entire encoder-discriminator network, and it is possible to learn the weights of the discriminator network using much less labelled data. Replacing Euclidean distance with DML network for feature matching improves matching score by 14% as seen in Table I.

V. Evaluation Setup

We now discuss the benchmark datasets used to evaluate the proposed networks.

A. Datasets

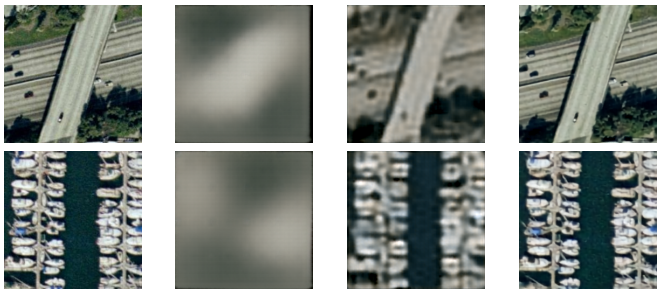
We use University of California Merced Land Use/Land Cover dataset (LandUse) [29] and High-resolution Satellite Scene dataset (SatScene) [12] to evaluate the proposed approach. In addition, we also use GTCrossView [28] dataset for feature learning.

1) The LandUse dataset contains 2100, aerial orthophotos, covering a total area of 42 thousand square kilometers. Each image is 256×256, and belongs to one of 21 classes (see Figure 8).
2) The SatScene dataset contains 19 diverse classes with aerial orthoimagery at various zoom levels with about 100 images per class. Each image is 600×600.
3) The GTCrossView dataset contains 1 million pairs (street-view/satellite-view). It does not contain any class information.

The purpose of using two datasets with variety in terms of pixel resolution, size, zoom level, and classes, is to develop a consolidated solution for challenging scenarios in unsupervised remote sensing matching.

B. Performance Metrics

We use Mean Average Precision (mAP) to capture image matching performance. We also measure the ranking efficiency for image retrieval using the proposed image matching technique. For this purpose we use two commonly used indicators: 1) Average Normalized Modified Retrieval Rank (ANMRR) [19] and 2) class-wise mAP [69].



(a) (b) (c) (d)

Fig. 6. (a) Input image. Visualization of reconstructed images from (b) $1 \times 1 \times 1024$ dimensional encoded features (c) $8 \times 8 \times 20$ dimensional encoded features (d) $8 \times 8 \times 512$ dimensional encoded features (ResDyadAE).

TABLE I. A comparison of networks with and without residual-dyad. Both autoencoder networks have been initially trained on GTCrossView dataset and fine tuned with LandUse dataset.

Unsupervised	**ResDML**		**ResDyadDML**	
Autoencoder	**ANMRR**	**mAP**	**ANMRR**	**mAP**
ResAE - LandUse	0.60	25.79	0.64	22.19
ResAE - GTCrossView	0.61	25.07	0.57	31.67
ResAE - Fine Tuned	0.54	33.80	0.57	32.32
ResDyadAE - LandUse	0.61	27.52	0.55	40.97
ResDyadAE - GTCrossView	0.40	43.97	0.19	72.19
ResDyadAE - Fine Tuned	**0.22**	**67.28**	**0.09**	**81.20**

1) Mean Average Precision (mAP): Precision can be defined as the fraction of retrieved images relevant to query image. It is usually evaluated in the cut-off rank, considering top k results yielded by RSIR system. This measure is termed as P@k. In this research we are calculating Mean Average Precision (mAP) and per class mAP values for the comparison with state-of-the-art methods. Mathematically, mAP can be computed as

$$\text{mAP} = \frac{1}{Q} \sum_{q=1}^{Q} AP(q), \tag{13}$$

where average precision (AP) is:

$$\text{AP} = \frac{\sum_{k=1}^{n} (\text{Precision}(k) \times z(k))}{\text{Number of relevant images}}, \tag{14}$$

where n is the number of retrieved images, k is their rank, and $z(k) \in \{0, 1\}$, equaling 1 if the feature at rank k belong to a relevant image, while zero otherwise.

2) ANMRR: ANMRR considers the number of similar images that are retrieved and quantifies them as per their rank. This also address the queries having varying relevant image sets in image retrieval problem. ANMRR removes the bias that arises during retrieving the set of relevant images give a query when the number of relevant images present in the given dataset for different query images vary. Mathematically Rank(k) is defined as

$$\text{Rank}(k) = \begin{cases} \text{Rank}(k), & \text{if Rank}(k) \leq K(q) \\ 1.25K(q), & \text{if Rank}(k) > K(q) \end{cases} \tag{15}$$

where Rank(k) is the kth position at which a similar item is retrieved. *K(q)* is the constant penalty term adapted to penalize

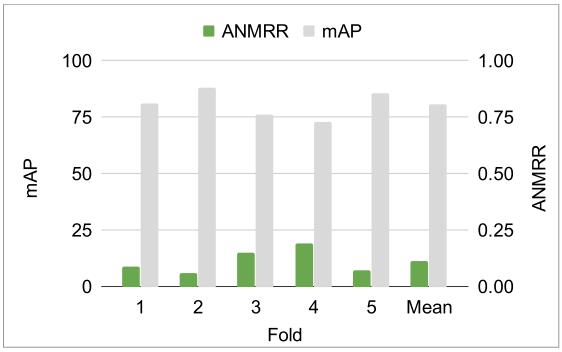


Fig. 7. 5-Fold cross validation of ResDyadDML network on LandUse dataset showing consistent ANMRR and mAP values for each fold, validating the performance and reliability of the discrimination network.

TABLE II. A comparison of mAP values with Euclidean distance and a pre-trained DML network applied to hand-crafted, supervised, and unsupervised (autoencoder) features of LandUse dataset.

Features	Feature Type	Euclidean mAP	DML (Ours) mAP
HoG [19]	Hand-crafted	17.85	21.88
LBP-RGB [19]		17.96	28.22
VGG-16 [70]	Supervised	28.26	58.69
GoogleNet [71]		55.86	62.23
SatResNet-50 [19]		**69.94**	**89.69**
Siamese [55]		10.39	65.87
ResDyadAE [Ours]	Unsupervised	4.64	81.20

the misclassified ranking results. It is commonly chosen to be $2G(q)$ where $G(q)$ is the set of relevant images. For all relevant images Mean Rank $\text{Rank}_{mean}(q)$ for the given query image q is computed using:

$$\text{Rank}_{mean}(q) = \frac{1}{G(q)} \sum_{k=1}^{G(q)} \text{Rank}(k). \tag{16}$$

$\text{Rank}_{mean}(q)$ is normalized by the amount of the difference between maximum penalty and minimum penalty value. This Normalized modified retrieval rank (NMRR) for query q is described as:

$$\text{NMRR} = \frac{\text{Rank}_{mean}(q) - 0.5[1 + G(q)]}{1.25K(q) - 0.5[1 + G(q)]}. \tag{17}$$

To generalize the results this operation is iterated over a number of different query images Q. For which average NMRR over all the queries can then be calculated as:

$$\text{ANMRR} = \frac{1}{Q} \sum_{q=1}^{Q} \text{NMRR}, \tag{18}$$

where Q indicates the number of queries q performed.

C. Experimental Setup

We trained two different autoencoders, a residual-dyad autoencoder (ResDyadAE) and a regular residual autoencoder (ResAE). For each of these autoencoders we trained two different discriminator networks, with residual-dyad blocks (ResDyadDML) and with regular residual connections (ResDML).

We carried out the following set of experiments to estimate the performance gain due to the use of the proposed residual-dyad blocks. We intially trained two autoencoders—one using residual blocks only and the second using the proposed residual-dyad blocks—on GTCrossView dataset [28] containing $200K$ pairs of satellite and street view images in an unsupervised manner. We used GTCrossView dataset instead of LandUse and SatScene datasets since the last two lacked the number of images needed to train a deep autoencoder network. The discriminator network, which uses deep metric learning, is trained only on LandUse and SatScene datasets using the 80/20 training/testing split. For the sake of completeness, we also trained the networks on the following training/testing splits: 70/30, 60/40, 50/50, 40/60, and 30/70. We discuss the results of these experiments in the next section.

We used the same hyper-parameters' values in all our experiment. Each network is optimized using Stochastic Gradient Decent (SGD) with initial learning rate of 0.001 and a rate decay of 0.2. Instead of using fixed number of epochs, we used early stopping criteria which terminates the training process in case there is no improvements for 10 consecutive epochs. All experiments are conducted on two systems having the following specifications: i) Intel Core 4.20 GHz i7-6700K processor with 32 GB RAM and NVIDIA GeForce GTX 1080Ti GPU, and ii) a Intel Xeon 48 core 2.1 GHz E7-4830 processor with 256 GB RAM and a Tesla K40 GPU.

VI. Results and Analysis

A. Comparison b/w Residual Unit and Residual-Dyad Unit

To measure the performance improvements due to proposed dyad block, we compare the dyad based networks (ResDyadAE and ResDyadDML) with the networks using regular residual blocks (ResAE and ResDML). Table I compares the performance of our proposed network architectures for image retrieval task on Landuse dataset. The residual-dyad based network in both cases of autoencoder and deep metric learning outperforms its other variants. When we used ResDML with both ResAE and ResDyadAE, the ResDyadAE achieves more than 18% improvement in mAP (from 25.07 to 43.97) as compared to ResAE. This improvement gap increased to approximately 35% (from 33.80 to 67.28) when these autoencoders were fine-tuned on LandUse dataset. The improvement of over 2x (from 0.54 to 0.22) in ANMRR is also observed after fine-tuning.

The last two columns of Table I show the improvements achieved after introducing dyad blocks in the discriminator network. The ResDyadDML improved the mAP by at least 2% with ResAE (from 25.07 to 31.67) and ResDyadAE fine-tuned features (from 67.28 to 81.20). Thus the overall improvement achieved using dyad based architecture (ResDyadAE with ResDyadDML) as compared to the one that does not use residual-dyad (ResAE with ResDML) is 47.44% (form 33.80 to 81.20) and 0.36% (from 0.54 to 0.09) respectively, in mAP and ANMRR values. This significant improvement in both mAP and ANMRR indicates the effectiveness of residual-dyad as compared to regular residual block.

Fig. 8. Top 10 retrieved images for the corresponding query images of LandUse dataset. The retrieval results shown for queries taken from the classes of Airplane, Dense Residential, Intersection, Tennis Court, Sparse Residential, and Storage Tanks. It could be clearly observed that images with very similar features have been misclassified as a match to original query image

.

TABLE III. Comparative evaluation of our proposed approach with other recent techniques on LandUse dataset [19].

Method	**Feature Type**	**ANMRR↓**	**mAP↑**	**P@5**	**P@10**	**P@50**	**P@100**	**P@1000**
HoG [19]	Hand-crafted	0.751	17.85	48.67	41.88	25.37	19.2	6.18
LBP-RGB [19]		0.751	17.96	58.73	49.83	28.12	19.62	6.07
Dense SIFT (VLAD) [20]		0.649	28.01	74.93	65.25	38.20	28.10	7.18
Dense SIFT (FV) [19]		0.639	29.18	75.34	66.28	39.09	28.54	7.88
GoogleNet [71]	Supervised	0.360	55.86	85.36	80.96	64.71	52.36	9.68
NetVLAD [72]		0.406	51.44	83.00	78.59	61.63	49.04	9.29
MLIR CNN-Fc7 [22]		0.322	62.73	80.76	71.00	30.80	17.77	-
SatResNet-50 [19]		0.239	69.94	92.06	89.02	77.23	64.42	9.86
ResDyadAE-ResDyadDML [Ours]	Unsupervised	**0.090**	**81.20**	**99.4**	**99.2**	**99.0**	**87.4**	**9.90**

To further study the effect of the proposed residual-dyad unit, we performed a 5-fold cross validation (see Figure 7). The obtained mean and standard deviation of ANMRR and mAP ranges from $0.1116|\pm0.05|$ and $80.786|\pm5.65|$, respectively. We also studied the effect of different train-test split illustrated in Figure 10. The ANMRR and mAP scores for 50% or more training data are better. As expected reducing the training data has an adverse effect on the overall performance. These results confirm the performance gain achieved by the proposed residual-dyad unit.

B. Comparison b/w Euclidean vs. Deep Metric Learning

Euclidean distance is one of the most commonly used measure in image matching [19]. In Table II we compare the mAP scores between Euclidean distance and our proposed DML based matching. The scores were computed for features extracted using hand-crafted (Histogram of Oriented Graphs (HoG) [19], Linear Binary Patterning for RGB Images (LBP-RGB) [19]), deep supervised (VGG16 [70], GoogleNet [71], Satellite ResNet (SatResNet) [19], Siamese [55]) and our deep unsupervised technique. For a fair comparison, in Siamese Network the encoder and discriminator follow the same architecture as used in ResDyadAE and ResDyadDML, respectively. Furthermore, we initialized the Siamese Network

TABLE IV. Comparative evaluation of our proposed approach with the state-of-the-art hand-crafted and supervised techniques on SatScene dataset.

Method	Feature Type	ANMRR↓	mAP↑	P@5	P@10	P@50	P@100	P@1000
HoG [19]	Hand-crafted	0.724	19.97	40.24	35.31	21.73	15.82	5.20
LBP-RGB [19]		0.664	24.95	50.33	43.98	26.33	19.40	5.20
Dense SIFT (VLAD) [20]		0.649	28.01	74.93	65.25	38.20	28.10	7.18
Dense SIFT (FV) [19]		0.552	35.89	71.30	62.78	36.19	25.03	5.20
GoogleNet [71]	Supervised	0.324	60.36	85.73	82.28	68.32	55.75	9.75
NetVLAD [72]		0.371	56.37	82.54	78.41	64.40	52.19	9.48
SatResNet-50 [19]		0.207	74.19	92.11	90.55	80.91	**68.02**	9.87
ResDyadAE-ResDyadDML [Ours]	Unsupervised	**0.06**	**96.6**	**100**	**99.1**	**94.3**	52.00	**9.92**

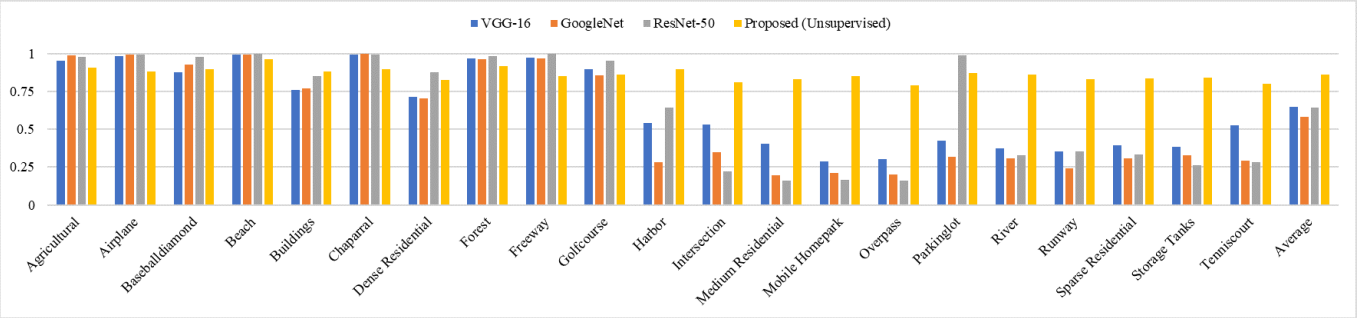


Fig. 9. Comparative results of class-wise mAP among supervised *(VGG-16, GoogleNet, SatResNet-50)*, and our proposed *unsupervised* approach. Our approach performs uniformly better and on average it surpasses the efficiency of supervised techniques on LandUse dataset.

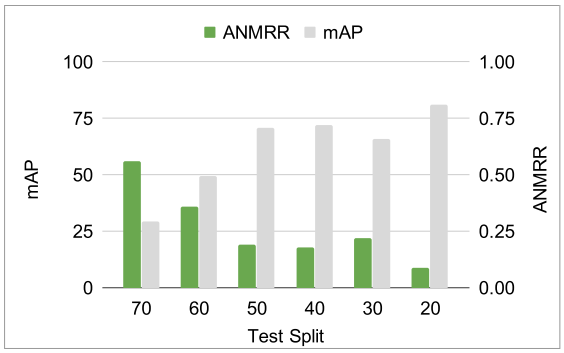


Fig. 10. ResDyadDML trained on ResDyadAE features with different train-test split ratios of LandUse dataset.

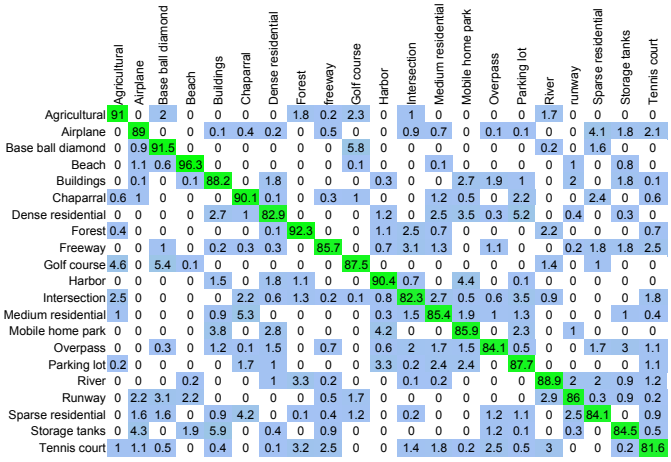
with the pre-trained weights of the ResDyadAE and then fine-tuned it on the LandUse dataset.

It can be observed that the proposed unsupervised features are not discriminative in the Euclidean space as they are trained to minimize reconstruction error only. However, when we use ResDyadDML instead of Euclidean distance, the mAP scores improve for all types of features. Overall, ResDyadDML has introduced about 5% to 30% improvement of mAP score for all the feature variants. The performance of supervised features particularly from SatResNet-50 has improved by 20% (from 69.94 to 89.69) as compare to its published score [19]. Similarly, for ResDyadAE features, it increases from 4.64 to 81.20 which is highest as compared to state-of-the-art approaches and 2nd highest as compared to our improved version of SatResNet-50 (i.e. SatResNet-50 with proposed DML). This improvement in mAP score demonstrates the efficacy of ResDyadDML for image matching.

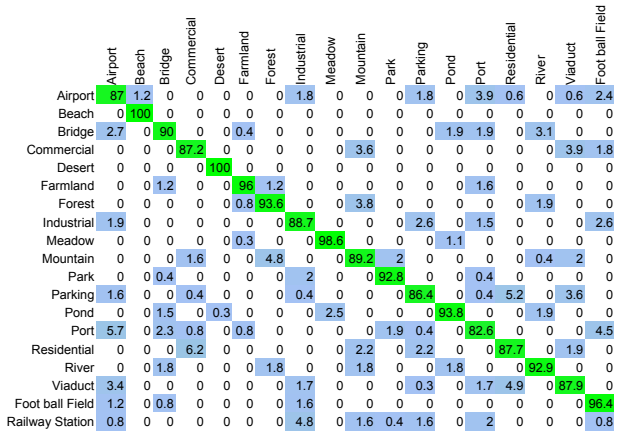
C. Comparison with State-of-the-art

We compared our proposed architecture consisting of ResDyadAE with ResDyadDML with 9 different techniques available in the literature: 1) Histogram of Oriented Graphs (HoG) [19]; 2) Linear Binary Patterning for RGB Images (LBP-RGB) [19]; 3) Dense SIFT (VLAD) [20] 4) Dense SIFT (FV); 5) GoogleNet [71]; 6) NetVLAD [72]; 7) Multi-label Image Retrieval (MLIR) [22]; 8) Satellite ResNet with 50 layers (SatResNet-50) [19]; and 9) Siamese Network. Out of these techniques HoG, LBP-RGB, and Dense SIFT use hand-crafted features; and GoogleNet, NetVLAD, MLIR, and SatResNet-50 use deep supervised features. This comparison is performed on both LandUse and SatScene benchmark datasets and is shown in Tables III and IV, respectively.

It can be observed from the Table III and IV that the ResDyadAE with ResDyadDML has outperformed the hand-crafted as well as supervised schemes. The hand-crafted features obtained least retrieval performance for both the benchmark datasets. In general it is expected that supervised approaches will perform better as compared to unsupervised techniques. However, it is interesting to observe that the our proposed unsupervised feature learning produced superior results as compared to supervised approaches such as GoogleNet, NetVLAD and SatResNet-50. As compared to the best performing supervised scheme (SatResNet-50), our



(a) LandUse Dataset with 21 classes



(b) SatScene Dataset with 19 classes

Fig. 11. Confusion metrics of (a) LandUse and (b) SatScene dataset averaged over 20 queries per class. This figure shows that the ambiguity between challenging classes has reduced to less than 7% percent. For example on LandUse only 5% golf course images are mismatched with those from agricultural class. Similarly, mismatch between river and forest is also reduced to just 3.35%.

proposed approach has approximately 11% and 22% higher mAP score on LandUse and SatScene datasets, respectively. Similarly, an improvement of over 14.5% can also be observed in ANMRR for both the benchmark datasets as compared to state-of-the-art.

Some of the detailed observations regarding remote sensing image retrieval could be highlighted by P@k measure which are enlisted for the values of $5, 10, 50, 100, 1000$ in Tables III and IV. For the proposed approach, top 5 retrieved images always belong to the class of the query image while top 10 images have also been recognized with 99.2% precision in LandUse and 100% precision in SatScene dataset. Even with top 50 images, our proposed technique superseded all other approaches. In case of top 100 images, SatResNet-50 obtained 16% better results on SatScene dataset whereas the proposed approach outperformed the state-of-the-art by 23% on LandUse dataset and approximately 14% on SatScene dataset.

D. Class-wise Analysis

To have a deeper look at the results, we compared the class-wise mAP of our proposed approach with 3 state-of-the-art supervised learning approaches: 1) VGG-16 [70]; 2) GoogleNet [71]; and 3) SatResNet-50 [19]. The results on 21 different classes as well as their average score are depicted in Figure 9. The average mAP score clearly shows that the proposed approach has outperformed other techniques in literature. Our technique performed uniformly well for all the classes of the LandUse dataset whereas the other approaches only performed well on 8 classes including agriculture, airplane, baseball diamond, beach, chaparral, forest, freeway and golf course. Many of the remaining 13 classes share visual features such as vegetation in agricultural and golf course classes, similarly dense trees in forest and river classes which makes it difficult to differentiate between them in Euclidean space. Furthermore, classes such as intersection, dense residential and sparse residential have high intra-class variations as structural content can appear at different locations and with different orientations in the given image.

The challenges due to intra-class variation, and orientation and spatial arrangements were addressed through deep metric learning employed in the proposed approach. Confusion matrix in Figure 11 shows that the ambiguity between challenging classes has reduced to less than 7 percentage. For example on LandUse only 5% golf course images are mismatched with those from agricultural class. Similarly, mismatch between river and forest is also reduced to just 3.3%. Likewise, on SatScene dataset the highest ambiguity of 6.2 is between images from residential and commercial classes, whereas the ambiguity between all the other classes is less than 5%.

E. Qualitative Evaluation

We analyse the reconstruction results of our proposed autoencoder (ResDyadAE) on both the test sets and randomly chosen images from other domains. The results on test set are shown in Figure 4 (bottom row), in which the triplets contain a query image, its deep features, and a decoded image. It can be seen that the autoencoder has successfully regenerated the query image without any blurring or loss of structural information.

The results on cross-disciplinary images are shown in Figure 12. It can be seen that our network generalizes to efficiently encode images from multiple domains, including but not limited to street view, satellite view, medical imagery, and synthetically generated images, into low dimensional space (see Figure 12).

Similarly, Figure 8 shows the top 10 retrieved images for a query image (left most) on LandUse dataset. It can be noticed that despite of having common visual features of trees in classes such as sparse residential, dense residential, and river and variation in zoom levels, our approach has retrieved images similar to the query image.

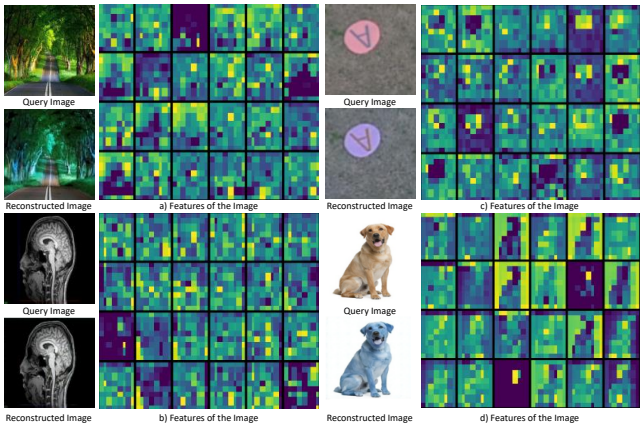


Fig. 12. Visualization of deep features of multi-disciplinary images showing that although our autoencoder is trained only on street view and satellite images, however it works equally well on images from other domains such as (a) street view image (b) top view/aerial image, (c) medical image and (d) synthetic image. Out of 512 features, we have visualized only 24 features each of size 8×8.

VII. Conclusion

We propose a remote sensing image matching system. The proposed system consists of two parts: a) an autoencoder that is pre-trained on GTCrossView data in an unsupervised manner, and b) a deep metric learning network trained on image pairs from LandUse and SatScene datasets. While the autoencoder is trained to minimize the reconstruction loss, the latent representation that it constructs succinctly captures salient characteristics of an image. We can, therefore, use its encoder sub-network to construct powerful image features. Given features for two images, the discriminator network decides if the two images belong to the same class. Unlike existing learning-based approaches that require labelled data for training networks responsible for image feature construction, our approached follows an unsupervised approach to train a network for image feature construction. Specifically, we use the encoder sub-network of an autoencoder that does not need labelled data for training. This suggests that the proposed approach requires much less labelled data for training purposes, since here we only need labelled examples to train the discriminator network.

Others have proposed to used *residual* connections to improve network performance. We propose a new network unit, called *residual-dyad*, which consists of two residual units stacked back to back. We find that networks using residual-dyad outperform networks that either do not use residual units or use traditional residual units. We provide an ablative study that confirms the benefits of using a residual-dyad over traditional dyad in our system. The proposed autoencoder contains 6 residual-dyad units: 3 in the encoder stage and 3 more in the decoder stage. The proposed discriminator also uses a residual-dyad unit. We also show the benefits of fine-tuning the encoder sub-network when training the discriminator network.

We have compared our approach on LandUse and SatScene benchmarks against both 1) traditional approaches that use hand-crafted features and 2) more recent learning-based approaches. Our method outperforms other schemes in terms of mAP and ANMRR metrics. Our method achieves an overall improvement in performance of 11.26% and 22.41% in mAP respectively, for LandUse and SatScene benchmark datasets over state-of-the-art. Similarly, we achieved over 14.5% improvement in ANMRR for both the benchmark datasets.

In the future, we intend to study the effects of residual-dyad unit in other deep learning settings. We also intend to explore the use of this RS image matching framework in a full-fledged RS image retrieval and management system, which includes ideas, such as query expansion and relevance feedback.

References

[1] R. P. Gupta, *Remote Sensing Geology*. Springer, 2017.
[2] R. C. Frohn and R. D. Lopez, *Remote Sensing for Landscape Ecology: New Metric Indicators: Monitoring, Modeling, and Assessment of Ecosystems*. CRC Press, 2017.
[3] F. Mao, Q. Min, G. Liu, C. Liu, S. Feng, S. Jin, J. Hu, W. Gong, and C. Li, "Assimilating moderate resolution imaging spectroradiometer radiance with the weather research and forecasting data assimilation system," *Jour. of Appl. Remote Sens.*, vol. 11, no. 3, p. 036002, 2017.
[4] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: a comprehensive review and list of resources," *IEEE Sens. and Remote Sens. Magazine*, vol. 5, no. 4, pp. 8–36, 2017.
[5] Q. Zou, "Research on cloud computing for disaster monitoring using massive remote sensing data," in *Proc. IEEE Int. Conf. on Cloud Computing and Big Data Analysis*, 2017, pp. 29–33.
[6] U. Nazir, N. Khurshid, M. Ahmed Bhimra, and M. Taj, "Tiny-inception-resnet-v2: Using deep learning for eliminating bonded labors of brick kilns in south asia," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition Workshops*, 2019, pp. 39–43.
[7] M. D'Urso, T. Isernia, and A. F. Morabito, "On the solution of 2-d inverse scattering problems via source-type integral equations," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 48, no. 3, pp. 1186–1198, 2009.
[8] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: benchmark and state of the art," *in Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.
[9] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Jour. of Sel. Topics in Appl. Earth Observ. and Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, 2019.
[10] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, 2016.
[11] F. Ye, H. Xiao, X. Zhao, M. Dong, W. Luo, and W. Min, "Remote sensing image retrieval using convolutional neural network features and weighted distance," *IEEE Geosci. and Remote Sens. Lett.*, pp. 1–5, 2018.
[12] D. Dai and W. Yang, "Satellite image classification via two-layer sparse coding with biased image representation," *IEEE Geosci. and Remote Sens. Lett.*, vol. 8, no. 1, pp. 173–176, 2011.
[13] M. Pesaresi and J. A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 39, no. 2, pp. 309–320, 2001.
[14] M. Song and D. Civco, "Road extraction using svm and image segmentation," *Photogrammetric Engineering & Remote Sens.*, vol. 70, no. 12, pp. 1365–1371, 2004.
[15] X.-Y. Wang, J.-F. Wu, and H.-Y. Yang, "Robust image retrieval based on color histogram of local feature regions," *Multimedia Tools and Applications*, vol. 49, no. 2, pp. 323–345, 2010.
[16] S. A. Chatzichristofis and Y. S. Boutalis, "CEDD: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval," *Lecture Notes in Comput. Sci.*, vol. 5008 LNCS, pp. 312–322, 2008.
[17] P. Bosilj, E. Aptoula, S. Lefèvre, and E. Kijak, "Retrieval of Remote Sensing Images with Pattern Spectra Descriptors," *ISPRS Int. Jour. of Geo-Information*, vol. 5, no. 12, p. 228, 2016.
[18] E. Aptoula, "Remote sensing image retrieval with global morphological texture descriptors," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 52, no. 5, pp. 3023–3034, 2014.

[19] P. Napoletano, "Visual descriptors for content-based retrieval of remote-sensing images," *Int. Jour. of Remote Sens.*, vol. 39, no. 5, pp. 1–34, 2018.

[20] W. Zhou, S. Newsam, C. Li, and Z. Shao, "Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval," *Remote Sens.*, vol. 9, no. 5, p. 489, 2017.

[21] N. Passalis and A. Tefas, "Learning neural bag-of-features for large-scale image retrieval," *IEEE Trans. on Sys., Man, and Cybernetics: Sys.*, vol. 47, pp. 2641–2652, 2017.

[22] Z. Shao, K. Yang, and W. Zhou, "Performance evaluation of single-label and multi-label remote sensing image retrieval using a dense labeling dataset," *Remote Sens.*, vol. 10, no. 6, p. 964, 2018.

[23] D. Lin, K. Fu, Y. Wang, G. Xu, and X. Sun, "Marta gans: Unsupervised representation learning for remote sensing image classification," *IEEE Geosci. and Remote Sens. Lett.*, vol. 14, no. 11, pp. 2092–2096, 2017.

[24] S. Xu, X. Mu, D. Chai, and X. Zhang, "Remote sensing image scene classification based on generative adversarial networks," *Remote Sens. Lett.*, vol. 9, no. 7, pp. 617–626, 2018.

[25] G. Xia, X. Tong, F. Hu, Y. Zhong, M. Datcu, and L. Zhang, "Exploiting deep features for remote sensing image retrieval: A systematic investigation," *arXiv preprint arXiv:1707.07321*, 2017.

[26] I. Dimitrovski, D. Kocev, I. Kitanovski, S. Loskovska, and S. Džeroski, "Improved medical image modality classification using a combination of visual and textual features," *Computerized Medical Imaging and Graphics*, vol. 39, pp. 14–26, 2015.

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *Proc. the AAAI Conf. on Artifi. Intell.*, vol. 4, 2017, p. 12.

[28] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *Proc. European Conf. on Comput. Vis.*, 2016, pp. 494–509.

[29] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," *Proc. SIGSPATIAL Int. Conf. on Adv. in Geographic Inf. Sys.*, p. 270, 2010.

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

[31] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[32] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern recognition*, vol. 40, no. 1, pp. 262–282, 2007.

[33] B. Thomee and M. S. Lew, "Interactive search in image retrieval: a survey," *Int. Jour. of Multimedia Inf. Retrieval*, vol. 1, no. 2, pp. 71–86, 2012.

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

[35] P. Thanh Noi and M. Kappas, "Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery," *Sensors*, vol. 18, no. 1, p. 18, 2018.

[36] T.-B. Jiang, G.-S. Xia, Q.-K. Lu, and W.-M. Shen, "Retrieving aerial scene images with learned deep image-sketch features," *Jour. of Comput. Sci. and Tech.*, vol. 32, no. 4, pp. 726–737, 2017.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," 2009.

[39] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. and Remote Sens. Lett.*, vol. 12, no. 12, pp. 2438–2442, 2015.

[40] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Jour. of Mach. Learn. Research*, vol. 11, pp. 3371–3408, 2010.

[41] P. Thomas, B. Price, C. Paine, and M. Richards, "Remote electronic examinations: student experiences," *British Jour. of Educational Tech.*, vol. 33, no. 5, pp. 537–549, 2002.

[42] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral-spatial feature learning via deep residual conv-deconv network for hyperspectral image classification," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 56, no. 1, pp. 391–406, 2018.

[43] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Proc. the Conf. on Deep Learning and Data Labeling for Medical Applications*, 2016, pp. 179–187.

[44] Q. Bao and P. Guo, "Comparative studies on similarity measures for remote sensing image retrieval," in *Proc. IEEE Int. Conf. on Sys., Man and Cybernetics*, vol. 1, Oct 2004, pp. 1112–1116 vol.1.

[45] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proc. ACM Int. Conf. on Multimedia*. ACM, 2014, pp. 157–166.

[46] S. C. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, vol. 2, 2006, pp. 2072–2078.

[47] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Jour. of Mach. Learn. Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[48] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *Jour. of Mach. Learn. Research*, vol. 11, no. Mar, pp. 1109–1135, 2010.

[49] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. Int. Conf. on Pattern Recognition*, 2014, pp. 34–39.

[50] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2016, pp. 4004–4012.

[51] G. Cheng, P. Zhou, and J. Han, "Duplex metric learning for image set classification," *IEEE Trans. on Image Process.*, vol. 27, no. 1, pp. 281–292, 2018.

[52] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2015, pp. 3279–3286.

[53] J. Han, G. Cheng, Z. Li, and D. Zhang, "A unified metric learning-based framework for co-saliency detection," *IEEE Trans. on Circuits and Sys. for Video Tech.*, vol. 28, no. 10, pp. 2473–2483, 2018.

[54] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *In Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2014, pp. 1875–1882.

[55] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. Deep Learning Workshop, Int. Conf. on Mach. Learn.*, Lille, France, 2015.

[56] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. Int. Workshop on Similarity-Based Pattern Recognition*, 2015, pp. 84–92.

[57] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, June 2016, pp. 1288–1296.

[58] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. Int. Conf. on Mach. Learn.*, 2016, pp. 1558–1566.

[59] B. McFee and G. R. Lanckriet, "Metric learning to rank," in *Proc. Int. Conf. on Mach. Learn.*, 2010, pp. 775–782.

[60] M. Engilberge, L. Chevallier, P. Pérez, and M. Cord, "Sodeep: a sorting deep net to learn ranking loss surrogates," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2019, pp. 10 792–10 801.

[61] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2019, pp. 1861–1870.

[62] W. Xiong, Y. Lv, Y. Cui, X. Zhang, and X. Gu, "A discriminative feature learning approach for remote sens. image retrieval," *Remote Sens.*, vol. 11, no. 3, p. 281, 2019.

[63] B. Chaudhuri, B. Demir, L. Bruzzone, and S. Chaudhuri, "Region-based retrieval of remote sensing images using an unsupervised graph-theoretic approach," *IEEE Geosci. and Remote Sens. Lett.*, vol. 13, no. 7, pp. 987–991, 2016.

[64] X. Tang, X. Zhang, F. Liu, and L. Jiao, "Unsupervised deep feature learning for remote sensing image retrieval," *Remote Sens.*, vol. 10, no. 8, p. 1243, 2018.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. European Conf. on Comput. Vis.*, 2016, pp. 630–645.

[66] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. European Conf. on Comp. Vis.*, 2016, pp. 646–661.

[67] Z. Wu, C. Shen, and A. van den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119 – 133, 2019.

[68] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2019, pp. 4340–4349.
[69] S. Özkan, T. Ates, E. Tola, M. Soysal, and E. Esen, "Performance analysis of state-of-the-art representation methods for geographical image retrieval and categorization," *IEEE Geosci. and Remote Sens. Lett.*, vol. 11, no. 11, pp. 1996–2000, 2014.
[70] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[71] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2015, pp. 1–9.
[72] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2016, pp. 5297–5307.

Numan Khurshid (Student Member IEEE) obtained his B.Sc. degree in Electrical Engineering from University of Engineering and Technology, Peshawar in 2011 and his Master's degree in Electrical Engineering from National University of Science and Technology (NUST), Islamabad. Currently, he is working as a Ph.D. candidate at the Computer Vision and Graphics Lab, which is a part of the Department of Computer Science, School of Science and Engineering, Lahore University of Management Sciences (LUMS), Lahore. His research interests include image processing, computer vision, natural language processing, machine learning and deep learning. He also enjoys working on large-scale visual recognition problems in the area of geoscience and remote sensing.

Mohbat received the B.S. degree in Electrical (Telecommunication) Engineering from COMSATS Institute of Information Technology, Lahore in 2014 and the M.S. degree in Electrical Engineering from Lahore University of Management Sciences (LUMS) in 2018. He is currently working as a research assistant with Computer Vision Lab at LUMS. His research interests include scene understanding, object detection and their applications in robotics and automation.

Murtaza Taj earned his Ph.D. and M.Sc. degrees in electronic engineering and computer science from the Queen Mary University of London (QMUL), United Kingdom, in 2009 and 2005, respectively. Currently, he is an Assistant Professor at the Lahore University of Management Sciences, Syed Babar Ali School of Science and Engineering, Pakistan. He is also an adjunct faculty at the Ontario Tech University, Canada. His research interest lies in the area of Computer Vision, Graphics and Image Processing. In particular, he is interested in detection and tracking of object in 2D and 3D scenes and in automatic generation of 3D models from raw point cloud data.

Faisal Z. Qureshi (Senior Member IEEE, Member ACM and Member CIPPRS) received the B.Sc. degree in Mathematics and Physics from Punjab University, Lahore, Pakistan, in 1993, the M.Sc. degree in Electronics from Quaid-e-Azam University, Islamabad, Pakistan, in 1995, and the M.Sc. and Ph.D. degrees in Computer Science from the University of Toronto, Toronto, Canada, in 2000 and 2007, respectively.

He is an Associate Professor of Computer Science in the Faculty of Science, Ontario Tech University, where he leads the Visual Computing Lab. His research focuses on computer vision, and his scientific and engineering interests center on the study of computational models of visual perception to support autonomous, purposeful behavior in the context of *ad hoc* networks of smart cameras. Dr. Qureshi is also active in journal special issues and conference organizations. He served as the general co-chair for the Workshop on Camera Networks and Wide-Area Scene Analysis (co-located with CVPR) in 2011-13. He also served as the co-chair of Computer and Robot Vision (CRV) conference 2015/16 meetings.