2021

인공지능세미나

김범채

AILab. 연구소장

2021.06.24.

Neighbourhood Consensus Networks

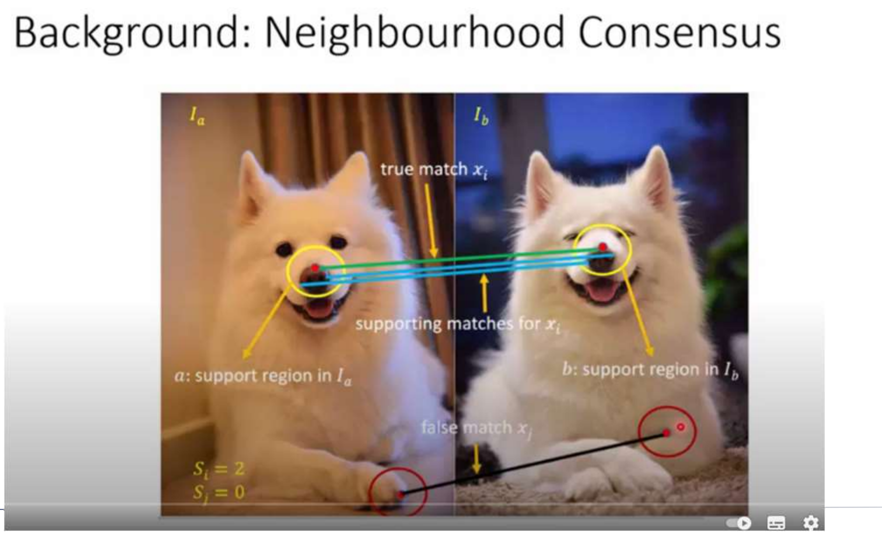
1. Intro
2. Related works
3. Methodology
4. Experiment
5. Conclusion

Problem

- Finding visual correspondences is one of the fundamental image understanding problems with applications in 3D reconstruction, visual localization or object recognition.

Related Works

- Matching with hand-crafted image descriptors.
- Matching with trainable descriptors.
- Trainable image alignment.
- Match filtering by neighbourhood consensus.
- Flow and disparity estimation.



Background: Neighbourhood Consensus

I_a
I_b
true match x_i
supporting matches for x_i
a: support region in I_a
b: support region in I_b
false match x_j
$S_i = 2$
$S_j = 0$

Methodology

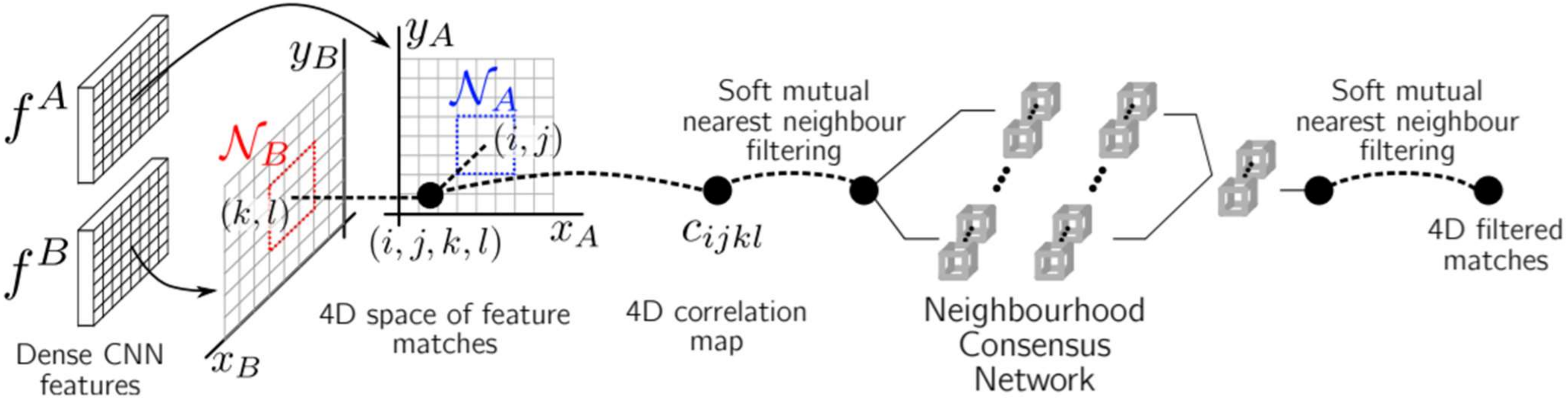
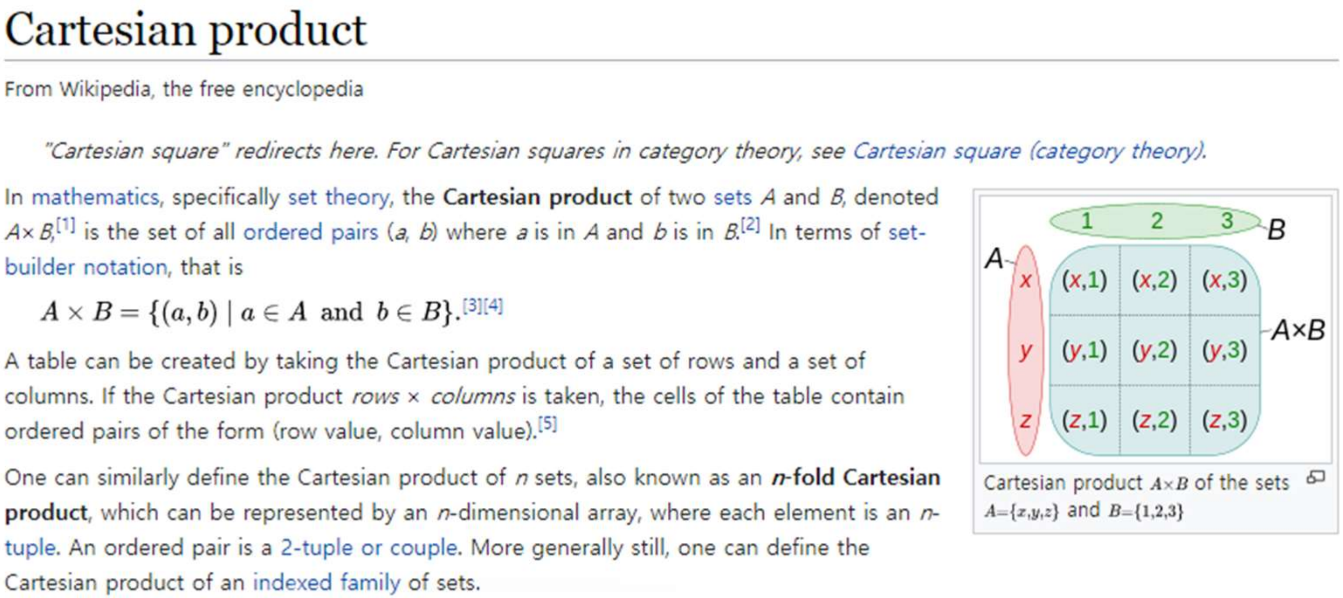


Figure 1: **Overview of the proposed method.** A fully convolutional neural network is used to extract dense image descriptors f^A and f^B for images I_A and I_B, respectively. All pairs of individual feature matches f^A_{ij} and f^B_{kl} are represented in the 4-D space of matches (i, j, k, l) (here shown as a 3-D perspective for illustration), and their matching scores stored in the 4-D correlation tensor c. These matches are further processed by the proposed soft-nearest neighbour filtering and neighbourhood consensus network (see Figure 2) to produce the final set of output correspondences.

Methodology



Cartesian product

From Wikipedia, the free encyclopedia

"Cartesian square" redirects here. For Cartesian squares in category theory, see Cartesian square (category theory).

In mathematics, specifically set theory, the **Cartesian product** of two sets *A* and *B*, denoted *A*× *B*,[1] is the set of all ordered pairs (*a*, *b*) where *a* is in *A* and *b* is in *B*.[2] In terms of set-builder notation, that is

$$A \times B = \{(a, b) \mid a \in A \ \text{ and } \ b \in B\}.$$ [3][4]

A table can be created by taking the Cartesian product of a set of rows and a set of columns. If the Cartesian product *rows* × *columns* is taken, the cells of the table contain ordered pairs of the form (row value, column value).[5]

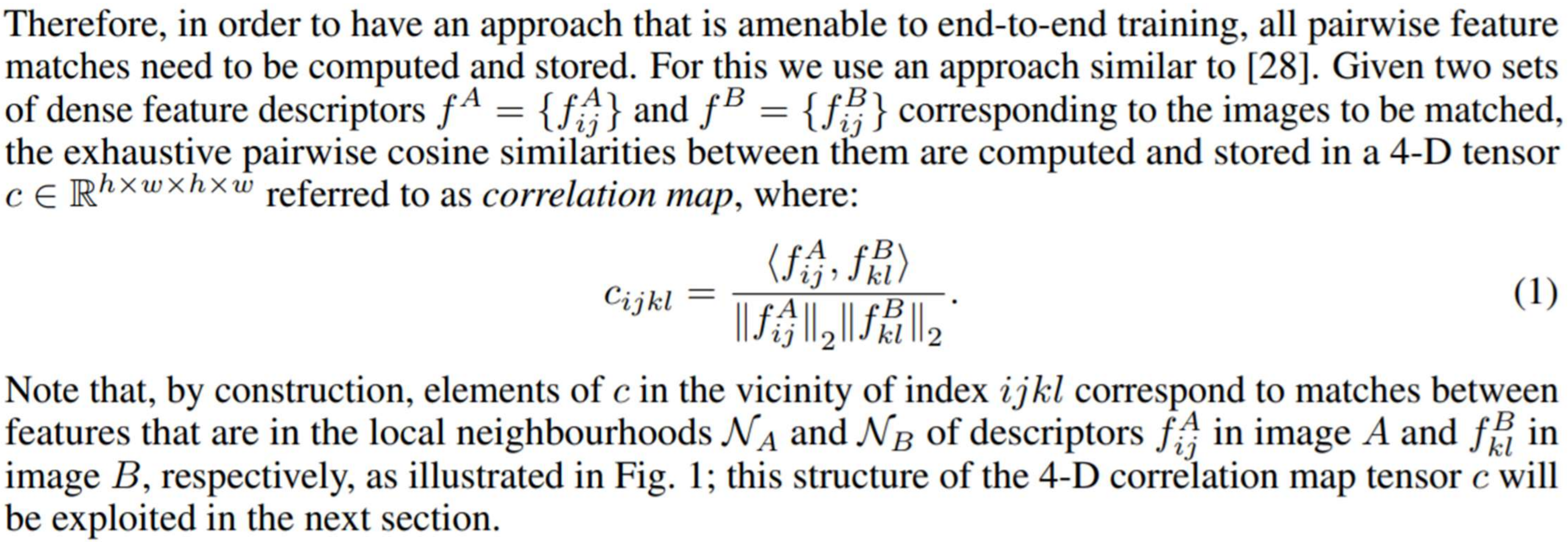
One can similarly define the Cartesian product of *n* sets, also known as an ***n*-fold Cartesian product**, which can be represented by an *n*-dimensional array, where each element is an *n*-tuple. An ordered pair is a 2-tuple or couple. More generally still, one can define the Cartesian product of an indexed family of sets.

A \ B	1	2	3
x	(x,1)	(x,2)	(x,3)
y	(y,1)	(y,2)	(y,3)
z	(z,1)	(z,2)	(z,3)

Cartesian product $A \times B$ of the sets $A = \{x,y,z\}$ and $B = \{1,2,3\}$

Methodology

Correlation map



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 $f^A = \{f^A_{ij}\}$ and $f^B = \{f^B_{ij}\}$ corresponding to the images to be matched, the exhaustive pairwise cosine similarities between them are computed and stored in a 4-D tensor $c \in \mathbb{R}^{h \times w \times h \times w}$ referred to as *correlation map*, where:

$$c_{ijkl} = \frac{\langle f^A_{ij}, f^B_{kl} \rangle}{\|f^A_{ij}\|_2 \|f^B_{kl}\|_2}. \tag{1}$$

Note that, by construction, elements of 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 f^A_{ij} in image A and f^B_{kl} in image B, respectively, as illustrated in Fig. 1; this structure of the 4-D correlation map tensor c will be exploited in the next section.

Methodology

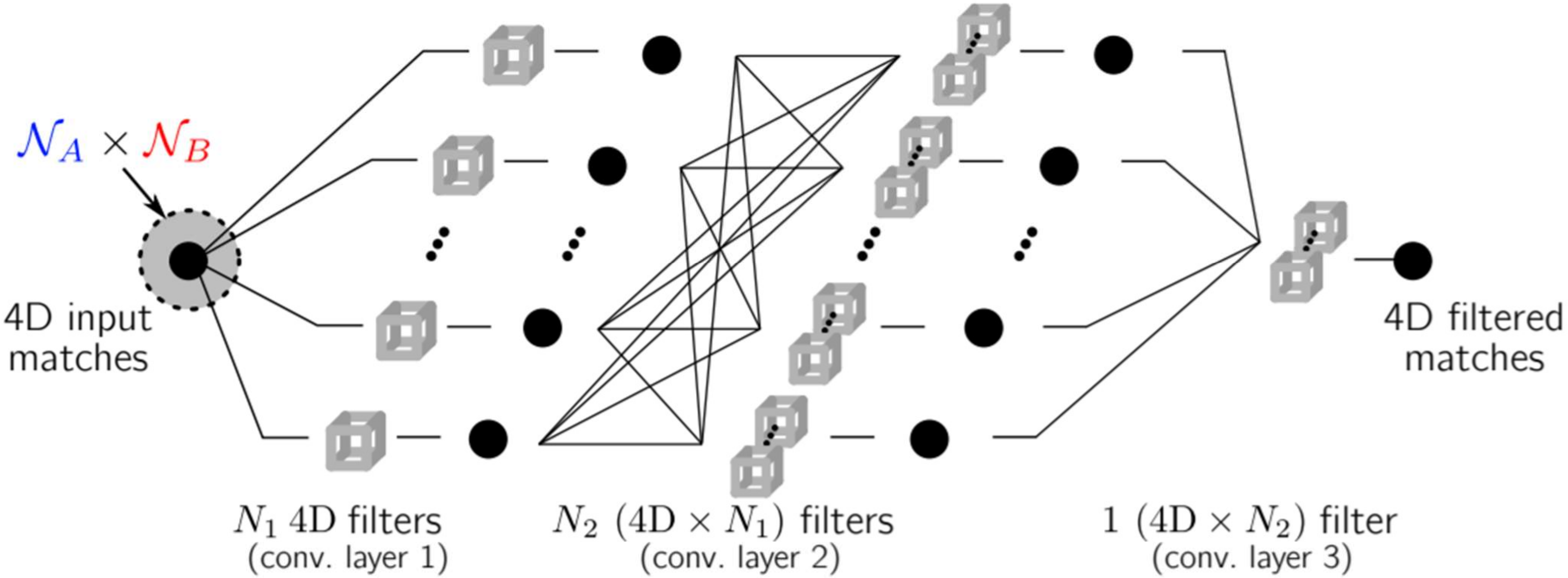
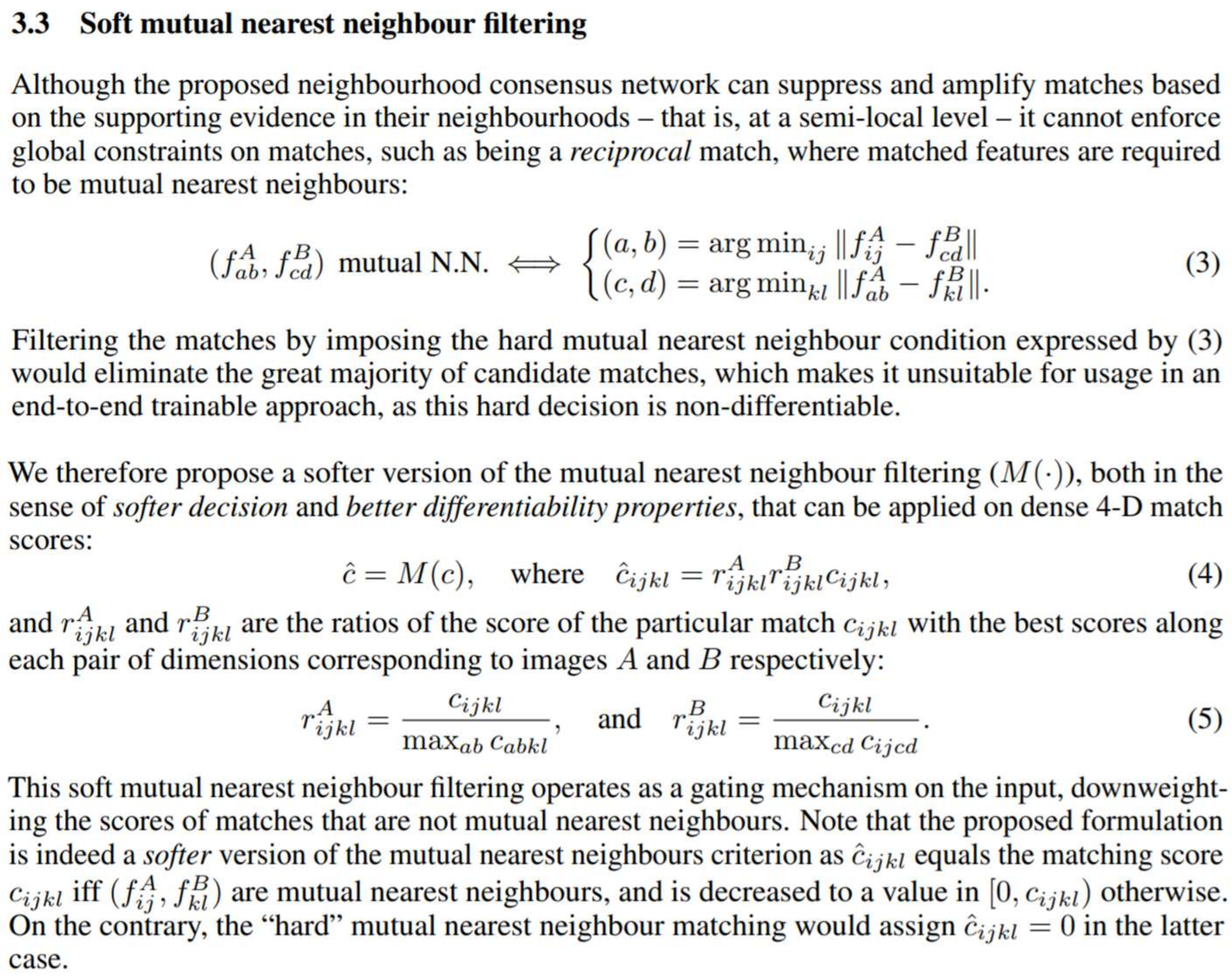

$\mathcal{N}_A \times \mathcal{N}_B$
4D input matches
4D filtered matches
N_1 4D filters (conv. layer 1)
N_2 (4D × N_1) filters (conv. layer 2)
1 (4D × N_2) filter (conv. layer 3)

Figure 2: **Neighbourhood Consensus Network (NC-Net).** A neighbourhood consensus CNN operates on the 4D space of feature matches. The first 4D convolutional layer filters span $\mathcal{N}_A \times \mathcal{N}_B$, the Cartesian product of local neighbourhoods $\mathcal{N}_A$ and $\mathcal{N}_B$ in images A and B respectively. The proposed 4D neighbourhood consensus CNN can learn to identify the matching patterns of reliable and unreliable matches, and filter the matches accordingly.

Methodology



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:

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

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:

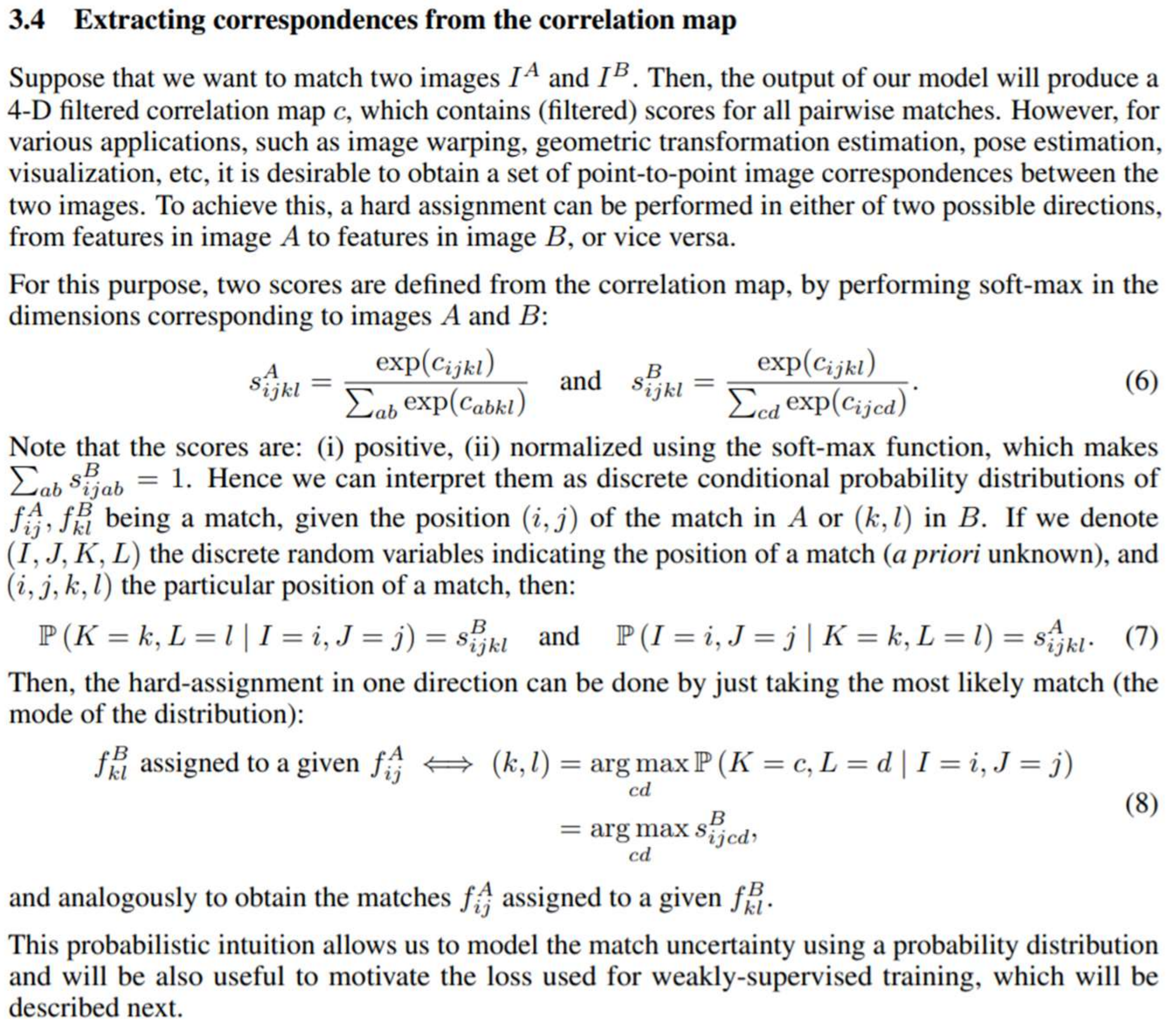
$$\hat{c} = M(c), \quad \text{where} \quad \hat{c}_{ijkl} = r^A_{ijkl} r^B_{ijkl} c_{ijkl}, \tag{4}$$

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

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

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 $\hat{c}_{ijkl}$ equals the matching score c_{ijkl} iff (f^A_{ij}, f^B_{kl}) are mutual nearest neighbours, and is decreased to a value in $[0, c_{ijkl})$ otherwise. On the contrary, the "hard" mutual nearest neighbour matching would assign $\hat{c}_{ijkl} = 0$ in the latter case.

Methodology



3.4 Extracting correspondences from the correlation map

Suppose that we want to match two images I^A and I^B. Then, the output of our model will produce a 4-D filtered correlation map c, 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:

$$s^A_{ijkl} = \frac{\exp(c_{ijkl})}{\sum_{ab}\exp(c_{abkl})} \quad \text{and} \quad s^B_{ijkl} = \frac{\exp(c_{ijkl})}{\sum_{cd}\exp(c_{ijcd})}. \tag{6}$$

Note that the scores are: (i) positive, (ii) normalized using the soft-max function, which makes $\sum_{ab} s^B_{ijab} = 1$. Hence we can interpret them as discrete conditional probability distributions of f^A_{ij}, 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:

$$\mathbb{P}\,(K = k, L = l \mid I = i, J = j) = s^B_{ijkl} \quad \text{and} \quad \mathbb{P}\,(I = i, J = j \mid K = k, L = l) = s^A_{ijkl}. \tag{7}$$

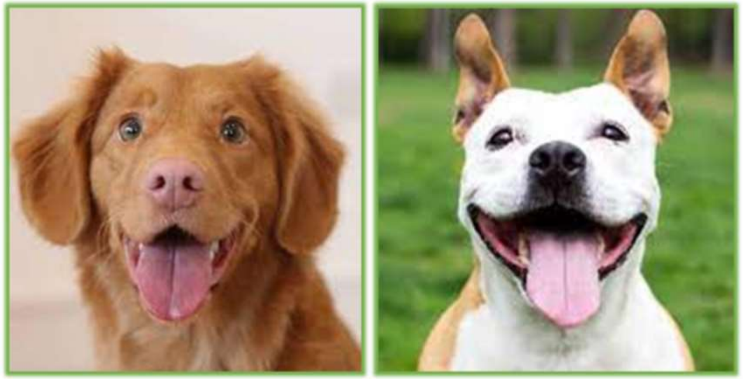
Then, the hard-assignment in one direction can be done by just taking the most likely match (the mode of the distribution):

$$\begin{aligned} f^B_{kl} \text{ assigned to a given } f^A_{ij} \iff (k, l) &= \arg\max_{cd} \mathbb{P}\,(K = c, L = d \mid I = i, J = j) \\ &= \arg\max_{cd} s^B_{ijcd}, \end{aligned} \tag{8}$$

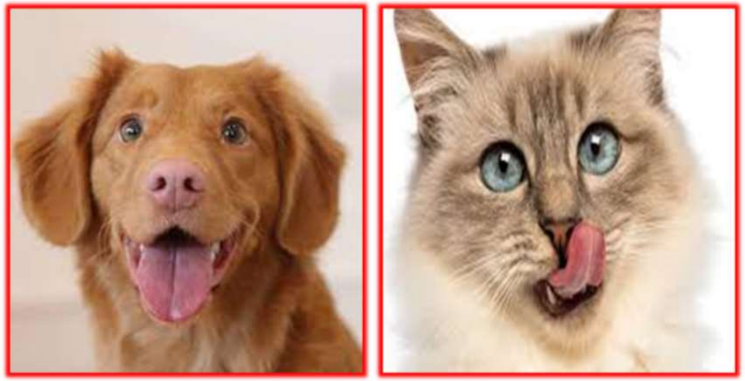
and analogously to obtain the matches f^A_{ij} assigned to a given f^B_{kl}.

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.

Methodology



Positive pair



Negative pair

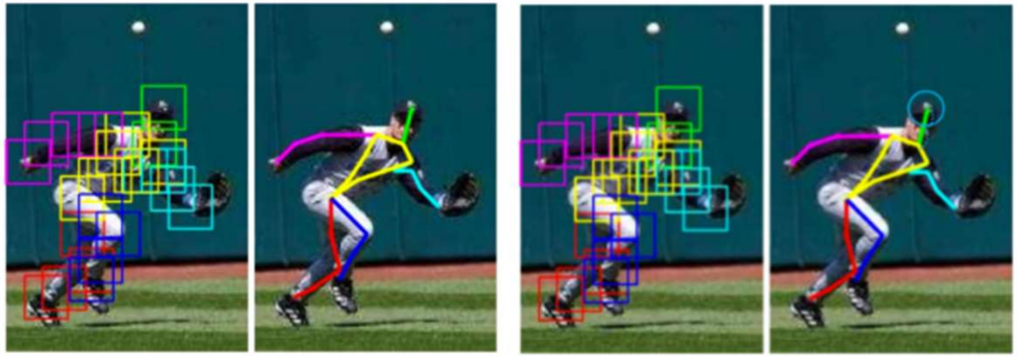
Results

Percentage of Correct Keypoints - PCK

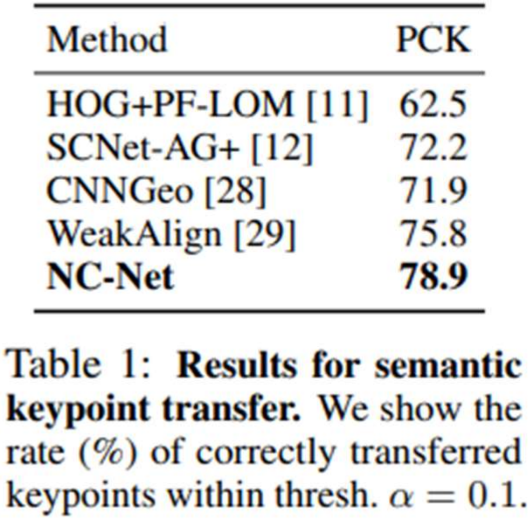
[3] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In CVPR, 2011. 의 논문 중 7.2장에 다음과 같이 나와있다.

『We define a candidate keypoint to be correct if it falls within α · max(h, w) pixels of the ground-truth keypoint, where h and w are the height and width of the bounding box respectively, and α controls the relative threshold for considering correctness.』

PCK에서는 관절점의 추정 좌표와 정답 좌표의 거리가 어느 임계값 보다 작다면 그 관절점이 옳다고 판단한다. PCK의 임계값은 인물 머리 크기에 따라 결정되는 경우가 많다. 이것을 PCKh라 부른다. 예를 들면 PCKh @0.5의 경우, 머리 사이즈의 0.5를 임계값으로 설정해 평가 한다.



좌 - PCK / 우 - PCKh



Method	PCK
HOG+PF-LOM [11]	62.5
SCNet-AG+ [12]	72.2
CNNGeo [28]	71.9
WeakAlign [29]	75.8
NC-Net	**78.9**

Table 1: **Results for semantic keypoint transfer.** We show the rate (%) of correctly transferred keypoints within thresh. $\alpha = 0.1$.

Results

Problem

- Deep Mutual Learning