IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 26, 2020, accepted December 7, 2020, date of publication December 23, 2020, date of current version January 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3046845

Semi-Global Context Network for Semantic Correspondence

HO-JUN LEE, HONG TAE CHOI, SUNG KYU PARK, AND HO-HYUN PARK
School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Ho-Hyun Park (hohyun@cau.ac.kr)

This work was supported in part by the Chung-Ang University Graduate Research Scholarship in 2019, in part by the Competency Development Program for Industry Specialists of the Korean Ministry of Trade, Industry and Energy (MOTIE), operated by the Korea Institute for Advancement of Technology (KIAT) under Project P0002397, in part by the Project—HRD Program for Industrial Convergence of Wearable Smart Devices, in part by the Korean Ministry of Science and Information and Communication Technology (ICT) (MSIT), operated by the National Research Foundation (NRF) under Project 2020R1F1A1069065, and in part by the Project—Research on 3D semantic matching and generative model for 3D visual localization, object transformation, and motion estimation.

ABSTRACT Estimating semantic correspondence between pairs of images can be challenging as a result of intra-class variation, background clutter, and repetitive patterns. This paper proposes a convolutional neural network (CNN) that attempts to learn rich semantic representations that contain the global semantic context to enable robust semantic correspondence estimation against intra-class variation and repetitive patterns. We introduce a global context fused feature representation that efficiently employs the global semantic context in estimating semantic correspondence as well as a semi-global self-similarity feature to reduce background clutter-induced distraction in capturing the global semantic context. The proposed network is trained in an end-to-end manner using a weakly supervised loss, which requires a weak level of supervision involving annotation on image pairs. This weakly supervised loss is supplemented with a historical averaging loss to effectively train the network. Our approach decreases running time by a factor of more than four and reduces the training memory requirement by a factor of three and produces competitive or superior results relative to previous approaches on the PF-PASCAL, PF-WILLOW, and TSS benchmarks.

INDEX TERMS Context fusion, historical averaging, neighborhood consensus network, semantic correspondence, semi-global self-similarity, weakly supervised learning.

I. INTRODUCTION

Semantic correspondence is the problem of establishing dense correspondence across images depicting different instances of a given object or scene category [1]–[3]. Relative to early correspondence tasks [3]–[6], which focus on finding correspondence between images depicting a given object or scene, semantic correspondence is challenging as a result of the need to process large intra-class variations, viewpoint changes, and background clutter. To find correspondence between related images, early approaches employ hand-crafted features such as SIFT [7] or HOG [8] in association with geometric regularizers [9] and have proven useful in various computer vision tasks such as image editing, scene understanding, object tracking, object detection, and 3D reconstruction. However, owing to a lack of high-level semantic information, matching using hand-crafted features often fails when large changes in appearance occur.

The development of convolutional neural networks [10], [11] (CNNs) has led to significant progress in the field of semantic correspondence in recent years. Recent approaches adopting CNNs [12]–[20] benefit from the ability of such networks to learn high-level features for accurate correspondence estimation. Among these approaches, Neighborhood Consensus Networks [21] (NC-Nets) and their variants [22]–[24] have shown excellent performance. An NC-Net [21] employs a pre-trained network to extract local feature maps from an image pair and calculates the cosine similarities between any two locations in the respective local feature maps, which are stored in a 4D tensor referred to as the correlation map. The NC-Net then refines the correlation map using a sequence of 4D convolutional kernels, which are trained to capture matching patterns between two different images and are highly effective in filtering incorrect matches.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Jiang.

The global semantic context or set of contextual information on inter-pixel relations within an image, can be used to help a visual system recognize an object's spatial layout to indicate where and how the object appears in the image. Recently [22], the global semantic context has been incorporated into the NC-Net, further improving its performance by enabling robust matching against repetitive patterns and intra-class variation. In this approach, context-aware features that contain the global semantic context are generated and then correlation maps derived from local and context-aware features are fused to apply the global semantic context to local features. A dynamical fusion mechanism [22] for fusing correlation maps to alleviate the performance degradation problem caused by background clutter has also been developed. However, this method requires a long execution time owing to its overuse of the 4D convolution kernels. Additional 4D convolution operations are needed to refine the correlation map derived from the context-aware features and to fuse the correlation maps dynamically. Since 4D convolution kernels are prohibitively expensive, these approaches are computationally intensive, resulting in lengthy execution time. Furthermore, their training method [22] is highly memory intensive since they require additional supervision from additional tasks to train the network.

Since the existing method of incorporating the semantic context into NC-Net requires a large amount of computation and memory, high-end or multiple graphics processing units (GPU) are required to run and train the model. Therefore, the existing method is inefficient in that it requires more GPU resources to incorporate the semantic context. To minimize the GPU resource requirement, this paper proposes a computationally efficient and memory-saving method of incorporating semantic context into the task of semantic correspondence estimation. To reduce the neural network's computational cost, the proposed method avoids combining correlation maps to incorporate the global semantic context by directly combining the global semantic context with the local features, thereby avoiding the high number of heavy 4D convolution operations needed to fuse the correlation maps. Furthermore, it efficiently mitigates performance degradation problems arising from background clutter by considering the inter-pixel spatial properties in capturing the global semantic context. The proposed network is trained with weak supervision from an image label-annotated dataset in an end-to-end manner, and a historical averaging loss [25] is used to efficiently train the network, reducing the training memory requirement relative to the previously developed methods.

We evaluated the accuracy of the proposed method in carrying out a weakly supervised semantic correspondence task through experiments using data from three public datasets—PF-PASCAL [26], PF-WILLOW [27], and TSS [28]. We further compared its training memory usage and execution time with those of other neighborhood consensus (NC)-based methods. The results revealed that the proposed method is four times faster and uses one-third the training memory compared to the prior approaches [22], while producing competitive results relative to state-of-the-art weakly supervised semantic correspondence methods. We summarize our contributions as follows: first, our proposed global context fused feature representation makes our method faster than existing state-of-the-art methods; second, we introduce a semi-global self-similarity feature to efficiently mitigate the effect of background clutter on the captured global context; and third, we achieve competitive performance while reducing training memory usage compared to existing methods.

II. RELATED WORK

Conventional correspondence tasks [3]–[6] focus on matching local descriptors around the interest points of an image. This is done through instance matching [7] or by estimating dense matches between images within a given scene using optical flow estimation [3], [4] or stereo matching [5], [6]. Unlike conventional tasks, semantic correspondence estimates dense matches across images depicting different instances of a given object or scene class. Early semantic correspondence approaches employed hand-crafted descriptors such as SIFT [7], [29], HOG [8], [27], [28], and DAISY [30] in conjunction with geometric models [9], [29], [31] or random sampling [32], [33]. As matching with hand-crafted features is easily distracted by background clutter or scale changes, several attempts have been made to estimate correspondence by using object proposal [27], [34] to generate matching elements or jointly perform co-segmentation [28] or performing matching in scale space [35] to obtain robust matching against background clutter and scale change. However, the lack of high-level semantic information in hand-crafted features often causes these approaches to fail when facing non-rigid deformation or large changes in appearance.

Recently developed semantic correspondence approaches [12]–[20] employ CNNs to obtain high-level semantic features that are robust to intra-class and shape variation. Several approaches [13]–[16], [18], [19], in which semantic correspondence is formulated as image alignment, employ CNN architecture to estimate transformation parameters between pairs of images. Rocco *et al.* [13] proposed a network that infers geometric transformation between two images by training in a self-supervised fashion. Seo *et al.* [16] proposed an attentive alignment method for filtering distracting regions. Inspired by RANSAC, Rocco *et al.* [14] further introduced an end-to-end trainable and weakly supervised CNN architecture using soft inlier counts. Their work was further improved by introducing joint learning with co-segmentation [15] and applying methods for predicting the foreground region and enforcing cycle consistency [18]. Kim *et al.* [19] designed a network with a recurrent structure to estimate geometric transformation iteratively. However, because only low-omplexity parametric transformations are inferred between images, these methods are highly sensitive to non-rigid deformation and local geometric variation.

Unlike image-alignment approaches, semantic flow [12], [17], [20], [36], [37] finds correspondence between individual pixels or patches by learning local features for semantic correspondence. These approaches are not significantly affected by non-rigid deformation. Choy *et al.* introduced the UCNet CNN model [20], which learns feature embedding for semantic correspondence problems through deep metric learning. Han *et al.* [12] proposed a CNN model that estimates the semantic correspondence between images using both appearance and geometry information. In [38] and [39], approaches to learning geometry-aware features through self- and weakly-supervised methods, respectively, were introduced. The FCSS [36] computes local self-similarity descriptors [40] with learned sparse sampling patterns in the object proposal and uses them to estimate a dense affine transformation flow at each feature location. The Hyper-pixel flow approach [37] applies the beam search algorithm to effectively combine features extracted from different layers, thereby improving the performance of the correspondence task. SFNet [17] uses images annotated with binary foreground masks and subjected to synthetic geometric deformation to train CNNs with mask and flow consistency and smoothness loss.

The NC-Net semantic flow approach [21] adopts a sequence of 4D convolutions that incorporate neighborhood consensus information to refine a 4D tensor that stores all matching scores. However, as the size of the 4D matching tensor increases quadratically with the size of the image, NC-Net is not scalable to high-resolution images. Furthermore, 4D convolution consumes large amounts of memory and suffers from long execution times. To address these issues, Sparse-NCNet [24] creates a sparse 4D matching tensor and processes using sub-manifold sparse convolution, whereas DRCNet [23] carries out a 4D convolution operation on a coarse-resolution image and incorporates the results into a 4D tensor obtained from the corresponding fine-resolution features. Other recent approaches [22], [41] apply semantic context to reduce intra-class variation and resolve the matching ambiguities in semantic correspondence estimation. DCCNet [22] uses a self-similarity descriptor to generate context-aware features that are fused with the local features using an attention mechanism. ANCNet [41] applies the multi-scale self-similarity feature as context-aware features and carries out processing using non-isotropic 4D convolution. These methods, however, require multiple 4D convolution operations to process additional context-aware features, an approach that degrades the inference time and is memory intensive. Unlike these approaches, the proposed method does not require additional 4D operations or tensors, making it much more efficient than previous approaches in terms of computational cost and memory usage [22].

III. METHOD

A. PROBLEM SETTING AND OVERVIEW

The goal of semantic correspondence is to estimate the pixel-wise correspondence between pairs of images depicting different instances from a given object or scene class. A common approach to the estimation of pixel-wise correspondence is matching the local features of two different images. In this process, local features are first extracted from an image pair and then similarity scores are calculated for all possible feature matches between the two images. Pixel-wise correspondence is inferred by selecting the feature match with the highest similarity score. However, the lack of global semantic context information in the local features can lead to the miscalculation of the similarities between different image points, primarily when large intra-class variation occurs, and repetitive patterns are present. To achieve robust matching against intra-class variation and repetitive pattern, the proposed method adds global contextual information to local features through the use of a CNN that incorporates semantic context in local features to achieve robust semantic correspondence estimation.

The overall architecture of the proposed network is shown in Fig. 1. It comprises a feature extractor, a local-context fusion module, and a neighborhood consensus module. As a first step, local features are extracted from input image pairs using the pre-trained CNN as the feature extractor. Each image pair $\left(\boldsymbol{I}^s, \boldsymbol{I}^t\right)$ is fed into a feature extractor to obtain a local feature map pair $\left(\boldsymbol{X}^s, \boldsymbol{X}^t\right)$, where $\boldsymbol{X}^s \in \mathbb{R}^{d \times h_l \times w_l}$ and $\boldsymbol{X}^t \in \mathbb{R}^{d \times h_l \times w_l}$ are the normalized local feature representations of input images $\boldsymbol{I}^s$ and $\boldsymbol{I}^t$, respectively, d is the dimension of local feature, and h_l, w_l are the height and width of the local feature map, respectively.

The local-context fusion module (LCF) is used to incorporate global semantic context into the local features. The LCF module first captures the global semantic context from the local feature map by extracting self-similarity features that convey the internal spatial layout of the image. The module then generates context-aware features by integrating self-similarity features with local semantic features. After multiplying the learnable scalar by the context-aware features to adjust the global semantic context information in the local features, the module combines the context-aware features with the local features to add global semantic context. In this process, the LCF module receives a local feature map pair $(\boldsymbol{X}^s, \boldsymbol{X}^t)$ as an input and produces a new feature map pair $(\boldsymbol{Z}^s, \boldsymbol{Z}^t)$ that contains local appearance and global semantic context information.

The cosine similarities of all possible matches between $\boldsymbol{Z}^s$ and $\boldsymbol{Z}^t$ are compared and stored in a 4D tensor referred to as the correlation map, $\boldsymbol{C}$, where $\boldsymbol{C} \in \mathbb{R}^{h_l \times w_l \times h_l \times w_l}$. The correlation map is then refined using the neighborhood consensus (NC) module [21], which applies a sequence of 4D convolution kernels to filter out incorrect matches in the correlation map.

Finally, the pixel-wise correspondence is inferred from the refined correlation map by extracting the most likely matches from it. The model is trained with a weakly supervised loss [21] that requires supervision at the level of image pairs in an end-to-end manner. To effectively train the network,

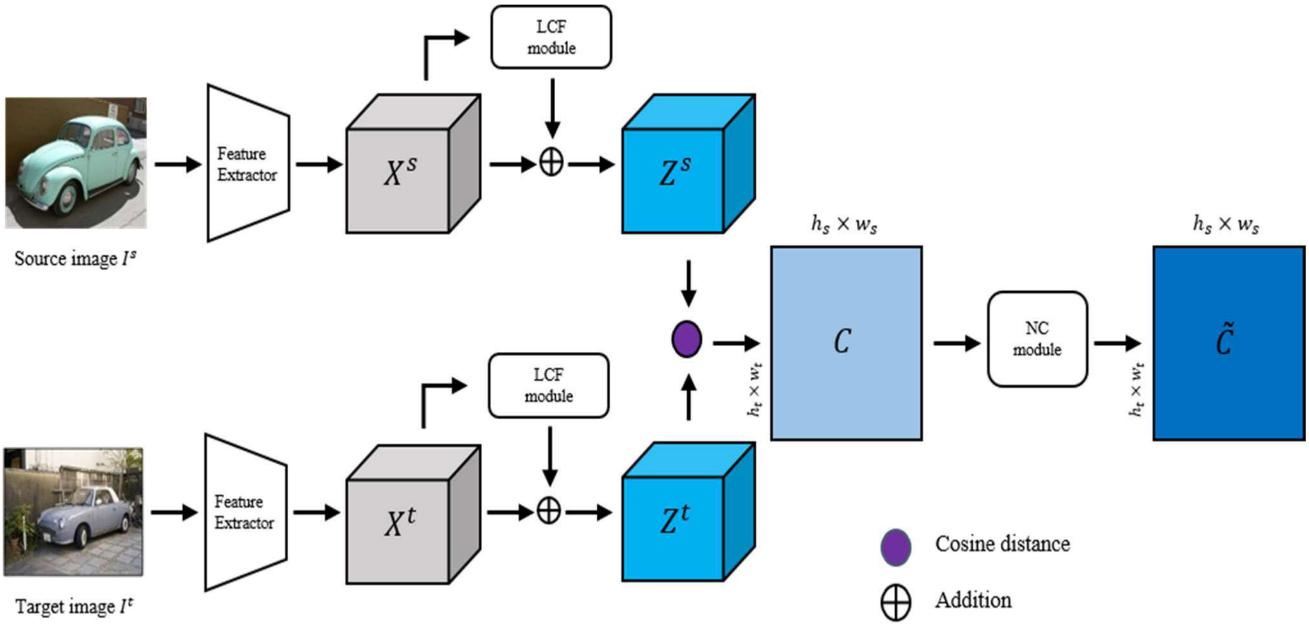

FIGURE 1. Overview of proposed network. Given an image pair (I^s, I^t), the network extracts a local feature map pair (X^s, X^t) using the feature extractor. The LCF module then captures the global semantic context from each local feature map and generates a global context fused feature map pair (Z^s, Z^t). A 4D correlation map C is then produced by computing all possible matching similarities between Z^s and Z^t. C is refined using the NC modules to filter incorrect matches within it. Finally, semantic correspondence is inferred from the refined correlation map $\tilde{C}$ by selecting the most-likely matches in $\tilde{C}$.

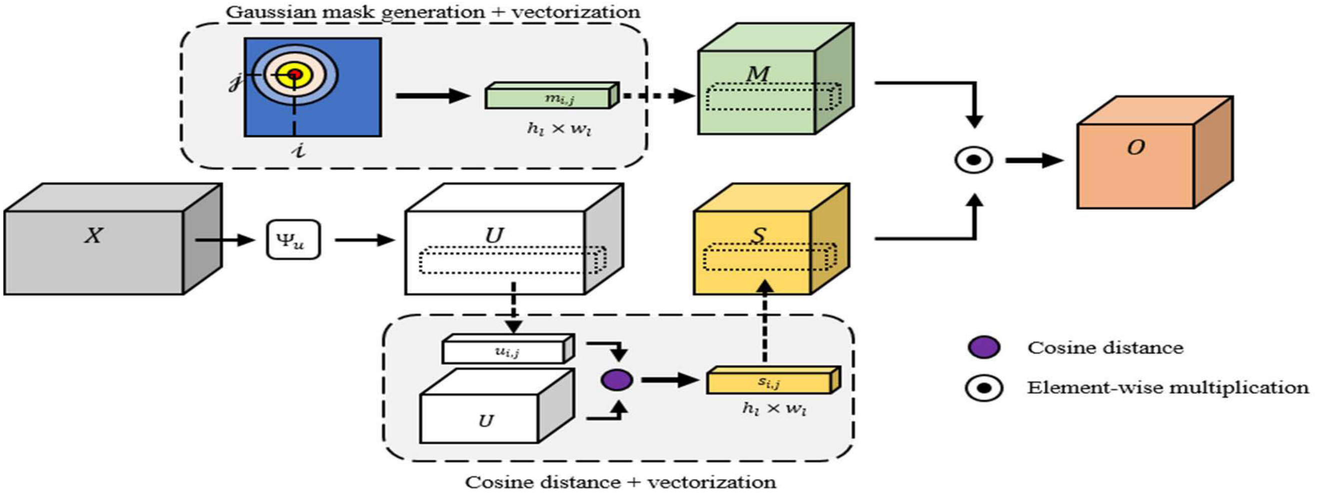

FIGURE 2. Overview of semi-global self-similarity feature generation process, in which local attention masks are applies as similarity features S to draw attention to local self-similarity patterns for mitigating the effect of background clutter.

a further historical averaging loss [25] is applied to the weakly supervised loss results.

B. LOCAL CONTEXT FUSION MODULE

The LCF module incorporates spatial context into the sets of local features established by the feature extractor by generating context-aware features from the local feature map and combining them with the local features. In subsection 1), we introduce a new self-similarity feature that effectively describes the global semantic context, as shown in Fig. 2. Following this, in subsection 2) we describe the generation of context-aware features using the local features and the self-similarity feature introduced in Subsection 1 and introduce a method for combining local and context-aware features.

1) SEMI-GLOBAL SELF-SIMILARITY FEATURE

To capture the global semantic context, global self-similarity features that measure the self-similarity patterns of an entire image are used as global spatial layout cues. These global self-similarity features are calculated in a manner similar to that used to generate attention maps [42]. First, the local feature map $\boldsymbol{X}$ is transformed into a new feature map $\boldsymbol{U}$. For each location (i, j) in $\boldsymbol{U}$, the global self-similarity features at (i, j) are obtained by calculating the cosine similarities between the feature $\boldsymbol{u}_{i,j}$ and all features in $\boldsymbol{U}$. The ReLU activation function is then used to suppress negative self-similarity scores to filter out irrelevant pixels. The self-similarity feature is calculated as follows:

$$\boldsymbol{u}_{i,j} = \boldsymbol{\Psi}_u(\boldsymbol{x}_{i,j}) \tag{1}$$

$$\boldsymbol{s}_{i,j} = \mathbf{ReLU}\left(\left[\frac{\langle \boldsymbol{u}_{i,j}, \boldsymbol{u}_{1,1}, \rangle}{\left\|\boldsymbol{u}_{i,j}\right\|_2 \left\|\boldsymbol{u}_{1,1}\right\|_2}, \ldots, \frac{\langle \boldsymbol{u}_{i,j}, \boldsymbol{u}_{h_l,w_l} \rangle}{\left\|\boldsymbol{u}_{i,j}\right\|_2 \left\|\boldsymbol{u}_{h_l,w_l}\right\|_2}\right]\right) \tag{2}$$

$$\boldsymbol{S} = \{\boldsymbol{s}_{1,1}, \ldots, \boldsymbol{s}_{h_l,w_l}\} \tag{3}$$

$$\boldsymbol{s}_{i,j} \in \mathbb{R}^{N\times 1}, \quad \boldsymbol{S} \in \mathbb{R}^{N\times h_l \times w_l} \tag{4}$$

where $\boldsymbol{\Psi}_u(\boldsymbol{x}_{i,j}) = \boldsymbol{\Phi}_u \boldsymbol{x}_{i,j}$ is a linear transformation in which $\boldsymbol{\Phi}_u \in \mathbb{R}^{d\times d}$ is the weight matrix to be learned, $\boldsymbol{s}_{i,j}$ is the self-similarity feature of location (i, j), $N = h_l \times w_l$, and $\boldsymbol{S}$ denote the self-similarity features of image I. In calculating self-similarity, it is insufficient to describe the correlation between features by measuring the similarity using the feature value alone without considering the spatial properties between features. Therefore, more precise self-similarity patterns are extracted by considering the spatial relations between features. Using the fact that the correlation between two points in an image increases as they approach each other, the self-similarity scores are increased as the spatial distance between two features is reduced. In this manner, the local self-similarity patterns can be used to increase the similarity score as the spatial distance between features diminishes.

A Gaussian function is used to build a local attention mask. For a self-similarity feature $\boldsymbol{s}_{i,j}$, a local attention mask $\boldsymbol{m}_{i,j}$ is generated using $g_{i,j}$, a Gaussian function with center position (i, j). The resulting local attention mask has a value of one at (i, j) and smoothly decreases to zero as the distance from (i, j) increases. This local attention mask is applied to the self-similarity patterns by performing element-wise multiplication between $\boldsymbol{s}_{i,j}$ and $\boldsymbol{m}_{i,j}$ to produce a new self-similarity feature, $\boldsymbol{o}_{i,j}$, which is referred to as a “semi-global” self-similarity feature because it captures the global self-similarity patterns while focusing on the local self-similarity patterns. The process for applying the local attention mask to self-similarity features is given by the following:

$$g_{i,j}(k, p) = \mathbf{exp}\left(-\frac{(i-k)^2 + (j-p)^2}{2\sigma^2}\right) \tag{5}$$

$$\boldsymbol{m}_{i,j} = \{g_{i,j}(1, 1), \ldots, g_{i,j}(h_l, w_l)\} \tag{6}$$

$$\boldsymbol{o}_{i,j} = \boldsymbol{m}_{i,j} \odot \boldsymbol{s}_{i,j} \tag{7}$$

$$\boldsymbol{o}_{i,j}, \quad \boldsymbol{m}_{i,j} \in \mathbb{R}^{N\times 1} \tag{8}$$

$$\boldsymbol{O} \in \mathbb{R}^{N\times h_l \times w_l} \tag{9}$$

where $\odot$ is element-wise multiplication, $g_{i,j}$ is a Gaussian function with center position (i, j) and standard deviation σ, $\boldsymbol{o}_{i,j}$ is the semi-global self-similarity feature of location (i, j), and $\boldsymbol{O}$ denotes the semi-global self-similarity features of image I.

Through this focusing on local self-similarity patterns, the proposed method obtains global semantic context that is robust to background clutter. The self-similarity feature should capture a self-similarity pattern within the global region that can represent the global semantic cue. As longer-range self-similarity patterns include more similarities with spatially distant background clutter, the results of capturing the global semantic context using self-similarity features can be easily affected by background clutter. However, by using local self-similarity patterns obtained from the local attention mask, it is possible to reduce the similarity scores with spatially distant background clutter, thereby reducing the impact of background clutter on the captured global semantic cue.

Unlike FCSS [36], our method does not rely on object proposal, which increases the estimation time needed to find matching elements. The proposed method measures self-similarity patterns over the global region to capture a global semantic context, whereas FCSS focuses on measuring the self-similarity pattern within a local region.

This approach is similar to the one used by DCCNet [22], which also uses self-similarity patterns within the global region as spatial layout cues. However, our method does not require the size of the local feature map to be increased for measuring the self-similarity pattern, making it more memory-efficient than DCCNet. In addition, unlike DCCNet the proposed method efficiently removes the effect of background clutter on the self-similarity. To mitigate the effect of background clutter, DCCNet uses an attention module [22] to dynamically fuse context-aware and local features. This approach is computationally expensive because it requires additional heavy 4D convolution operations to carry out the fusion. The proposed approach is more computationally efficient because it requires only element-wise multiplication in applying the local attention mask to the mitigation of background clutter.

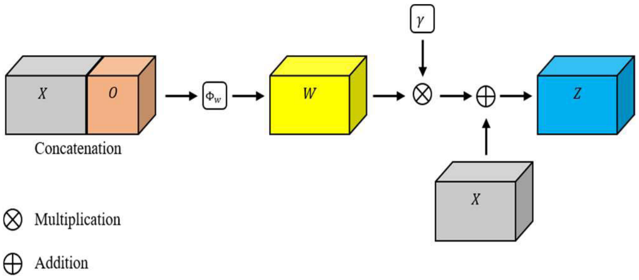

FIGURE 3. **Overview of procedure for generating fused global semantic context features.**

2) LOCAL CONTEXT FUSION

Here, we describe the process for generating context-aware features from the semi-global self-similarity features and combining them with local features (Fig. 3). The semi-global self-similarities contain only the spatial layout information of the image and lacks the appearance information represented by local features. To capture different aspects of the semantic object, therefore, the proposed method applies the fusion step from [22] to generate context-aware features. The semi-global self-similarity features $\boldsymbol{O}$ are then integrated with the local feature map by applying a linear transformation over the concatenation of $\boldsymbol{O}$ and $\boldsymbol{X}$ as follows:

$$\boldsymbol{w}_{i,j} = \boldsymbol{\Phi}_w[\boldsymbol{x}_{i,j}, \boldsymbol{o}_{i,j}] \tag{10}$$

$$\boldsymbol{w}_{i,j} \in \mathbb{R}^{d\times 1} \tag{11}$$

where $\boldsymbol{\Phi}_w \in \mathbb{R}^{d\times(d+N)}$ is the weight matrix that transforms the concatenated features into d -dimensional space, and $\boldsymbol{w}_{i,j}$ is the context-aware feature of location (i, j). The context-aware semantic features are then added to the local features to incorporate the global semantic context into the local features. Before adding the context-aware features to the local features, the former are multiplied by the learnable scalar γ to adjust the global semantic context in the local features:

$$z_{i,j} = \boldsymbol{x}_{i,j} + \gamma \boldsymbol{w}_{i,j} \tag{12}$$

$$\boldsymbol{Z} = \{z_{1,1}, \ldots, z_{h_l,w_l}\} \tag{13}$$

$$z_{i,j} \in \mathbb{R}^{d\times 1}, \quad \boldsymbol{Z} \in \mathbb{R}^{d\times h_l \times w_l} \tag{14}$$

where $\boldsymbol{Z}$ denotes the global semantic context fused features of image I and the additional superscripts represent the global semantic context fused features $\boldsymbol{Z}^s$ and $\boldsymbol{Z}^t$ obtained from images $\boldsymbol{I}^s$ and $\boldsymbol{I}^t$, respectively.

However, adding untrained context-aware features directly to local features in the early stage of training results in unstable training. To prevent this, the learnable scalar γ is initialized to zero to ensure that the network relies on local features to find correspondences between images in the early stage of training. As the network learns to capture the global semantic context from images, it gradually incorporates a more global semantic context by assigning more weight to γ. Thus, the network can be trained stably by relying on local features at the early training stage and gradually adding the global semantic context as the training proceeds.

This approach is much lighter computationally than that used by DCCNet. As DCCNet generates an additional 4D correlation map for context-aware feature map pairs, it becomes computationally intensive as a result of the heavy 4D convolution operations needed to fuse the correlation maps. By contrast, our approach directly adds context-aware features to local features to avoid generating an additional 4D correlation map, allowing it to incorporate semantic context more efficiently and increase its speed relative to that of DCCNet.

C. NEIGHBORHOOD CONSENSUS MODULE

To infer pixel-wise correspondence between $\boldsymbol{Z}^s$ and $\boldsymbol{Z}^t$, cosine similarities are calculated for all possible matches between the features and stored in the 4D correlation maps $\boldsymbol{C} \in \mathbb{R}^{h_l\times w_l\times h_l\times w_l}$:

$$\boldsymbol{c}_{i,j,k,p} = \frac{\left\langle z^s_{i,j}, z^t_{k,p}\right\rangle}{\left\| z^s_{i,j}\right\|_2 \left\| z^t_{k,p}\right\|_2} \tag{15}$$

where $\boldsymbol{c}_{i,j,k,p}$ is the matching similarity scores between $z^s_{i,j}$ and $z^t_{k,p}$. To enable precise pixel-wise correspondence, the correlation map $\boldsymbol{C}$ is further refined by filtering out incorrect matches. To do this, the NC module from NC-Net [21] is adopted. The NC module comprises a stack of 4D convolutions that filter out incorrect matches by analyzing local neighborhood matching patterns. The proposed method applies this module to both matching directions (i.e., to matching $\boldsymbol{I}^s$ to $\boldsymbol{I}^t$ and matching $\boldsymbol{I}^t$ to $\boldsymbol{I}^s$), making the model invariant with respect to the order of images. The refined 4D correlation map is obtained as

$$\tilde{\boldsymbol{C}} = \mathcal{N}(\boldsymbol{C}) + \left(\mathcal{N}\left(\boldsymbol{C}^T\right)\right)^T \tag{16}$$

where $\mathcal{N}$ is the NC module, T denotes the swapping of the matching direction for an image pair, i.e., $\left(\boldsymbol{c}^T\right)_{i,j,k,p} = \boldsymbol{c}_{k,p,i,j}$, and $\tilde{\boldsymbol{C}}$ is the refined correlation map, which is the final output of the network. A mutual nearest neighbor consistency constraint [21] is also applied before and after $\mathcal{N}$ to down-weight the scores of matches that are not mutually nearest neighbors; the reader is referred to [21] for further details.

1) MOST-LIKELY MATCHES

Finally, the pixel-wise correspondence between images is inferred by selecting the most-likely matches from the refined correlation map. Before extracting these matches, softmax normalization is applied to the similarity scores to convert matching scores into matching probabilities. The matching probability of a given point (i, j) in $\boldsymbol{I}^s$ to an arbitrary point (k, p) in $\boldsymbol{I}^t$ is

$$\boldsymbol{v}^s_{i,j,k,p} = \frac{\mathbf{exp}(\tilde{\boldsymbol{c}}_{i,j,k,p})}{\sum_{a,b}\mathbf{exp}(\tilde{\boldsymbol{c}}_{a,b,k,p})} \tag{17}$$

Similarly, the matching probability of a given point at (k, p) in $\boldsymbol{I}^t$ to an arbitrary point (i, j) in $\boldsymbol{I}^s$ is

$$\boldsymbol{v}^t_{i,j,k,p} = \frac{\mathbf{exp}(\tilde{\boldsymbol{c}}_{i,j,k,p})}{\sum_{c,d}\mathbf{exp}(\tilde{\boldsymbol{c}}_{i,j,c,d})} \tag{18}$$

After performing the softmax normalization, pixel-wise correspondence between the pair of images is attained by performing hard assignments [21] over the refined correlation map in either of two possible directions—from $\boldsymbol{I}^s$ to $\boldsymbol{I}^t$ or vice versa—to select the most-likely matches. A given position (i, j) in $\boldsymbol{I}^s$ will correspond to (k, p) in $\boldsymbol{I}^t$ if

$$(k, p) = \underset{c,d}{\arg\max}\, \boldsymbol{v}^s_{i,j,c,d} \tag{19}$$

Similarly, a given position (k, p) in $\boldsymbol{I}^t$ will correspond to (i, j) in $\boldsymbol{I}^s$ if

$$(i, j) = \underset{a,b}{\arg\max}\, \boldsymbol{v}^t_{a,b,k,p} \tag{20}$$

D. LEARNING OBJECTIVE

The model parameter is trained in a weakly supervised manner that requires only a weak level of supervision comprising annotation on image pairs. For this, the weakly-supervised training loss proposed in NC-Net [21], which has the functional form

$$\ell_\omega \left(\boldsymbol{I}^s, \boldsymbol{I}^t\right) = -y \left(\overline{\boldsymbol{v}^s} + \overline{\boldsymbol{v}^t}\right) \tag{21}$$

is adopted, where y denotes the ground-truth label of the image pair $(\boldsymbol{I}^s, \boldsymbol{I}^t)$, with $y = +1$ and $y = -1$ corresponding to positive and negative matching, respectively, and $\overline{\boldsymbol{v}^s}$ and $\overline{\boldsymbol{v}^t}$ are the mean matching scores over all of the hard-assigned matches of a given image pair $(\boldsymbol{I}^s, \boldsymbol{I}^t)$ in both matching directions. Minimization of this loss maximizes and minimizes the scores of the positive and negative image pairs, respectively.

To stabilize the training process, the network gradually adds the global semantic context to the local features as the training proceeds. This allows the NC module to learn to filter out incorrect matches between local feature maps during the early training process and to gradually learn to filter out matches between feature maps containing a global semantic context. However, as training proceeds the NC module slowly forgets to filter incorrect matches between local feature maps. To train the NC module more effectively, the past-learned information is used to induce the NC module to preserve its ability to filter incorrect matches between local feature maps. To reuse the past-learned information, the historical averaging loss [25] on the NC module is applied as follows:

$$\ell_h \left(\boldsymbol{\theta}_{NC}\right) = \left\| \boldsymbol{\theta}_{NC} - \frac{1}{t} \sum\nolimits_{i=1}^{t} \boldsymbol{\theta}_{NC} \left[i\right] \right\|_2 \tag{22}$$

where $\boldsymbol{\theta}_{NC}$ are the current parameters of the NC module, and $\boldsymbol{\theta}_{NC}[i]$ are the parameters of the module at past time i. By continuously updating the past averages of the NC module parameters, it is possible for the module to retain the information learned during the training process. The overall loss of the proposed model can be written as

$$\ell \left(\boldsymbol{I}^s, \boldsymbol{I}^t\right) = \ell_\omega \left(\boldsymbol{I}^s, \boldsymbol{I}^t\right) + \lambda \ell_h \left(\boldsymbol{\theta}_{NC}\right) \tag{23}$$

where λ is a weight balancing term. To effectively train the network, DCCNet [22] uses additional supervision from two additional tasks involving the addition of training networks with correlation maps derived from the local and context-aware features, respectively. Adding these auxiliary tasks has shown to produce outstanding performance but requires a large amount of training memory to train the network with the additional correlation maps. The proposed method is more memory-efficient than DCCNet [22] because it avoids using additional correlation maps to train the network.

IV. EXPERIMENT

A. IMPLEMENTATION DETAILS

We implemented the proposed method using the PyTorch [43] framework. As a feature extractor, we used ResNet101 [11] pre-trained on ImageNet [44] with the parameters fixed and cropped at the conv4-23 layer. In the local-fusion context module, we set the value of σ in Eq. (5) to 7 and the output dimension of the number of context-aware semantic features, d, to 1,024, which is the same dimension of features established by the feature extractor. For the NC module, we followed [21] and stacked three 4D convolutional layers with the kernel size set to 5 × 5 × 5 × 5 and the channel number of the intermediate layer set to 16.

To train the network, we set the value of λ in the historical averaging loss to 1 by validation. We kept the pre-trained feature extractor weights fixed and initialized the weights of the LCF module randomly. The neighbor consensus module was initialized using the pre-trained weight from [21]. The network was trained for five epochs on a Tesla P100 with early stopping to avoid overfitting. An Adam optimizer [45] with a learning rate of 0.0005, no weight decay and batch size of 16 was used.

We then trained the model using the PF-PASCAL [26] dataset and evaluated its performance in carrying out weakly supervised semantic correspondence tasks using the PF-PASCAL [26], PF-WILLOW [27], and TSS datasets [28]. To evaluate the model on the three different datasets, we reshaped the size of all images to 400 × 400.

B. BENCHMARK COMPARISONS

1) PF-PASCAL BENCHMARK

The PF-PASCAL benchmark contains 1,351 keypoint annotated image pairs classified into 20 categories. Following the split in [12], we divided the dataset into 700 training pairs, 300 validation pairs, and 300 test pairs. To train the network using weakly supervised loss, we followed the procedure in [21] by using the 700 training pairs as positive training pairs and generating negative pairs by randomly paring images from different categories.

We evaluated the model using the percentage of correct keypoints (PCK) matrix [46], which counts the number of correct keypoints whose distance from ground-truth lies within $\alpha \max(h, w)$ pixels, where h and w are height and width, respectively, of the image or bounding box, and then divides by the total number of image pairs. In line with previous work, we evaluated PCK($\alpha = 0.1$) with respect to image size.

Table 1 compares the results obtained using our method with those obtained using recently developed state-of-the-art methods including NC-Net [21], DCCNet [22], WeakAlign [14], A2Net [16], CNNGeo [13], Proposal Flow [27], UCN [20], and different versions of SCNet [12]. The proposed method achieves an overall PCK of 82%, which is only 0.3% lower than that of the best state-of-the-art method. It outperforms recent image alignment approaches (CNNGeo, A2Net, WeakAlign) because they are sensitive to non-rigid deformation and local geometric variations as they only estimate low-complexity parametric global transformation between two images. The proposed method avoids this problem by inferring pixel-wise correspondence from the correlation map. The proposed method outperforms recent

TABLE 1. **Per-class PCK($\alpha = 0.1$) by image size on the PF-PASCAL [26] dataset.**

Method	aero	bike	bird	boat	bottle	bus	car	Cat	chair	cow	d.table	dog	horse	moto	person	plant	sheep	sofa	train	tv	all
UCN [20]	64.8	58.7	42.8	59.6	47	42.2	61	45.6	49.9	52	48.5	49.5	53.2	72.7	53	41.4	83	49	73	66	55.6
HOG+PF-LOM [27]	73.3	74.4	54.4	50.9	49.6	73.8	72.9	63.6	46.1	79.8	42.5	48	68.3	66.3	42.1	62.1	65.2	57.1	64.4	58	62.5
SCNet$_{vgg16}$-AG [12]	67.6	72.9	69.3	59.7	74.5	72.7	73.2	59.5	51.4	78.2	39.4	50.1	67.0	62.1	69.3	68.5	78.2	63.3	57.7	59.8	66.3
SCNet$_{vgg16}$-A [12]	83.9	81.4	70.6	62.5	60.6	81.3	81.2	59.5	53.1	81.2	**62.0**	58.7	65.5	73.3	51.2	58.3	60.0	73.7	66.5	76.7	72.2
SCNet$_{vgg16}$-AG+ [12]	85.5	84.4	66.3	70.8	57.4	82.7	82.3	71.6	54.3	95.8	55.2	59.5	68.6	75.0	56.3	60.4	60.0	**73.7**	66.5	76.7	72.2
A2Net$_{res101}$ [16]	83.2	82.8	83.8	44.4	57.8	81.3	89.4	86.1	40.1	91.7	21.4	73.2	33.8	76.3	74.3	63.3	**100**	45	45.3	60	70.8
GeoCNN$_{res101}$ [13]	82.4	80.9	85.9	47.2	57.8	83.1	92.8	86.9	43.8	91.7	28.1	76.4	70.2	76.6	68.9	65.7	80	50.1	46.3	60.6	71.9
Weakalign$_{res101}$ [14]	83.7	88	83.4	58.3	68.8	90.3	92.3	83.7	47.4	91.7	28.1	76.3	77	76	71.4	**76.2**	80	59.5	62.3	63.9	75.8
NC-Net [21]	86.8	86.7	86.7	55.6	82.8	88.6	93.8	87.1	54.3	87.5	43.2	82	64.1	79.2	71.1	71	60	54.2	75	**82.8**	78.9
DCCNet [22]	87.3	**88.6**	82	**66.7**	84.4	89.6	**94**	**90.5**	64.4	**91.7**	51.6	84.2	**74.3**	**83.5**	72.5	72.9	60	68.3	**81.8**	81.1	**82.3**
Our method	**87.4**	87.9	**86.7**	58.3	**84.4**	**90.4**	93.8	89.9	**65.6**	89.6	51.6	**84.2**	72.5	83.2	**78.9**	69	60	62.9	79	82.2	82

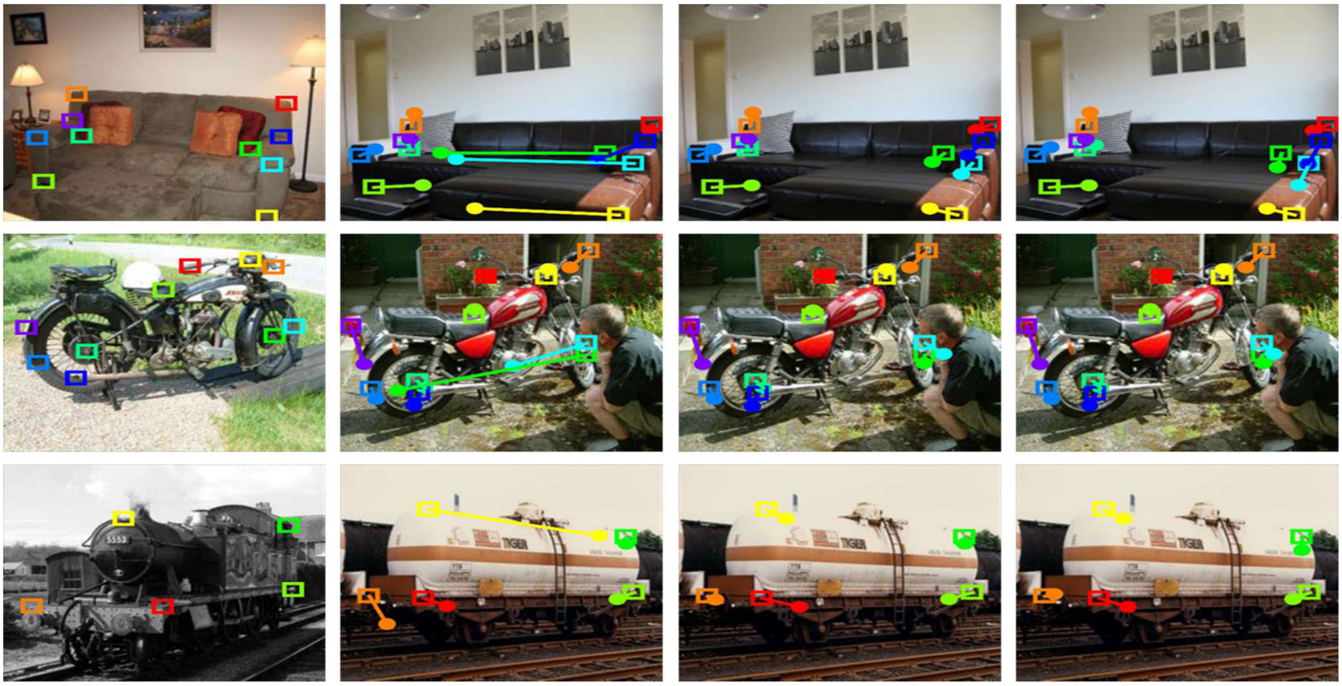

FIGURE 4. **Qualitative comparisons on the PF-PASCAL dataset. The leftmost column shows source images. The second, third, and fourth columns show predictions from NC-Net [21], DCCNet [22], and the proposed model, respectively. The ground truth and predicted key-points are indicated by squares and dots, respectively, with their distances in the target image representing matching error. It is seen from the figures that the proposed and DCCNet methods are more robust than NC-Net to repetitive patterns and intra-class variation.**

semantic flow approaches (Proposal Flow, UCN, SCNet), which indicates that further processing the correlation tensor with 4D convolutional kernels to filter out incorrect matches significantly improves the accuracy of the correspondence estimation. We compare the proposed model with other neighborhood consensus-based approaches (NC-Net, DCC-Net) and present a qualitative comparison between the results obtained using these approaches (Fig. 4). It is observed that DCCNet and our method are more robust against repetitive patterns than NC-Net, as both DCCNet and the proposed model use the global context information at each image location in conjunction with local semantic features, whereas NC-Net solely relies on local semantic features.

2) PF-WILLOW BENCHMARK

The PF-WILLOW [27] dataset contains 900 image pairs, selected from 100 images, with corresponding ground-truth bounding boxes. We computed the PCK scores with respect to bounding box size at multiple thresholds ($\alpha = 0.05$, 0.10, 0.15) and compared the PCK accuracies of the proposed method with those of the state-of-the-art semantic

TABLE 2. **Evaluation results obtained using PF-WILLOW [27] dataset.**

Method	α = 0.05	α = 0.10	α = 0.15
HOG+PF-LOM [26]	28.4	56.8	68.2
DCTM [47]	38.1	61	72.1
UCN-ST [20]	24.1	54	66.5
CAT-FCSS [36]	36.2	54.6	69.2
SCNet [12]	38.6	70.4	85.3
CNNGeo$_{res101}$ [13]	36.9	69.2	77.8
Weakalign$_{res101}$ [14]	38.2	71.2	85.8
RTN [19]	41.3	71.9	86.2
NC-Net [21]	44	72.7	85.4
DCCNet [22]	43.6	73.8	**86.5**
Our method	**44.88**	**73.81**	86.44

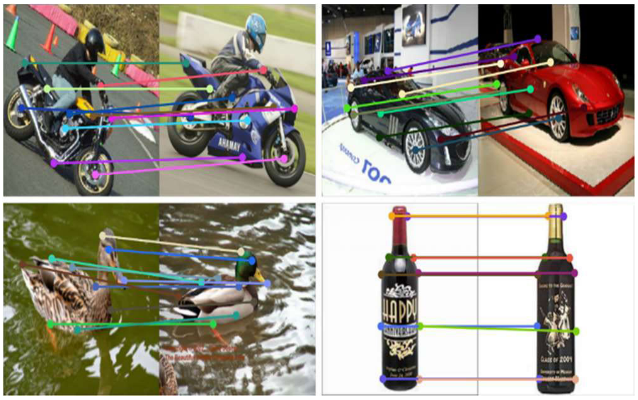

FIGURE 5. Qualitative results on PF-WILLOW [27] dataset.

correspondence methods (Table 2). It is seen that, for $\alpha = 0.05$ and 0.10, our method improves the PCK accuracies compared to the best- performing state-of-the-art approaches by 0.88% and 0.01% respectively. Furthermore, at $\alpha = 0.15$ our method achieves a competitive PCK of 86.44%, which is only 0.06% lower than that achieved by DCCNet. As shown by the results at $\alpha = 0.05$, our method has a more precise localization ability than previous semantic correspondence outlines. Fig 5 shows the qualitative obtained on the PF-WILLOW dataset.

3) TSS BENCHMARKS

We further evaluated the proposed model on the TSS benchmarks [28], which contain a total of 400 image pairs divided into three groups: FG3DCAR, JODS, and PASCAL. The TSS benchmarks provide dense flow fields obtained by interpolating sparse keypoint matches and co-segmentation masks for each image pair. Following the experimental protocol protocol used in [28], we computed the PCK over the foreground objects with respect to image size for $\alpha = 0.05$.

TABLE 3. Evaluation of results obtained using TSS [28] dataset.

Method	FG3D.	JODS	PASC.	avg.
HOG+PF-LOM [26]	78.6	65.3	53.1	65.7
HOG+TSS [28]	83	59.5	48.3	63.6
FCSS+SIFT Flow [36]	83	65.6	49.4	66
FCSS+PF-LOM [36]	83.9	63.5	58.2	68.5
HOG+OADSC [34]	87.5	70.8	72.9	77.1
FCSS+DCTM [47]	89.1	72.1	61	74
VGG-16+CNNGeo [13]	83.9	65.8	52.8	67.5
CNNGeo$_{\text{res101}}$ [13]	83.9	76.4	56.3	74.3
Weakalign$_{\text{res101}}$ [14]	90.3	76.4	56.5	74.4
RTN [19]	90.1	78.2	63.3	77.2
NC-Net [21]	94.5	81.4	57.1	77.7
DCCNet [22]	93.5	**82.6**	57.6	77.9
Our Method	**94.5**	82.2	**57.6**	**78.1**

Table 3 and Fig. 6 show the quantitative and qualitative results obtained on the TSS benchmarks, respectively. Our method achieves the same performance as a state-of-the-art result on FG3DCAR and a competitive PCK accuracy of 82.2% on JODS, which is only 0.4% less than the result obtained by DCCNet. Furthermore, our method improves on the state-of-the-art results in terms of average performance over the three groups on the TSS dataset. These TSS benchmarks results demonstrate our method's ability to generalize to novel datasets outside of the training domain.

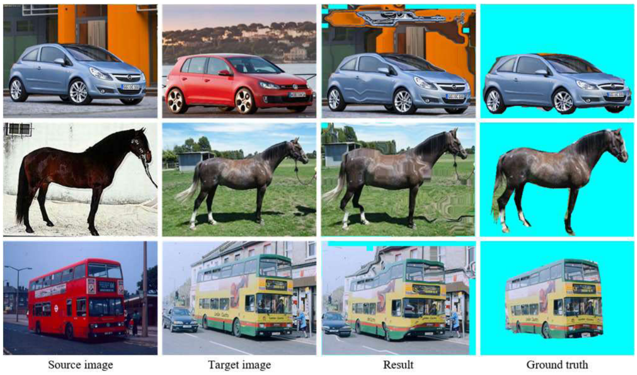

FIGURE 6. Qualitative results obtained using TSS [28] dataset.

TABLE 4. Runtime comparison of local region matching.

Approach	Model	PCK(α = 0.1)	Time(ms)
Local region matching	NC-Net	78.9	272
	DCCNet	82.3	1116.5
	Our method	82	280.9

C. COMPARISON WITH OTHER NEIGHBORHOOD CONSENSUS-BASED APPROACHES

1) RUNTIME COMPARISON

Table 4 compares the runtime performance of several NC-based methods. To enable a straight comparison, we implemented all source codes using PyTorch [43] and measured the average runtimes on the same machine with an NVIDIA Tesla P100 GPU. It is seen that DCCNet [22] requires more 4D convolution operations than NC-Net and our method to combine the local and context-aware feature convolutions. As 4D convolution itself requires a large amount of computation, this significantly degrades its runtime performance, as shown in Table 4. By contrast, our method requires only the use of additional 2D convolution operations to generate local-context fusion features. As 2D convolution is computationally much lighter than 4D convolution, it does not significantly slow runtime performance relative to DCCNet (Table 4).

2) TRAINING MEMORY USAGE COMPARISON

We evaluated the training memory requirements of NC-Net [21], DCCNet [22], and our model (Fig. 7); the number of training parameters are presented in Table 5. The training memory usages were measured in the same environment used for the runtime comparisons. We selected NC-Net

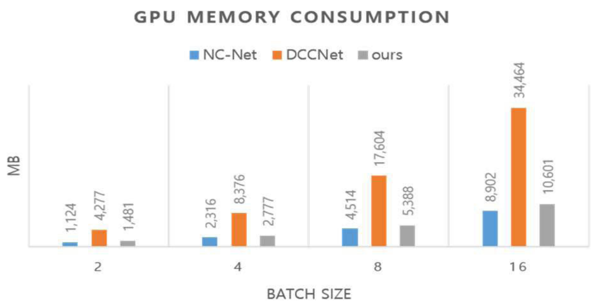

FIGURE 7. GPU training memory consumption as a function of batch size.

TABLE 5. Number of trainable parameters.

Method	#Parameters
NC-Net	0.18M
DCCNet	2.05M
Proposed model	1.86M

as the baseline because it trains only as a network with weakly supervised loss. Because the input and output tensor of each layer are temporarily stored on the memory for calculating the weight gradients via backpropagation, the dimensions of the input and output tensors considerably affect the training memory. Therefore, despite the small number of training parameters, NC-Net requires large amounts of training memory owing to processing high-dimensional 4D tensors. The results in Fig. 7 indicate that DCCNet requires approximately 370% more GPU memory than NC-Net at each batch size owing to storing more 4D tensors in the memory for its multi-auxiliary task loss and additional weights provided by the dynamic fusion network and spatial context encoder [22]. By contrast, our method requires only approximately 19% more GPU memory than NC-Net owing to its historical averaging loss term and additional weights provided by the LCF module. These results indicate that our training method is much more efficient than that used by DCCNet in terms of memory usage.

TABLE 6. Ablation study experimental results.

Method	Self-similarity	γ	Historical averaging	PCK(α=0.1)
NC-Net				78.9
GL_0	Global	Learnable, initialize to 0	×	79.71
SL_0	Semi-global	Learnable, initialize to 0	×	81.03
SC_1H	Semi-global	Constant value 1	✓	68.61
SL_1H	Semi-global	Learnable, initialize to 1	✓	78.1
SL_0H	Semi-global	Learnable, initialize to 0	✓	**82.0**

D. ABLATION STUDY

We conducted ablation studies on different components of the LCF module and on the losses used in our model. We selected NC-Net [21] as the baseline and evaluated the PCK ($\alpha = 0.1$) results obtained on the PF-PASCAL [26] test split (Table 6).

TABLE 7. Effect of σ in LCF module on PF-PASCAL [26] dataset.

σ	PCK(α=0.1)
3	81.7
7	82
12	80.38
25	80.37

1) SELF-SIMILARITY FEATURES

We conducted a series of experiments using different types of self-similarity feature to encode the context-aware features. Adding context-aware features encoded with global self-similarity features to local features (GL_0) improved the overall PCK by 0.81% relative to NC-Net and adding context-aware features encoded with semi-global self-similarity features to the local features (SL_0) further improved the performance by 1.32% relative to using context-aware features encoded with a global self-similarity feature (GL_0). These improvements indicate the effectiveness of using semi-global self-similarity features as a global context cue. Table 7 further lists the effects of semi-global self-similarity features with different σ values.

2) FUSION METHOD

We subsequently conducted experiments in which the settings of γ in Eq. (12) were varied to obtain different fuse local and context-aware features. As seen from Table 6, setting γ to a constant value of 1.0 for combining local and context-aware features (SC_1H) downgrades the performance significantly, producing an overall reduction in PCK of 10.29% relative to NC-Net. Furthermore, setting γ as a learnable scalar and initializing it to 1.0 (SL_1H) downgrades performance relative to NC-Net—in this case by an overall reduction in PCK of 0.8%—owing to unstable training at the early training stage. By contrast, combining local and context-aware features while using γ as a learnable scalar initialized to zero yields significantly better results (82.0%), illustrating the necessity of gradually adding context-aware features to local features to prevent the large degradation induced by unstable training.

3) HISTORICAL AVERAGING

To analyze the effects of historical averaging loss, we compared the model between being trained with and without an historical averaging term. It is seen from Table 6 that, with an historical averaging loss term (SL_0H), the proposed model obtains a PCK value 0.97% higher than that without such a term (SL_0), reaching an overall PCK of 82.0%. This result illustrates the effectiveness of historical averaging loss in regularizing the weakly supervised loss term in Eq. (21). With the addition of the historical averaging loss term, the NC module retains information learned at the beginning of the training phase, enabling it to combine local and context information more efficiently to achieve better semantic correspondence estimation.

V. CONCLUSION

In this paper, we introduced a CNN for semantic correspondence estimation. Our focus in developing this network was on efficiently incorporating the global context cue into local semantic features to achieve accurate semantic matching. To this end, we introduced a semi-global self-similarity feature for capturing the global context semantic cue while reducing the sensitivity to background clutter. We further proposed the LCF module, which effectively fuses the global context cue to local features. Finally, we applied historical averaging loss to train our network efficiently. We evaluated our method on the PF-PASCAL [26], PF-WILLOW [27], and TSS [28] benchmarks and compared its performance with that of prior methods [22] in terms of inference time and training memory usage. The results demonstrated that the proposed model is much faster and more memory-efficient than existing approaches and achieves comparable accuracy.

However, training the model with weak supervision can be easily overfitted, especially when training the model with a small dataset such as PF-PASCAL. The proposed method also requires a significant amount of computation to refine correlation maps with sequences of 4D convolution kernels. In future work, we will train the model with other spatial regularizers that enforce strong spatial constraints for semantic correspondence, such as a cycle-consistency constraint or smoothness constraint, to overcome the overfitting problem and improve the accuracy of the model. We will train the model with other training strategies such as self-supervised or semi-supervised training methods to train a model with stronger supervision than the weakly supervised method on a small dataset. We will also attempt to develop a simple method of refining the correlation map to reduce the computation needed for correspondence estimation. We will develop an effective way to reduce the size of the correlation map while maintaining the pixel-wise correlations between images and develop low complex kernels to filter out incorrect matches in the correlation map.

REFERENCES

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, Apr. 2002.

[2] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[3] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. ECCV*, 2004, pp. 25–36.

[4] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 41–48.

[5] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-, vol. filtering, for visual correspondence and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 504–511, Feb. 2013.

[6] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 353–363, Apr. 1993.

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

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 886–893.

[9] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2307–2314.

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

[11] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*. [Online]. Available: http://arxiv.org/abs/1602.07261

[12] K. Han, R. S. Rezende, B. Ham, K.-Y.-K. Wong, M. Cho, C. Schmid, and J. Ponce, "SCNet: Learning semantic correspondence," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1831–1840.

[13] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2553–2567, Nov. 2019.

[14] I. Rocco, R. Arandjelovic, and J. Sivic, "End-to-end weakly-supervised semantic alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6917–6925, doi: 10.1109/cvpr.2018.00723.

[15] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 6, 2020, doi: 10.1109/TPAMI.2020.2985395.

[16] P. H. Seo, J. Lee, D. Jung, B. Han, and M. Cho, "Attentive semantic alignment with offset-aware correlation kernels," in *Proc. ECCV*, 2018, pp. 367–383, doi: 10.1007/978-3-030-01225-0_22.

[17] J. Lee, D. Kim, W. Lee, J. Ponce, and B. Ham, "Learning semantic correspondence exploiting an object-level prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 3, 2020, doi: 10.1109/TPAMI.2020.3013620.

[18] Y.-C. Chen, P.-H. Huang, L.-Y. Yu, J.-B. Huang, M.-H. Yang, and Y.-Y. Lin, "Deep semantic matching with foreground detection and cycle-consistency," in *Proc. ACCV*, vol. 2019, pp. 347–362.

[19] S. Kim, S. Lin, S. R. Jeon, D. Min, and K. Sohn, "Recurrent transformer networks for semantic correspondence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 6126–6136.

[20] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2414–2422.

[21] I. Rocco, M. Cimpoi, R. Arandjelovic, A. Torii, T. Pajdla, and J. Sivic, "NCNet: Neighbourhood consensus networks for estimating image correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 14, 2020, doi: 10.1109/TPAMI.2020.3016711.

[22] S. Huang, Q. Wang, S. Zhang, S. Yan, and X. He, "Dynamic context correspondence network for semantic alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2010–2019, doi: 10.1109/iccv.2019.00210.

[23] X. Li, K. Han, S. Li, and V. Adrian Prisacariu, "Dual-resolution correspondence networks," 2020, *arXiv:2006.08844*. [Online]. Available: http://arxiv.org/abs/2006.08844

[24] I. Rocco, R. Arandjelovi, and J. Sivic, "Efficient neighbourhood consensus networks via submanifold sparse convolutions," 2020, *arXiv:2004.10566*. [Online]. Available: http://arxiv.org/abs/2004.10566

[25] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[26] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow: Semantic correspondences from object proposals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 7, pp. 1711–1725, Jul. 2018.

[27] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3475–3484.

[28] T. Taniai, S. N. Sinha, and Y. Sato, "Joint recovery of dense correspondence and cosegmentation in two images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4246–4255, doi: 10.1109/cvpr.2016.460.

[29] C. Liu, J. Yuen, and A. Torralba, "SIFT Flow: Dense correspondence across scenes and its applications," in *Proc. Dense Image Correspondences Comput. Vis.*, 2016, pp. 15–49, doi: 10.1007/978-3-319-23048-1_2.

[30] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.

[31] J. Hur, H. Lim, C. Park, and S. C. Ahn, "Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1392–1400.

[32] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[33] H. Yang, W.-Y. Lin, and J. Lu, "DAISY filter flow: A generalized discrete approach to dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3406–3413.

[34] F. Yang, X. Li, H. Cheng, J. Li, and L. Chen, "Object-aware dense semantic correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2777–2785.

[35] W. Qiu, X. Wang, X. Bai, A. Yuille, and Z. Tu, "Scale-space SIFT flow," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 1112–1119.

[36] S. Kim, D. Min, B. Ham, S. Lin, and K. Sohn, "FCSS: Fully convolutional self-similarity for dense semantic correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 581–595, Mar. 2019.

[37] J. Min, J. Lee, J. Ponce, and M. Cho, "Hyperpixel flow: Semantic correspondence with multi-layer neural features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3395–3404.

[38] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Self-supervised learning of geometrically stable features through probabilistic introspection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3637–3645, doi: 10.1109/cvpr.2018.00383.

[39] D. Novotny, D. Larlus, and A. Vedaldi, "AnchorNet: A weakly supervised network to learn geometry-sensitive features for semantic matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5277–5286.

[40] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[41] S. Li, K. Han, T. W. Costain, H. Howard-Jenkins, and V. Prisacariu, "Correspondence networks with adaptive neighbourhood consensus," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10196–10205.

[42] A. Vaswani, N. Shazeer, and N. Parmar, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[43] A. Paszke, S. Gross, S. Chintala, and G. Chanan. (2017). *Pytorch*. [Online]. Available: http://pytorch.org

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[46] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2878–2890, Dec. 2013.

[47] S. Kim, D. Min, S. Lin, and K. Sohn, "DCTM: Discrete-continuous transformation matching for semantic flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4529–4538, doi: 10.1109/iccv.2017.485.

HO-JUN LEE received the B.S. degree in electrical and electronics engineering from Chung-Ang University, Seoul, South Korea, in 2019, where he is currently pursuing the M.S. degree. His research interests include computer vision, deep reinforcement learning, and deep meta-learning.

HONG TAE CHOI received the bachelor's degree in engineering from the Department of Electronic Communication Engineering, Daelim University College. He is currently pursuing the M.S. degree in electrical and electronic engineering with Chung-Ang University, Seoul, South Korea. His research interests include computer vision, attention mechanism, and weakly supervised learning.

SUNG KYU PARK received the Ph.D. degree from The Pennsylvania State University, USA, in 2007. He was employed with the Korea Electronics Technology Institute (KETI), from 1997 to 2003, and Eastman Kodak Company, Rochester, NY, USA, from 2007 to 2008. He is currently a Professor with the School of Electrical and Electronics Engineering, Chung-Ang University. His current research interests include display technology and deep-learning based sensor networks.

HO-HYUN PARK received the B.S. and M.S. degrees from Seoul National University, in 1987 and 1995, respectively, and the Ph.D. degree in computer science and engineering from KAIST, in 2001. From 1987 to 2003, he was a Principal Engineer with Samsung Electronics. He is currently a Professor of Electrical and Electronics Engineering with Chung-Ang University. His research interests include big data, deep learning, machine vision, information security, and real time and embedded systems.