Learning to Match Aerial Images with Deep Attentive Architectures

Hani Altwaijry[1,2], Eduard Trulls[3], James Hays[4], Pascal Fua[3], Serge Belongie[1,2]

[1] Department of Computer Science, Cornell University [2] Cornell Tech
[3] Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne (EPFL)
[4] School of Interactive Computing, College of Computing, Georgia Institute of Technology

Abstract

Image matching is a fundamental problem in Computer Vision. In the context of feature-based matching, SIFT and its variants have long excelled in a wide array of applications. However, for ultra-wide baselines, as in the case of aerial images captured under large camera rotations, the appearance variation goes beyond the reach of SIFT and RANSAC. In this paper we propose a data-driven, deep learning-based approach that sidesteps local correspondence by framing the problem as a classification task. Furthermore, we demonstrate that local correspondences can still be useful. To do so we incorporate an attention mechanism to produce a set of probable matches, which allows us to further increase performance. We train our models on a dataset of urban aerial imagery consisting of 'same' and 'different' pairs, collected for this purpose, and characterize the problem via a human study with annotations from Amazon Mechanical Turk. We demonstrate that our models outperform the state-of-the-art on ultra-wide baseline matching and approach human accuracy.

1. Introduction

Finding the relationship between two images depicting a 3D scene is one of the fundamental problems of Computer Vision. This relationship can be examined at different granularities. At a coarse level, we can ask whether two images show the same scene. At the other extreme, we would like to know the dense pixel-to-pixel correspondence, or lack thereof, between the two images. These granularities are directly related to broader topics in Computer Vision; in particular, one can look at the coarse-grained problem as a recognition/classification task, whereas the pixel-wise problem can be viewed as one of segmentation. Traditional geometry-based approaches live in a middle ground, relying on a multi-stage process that typically involves keypoint matching and outlier rejection, where image-level correspondence is derived from local correspondence.

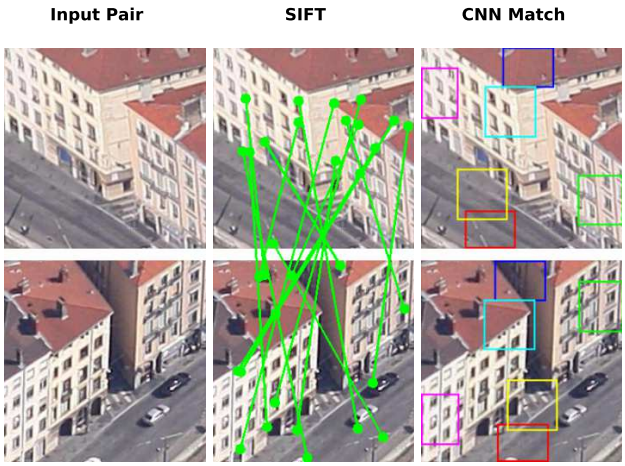

Figure 1. Matching ultra-wide baseline aerial images. Left: The pair of images in question. Middle: Local correspondence matching approaches fail to handle this baseline and rotation. Right: The CNN matches the pair and proposes possible region matches.

In this paper we focus on pairs of oblique aerial images acquired by distant cameras from very different angles, as shown in Fig. 1. These images are challenging for geometry-based approaches for a number of reasons—chief among them are dramatic appearance distortions due to viewpoint changes and ambiguities due to repetitive structures. This renders methods based on local correspondence insufficient for ultra-wide baseline matching.

In contrast, we follow a data-driven approach. Specifically, we treat the problem from a recognition standpoint, without appealing specifically to hand-crafted, feature-based approaches or their underlying geometry. Our aim is to learn a discriminative representation from a large amount of instances of *same* and *different* pairs, which separates the genuine matches from the impostors.

We propose two architectures based on Convolutional Neural Networks (CNN). The first architecture is only concerned with learning to discriminate image pairs as *same* or *different*. The second one extends it by incorporating a Spatial Transformer module [16] to propose *possible* matching

Figure 2. Sample pairs from one of our datasets, collected from Google Maps [13] 'Birds-Eye' view. Pairs show an area or building from two widely separated viewpoints.

regions, in addition to the classification task. We learn both networks given only *same* and *different* pairs, *i.e.*, we learn the spatial transformations in a semi-supervised manner.

To train and validate our models, we use a dataset with 49k ultra-wide baseline pairs of aerial images compiled from Google Maps specifically for this problem: example pairs are shown in Fig. 2. We benchmark our models against multiple baselines, including human annotations, and demonstrate state-of-the-art performance, close to that of the human annotations.

Our main contributions are as follows. **First,** we demonstrate that deep CNNs offer a solution for ultra-wide baseline matching. Inspired by recent efforts in patch matching [14, 43, 31] we build a siamese/classification hybrid model using two AlexNet networks [19], cut off at the last pooling layer. The networks share weights, and are followed by a number of fully-connected layers embodying a binary classifier. **Second,** we show how to extend the previous model with a Spatial Transformer (ST) module, which embodies an attention mechanism that allows our model to propose *possible* patch matches (see Fig. 1), which in turn increases performance. These patches are described and compared with MatchNet [14]. As with the first model, we train this network end-to-end, and only with *same* and *different* training signal, *i.e.*, the ST module is trained in a semi-supervised manner. In sections 3.2 and 4.6 we discuss the difficulties in training this network, and offer insights in this direction. **Third,** we conduct a human study to help us characterize the problem, and benchmark our algorithms against human performance. This experiment was conducted on Amazon Mechanical Turk, where participants were shown pairs of images from our dataset. The results confirm that humans perform exceptionally while responding relatively quickly. Our top-performing model falls within 1% of human accuracy.

2. Related Work

2.1. Correspondence Matching

Correspondence matching has been long dominated by feature-based methods, led by SIFT [23]. Numerous descriptors have been developed within the community, such as SURF [5], BRIEF [8], and DAISY [36]. These descriptors generally provide excellent performance in narrow baselines, but are unable to handle the large distortions present in ultra-wide baseline matching [25].

Sparse matching techniques typically begin by extracting keypoints, *e.g.*, Harris Corners [15]; followed by a description step, *e.g.*, computing SIFT descriptors; then a keypoint matching step, which gives us a pool of probable keypoint matches. These are then fed into a model-estimation technique, *e.g.*, RANSAC [11] with a homography model. This pipeline assumes certain limitations and demands assumptions to be made. Relying on keypoints can be limiting—dense techniques have been successful in wide-baseline stereo with calibration data [36, 38, 40], scene alignment [21, 40] and large displacement motion [38, 40].

The descriptor embodies assumptions about the topology of the scene, *e.g.*, SIFT is not robust against affine distortions, a problem addressed by Affine-SIFT [42]. Further assumptions are made in the matching step: do we consider only unique keypoint matches? What about repetitive structures? Finally, the robust model estimation step is expected to tease out a correct geometric model. We believe that these assumptions play a major role in why feature-based approaches are currently incapable of matching images across very wide baselines.

2.2. Ultra-wide Baseline Feature-Based Matching

Ultra-wide baseline matching generally falls under the umbrella of correspondence matching problems. There have been several works on wide-baseline matching [35, 24]. For urban scenery, Bansal *et al*. [4] presented the Scale-Selective Self-Similarity (S^4) descriptor which they used to identify and match building facades for image geo-localization purposes. Altwaijry and Belongie [1] matched urban imagery under ultra-wide baseline conditions with an approach involving affine invariance and a controlled matching step. Chung *et al*. [9] calculate sketch-like representations of buildings used for recognition and matching. In general, these approaches suffer from poor performance due to the difficulty of the problem.

2.3. Convolutional Neural Networks

Neural Networks have a long history in the field of Artificial Intelligence, starting with [30]. Recently, Deep Convolutional Neural Networks have achieved state-of-the-art results and become the dominant paradigm in multiple fronts of Computer Vision research [19, 33, 34, 12].

Several works have investigated aspects of correspondence matching with CNNs. In [22], Long *et al*. shed some light on feature localization within a CNN, and determine that features in later stages of the CNN correspond to features finer than the receptive fields they cover. Toshev and Szegedy [37] determine the pose of human bodies using

CNNs in a regression framework. In their setting, the neural network is trained to regress the locations of body joints in a multi-stage process. Lin *et al*. [20] use a siamese CNN architecture to put aerial and ground images in a common embedding for ground image geo-localization.

The literature has seen a number of approaches to learning descriptors prior to neural networks. In [7], Brown *et al*. introduce three sets of matching patches obtained from structure-from-motion reconstructions, and learn descriptor representations to match them better. Simonyan *et al*. [32] learn the placement of pooling regions in image-space and dimensionality reduction for descriptors. However, with the rise of CNNs, several lines of work investigated learning descriptors with deep networks. They generally rely on a two-branch structure inspired by the siamese network of [6], where two networks are given pairs of matching and non-matching patches. This is the approach followed by Han *et al*. with MatchNet [14], which relies on a fully connected network after the siamese structure to learn the comparison metric. DeepCompare [43] uses a similar architecture and focuses on the center of the patch to increase performance. In contrast, Simo-Serra *et al*. [31] learn descriptors that can be compared with the L_2 distance, discarding the siamese network after training. These three methods relied on data from [7] to learn their representations. They assume that salient regions are already determined, and deliver a better approach to feature description for feature-based correspondence matching techniques. The question of obtaining CNN-borne correspondences between two input pairs, however, remains unexplored.

Lastly, attention models [26, 3] have been developed to recognize objects by an attention mechanism examining sub-regions of the input image sequentially. In essence, the attention mechanism embodies a saliency detector. In [16], the Spatial Transformer (ST) network was introduced as an attention mechanism capable of warping the inputs to increase recognition accuracy. In section 3.2 we discuss how we employ an ST module to let the network produce guesses for probable region matches.

3. Deep-Learning Architectures

3.1. Hybrid Network

We introduce an architecture which, given a pair of images, estimates the likelihood that they belong to the same scene. Inspired by the recent success of patch-matching approaches based on CNNs [43, 14, 31], we use a hybrid siamese/classification network. The network comprises two parts: two feature extraction arms that share weights (the *siamese* component) and process each input image separately, and a *classifier* component that produces the matching probability. For the siamese component we use the convolutional part of AlexNet [19], *i.e*., cutting off the fully connected layers. For the classifier we use a set of fully-connected layers that takes as input the concatenation of the siamese features and ends with a binary classifier, for which we minimize the binary cross-entropy loss. Fig. 3 illustrates the structure of the 'Hybrid' network.

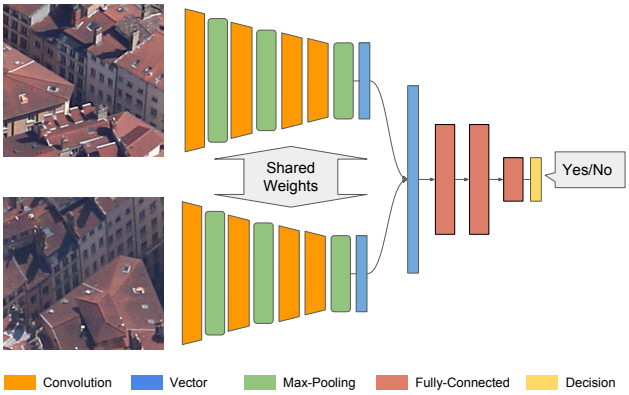

Figure 3. The siamese/classification Hybrid network. Weights are shared between the convolutional arms. ReLU and LRN (Local Response Normalization) layers are not shown for brevity.

The main motivation behind this design is that it allows features with local information from both images to be considered jointly. This is achieved where the two convolutional features are concatenated. At that layer, the features from both images retain correspondence to specific regions within the input images.

3.2. Hybrid++

Unlike traditional geometry-based approaches, the hybrid network proposed in the previous section does not model local similarity explicitly, making it difficult to draw conclusions about corresponding image regions. We would like to determine whether modeling local similarities more explicitly can produce more discriminative models.

We therefore sought to expand our hybrid architecture to allow for predictions of *probable* region matches, in addition to the classification task. To accomplish this, we leverage the Spatial Transformer (ST) network described in [16]. Spatial transformers consist of a network used for localization, which takes as input the image and produces the parameters for a pre-determined transformation model (*e.g*., translation, affine, etc.) which is used in turn to transform the image. It relies on a grid generator and a differentiable sampling kernel to keep track of the gradient propagation to the localization network. The model can be trained with standard back-propagation, unlike the attention mechanisms of [3, 26] that relied on reinforcement learning techniques. The *spatial transformer* is typically a standard CNN followed by a set of fully-connected layers with the required number of outputs, *i.e*., the number of transformation parameters, *e.g*., two for translation, six for affine.

The spatial transformer allows for any transformation as long as it is differentiable. However, in this work we

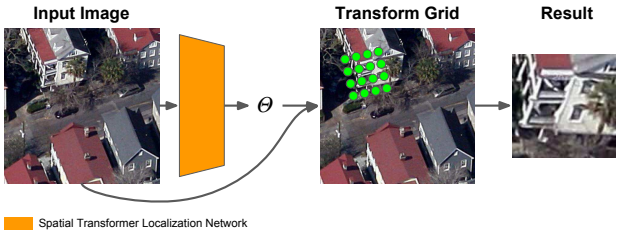

Figure 4. Overview of a Spatial Transformer module operating on a single image. The module uses the regressed parameters Θ to generate and sample a grid of pixels in the original image.

only consider extracting patches at a fixed scale, *i.e.*, translations, which are used to generate patch proposals over both images—richer models, such as perspective transformations, can potentially be more descriptive, but are also more difficult to train.

We build the spatial transformer with the same convolutional network used for the 'arms' of the siamese component of our hybrid network, plus a set of fully connected layers that regress the transformation parameters $\Theta = \{\Theta_1, \Theta_2\}$, which are used to transform the input images, effectively sampling patches. Note that patch locations for each individual image are a function of *both* images. The number of extracted patches is reflected in the number of regressed parameters specified. Fig. 4 illustrates how the spatial transformer module operates.

The spatial transformer modules allow us to explicitly model regions within each input image, permitting the network to propose similar regions given an architecture that demands such a goal. The overall structure of this model, which we call 'Hybrid++', is shown in Fig. 5.

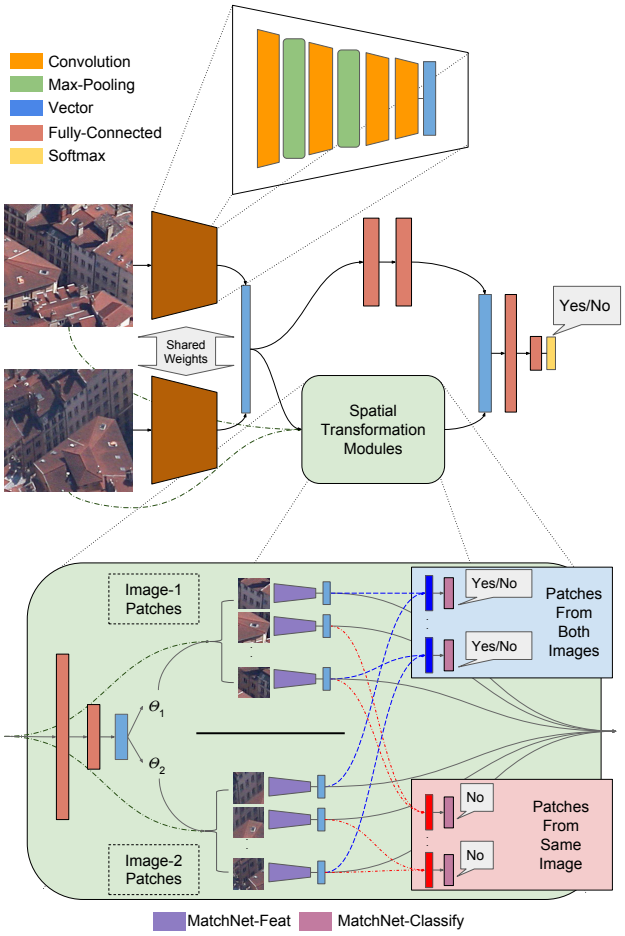

Figure 5. The 'Hybrid++' Network. Spatial Transformer modules are incorporated into the 'Hybrid' model to predict probable patch matches.

3.2.1 Describing Patches

In our model, we pair a ST module which produces a predetermined number of fixed-scale patch proposals with our hybrid network. The extracted patches are given to a MatchNet [14] network, which was trained with interest points from Structure-from-Motion data [7] and thus already has a measure of invariance against perspective changes built-in.

MatchNet has two components in its network, a feature extractor modeled as a series of convolutional layers, and a classifier network that takes the outputs of two feature extractors and produces a similarity score. We pass each extracted patch, after converting it to grayscale, through the MatchNet feature extractor network (MatchNet-Feat) and arrive at a 4096-dimensional descriptor vector.

These descriptors are then used for three different objectives. The first objective is to supplement the global feature description extracted by the original hybrid architecture. In this manner, the extracted descriptors provide the classifier with information extracted at a dedicated higher-resolution mode. The second objective is to match patches in the *other image*. This objective encourages the network to use the spatial transformer to focus on similar patches in both images simultaneously. The third objective is for the patch to *not match* other patches extracted from the *same image*, which we mainly use to discourage the network from collapsing onto a single patch. For the last two tasks, we use the MatchNet classification network (MatchNet-Classify).

3.2.2 Optimization

Combining the image-wise classification objective with the regional descriptor objectives yields an objective function with four components:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{L}_{\text{class}} + \alpha\mathcal{L}_{\text{patch}} + \beta\mathcal{L}_{\text{pairwise}} + \gamma\mathcal{L}_{\text{bounds}}\right) \quad (1)$$

where N is the size of the training batch and α, β, γ are used to adjust the weights. The first component of the loss

function encodes the image classification objective:

$$\mathcal{L}_{\text{class}} = y_i \log p_i + (1 - y_i) \log(1 - p_i) \quad (2)$$

where p_i is the probability of the images matching and $y_i \in \{0, 1\}$ is the label. The second component encodes the match of each pair of patches across both images:

$$\mathcal{L}_{\text{patch}} = \frac{1}{M} \sum_{m=1}^{M} \Big[y_i \log q_m + (1 - y_i) \log(1 - q_m) \Big] \quad (3)$$

where M is the number of patches, and q_m is the probability of patch $\mathbf{x}_m^1$ on image 1 matching patch $\mathbf{x}_m^2$ on image 2. The third component is a pairwise penalty function that discourages good matches among the patches within the same image, to prevent the network from collapsing the transformations on top of each other:

$$\mathcal{L}_{\text{pairwise}} = \frac{4}{M(M-1)} \sum_{t=1}^{2} \sum_{m=1}^{M} \sum_{k=m+1}^{M} \log(1 - u_{m,k}^t) \quad (4)$$

where $u_{m,k}^t$ is the probability of patch $\mathbf{x}_m^t$ matching patch $\mathbf{x}_k^t$ on image $t = \{1, 2\}$. The last component is a penalty function that discourages spatial transformations that fall out of bounds:

$$\mathcal{L}_{\text{bounds}} = \frac{2}{M} \sum_{t=1}^{2} \sum_{m=1}^{M} f(\mathbf{x}_m^t) \quad (5)$$

where $f(\mathbf{x}_m^t)$ is a function that computes the ratio of pixels sampled out of bounds for patch $\mathbf{x}_m^t$. The out-of-bounds loss term discourages the model from stepping outside the image, which may minimize the patch-matching loss, given an appropriate weight—with this penalty function we gain more control over the optimization process.

3.3. Training Procedure

To train the hybrid network, we follow a standard training procedure by fine-tuning the model after loading pre-trained AlexNet weights into the convolutional arms only. However, training the Hybrid++ network is more subtle, as the network needs to get started on the right foot. We initially train the non-ST and ST sides separately with the global *yes/no* matching signal. Afterwards, we train the networks jointly. We learned this is necessary to prevent the network from shutting off one side while minimizing the objective. Similar to the Hybrid case, we use pre-trained weights for the convolutional arms.

We use MatchNet as a pure feature descriptor, with frozen weights, *i.e.*, no learning. This is primarily done to prevent the network from minimizing the loss by changing the descriptors themselves without moving the attention mechanism. Our training procedure does not have pixel-to-pixel correspondence labels, and hence we do not know if the network is examining similar patches. We rely on the power provided by MatchNet to determine patch similarity. The global matching label in turn becomes a semi-supervised cue. Therefore, the network can only minimize the loss component for patch matching by moving the attention mechanism to examine patches that appear to be similar, as per MatchNet.

The reliance on MatchNet is a double-edged sword, as it is our only means of moving the attention mechanism without explicit knowledge of labeled patch correspondences. That means if MatchNet cannot find correspondence for two patches that do match, then the attention mechanism cannot learn to look for these two patches.

4. Experiments

4.1. Dataset

We compiled 49,271 matching pairs (98,542 images) of oblique aerial imagery through Google Maps [13]. The images were collected using an *automated process* that looks for planar surfaces such that the normal vector of the surface is within 40° to 75° of one cardinal direction. This guarantees the visibility of the surface from two different viewpoints. The pairs were collected non-uniformly from: San Francisco, Boston and Milan. Those locations were chosen with a goal of diversifying the scenery.

We split the dataset into roughly ∼39K/∼10K training/testing positive pairs. For training we generate samples in an online manner by sampling from the reservoir of positive matching pairs. The sampling procedure is set to produce samples with a 1:1 positive:negative ratio. Therefore, a random classifier would score 50% on the test-set. We call this the 'aerial' dataset.

4.2. Human Performance

We ask ourselves: How well do humans perform when matching such images? To this end, we conducted a small experiment with human participants on Amazon Mechanical Turk [2]. We picked a subset of 1,000 pairs from our test set and presented them to the human subjects. Each participant was shown 10 pairs of different images, and was asked to determine whether each pair showed the same area or building, as a binary question. We show a screenshot of the interface presented to the participants in Fig. 6. Each pair of images was presented at least 5 times to different participants, giving us a total of 5000 labels, 5 per pair.

Our interface was prone to adversarial participants, those answering randomly or giving a constant answer all the time. To mitigate the effect of unfaithful workers, we took the majority vote of the 5 labels per-pair. Human accuracy was then calculated to be 93.3%, with a precision of 98% and a recall of 89.4%.

We observed that the average response time for humans was less than 4.5 seconds/pair, with a minimum re-

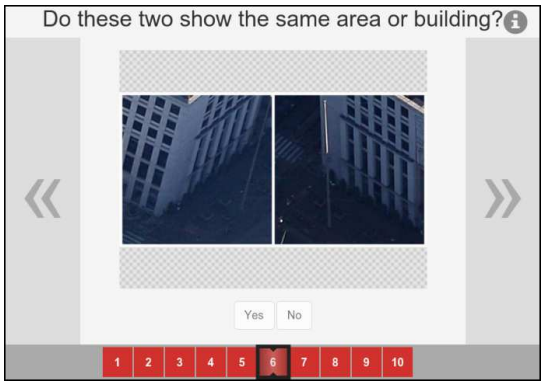

Figure 6. The user interface presented to our human subjects through Amazon Mechanical Turk.

sponse time of half a second. This quick response average prompted us to examine mislabeled pairs: we show examples of False-Positives in Fig. 7 and False-Negatives in Fig. 8. Most of the False-Positive pairs have a similar general structure, a cue that humans relied on hastily—notice that these examples require deliberate correspondence matching. This is a non-trivial, time-consuming task, which explains why the human subjects, who operate in an environment that favors lower response times, labeled them as False. This is also corroborated by the high precision and lower recall of the human labelers, which is another indication that humans are performing high-level image comparisons. All in all, we believe this indicates that the human participants were relying mostly on global appearance cues, which indicates the need for local correspondence matching.

4.3. Training Framework

We train our networks with *Torch7* [10]. We transplant weights in our models from the pre-trained reference model *CaffeNet* available from *Caffe* [18]. For the convolutional feature arms, we keep the AlexNet layers up to 'pool5' and discard the rest. The fully connected layers of our classifier component are trained from scratch. For the patch descriptor network, *i.e.*, MatchNet [14], we transplant the 'feature'-network and the 'classification'-network as-is and freeze the learning for both.

We use Rectified Linear Units (ReLU) for all our non-linearities, and train the networks with Stochastic Gradient Descent. The spatial transformer modules are trained specifically without momentum.

4.4. Spatial Transformer Details

The spatial transformer regresses $|\Theta| = 4n$ parameters, where n is the number of patches per image. Each 2 parameters are taken for an x-y location in the image plane in the range $[-1, 1]$. We specify a fixed-scale interpretation, where extracted patches are always 64×64, the resolution

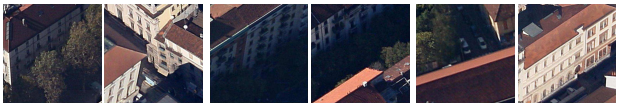

Figure 7. False-Positive pairs from the human experiment.

Figure 8. False-Negative pairs from the human experiment.

required by MatchNet.

In the Hybrid++ network, we remove the 'pool5' and 'conv5' layers provided by AlexNet from the convolutional arms, and learn a new 1×1 convolutional layer with an output size of $64 \times 13 \times 13$, performing dimensionality reduction from the 384-channel output of 'conv4'. The localization network takes a $2 \times 64 \times 13 \times 13$ input from the two convolutional arms and follows up with 3 fully-connected layers as follows: $21632 \rightarrow 1024 \rightarrow 256 \rightarrow 4n$. The initialization of the last fully-connected layer is not random; as recommended in [16], we initialize it with a zero-weight matrix and a bias specifying initial locations for the patches. In our experiments, we predict $M = 6$ patches per image, initialized to non-overlapping grid locations.

4.5. Matching Results

We compare our CNN models with a variety of baselines on the 'aerial' dataset. Our first baseline was a feature-based correspondence-matching method. We chose A-SIFT [42] as it offers all the capabilities of SIFT with the addition of affine invariance. In aerial images we mainly observe affine distortion effects, which makes A-SIFT's invariance properties particularly relevant. We use the implementation offered by the authors, which computes the matches and performs outlier rejection to estimate the fundamental matrix between the views, providing a yes/no answer, given a threshold. The accuracy of A-SIFT is better than random by 11%, but suffers from low accuracy for the positive samples (*i.e.*, low recall), as it is unable to find enough correspondences to perform the fundamental matrix estimation for a large number of positive pairs. This illustrates the difficulty of this problem with local correspondence matching.

Our second set of baselines are a measure of the performance of holistic representation methods used in the image classification and retrieval literature. We chose to compare the performance of GIST [27], Fisher Vectors [28], and VLAD [17]. The GIST-based classifier predicted most image pairs to be non-matching. Fisher Vectors surpassed A-SIFT performance by showing a better ability to recognize positive matches, but performed worse than A-SIFT in distinguishing negative pairs. VLAD performed the best out of these three holistic approaches with an average accuracy of 78.6%. For GIST we use the authors' implementation, and

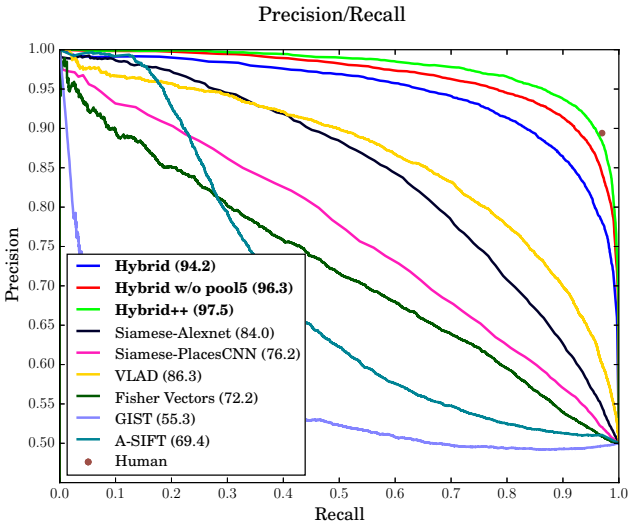

Figure 9. Precision/Recall curves for the 'aerial' dataset. The number between parenthesis denotes the average precision (%).

for Fisher Vectors and VLAD we use VLFeat [39].

The third set of baselines are vanilla CNN models used in a siamese fashion (without fine-tuning). We compare against AlexNet [19], trained on ImageNet, and PlacesCNN [44], which is an instance of the AlexNet architecture trained on the Places205 dataset [44]. We extract the 'fc7' layer outputs as descriptor vectors for input images, and use the L_2 distance as a similarity metric. This group of baselines explores the applicability of pre-trained networks as generic feature descriptors, for which there is mounting evidence [29]. Both CNNs performed well, considering the lack of fine-tuning. We note that while VLAD surpassed the performance of these two CNN approaches, both VLAD and Fisher Vectors require training with our dataset. This shows the power of CNNs generalizing to other domains.

Finally we measure the classification accuracy of our proposed architectures. Our Hybrid CNN outperforms all the baselines. A variant of the Hybrid CNN was trained without the 'conv5' and 'pool5' layers, with a 1×1 convolution layer after 'conv4' to reduce the dimensionality of its output. This variant outperforms the base Hybrid CNN by a small margin. Our Hybrid++ model with Spatial Transformers gives us a further boost, and performs nearly as well as the human participants in our study.

Table 1 summarizes the accuracy for every method, and Fig. 9 shows precision/recall curves, along with the average precision, expressed as a percentage.

4.6. Insights and Discussion

One of the main difficulties in the application of CNNs to real-world problems lies in designing and training the networks. This is particularly true for complex architectures with multiple components, such as our Hybrid++ network. In this section we discuss our experience and attempt to offer insights that may not be immediately obvious.

Method	Acc.	Acc. pos	Acc. neg	AP
Human*	.933	.894	.972	—
A-SIFT [42]	.613	.353	.874	.694
GIST [27]	.549	.242	.821	.553
Fisher Vectors [28]	.659	.605	.713	.722
VLAD [17]	.786	.769	.803	.863
Siamese PlacesCNN [44]	.690	.626	.754	.762
Siamese AlexNet [19]	.754	.697	.811	.840
Hybrid CNN	.881	.901	.861	.942
Hybrid w/o pool5	.909	.928	.891	.963
Hybrid++	.926	.927	.925	.975

Table 1. Classification performance on the 'aerial' dataset. AP denotes Average Precision. (*Human performance was measured on a subset of the samples.)

We obtained a small improvement by removing the 'pool5' layer from the AlexNet model, and replacing 'conv5' by a 1×1 dimensionality reduction convolution. We believe this is mainly due to the increased resolution of 13×13 presented to the classifier. This resolution would typically allow for more local detail to be considered jointly. In particular, this detail appears to be crucial to training the Hybrid++ model, as it provided the Spatial Transformer module with more resolution to work with. In Fig. 10 we show a sample of matched images with probable patch matches highlighted. Even with the increase in resolution, the receptive field for each neuron is still quite large in the original image space. This suggests that higher resolution features would be needed for finer localization of similar patches. This aspect is reflected in the network learning regions of interest for each of its attention mechanisms.

We attempted to use transformations with more degrees of freedom with the Spatial Transformer module, such as affine transforms, but we found the task increasingly difficult without higher levels of supervision and additional constraints. This was the origin of our 'out-of-bounds' penalty term. For example, the network would learn to stretch parts of each image into seemingly similar looking patches, effectively minimizing the pairwise patch similarity loss term.

To train the pairwise patch similarity portion of the network, we only have the image-level match label, with no information regarding pixel-wise correspondence. It might seem unclear what target labels should be presented to the pairwise similarity loss. However, by studying the loss function we can see that the attention mechanism would not be able to find matching patches unless we actively look for correspondences; hence it is sensible to use the image-level label for patch correspondence. Given that MatchNet modules are frozen, the network will not induce a high loss for non-corresponding patches over negative samples, but only for non-corresponding patches over positive samples.

Figure 10. Image pairs from 'aerial', matched with Hybrid++. The overlaying boxes indicate patch proposals. Red boxes denote patches that do not match, according to MatchNet. Boxes with colors other than red indicate matches, with the color encoding the correspondence.

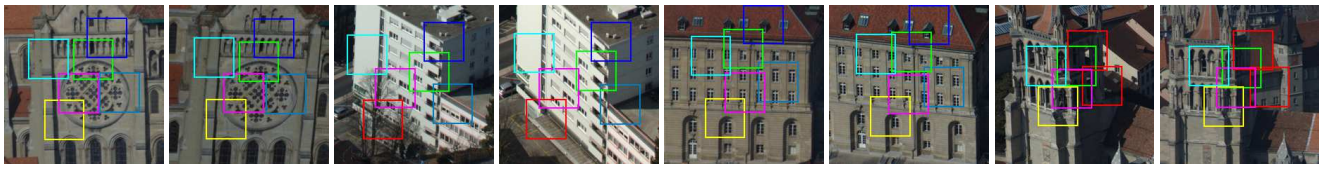

Figure 11. Image pairs from 'Lausanne', matched with Hybrid++. Color coding follows the same conventions are the figure above.

4.7. Investigating the Spatial Transformers

The patch proposal locations of Fig. 10 are meaningful from pair to pair, and across the images for a given pair. However, while the baseline between the two images in a pair is very large, it does not change much from pair to pair—an inevitable artifact of the dataset collection process. This results in patch proposals with similar configurations and raises questions about the Spatial Transformers.

We thus set up a second experiment to study the effect of varying viewpoint changes explicitly. To this end we used several high-resolution aerial images from the city of Lausanne, Switzerland, to build a Structure-from-Motion dataset [41] and extract corresponding patches, with 8.7k training pairs and 3.6k test pairs. Patches were extracted around SIFT locations and are thus significantly easier to match than those in the 'aerial' dataset. However, the viewpoint changes from pair to pair are much more pronounced.

We followed the same methodology as before to train our models on this new dataset. In Fig. 11 we show different pairs from the new dataset, along with the probable patch matches suggested by the model. The model learns to predict patch locations that are consistent with the change in perspective, while also differing from pair to pair. MatchNet results on the proposals corroborate the findings when the contents of those patches do match (non-red boxes), and when they do not (red boxes). Numerical results are provided in Table 2. As this data is significantly easier, the baselines (notably A-SIFT) perform much better, but our method achieves the highest accuracy of 96%. The performance gain from Hybrid to Hybrid++ is however negligible.

Method	Acc.	Acc. pos	Acc. neg	AP
A-SIFT [42]	.947	.896	.998	.968
GIST [27]	.856	.798	.914	.937
Fisher Vectors [28]	.769	.723	.816	.867
VLAD [17]	.898	.867	.930	.965
Siamese PlacesCNN [44]	.690	.626	.754	.958
Siamese AlexNet [19]	.754	.697	.811	.968
Hybrid CNN	.959	.960	.957	.992
Hybrid++	.959	.962	.956	.992

Table 2. Classification performance on the 'Lausanne' dataset.

5. Conclusions and Future Work

We present two neural network architectures to address the problem of ultra-wide baseline image matching. First, we fine-tune a pre-trained AlexNet model over aerial data, with a siamese architecture for feature extraction, and a binary classifier. This network proves capable of discerning image-level correspondence, but is agnostic to local correspondence. We then show how to integrate Spatial Transformer modules to predict probable patch matches in addition to the classification task, which further boosts performance. Our models achieve state-of-the-art accuracy in ultra-wide baseline matching, and close the gap with human performance. We also demonstrate the adaptability of our approach on a new dataset with varied viewpoint changes which the ST modules can adapt to.

This work is a step towards bridging the gap between neural networks and traditional image-matching techniques based on local correspondence, in a framework that is trainable end-to-end. We intend to build on it in the following directions. First, we plan to explore means to increase the resolution of the localization network to obtain finer-grained patch proposals. Second, we plan to replace MatchNet with 'descriptor' networks trained for this specific purpose. Third, we are interested in richer transformations for the ST modules, *e.g.*, affine, and in exploring constraints in order to do so. Finally, we want to study the use of higher supervision for a better feature-localization step, bringing neural networks closer to local correspondence techniques.

Acknowledgments

We would like to thank Kevin Matzen and Tsung-Yi Lin for their valuable input. This work was supported by the KACST Graduate Studies Scholarship and EU FP7 project MAGELLAN under grant number ICT-FP7-611526.

References

[1] H. Altwaijry and S. Belongie. Ultra-wide baseline aerial imagery matching in urban environments. In *BMVC*, 2013. 2

[2] Amazon.com. Amazon mechanical turk. 5

[3] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. In *ICLR*, 2015. 3

[4] M. Bansal, K. Daniilidis, and H. Sawhney. Ultra-wide baseline facade matching for geo-localization. In *ECCV*, 2012. 2

[5] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. In *ECCV*, 2006. 2

[6] J. Bromley, I. Guyon, Y. Lecun, E. Sckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. In *NIPS*, 1994. 3

[7] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *PAMI*, 2011. 3, 4

[8] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a local binary descriptor very fast. *PAMI*, 2012. 2

[9] Y.-C. Chung, T. Han, and Z. He. Building recognition using sketch-based representations and spectral graph matching. In *ICCV*, 2009. 2

[10] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A MATLAB-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 6

[11] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 1981. 2

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[13] Google Inc. Google maps. 2, 5

[14] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015. 2, 3, 4, 6

[15] C. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, 1988. 2

[16] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *NIPS*, 2015. 1, 3, 6

[17] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 6, 7, 8

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 6

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*. 2012. 2, 3, 7, 8

[20] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *CVPR*, 2015. 3

[21] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT Flow: Dense correspondence across different scenes. In *ECCV*, 2008. 2

[22] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *NIPS*, 2014. 2

[23] D. G. Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999. 2

[24] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002. 2

[25] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 2005. 2

[26] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In *NIPS*, 2014. 3

[27] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV 2001*. 6, 7, 8

[28] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 6, 7, 8

[29] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshop*, 2014. 7

[30] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958. 2

[31] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015. 2, 3

[32] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *PAMI*, 2014. 3

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015. 2

[34] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 2

[35] D. Tell and S. Carlsson. Combining appearance and topology for wide baseline matching. In *ECCV*, 2002. 2

[36] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *PAMI*, 2010. 2

[37] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 2

[38] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer. Dense segmentation-aware descriptors. In *CVPR*, 2013. 2

[39] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. In *ACM International Conference on Multimedia*, 2010. 7

[40] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *ICCV*, 2013. 2

[41] C. Wu. Towards linear-time incremental structure from motion. In *3DV*, 2013. 8

[42] G. Yu and J.-M. Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*, 2011. 2, 6, 7, 8

[43] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. *CVPR*, 2015. 2, 3

[44] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 7, 8