NCNet: Neighbourhood Consensus Networks for Estimating Image Correspondences

Ignacio Rocco, Mircea Cimpoi, Relja Arandjelovic, Akihiko Torii, Tomas Pajdla, Josef Sivic

NCNet: Neighbourhood Consensus Networks for Estimating Image Correspondences

Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic

Abstract—We address the problem of finding reliable dense correspondences between a pair of images. This is a challenging task due to strong appearance differences between the corresponding scene elements and ambiguities generated by repetitive patterns. The contributions of this work are threefold. First, inspired by the classic idea of disambiguating feature matches using semi-local constraints, we develop an end-to-end trainable convolutional neural network architecture that identifies sets of spatially consistent matches by analyzing neighbourhood consensus patterns in the 4D space of all possible correspondences between a pair of images without the need for a global geometric model. Second, we demonstrate that the model can be trained effectively from weak supervision in the form of matching and non-matching image pairs without the need for costly manual annotation of point to point correspondences. Third, we show the proposed neighbourhood consensus network can be applied to a range of matching tasks including both category- and instance-level matching, obtaining the state-of-the-art results on the PF, TSS, InLoc and HPatches benchmarks.

Index Terms—Neighbourhood consensus, geometric matching, image alignment, category-level matching.

1 Introduction

Finding visual correspondences is one of the fundamental image understanding problems with applications in 3D reconstruction [1], visual localization [2], [3] or object recognition [4]. In recent years, significant effort has gone into developing trainable image representations for finding correspondences between images under strong appearance changes caused by viewpoint or illumination variations [5], [6], [7], [8], [9], [10], [11], [12], [13]. However, unlike in other visual recognition tasks, such as image classification or object detection, where trainable image representations have become the *de facto* standard, the performance gains obtained by trainable features over the classic hand-crafted ones have been only modest at best [14].

One of the reasons for this plateauing performance could be the currently dominant approach for finding image correspondence based on matching *individual* image features. While we have now better local patch descriptors, the matching is still performed by variants of the nearest neighbour assignment in a feature space followed by separate disambiguation stages based on geometric constraints. This approach has, however, fundamental limitations. Imagine a scene with textureless regions or repetitive patterns, such as a corridor with almost textureless walls and only few distinguishing features. A small patch of an image, depicting a repetitive pattern or a textureless area, is indistinguishable from other portions of the image depicting the same repetitive or textureless pattern. Such matches will be either discarded [15] or incorrect. As a result, matching individual patch descriptors will often fail in such challenging situations.

In this work we take a different direction and develop a trainable neural network architecture that disambiguates such challenging situations by analyzing local neighbourhood patterns in a full set of dense correspondences. The intuition is the following: in order to disambiguate a match on a repetitive pattern, it is necessary to analyze a larger context of the scene that contains a unique non-repetitive feature. The information from this unique match can then be propagated to the neighbouring uncertain matches. In other words, the certain unique matches will *support* the close-by uncertain ambiguous matches in the image.

This powerful idea goes back to at least 1990s [16], [17], [18], [19], [20], and is typically known as *neighbourhood consensus* or more broadly as *semi-local constraints*. The neighbourhood consensus has been typically carried out on sparsely detected local invariant features as a filtering step performed *after* a hard assignment of features by nearest neighbour matching using the Euclidean distance in the feature space. Furthermore, the neighbourhood consensus has been evaluated by manually engineered criteria, such as a certain number of locally consistent matches [18], [19], [20], or consistency in geometric parameters including distances and angles between matches [16], [17].

In this work, we go one step further and propose a way of *learning* neighbourhood consensus constraints directly from training data. Moreover, we perform neighbourhood consensus *before* hard assignment of feature correspondence; that is, on the complete set of dense pair-wise matches. In

- *I. Rocco is with Inria and the Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, Paris, France. E-mail: ignacio.rocco@inria.fr.*
- *M. Cimpoi and T. Pajdla are with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague. E-mail: {mircea.cimpoi,pajdla}@cvut.cz.*
- *M. Cimpoi is currently at Oculus Zurich.*
- *R. Arandjelović is with DeepMind. E-mail: relja@google.com*
- *A. Torii, deceased, was with the Department of Mechanical and Control Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology, Tokyo 152-8550, Japan.*
- *J. Sivic is with Inria, the Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, Paris, France and with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague. E-mail: josef.sivic@inria.fr.*

this way, the decision on matching assignment is done only after taking into account the spatial consensus constraints, hence avoiding errors due to early matching decisions on ambiguous, repetitive or textureless matches.

Contributions. The contributions of this work are threefold: **First**, we develop a neighbourhood consensus network – a convolutional neural network architecture for dense matching that learns local geometric constraints between neighbouring correspondences without the need for a global geometric model. **Second**, we show that parameters of this network can be trained from scratch using a weakly supervised loss-function that requires supervision at the level of image pairs without the need for manually annotating individual correspondences. **Third**, we show that the proposed model is applicable to a range of matching tasks producing high-quality dense correspondences, achieving state-of-the-art results on both category- and instance-level matching benchmarks. Code, training data and models are available at `http://www.di.ens.fr/willow/research/ncnet/`.

This paper is an extended version of our previous work [21], with several improvements: (i) introducing a lightweight version of the model, which can be enabled without retraining, and allows for higher input feature resolution, (ii) providing a more in depth explanation of the model and implementation details, and (iii) extended evaluation on category- and instance-level matching, on several benchmarks: PF-PASCAL, Caltech, TSS, InLoc and HPatches, together with Ablation studies, to assess the contribution of the components of the proposed formulation.

2 Related work

This work relates to several lines of research, which we review below.

Matching with hand-crafted image descriptors. Traditionally, correspondences between images have been obtained by hand crafted local invariant feature detectors and descriptors [15], [22], [23] that were extracted from the image with a controlled degree of invariance to local geometric and photometric transformations. Candidate (tentative) correspondences were then obtained by variants of nearest neighbour matching. Strategies for removing ambiguous and non-distinctive matches include the widely used second nearest neighbour ratio test [15], or enforcing matches to be mutual nearest neighbours. Both approaches work well for many applications, but have the disadvantage of discarding many correct matches, which can be problematic for challenging scenes, such as indoor spaces considered in this work that include repetitive and textureless areas. While successful, handcrafted descriptors have only limited tolerance to large appearance changes beyond the built-in invariance.

Matching with trainable descriptors. The majority of trainable image descriptors are based on convolutional neural networks (CNNs) and typically operate on patches extracted using a feature detector such as DoG [15], yielding a sparse set of descriptors [5], [6], [9], [10], [11], [12] or use a pre-trained image-level CNN feature extractor [24], [25]. Others have recently developed trainable methods that comprise both feature detection and description [13], [24], [26]. The extracted descriptors are typically compared using the Euclidean distance, but an appropriate similarity score can be also learnt in a discriminative manner [7], [8], where a trainable model is used to both extract descriptors and produce a similarity score. Finding matches consistent with a geometric model is typically performed in a separate post-processing stage [5], [6], [9], [10], [11], [12], [13], [24], [26], [27].

Trainable image alignment. Recently, end-to-end trainable methods have been developed to produce correspondences between images according to a parametric geometric model, such as an affine, perspective or thin-plate spline transformation [28], [29]. In addition, Recurrent Transformer Nets (RTN) [30] employ locally-varying affine deformation fields. In these works, all pairwise feature matches are computed and used to estimate the geometric transformation parameters using a CNN. Unlike previous methods that capture only a sparse set of correspondences, this geometric estimation CNN captures interactions between a full set of dense correspondences. However, these methods currently only estimate a low complexity parametric transformation, and therefore their application is limited to only coarse image alignment tasks. In contrast, we target a more general problem of identifying reliable correspondences between images of a general 3D scene. Our approach is not limited to a low dimensional parametric model, but outputs a generic set of locally consistent image correspondences, applicable to a wide range of computer vision problems ranging from category-level image alignment to camera pose estimation. The proposed method builds on the classical ideas of neighbourhood consensus, which we review next.

Match filtering by neighbourhood consensus. Several strategies have been introduced to decide whether a match is correct or not, given the supporting evidence from the neighbouring matches. The early examples analyzed the patterns of distances [16] or angles [17] between neighbouring matches. Later work simply counts the number of consistent matches in a certain image neighbourhood [18], [19], which can be built in a scale invariant manner [31], using a regular image grid [20], or an adaptive neighbourhood size by considering a certain number of nearby matches [32]. While simple, these techniques have been remarkably effective in removing random incorrect matches and disambiguating local repetitive patterns [31]. Inspired by this simple yet powerful idea we develop a neighbourhood consensus network – a convolutional neural architecture that (i) analyzes the *full set of dense matches* between a pair of images and (ii) *learns* patterns of locally consistent correspondences directly from data.

Other modern match filtering methods. While the idea of using neighbourhood consensus to remove outlier matches dates back to the 1990s [16], [17], it is still an actively researched topic. Recently, Bian et al. [20] proposed the Grid-based Motion Statistics (GMS) approach, where the images to be matched are partitioned into a set of cells and the number of matches between each cell are used to distinguish inliers from outliers. Also, Ma et al. [32] propose the Locality Preserving Matching (LPM) approach, where the sizes of the neighbourhoods are not explicitly defined, but rather inferred by nearest-neighbour search, and which proposes an optimization scheme to determine inliers by

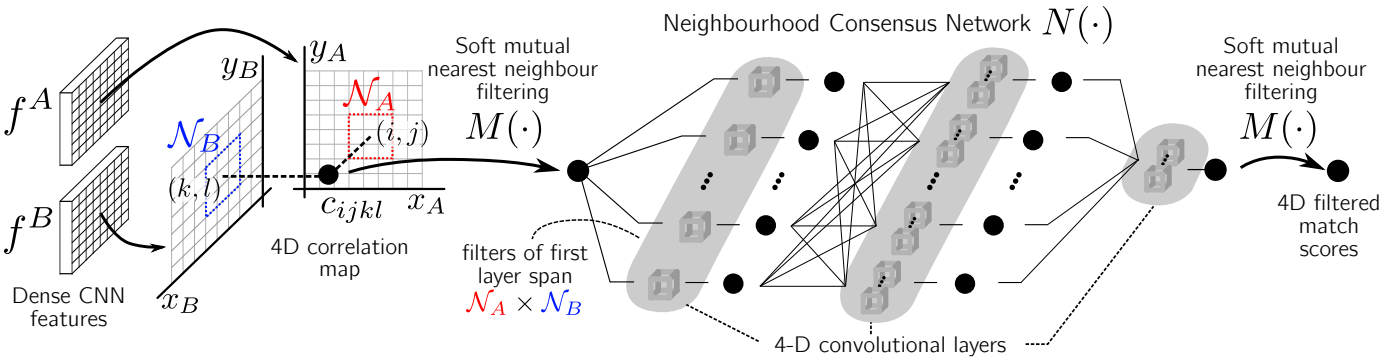


Fig. 1: *Overview of the proposed method.* A fully convolutional neural network is used to extract dense image descriptors f^A and f^B for images I_A and I_B, respectively. Scores for all pairs of individual feature matches f^A_{ij} and f^B_{kl} are stored in the 4-D correlation map c_{ijkl} (here shown as a 3-D illustration). These matches are further processed by the proposed soft-nearest neighbour filtering and neighbourhood consensus network to produce the final set of output correspondences.

minimizing a global cost function. While these methods build on the idea of neighbourhood consensus, they are manually engineered and have no trainable parameters. Our proposed neighbourhood consensus network seeks to combine the power of the neighbourhood consensus approach with that of trainable convolutional neural networks. Other recent trainable methods for match filtering have also been proposed [33], [34], [35], although they are particularly focused on robust estimation of the essential and fundamental matrices of two-view geometry. In particular, the Context Normalization Network (CNe) of Yi et al. [33] combines a deep fully connected network (MLP) that operates on the level of individual 4D match coordinates with a Context Normalization (CN) Layer which acts at a global, image level. We compare our proposed method with GMS [20], LPM [32] and CNe [33].

Flow and disparity estimation. Related are also methods that estimate optical flow or stereo disparity such as [36], [37], [38], [39], [40], or their trainable counterparts [41], [42], [43]. These works also aim at establishing reliable point to point correspondences between images. However, we address a more general matching problem where images can have large viewpoint changes (indoor localization) or major changes in appearance (category-level matching). This is different from optical flow where image pairs are usually consecutive video frames with small viewpoint or appearance changes, and stereo where matching is often reduced to a local search around epipolar lines. The optical flow and stereo problems are well addressed by specialized methods that explicitly exploit the problem constraints (such as epipolar line constraint, small motion, smoothness, etc.).

3 Proposed Approach

In this work, we combine the robustness of neighbourhood consensus filtering with the power of trainable neural architectures. We design a model which learns to discriminate a reliable match by recognizing patterns of supporting matches in its neighbourhood. Furthermore, we do this in a fully differentiable way, such that this trainable matching module can be directly combined with strong CNN image descriptors. The resulting pipeline can then be trained in an end-to-end manner for the task of feature matching. An overview of our proposed approach is presented in Fig. 1. There are five main components: (i) dense feature extraction and matching, (ii) the neighbourhood consensus network, (iii) a soft mutual nearest neighbour filtering, (iv) extraction of correspondences from the output 4D filtered match tensor, and (v) weakly supervised training loss. These components are described next.

3.1 Dense feature extraction and matching

In order to produce an end-to-end trainable model, we follow the common practice of using a deep convolutional neural network (CNN) as a dense feature extractor.

Then, given an image I, this feature extractor will produce a dense set of descriptors, $\{\mathbf{f}^I_{ij}\} \in \mathbb{R}^d$, with indices $i = 1, \dots, h$ and $j = 1, \dots, w$, and (h, w) denoting the number of features along image height and width (*i.e.* the spatial resolution of the features), and d the dimensionality of the features.

While classic hand-crafted neighbourhood consensus approaches are applied *after* a hard assignment of matches is done, this is not well suited for developing a matching method that is differentiable and amenable for end-to-end training. The reason is that the step of selecting a particular match is not differentiable with respect to the set of all the possible features. In addition, in case of repetitive features, assigning the match to the first nearest neighbour might result in an incorrect match, in which case the hard assignment would lose valuable information about the subsequent closest neighbours.

Therefore, in order to have an approach that is amenable to end-to-end training, all pairwise feature matches need to be computed and stored. For this we use an approach similar to [28]. Given two sets of dense feature descriptors $\mathbf{f}^A = \{\mathbf{f}^A_{ij}\}$ and $\mathbf{f}^B = \{\mathbf{f}^B_{ij}\}$ corresponding to the images to be matched, the exhaustive pairwise cosine similarities between them are computed and stored in a 4-D tensor $\mathbf{c} \in \mathbb{R}^{h \times w \times h \times w}$ referred to as *correlation map*, where:

$$\mathbf{c}_{ijkl} = \frac{\langle \mathbf{f}^A_{ij}, \mathbf{f}^B_{kl} \rangle}{\|\mathbf{f}^A_{ij}\|_2 \|\mathbf{f}^B_{kl}\|_2}. \quad (1)$$

Note that, by construction, elements of $\mathbf{c}$ in the vicinity of index $ijkl$ correspond to matches between features that are

in the local neighbourhoods $\mathcal{N}_A$ and $\mathcal{N}_B$ of descriptors $\mathbf{f}^A_{ij}$ in image A and $\mathbf{f}^B_{kl}$ in image B, respectively, as illustrated in Fig. 1; this structure of the 4-D correlation map tensor $\mathbf{c}$ will be exploited in the next section.

3.2 Neighbourhood consensus network

The correlation map contains the scores of *all* pairwise matches. In order to further process and filter the matches, we propose to use 4-D convolutional neural network (CNN) for the neighbourhood consensus task (denoted by $N(\cdot)$), which is illustrated in Fig. 1.

Determining the correct matches from the correlation map is, *a priori*, a significant challenge. Note that the number of correct matches are of order of hw, while the size of the correlation map is of the order of $(hw)^2$. This means that the great majority of the information in the correlation map corresponds to *matching noise* due to incorrectly matched features.

However, supported by the idea of neighbourhood consensus presented in Sec. 1, we can expect correct matches to have a coherent set of supporting matches surrounding them in the 4-D space. These geometric patterns are equivariant with translations in the input images; that is, if the images are translated, the matching pattern is also translated in the 4-D space by an equal amount. This property motivates the use of 4-D convolutions for processing the correlation map as the same operations should be performed regardless of the location in the 4-D space. This is analogous to the motivation for using 2-D convolutions to process individual images – it makes sense to use convolutions, instead of for example a fully connected layer, in order to profit from weight sharing and keep the number of trainable parameters low. Furthermore, it facilitates sample-efficient training as a single training example provides many error signals to the convolutional weights, since the *same* weights are applied at all positions of the correlation map. Finally, by processing matches with a 4D convolutional network we establish a strong locality prior on the relationships between the matches. That is, by design, the network will determine the quality of a match by examining only the information in a local 2D neighbourhood in each of the two images.

The proposed neighbourhood consensus network has several convolutional layers, as illustrated in Fig. 1, each followed by ReLU non-linearities. The convolutional filters of the first layer of the proposed CNN span a local 4-D region of the matches space, which corresponds to the Cartesian product of local neighbourhoods $\mathcal{N}_A$ and $\mathcal{N}_B$ in each image, respectively. Therefore, each 4-D filter of the first layer can process and detect patterns in all pairwise matches of these two neighbourhoods. This first layer has N_1 filters that can specialize in learning different local geometric deformations, producing N_1 output channels, that correspond to the agreement with these local deformations at each 4-D point of the correlation tensor. These output channels are further processed by subsequent 4-D convolutional layers. The aim is that these layers capture more complex patterns by combining the outputs from the previous layer, analogously to what has been observed for 2-D CNNs [44]. Finally, the neighbourhood consensus CNN produces a single channel output, which has the same dimensions as the 4D input matches.

To make the method invariant to the particular order of the input images, that is, that it will produce the same matches regardless of whether an image pair is input to the net as (I^A, I^B) or (I^B, I^A), we define the following *symmetric* version of the network N by applying it twice in the following way:

$$S(\mathbf{c}) = N(\mathbf{c}) + \big(N(\mathbf{c}^{\mathrm{T}})\big)^{\mathrm{T}}, \tag{2}$$

where $\mathbf{c}$ is the correlation map defined in (1) and by $\mathbf{c}^{\mathrm{T}}$ we mean swapping the pairs of dimensions corresponding to the first and second images: $\big(\mathbf{c}^{\mathrm{T}}\big)_{ijkl} = \mathbf{c}_{klij}$. This final output constitutes the *filtered matches* using the neighbourhood consensus network, where matches with inconsistent *local* patterns are downweighted or removed. Further filtering can be done by means of a *global* filtering strategy, as presented next.

3.3 Soft mutual nearest neighbour filtering

Although the proposed neighbourhood consensus network can suppress and amplify matches based on the supporting evidence in their neighbourhoods – that is, at a semi-local level – it cannot enforce global constraints on matches, such as being a *reciprocal* match, where matched features are required to be mutual nearest neighbours:

$$(\mathbf{f}^A_{ab}, \mathbf{f}^B_{cd}) \text{ mutual N.N.} \iff \begin{cases} (a,b) = \arg\min_{ij} \|\mathbf{f}^A_{ij} - \mathbf{f}^B_{cd}\| \\ (c,d) = \arg\min_{kl} \|\mathbf{f}^A_{ab} - \mathbf{f}^B_{kl}\|. \end{cases} \tag{3}$$

Filtering the matches by imposing the hard mutual nearest neighbour condition expressed by (3) would eliminate the great majority of candidate matches, which makes it unsuitable for usage in an end-to-end trainable approach, as this hard decision is non-differentiable.

We therefore propose a softer version of the mutual nearest neighbour filtering ($M(\cdot)$), both in the sense of *softer decision* and *better differentiability properties*, that can be applied on dense 4-D match scores:

$$\mathbf{c}' = M(\mathbf{c}), \quad \text{where} \quad \mathbf{c}'_{ijkl} = \mathbf{r}^A_{ijkl}\mathbf{r}^B_{ijkl}\mathbf{c}_{ijkl}, \tag{4}$$

and $\mathbf{r}^A_{ijkl}$ and $\mathbf{r}^B_{ijkl}$ are the ratios of the score of the particular match $\mathbf{c}_{ijkl}$ with the best scores along each pair of dimensions corresponding to images A and B respectively:

$$\mathbf{r}^A_{ijkl} = \frac{\mathbf{c}_{ijkl}}{\max_{ab} \mathbf{c}_{abkl}}, \quad \text{and} \quad \mathbf{r}^B_{ijkl} = \frac{\mathbf{c}_{ijkl}}{\max_{cd} \mathbf{c}_{ijcd}}. \tag{5}$$

This soft mutual nearest neighbour filtering operates as a gating mechanism on the input, downweighting the scores of matches that are not mutual nearest neighbours. Note that the proposed formulation is indeed a *softer* version of the mutual nearest neighbours criterion as $\mathbf{c}'_{ijkl}$ equals the matching score $\mathbf{c}_{ijkl}$ iff $(\mathbf{f}^A_{ij}, \mathbf{f}^B_{kl})$ are mutual nearest neighbours, and is decreased to a value in $[0, \mathbf{c}_{ijkl})$ otherwise. On the contrary, the "hard" mutual nearest neighbour matching would assign $\mathbf{c}'_{ijkl} = 0$ in the latter case.

While this filtering step has no trainable parameters, it can be inserted in the CNN pipeline at both training and evaluation stages, and it will help to enforce the global *reciprocity* constraint on matches. In the proposed approach, the soft mutual nearest neighbour filtering is used to filter both the correlation map, as well as the output of the neighbourhood consensus CNN, as illustrated in Fig. 1.

3.4 Lightweight model

Given the correlation tensor $\mathbf{c}$, and the previously defined symmetric neighbourhood-consensus network (S) and soft mutual nearest neighbour filtering (M) operations, the full proposed method can be expressed as:

$$\text{NCNet}(\mathbf{c}) = (M \circ S \circ M)(\mathbf{c}). \tag{6}$$

However, due to memory requirements, one might prefer to use *lighter-weight* neighbourhood consensus network N instead of its symmetric version S:

$$\text{L-NCNet}(\mathbf{c}) = (M \circ N \circ M)(\mathbf{c}), \tag{7}$$

Due to the lower memory requirements, L-NCNet is useful for running the network on higher resolution images. Note that switching from NCNet to the lightweight L-NCNet simply results in neglecting the second term of (2) and can be done without retraining.

3.5 Extracting correspondences from the correlation map

Suppose that we want to match two images I^A and I^B, whose *raw* correlation map is $\mathbf{c}$. Then, the output of our model $\mathbf{c}' = \text{NCNet}(\mathbf{c})$ is a 4-D *filtered* correlation map, which contains (filtered) scores for all pairwise matches. However, for various applications, such as image warping, geometric transformation estimation, pose estimation, visualization, etc, it is desirable to obtain a set of point-to-point image correspondences between the two images. To achieve this, a hard assignment can be performed in either of two possible directions, from features in image A to features in image B, or vice versa.

For this purpose, two scores are defined from the correlation map, by performing soft-max in the dimensions corresponding to images A and B:

$$\mathbf{s}^A_{ijkl} = \frac{\exp(\mathbf{c}'_{ijkl})}{\sum_{ab}\exp(\mathbf{c}'_{abkl})} \quad \text{and} \quad \mathbf{s}^B_{ijkl} = \frac{\exp(\mathbf{c}'_{ijkl})}{\sum_{cd}\exp(\mathbf{c}'_{ijcd})}. \tag{8}$$

Note that the scores are: (i) positive, (ii) normalized using the soft-max function, which makes $\sum_{ab}\mathbf{s}^B_{ijab} = 1$. Hence we can interpret them as discrete conditional probability distributions of $\mathbf{f}^A_{ij}, \mathbf{f}^B_{kl}$ being a match, given the position (i,j) of the match in A or (k,l) in B. If we denote (I,J,K,L) the discrete random variables indicating the position of a match (*a priori* unknown), and (i,j,k,l) the particular position of a match, then:

$$\begin{aligned} &\mathbb{P}\left(K=k, L=l \mid I=i, J=j\right) = \mathbf{s}^B_{ijkl}, \quad \text{and} \\ &\mathbb{P}\left(I=i, J=j \mid K=k, L=l\right) = \mathbf{s}^A_{ijkl}. \end{aligned} \tag{9}$$

Then, the hard-assignment in one direction can be done by just taking the most likely match (the mode of the distribution) as follows:

$\mathbf{f}^A_{ij}$ matches $\mathbf{f}^B_{kl}$ with score ρ^A_{ij} iff:

$$\begin{aligned} &(k,l) = \arg\max_{cd} \mathbb{P}\left(K=c, L=d \,|\, I=i, J=j\right) = \arg\max_{cd} \mathbf{s}^B_{ijcd}, \\ &\text{with} \quad \rho^A_{ij} := \mathbf{s}_{ijkl}. \end{aligned} \tag{10}$$

The matches of $\mathbf{f}^B_{kl}$ over $\mathbf{f}^A$ are obtained analogously.

This probabilistic intuition allows us to model the match uncertainty using a probability distribution and will be also useful to motivate the loss used for weakly-supervised training, which will be described next.

3.6 Weakly-supervised training

In this section we define the loss function used to train our network. One option is to use a strongly-supervised loss, but this requires dense annotations consisting of all pairs of corresponding points for each training image pair. Obtaining such exhaustive ground-truth is complicated – dense manual annotation is impractical, while sparse annotation followed by an automatic densification technique typically results in imprecise and erroneous training data. Another alternative is to resort to synthetic imagery which would provide point correspondences by construction, but this has the downside of making it harder to generalize to larger appearance variations encountered in real image pairs we wish to handle. Therefore, it is desirable to be able to train directly from pairs of real images, and using as little annotation as possible.

For this we propose to use a training loss that only requires a weak-level of supervision consisting of annotation on the level of image pairs. These training pairs (I^A, I^B) can be of two types, positive pairs, labelled with $y = +1$, or negative pairs, labelled with $y = -1$. Then, the following loss function is proposed:

$$\mathcal{L}(I^A, I^B) = -y\left(\bar{\rho}^A + \bar{\rho}^B\right), \tag{11}$$

where $\bar{\rho}^A$ and $\bar{\rho}^B$ are the mean matching scores over all hard assigned matches as per (10) of a given image pair (I^A, I^B) in both matching directions.

Note that the minimization of this loss maximizes the scores of positive and minimizes the scores of negative image pairs, respectively. As explained in 3.5, the hard-assigned matches correspond to the modes of the distributions of (9). Therefore, maximizing the score forces the distribution towards a Kronecker delta distribution, having the desirable effect of producing well-identified matches in positive image pairs. Similarly, minimizing the score forces the distribution towards the uniform one, weakening the matches in the negative image pairs. Note that while the only scores that directly contribute to the loss are the ones coming from hard-assigned matches, all matching scores affect the loss because of the normalization in (8). Therefore, all matching scores will be updated at each training step.

3.7 Feature relocalization

The localization precision of the extracted features $\mathbf{f}^I_{ij}$ depends on the spatial resolution $h \times w$ of the dense feature map $\mathbf{f}^I$. For some tasks, such as pose estimation, precisely localized features are needed. However, in some cases, given hardware constraints, one cannot increase the spatial resolution $h \times w$ to obtain the required precision, as increasing h and w by a factor of two results in a sixteen-fold increase in the memory consumption and computation time of the whole NCNet model. Therefore, we devise a method to

increase the localization precision, with a less severe impact on the memory consumption and computation time.

In this approach, the correlation map $\mathbf{c}$ from (1) is computed with higher resolution features $2h \times 2w$ leading to a $2h \times 2w \times 2h \times 2w$ correlation map. This correlation map $\mathbf{c}$ is then downsampled to resolution $h \times w \times h \times w$ before further processing by the neighbourhood consensus network. Note that by doing this the memory requirements of the correlation tensor are still increased by a factor of 16, but the memory requirements of the 4D convolutional network are kept constant. The downsampling is performed by a 4-D max-pooling operation, with the kernel of size 2:

$$\mathbf{c}'_{abcd} = \max_{i\in[2a,2a+1],j\in[2b,2b+1],k\in[2c,2c+1],l\in[2d,2d+1]} \mathbf{c}_{ijkl}. \tag{12}$$

The downsampled correlation map $\mathbf{c}'$ is then processed and used to compute the final matches, which are localized with a precision given by the downsampled resolution $h \times w$. However, one can *re-localize* these features, and reduce the localization error, by simply using the positions of the features that yielded the locally maximal correlation value in the 4-D max-pooling operation given by (12). In other words, for a match $(\mathbf{f}^A_{ab}, \mathbf{f}^B_{cd})$, the final *re-localized* feature positions (a', b') and (c', d') are computed by:

$$a', b', c', d' = \underset{i\in[2a,2a+1],j\in[2b,2b+1],k\in[2c,2c+1],l\in[2d,2d+1]}{\arg\max} \mathbf{c}_{ijkl}. \tag{13}$$

Note that a similar approach was used in [45] for upsampling feature maps for the task of semantic segmentation, and in [46] for feature localization.

4 Experimental results

The proposed approach was evaluated on both category and instance-level matching problems. The same approach is used to obtain reliable matches for both types of matching problems, which are then used to solve the different task proposed by each particular benchmark.

4.1 Category-level matching

The proposed method was evaluated on the task of category-level matching, where, given two images containing different instances from the same category (*e.g.* two different cat images) the goal is to match or align the similar semantic parts. Three different standard benchmarks were used and evaluated using their respective metrics. These benchmarks will be presented next.

Proposal Flow. The Proposal Flow benchmark was used for evaluating the task of semantic keypoint transfer, where given annotated keypoints in the source image the task is to determine their positions in the target image. Both PF-Pascal and PF-Willow variants of the PF dataset [47] were used, which respectively contain 1251 and 900 semantically related image pairs annotated with sparse keypoints. The performance is measured using the percentage of correct keypoints (PCK), that is, the number of correctly matched annotated keypoints within a tolerance threshold of the ground-truth position. In both cases, the evaluation protocol of [29], [48] is followed, where the PCK is computed using normalized keypoint coordinates $(\hat{x}, \hat{y}) = (x/w, y/h)$ with (h, w) being the image resolution and the normalized distance threshold $\hat{L} = \alpha$. Note that [28], [47] used slightly different definitions of PCK, so in order to make a fair comparison, for these methods we report the results from [30], [48] which re-evaluated them using the presented evaluation procedure.

Caltech-101. The Caltech-101 [49] dataset was used for evaluating the task of label transfer, which consists of transferring the semantic segmentation labels of the source image onto the target image. The same evaluation data and procedure as in [47], [50] was used, which includes 1515 evaluation pairs. The alignment accuracy is quantitatively measured using the label transfer accuracy (LT-ACC), which measures the alignment correctness of both foreground and background labels; and the Jaccard similarity coefficient (IoU), which only measures the alignment correctness of the foreground object. The previously employed localization error metric (LOC-ERR) is not considered here as it was shown to be unrepresentative of the alignment quality [29]. For the mathematical definitions of these metrics we refer the reader to [50]. For qualitative evaluation, the output aligned images are presented, which provides better qualitative assessment of the alignment than visualizing the transferred segmentation masks.

TSS. Finally, we report results on the TSS dataset [51] which consists of 400 semantically related image pairs which are subdivided into three subgroups: FG3D, JODS and PASCAL. By employing a semi-automated method requiring human intervention, dense ground-truth flow maps were computed for each image pair, which enables dense evaluation of the alignment, in contrast to sparse keypoints or segmentation masks used in the previous two cases. The metric employed to assess the dense alignment is also the percentage of correct keypoints (PCK) – but evaluated densely and not sparsely – with distance threshold $L = \alpha \max(h, w)$, where (h, w) is the target image resolution.

Training. From the three category-level benchmarks, only the PF-Pascal provides a training split [48], which divides the data into approximately 700 pairs for training, 300 for validation and 300 for testing. In order to train the network in a weakly-supervised manner using the proposed loss (11), the 700 training pairs of PF-Pascal are used as positive training pairs, and negative pairs are generated by randomly pairing images of different categories, such as a car with a dog image. The same model trained on the PF-Pascal training split was used for evaluation on all the category-level benchmarks.

Results. Quantitative results are presented in Table 1. The proposed neighbourhood consensus network (NCNet) obtains state-of-the-art results in several of the evaluated benchmarks. Qualitative examples of the semantic keypoint transfer on the PF dataset are shown in Fig. 2. Additional qualitative examples of dense alignment on the Caltech-101 and TSS datasets are presented in Fig. 3. These qualitative results demonstrate how our approach can correctly match semantic object parts in challenging situations with large changes of appearance and non-rigid geometric deformations. Note that the dense alignments in Fig. 3 are obtained directly by bilinear interpolation of the matches outputted

TABLE 1: Results for semantic matching on different datasets.

	PF-Pascal	PF-Willow	Caltech		TSS			
Method	PCK ($\alpha = 0.1$)	PCK ($\alpha = 0.1$)	IOU	LT-ACC	FG3D	JODS	PASCAL	avg.
HOG+PF-LOM [47]	62.5	56.8	0.50	0.78	0.786	0.653	0.531	0.657
SCNet-AG+ [48]	72.2	70.4	0.51	0.79	0.776	0.608	0.474	0.619
CNNGeo [28]	71.9	81.1	0.61	0.83	0.901	0.764	0.563	0.743
WeakAlign [29]	75.8	**84.3**	0.63	0.85	0.903	0.764	0.565	0.744
RTN [30]	75.9	71.9	**0.64**	**0.87**	0.901	0.782	**0.633**	0.772
NCNet ($r = 100$, FR)	**78.9**	**84.3**	0.62	0.85	**0.945**	**0.814**	0.571	**0.777**

We evaluate on the tasks of keypoint transfer (PF-Pascal/Willow), label transfer (Caltech) and dense alignment (TSS).

by NCNet, not requiring the use of any global geometric model or regularization technique, in contrast with other methods.

4.2 Instance-level matching

Next we show that our method is also suitable for instance level matching and evaluate it on two different benchmarks, (i) HPatches [57], consisting mostly of pictures of outdoor planar scenes, paintings or printed photographs and, (ii) InLoc [3], consisting in indoor images taken at different times. While in both benchmarks the image pairs contain strong variations of illumination conditions and viewpoint, the InLoc dataset is particularly challenging as indoor spaces are often self-similar and contain large textureless areas. Furthermore, while the HPatches dataset allows for a direct evaluation of the matching accuracy, the InLoc dataset evaluates the matching task as a module in an indoor visual localization pipeline, where the goal is to estimate an accurate 6DoF camera pose of a query photograph given a large-scale 3D model of a building.

HPatches. We employ the HPatches benchmark [57] to directly evaluate the matching accuracy in the instance-level matching case. The dataset contains 116 sequences, each belonging to a different planar scene and each containing 6 images which are used to form 5 image pairs. These sequences are divided into the *illumination* (57 sequences) and *viewpoint* (59 sequences) subsets, which only contain changes along these factors of variation. The dataset provides a ground-truth homography transformation for each image pair, which is used to assess the accuracy of the extracted matches. Two different measures are used. The first one is the mean matching accuracy (MMA), following the standard evaluation protocol for this dataset [56], inspired by [58]:

$$\text{MMA}\big(\{(p_i^A, p_i^B)\}_{i=1}^N; t\big) = \frac{\sum_{i=1}^N \mathbb{1}_{>0}\big(t - \|\mathcal{T}(p_i^A; H_{GT}) - p_i^B\|_2\big)}{N}, \quad (14)$$

where $\{(p_i^A, p_i^B)\}_{i=1}^N$ is the set of matches between points p_i^A and p_i^B in images A and B respectively, $\mathcal{T}(\cdot, H_{GT})$ is the transformation with the ground-truth homography H_{GT}, $\mathbb{1}_{>0}$ is the indicator function for positive numbers and t is the distance threshold parameter. Similarly to PCK, MMA measures the proportion of matches that are correct up to a certain tolerance threshold t, but contrary to PCK, t is defined as an absolute number of pixels in the original image resolution. The MMA directly evaluates the matching accuracy. The second evaluation metric assesses whether the matches can be used to accurately estimate the homography transformation between each image pair. For this, the obtained matches are used to estimate the homography matrix $\hat{H}$ by running a modern RANSAC variant [59], [60]. The average transfer error (TE) of the estimated homography $\hat{H}$ with respect to the ground-truth homography H_{GT} is computed as:

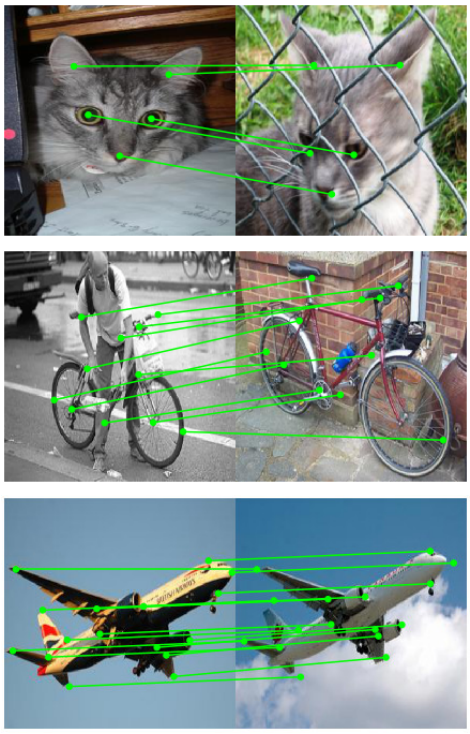
$$\text{TE}\big(\hat{H}; t\big) = \frac{\sum_{p^A \in \Omega} \|\mathcal{T}(p^A; H_{GT}) - \mathcal{T}(p^A; \hat{H})\|_2}{|\Omega_A|}, \quad (15)$$

where Ω_A is the set of all pixels coordinates over image A, $|\Omega_A|$ the number of pixels of image A, $\mathcal{T}(\cdot, H_{GT})$ is the transformation with the ground-truth homography H_{GT} and $\mathcal{T}(\cdot, \hat{H})$ is the transformation with the estimated homography $\hat{H}$. We use the transfer error to evaluate the quality of $\hat{H}$ as it represents a meaningful geometric distance which is measured in pixels and is invariant to the homography parametrization, which is not the case if the error is computed directly on the homography matrix entries.

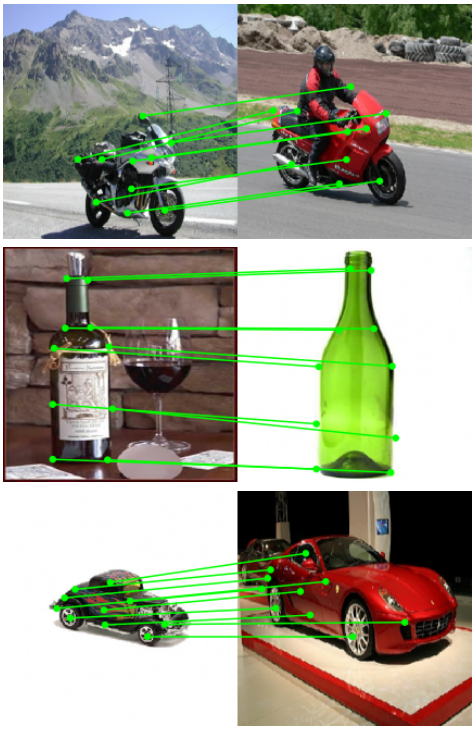
InLoc. We use the InLoc dataset [3], which consists of 10K database images (perspective cutouts) extracted from 227 RGBD panoramas, and an additional set of 356 query images captured with a smart-phone camera at a different time (several months later) from the database images. Here, the goal is to estimate an accurate 6DoF camera pose of a query photograph within a large-scale 3D model of a building. We follow the same evaluation protocol as in [3] and report the percentage of correctly localized queries at a given camera position error threshold.

Training. As both the HPatches and the InLoc were designed for evaluation and do not provide a training set, we collected an Indoor Venues Dataset (IVD) [21], consisting of user-uploaded photos, captured at public places such as restaurants, cafes, museums or cathedrals, by crawling Google Maps. It features not only viewpoint and illumination changes, such as the variations present in the HPatches dataset, but also scene modifications due to the passage of time as in the InLoc dataset. The IVD dataset contains 3861 positive image pairs from 89 different venues in 6 different cities, split into *train*: 3481 pairs (80 places) and *validation*: 380 pairs (from the remaining 9 places). The same model trained on IVD was used for evaluation on both the HPatches and InLoc benchmarks.

Results. We use the trained NCNet model to find correspondences in each pair of the HPatches dataset and

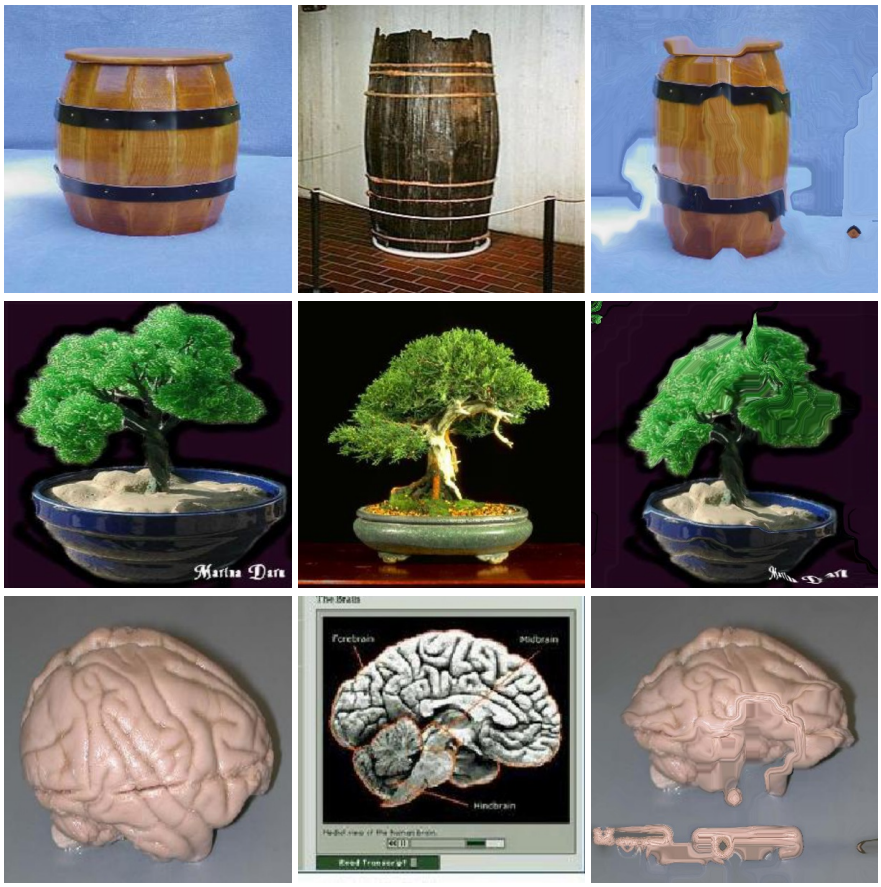


(a) PF-Pascal



(b) PF-Willow

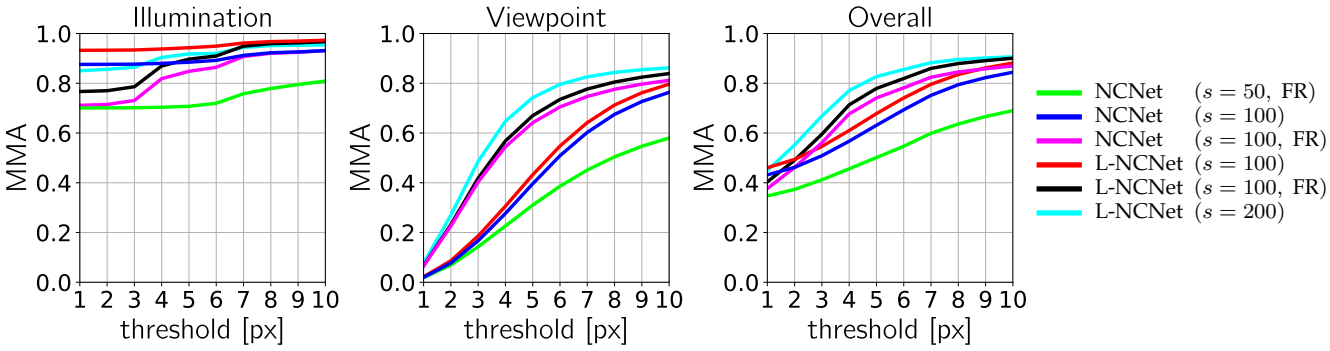
Fig. 2: *Semantic keypoint transfer.* The annotated (ground truth) keypoints in the left image are automatically transferred to the right image using the dense correspondences between the two images obtained from our NCNet.



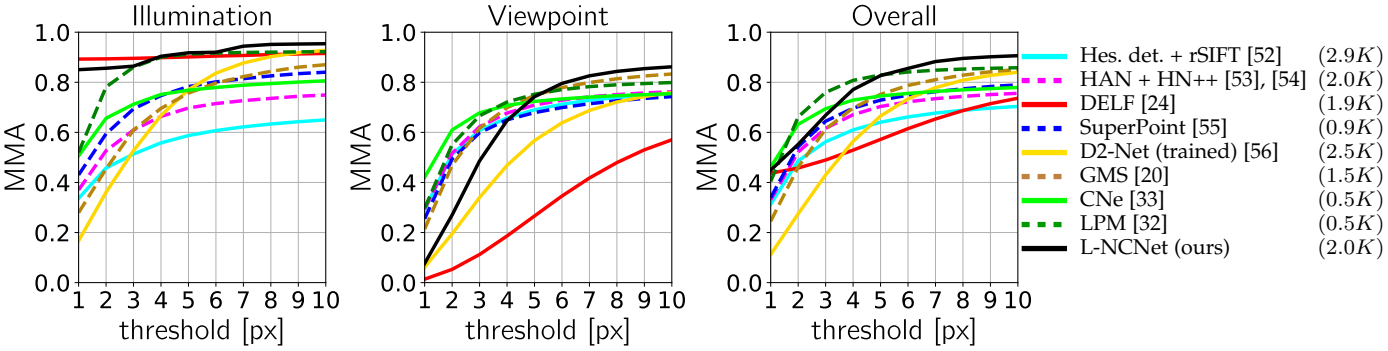
(a) Caltech-101

(b) TSS

Fig. 3: *Dense semantic alignment.* The first two columns show the source and target images, respectively. The right-most column shows the result of transforming the source image by bilinear interpolation using the matches obtained by NCNet such that the result is aligned to the target image. Note that no global geometric model is used for the warping.



(a) *Variants of the proposed method.* We present results when matching using different feature map sizes (s is the size of the feature map along the longest edge), using both NCNet and the lighter version L-NCNet with or without the feature relocalization operation (FR).



(b) *Comparison with state-of-the-art methods.* The proposed L-NCNet (with a feature map size $s = 200$) method obtains the best overall results for threshold values above 5px and the best viewpoint results for thresholds above 5.5px.

Fig. 4: *HPatches benchmark results.* We report the Mean Matching Accuracy (MMA) as a function of the tolerance threshold for the *illumination* and *viewpoint* subsets, and well as the *overall* results.

evaluated the accuracy of these correspondences. The results using the mean matching accuracy (MMA) metric from (14) are presented in Fig. 4. In Fig. 4a, the results of several variants of the proposed method are presented. These variations include: (i) different sizes of the feature maps (s=50, 100 or 200 along the longest edge), which correspond to running the method with different input image resolutions; (ii) the use of a feature relocalization operation described in Sec. 3.7; and (iii) either employing the symmetric NCNet or the lightweight L-NCNet model, which allows for evaluation with larger feature map sizes. Increasing the feature map size improves the localization precision of the matches, increasing the matching accuracy. However the vanilla NCNet cannot handle the larger feature map sizes due to GPU memory constraints. Using the feature relocalization method enables the NCNet to improve the localization precision, improving the MMA over the variants using the same feature map sizes and no relocalization. The best results are obtained with the lighter L-NCNet using the largest feature size of $s = 200$.

In Fig. 4b, the best performing variant of NCNet is compared against other state-of-the-art methods on the HPatches benchmark. Our method obtains the best results for viewpoint changes for the larger thresholds. For smaller thresholds, our method suffers from a localization precision that is limited by the resolution of the features used for matching. This is a common issue for all methods that rely on CNN descriptors, contrary to handcrafted descriptors that can be run on much higher resolutions. Note that, while some methods present better accuracy at small threshold values in the viewpoint subset, they tend to perform worse than our method in the illumination subset. Overall, our method obtains the best results for threshold values above 5px.

Next, we evaluate the quality of the homographies estimated with RANSAC using the transfer error (15) on the HPatches dataset. An image pair is counted as correctly aligned if the transfer error for that pair is smaller than a 5px threshold. In Tab. 3 we present the number of pairs that each method is able to correctly align, from both the illumination and viewpoint subsets, and overall. For the correctly aligned pairs, we also show the average number of inliers, and the effective transfer error for these pairs. NCNet can align more pairs overall than any other method, while obtaining the smallest average transfer error overall.

For evaluation on the InLoc benchmark, we plug-in our trainable neighbourhood consensus network (NCNet) as a correspondence module into the InLoc indoor localization pipeline [3]. This pipeline consists of the following steps: i. retrieval, ii. re-ranking, iii. pose estimation for shortlisted images, and iv. dense pose verification by view synthesis. We integrate NCNet as an intermediate step between the re-ranking (ii) and pose estimation steps (iii) of the InLoc pipeline. In the combined approaches DensePE+NCNet and InLoc+NCNet, the matches generated by NCNet are used to compute the poses in step (iii). DensePE+NCNet contains

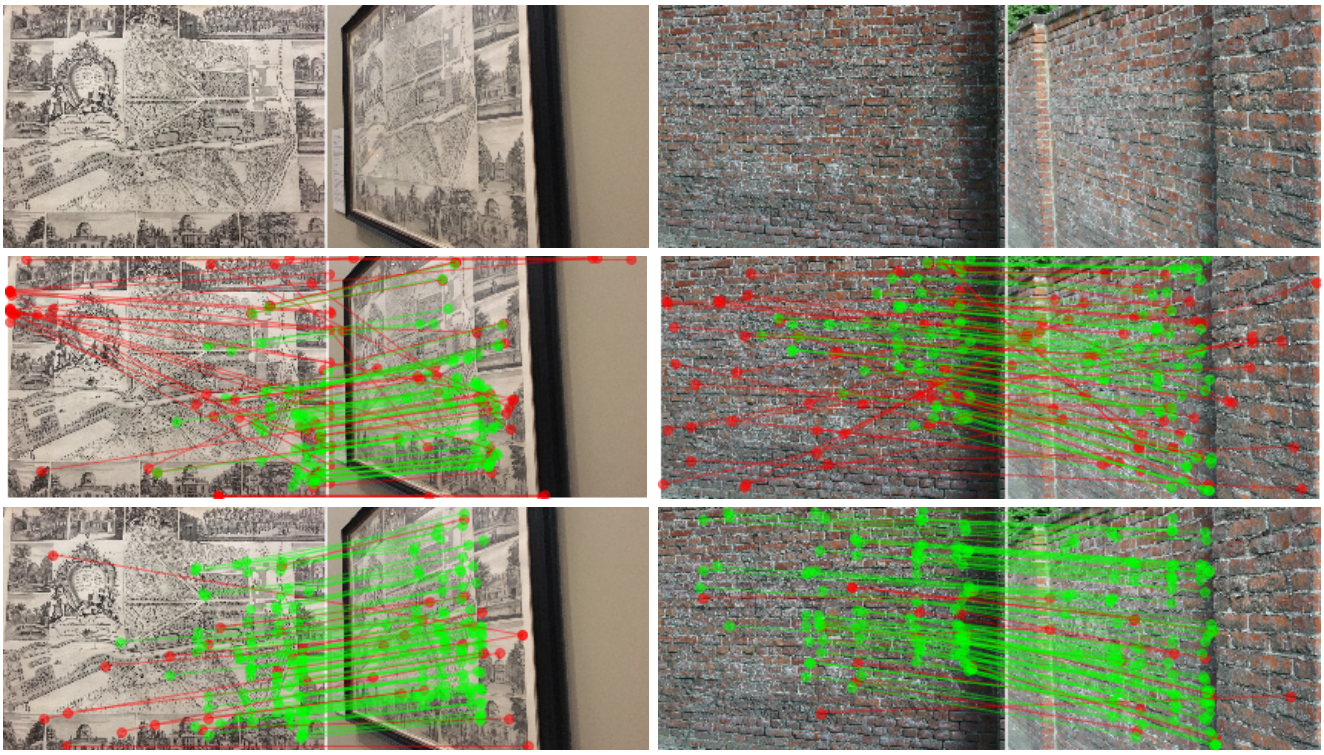


Fig. 5: *Correspondences on HPatches images.* The top row shows the image pair; the middle row shows the matches obtained directly from the correlation map **c** (before NCNet filtering); the bottom row shows the matches obtained by the proposed method (after NCNet filtering). Correspondences have been coloured as inliers (green) and outliers (red) w.r.t. the ground-truth homography using a threshold of 5px. Each image shows 100 randomly sampled matches from the top 2000 matches. The proposed NCNet method tends to obtain a larger fraction of correct matches which span a larger portion of the image with respect to the raw matches before NCNet filtering.

TABLE 3: Homography estimation on HPatches

	Illumination (260 pairs)			Viewpoint (280 pairs)			Overall (540 pairs)		
Method	# correct	# inliers	TE [px]	# correct	# inliers	TE [px]	# correct	# inliers	TE [px]
NCNet	256	**9124**	**0.34**	**268**	**8227**	1.35	**524**	**8665**	**0.86**
SuperPoint	**257**	574	0.96	264	914	**1.16**	521	746	1.06
GMS	224	2478	1.27	235	2782	1.34	459	2634	1.31
CNe	254	486	1.05	252	438	1.26	506	462	1.16
LPM	222	199	1.15	223	945	1.24	445	572	1.20

The number of correctly aligned pairs (within a 5px average transfer error threshold) for each method and benchmark subset (illumination, viewpoint or overall) is computed. For the correct pairs, we also present the average number of inliers and transfer error (TE). NCNet obtains the largest number of correctly aligned pairs overall (524 out of 540) with the smallest average transfer error (0.86px).

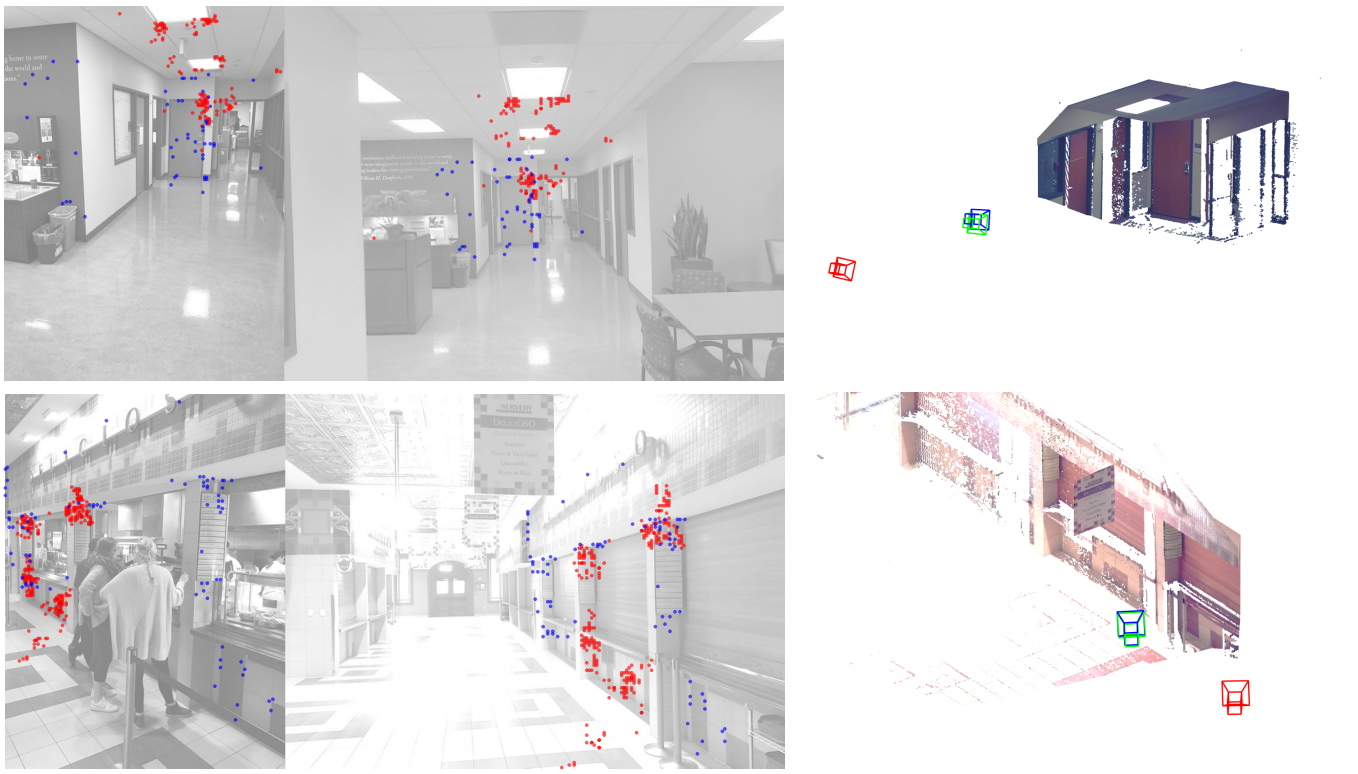
steps (i-iii) while InLoc+NCNet also contains the dense pose verification step (iv).

In order to evaluate the contribution of NCNet separately from that of the feature extractor network, two additional experiments are performed where NCNet is replaced with hard mutual nearest neighbours matching (MNN), using the same base CNN network (ResNet-101). Results are summarized in Table 2 and clearly demonstrate benefits of our approach (DensePE+NCNet) compared to both sparse keypoint (DoG+SIFT) matching (SparsePE) and the CNN feature matching used in [3] (DensePE). When inserted into the entire localization pipeline, our approach (InLoc + NCNet) obtains state-of-the-art results on the indoor localization benchmark. For these experiments an input resolution of $r = 3200$ pixels along the longest image edge together with the feature relocalization operation from Sec. 3.7 were used. Qualitative examples of the correspondences obtained by our method in challenging indoor scenes with repetitive structures and texture-less areas are presented in Fig. 6.

4.3 Ablation studies.

In this section we assess the relevance of two different components in the proposed NCNet formulation (6), namely the symmetric 4D convolutional network S that implements neighbourhood consensus, and the soft mutual nearest neighbour filtering operation M. For this, we train different variants of the proposed method and evaluate their performance on the PF-Pascal benchmark. No finetuning of the feature extraction network is performed during in this ablation. Results are presented in Table 4.

The top section of the table shows the performance of two networks containing only the isolated symmetric neighbourhood-consensus S and soft mutual nearest neighbour filtering M components. As it can be observed, the



(a) (b) (c)

Fig. 6: *Correspondences and poses on InLoc.* Each row shows (a-b) the correspondences used for pose estimation in the case of the proposed NCNet method (blue) against those of the baseline InLoc method (red); and (c) the resulting obtained poses for the proposed NCNet (blue) and InLoc baseline (red) compared to the ground-truth pose (green). In both cases the InLoc baseline produces many mismatches due to repetitive structures (ceiling lamps in the top, and columns in the bottom example) that result in a large pose error. On the other hand, NCNet obtains mostly correct matches resulting in a small pose error.

TABLE 2: Comparison of indoor localization methods.

	Distance (m)			
Method	0.25	0.50	1.00	2.00
SparsePE [3]	21.3	30.7	42.6	48.3
DensePE [3]	35.3	47.4	57.1	61.1
DensePE+MNN	31.9	50.5	62.0	64.7
DensePE+**NCNet**	37.1	53.5	62.9	66.3
InLoc [3]	38.9	56.5	69.9	74.2
InLoc+MNN	37.1	60.2	72.0	76.3
InLoc+**NCNet**	**44.1**	**63.8**	**76.0**	**78.4**

We show the rate (%) of correctly localized queries within a given distance (m) and 10° angular error.

TABLE 4: Ablation studies on the PF-Pascal dataset.

	PF-Pascal
Method	PCK ($\alpha = 0.1$)
$S(\mathbf{c})$	12.5
$M(\mathbf{c})$	44.0
$(M \circ S)(\mathbf{c})$	13.6
$(S \circ M)(\mathbf{c})$	77.2
$(M \circ N \circ M)(\mathbf{c})$ (**L-NCNet**)	74.3
$(M \circ S \circ M)(\mathbf{c})$ (**NCNet**)	78.0

We evaluate different model configurations for filtering the correlation map scores $\mathbf{c}$ *using the neighbourhood consensus* (S *or the lightweight* N*) and soft mutual nearest neighbour filtering* (M) *modules.*

isolated components do not perform well in the task of keypoint transfer. The middle section of the table shows the combinations of these two components. Note that applying the soft mutual nearest neighbour filtering M first and then symmetric neighbourhood consensus module S produces much better results than doing so in the opposite order. The bottom section of the table shows the full proposed model including two stages of soft mutual nearest neighbour filtering M, using both the lightweight L-NCNet and the symmetric NCNet versions.

4.4 Implementation details.

Model details. The model was implemented in PyTorch [61], and a ResNet-101 network [62] initialized on ImageNet was used for feature extraction (up to the `conv4_23` layer). Different architectures of the neighbourhood consensus network $N(\cdot)$ are used for category- and instance-level matching, as these two problems present different challenges. For category-level matching, a more complex network is used in order to capture the strong appearance differences in these matching problems. For instance-level matching, a simpler

network is used, allowing to process the images in higher resolution and obtaining more precisely localized matches, which is required for tasks such as pose estimation. In all cases, the input and output tensors have a single channel, and the intermediate results have 16 channels. For category-level matching, $N(\cdot)$ contains three layers of $5 \times 5 \times 5 \times 5$ filters, resulting in 180K trainable parameters. In the case of instance-level matching, $N(\cdot)$ has two layers of $3 \times 3 \times 3 \times 3$ filters, resulting in 2.6K trainable parameters. Both models are trained using dense feature maps $\mathbf{f}^A$ and $\mathbf{f}^B$ of size 25×25. For evaluation, the category-level matching model also uses 25×25 dense feature maps, processes an image pair in 0.5s and requires 240MB of memory. For instance-level matching, evaluation is performed using larger feature maps, such as 100×75, in order to obtain a higher localization precision which is required for instance-level matching. In this case, the execution takes 9.3s and requires 5700MB of memory.

Training details. The model is initially trained for 5 epochs using Adam optimizer [63], with a learning rate of 5×10^{-4} and keeping the feature extraction layer weights fixed. We used a batch size of 16 and the training takes 9 hours on a standard Tesla T4 GPU. For category level matching, the model is then subsequently finetuned for 5 more epochs, training both the feature extraction and the neighbourhood consensus network, with a learning rate of 1×10^{-5}. In the case of instance level matching, finetuning the feature extraction did not improve the performance.

4D convolutions. As 4D convolutions ($\overset{*}{}_{4D}$) are not currently supported by PyTorch, they were implemented by aggregating the results of multiple 3D convolutions ($\overset{*}{}_{3D}$) over the remaining fourth dimension. Given a 4D input tensor $X \in \mathbb{R}^{h \times w \times d \times t}$ and a 4D weight tensor $W \in \mathbb{R}^{k \times k \times k \times k}$ with k odd (channel and batch dimensions are omitted for simplicity), their convolution can be then computed by:

$$\left(X \underset{4D}{*} W\right)_{i:::} = \sum_{j=0}^{k-1} X_{j':::} \underset{3D}{*} W_{j:::} \text{ with } j' = i+j-(k-1)/2 \ , \tag{16}$$

and considering 0-indexed tensors and that X takes the value 0 when the index j' is out of range ($j' < 0$ or $j' \geq h$).

In addition, the memory requirements can be reduced by exploiting the fact that the 4D convolutional network N has multiple channels in the hidden layers but single channel input and output. If the correlation tensor can be fitted into the GPU memory, the memory requirements of the forward pass through N can be limited by computing the output tensor $Y = N(X)$ in *chunks* which can then be stacked to obtain the full output tensor. In order to do this, a set of *slices* X_s $s = 1, \ldots, n$ are generated from the input tensor X, and fed to the network progressively. Note that, while the output slices are non-overlapping, the inputs will have an overlap due to the overlapping receptive fields of the output slices. The slicing can be then performed in the following way:

$$\begin{aligned} \{Y_{i:::}\}_{i=a,\ldots,b} &= Y_s = N(X_s) \\ &\text{with } X_s = \{X_{i:::}\}_{i=a-p,\ldots,b+p} \ , \end{aligned} \tag{17}$$

where the input slices X_s are larger than the output slices Y_s by $2p + 1$, which corresponds to the receptive field of the last layer of the network N. In consequence, when using this approach no padding should be performed in the convolutional layers of N.

4.5 Limitations

While our method identifies correct matches in many challenging cases, some situations remain difficult. The two typical failure modes include: repetitive patterns combined with large changes in scale, and locally geometrically consistent groups of incorrect matches. Furthermore, the proposed method has quadratic $O(N^2)$ complexity with respect to the number of image pixels (or CNN features) N. This limits the resolution of the images that we are currently able to handle in a 16GB GPU to 1600×1200px (or 3200×2400px if using feature relocalization or slicing), and renders the method relatively slow: the processing time of a 3200×2400px image pair using feature relocalization is ≈ 7 seconds.

5 Conclusion

We have developed a neighbourhood consensus network a CNN architecture that learns local patterns of correspondences for image matching without the need for a global geometric model. We have shown the model can be trained effectively from weak supervision and obtains strong results outperforming state-of-the-art on two very different matching tasks. These results open up the possibility for end-to-end learning of other challenging visual correspondence tasks, such as 3D category-level matching [64], or visual localization across day/night illumination [2].

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Numbers 15H05313, 16KK0002, EU-H2020 project LADIO No. 731970, ERC grant LEAP No. 336845, CIFAR Learning in Machines & Brains program and the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15 003/0000468). We gratefully acknowledge the support of NVIDIA Corporation with the donation of Quadro P6000 GPU.

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[2] T. Sattler, W. Maddern, A. Torii, J. Sivic, T. Pajdla, M. Pollefeys, and M. Okutomi, "Benchmarking 6dof urban visual localization in changing conditions," in *Proc. CVPR*, 2018.

[3] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "InLoc: Indoor visual localization with dense matching and view synthesis," in *Proc. CVPR*, 2018.

[4] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "Sift-flow: Dense correspondence across different scenes," in *Proc. ECCV*, 2008.

[5] M. Jahrer, M. Grabner, and H. Bischof, "Learned local descriptors for recognition and matching," in *Computer Vision Winter Workshop*, 2008.

[6] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: a comparison to SIFT," *arXiv preprint arXiv:1405.5769*, 2014.

[7] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. CVPR*, 2015.

[8] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. CVPR*, 2015.

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

[10] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE PAMI*, vol. 36, no. 8, pp. 1573–1585, 2014.

[11] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. ICCV*, 2015.

[12] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proc. BMVC.*, 2016.

[13] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Proc. ECCV*, 2016.

[14] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proc. CVPR*, 2017.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[16] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial intelligence*, vol. 78, no. 1-2, pp. 87–119, 1995.

[17] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE PAMI*, vol. 19, no. 5, pp. 530–535, 1997.

[18] F. Schaffalitzky and A. Zisserman, "Automated scene matching in movies," in *International Conference on Image and Video Retrieval*, 2002.

[19] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. ICCV*, 2003.

[20] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. CVPR*, 2017.

[21] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," in *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 2018.

[22] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. ECCV*, 2002.

[23] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[24] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proc. ICCV*, 2017.

[25] N. Savinov, L. Ladicky, and M. Pollefeys, "Matching neural paths: transfer from recognition to correspondence search," in *NIPS*, 2017.

[26] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," in *NIPS*, 2016.

[27] J. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?" in *NIPS*, 2014.

[28] I. Rocco, R. Arandjelović, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proc. CVPR*, 2017.

[29] ——, "End-to-end weakly-supervised semantic alignment," in *Proc. CVPR*, 2018.

[30] S. Kim, S. Lin, S. Jeon, D. Min, and K. Sohn, "Recurrent transformer networks for semantic correspondence," in *NeurIPS*, 2018.

[31] T. Sattler, B. Leibe, and L. Kobbelt, "SCRAMSAC: Improving RANSAC's Efficiency with a Spatial Consistency Filter," in *Proc. ICCV*, 2009.

[32] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, "Locality preserving matching," *IJCV*, vol. 127, no. 5, pp. 512–531, 2019.

[33] K. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. CVPR*, 2018.

[34] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao, "Learning two-view correspondences and geometry using order-aware network," *Proc. ICCV*, 2019.

[35] E. Brachmann and C. Rother, "Neural- Guided RANSAC: Learning where to sample model hypotheses," in *Proc. ICCV*, 2019.

[36] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI*, 1981.

[37] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[38] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE PAMI*, vol. 30, no. 2, pp. 328–341, 2007.

[39] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. CVPR*, 2010.

[40] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE PAMI*, vol. 33, no. 3, pp. 500–513, 2010.

[41] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. ICCV*, 2015.

[42] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. CVPR*, 2018.

[43] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proc. ICCV*, 2017.

[44] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014.

[45] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE PAMI*, vol. 39, no. 12, pp. 2481–2495, 2017.

[46] A. R. Widya, A. Torii, and M. Okutomi, "Structure from motion using dense cnn features with keypoint relocalization," *IPSJ Transactions on Computer Vision and Applications*, vol. 10, no. 1, p. 6, 2018.

[47] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow: Semantic correspondences from object proposals," *IEEE PAMI*, vol. 40, no. 7, pp. 1711–1725, 2017.

[48] K. Han, R. S. Rezende, B. Ham, K.-Y. K. Wong, M. Cho, C. Schmid, and J. Ponce, "SCNet: Learning Semantic Correspondence," in *Proc. ICCV*, 2017.

[49] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE PAMI*, vol. 28, no. 4, pp. 594–611, 2006.

[50] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," *IEEE PAMI*, 2018.

[51] T. Taniai, S. N. Sinha, and Y. Sato, "Joint recovery of dense correspondence and cosegmentation in two images," in *Proc. CVPR*, 2016.

[52] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. CVPR*, 2012.

[53] D. Mishkin, F. Radenović, and J. Matas, "Repeatability Is Not Enough: Learning Discriminative Affine Regions via Discriminability," in *Proc. ECCV*, 2018.

[54] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *NIPS*, 2017.

[55] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *CVPR Workshops*, 2018.

[56] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-Net: A Trainable CNN for Joint Detection and Description of Local Features," in *Proc. CVPR*, 2019.

[57] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors," in *Proc. CVPR*, 2017.

[58] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[59] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *Pattern Recognition*, 2003.

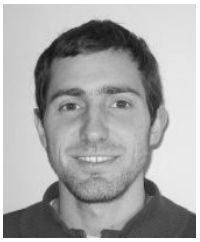
[60] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane," in *CVPR*, 2005.

[61] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[64] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *Proc. ECCV*, 2018.



Ignacio Rocco received the MSc degree in Vision, Mathematics and Machine Learning (MVA) from École Normale Supérieure Paris-Saclay in 2016 and is currently pursuing a PhD degree in computer vision in the WILLOW research group at Inria / École Normale Supérieure, Paris. His current research is focused on the image alignment problem, investigating different trainable methods to efficiently solve this task.

Mircea Cimpoi is currently a postdoctoral researcher at Czech Technical University in Prague, member of IMPACT Project. He received his PhD degree from University of Oxford, in 2015. Prior to PhD, he was with Microsoft, part of Bing Augmented Reality, until 2011. His research interests include indoor vision-based navigation, large scale indoor localization, local image features.

Relja Arandjelović received the BA and MEng degrees from the University of Cambridge in 2009, and PhD from the University of Oxford in 2013. He then spent one year as a postdoctoral researcher at the University of Oxford, and two years as a postdoctoral researcher at Inria / École Normale Supérieure, Paris. Since 2016, he is a staff research scientist at DeepMind. His research is focused on learning from little or no labelled data, including work on multimodal learning, large scale image retrieval and place recognition, and image alignment.

Akihiko Torii received the masters and the PhD degrees from Chiba University, in 2003 and 2006. He then spent four years as a postdoctoral researcher in Czech Technical University in Prague. From 2010 to 2020, he was with Tokyo Institute of Technology, where he was an assistant professor in the Department of Systems and Control Engineering, the School of Engineering.

Tomas Pajdla received the MSc and PhD degrees from the Czech Technical University in Prague. He works in geometry and algebra of computer vision and robotics with emphasis on nonclassical cameras, 3D reconstruction, and industrial vision. He contributed to introducing epipolar geometry of panoramic cameras, noncentral camera models generated by linear mapping, generalized epipolar geometries, to developing solvers for minimal problems in structure from motion and to solving image matching problem. He coauthored works awarded the best paper prizes at OAGM 1998 and 2013, BMVC 2002 and ACCV 2014. He is a member of the IEEE.

Josef Sivic received MSc degree from the Czech Technical University in Prague, PhD from the University of Oxford and Habilitation from Ecole Normale Superieure in Paris. He currently holds a joint senior researcher position at Inria in Paris and Czech Technical University in Prague, where he leads a newly created team on Intelligent Machine Perception spanning both institutions. He has published more than 60 scientific publications and his papers have been awarded the Longuet-Higgins prize (CVPR07) and the Helmholtz prize (ICCV03 and ICCV05) for fundamental contributions to computer vision that withstood the test of time. He has served as an area chair for major computer vision conferences and as a program chair for ICCV15. In 2013, he has received an ERC starting grant.