Cross-Time and Orientation-Invariant Overhead Image Geolocalization Using Deep Local Features

Yuxin Tian Xueqing Deng Yi Zhu Shawn Newsam
Electrical Engineering and Computer Science, University of California, Merced
{ytian8, xdeng77, yzhu25, snewsam}@ucmerced.edu

Abstract

Overhead image geolocalization is becoming increasingly important due to the growing collection of drone imagery without location information. In this paper, we perform large-scale overhead image geolocalization by matching a query image to wide-area reference imagery with known location. We use deep local features so that the query image need not align with but only overlap the tiled reference imagery. We further address two key challenges. For when the query and reference imagery are from different dates, we perform cross-time geolocalization using time invariant features learned using a Siamese network. For when the query and reference imagery are oriented differently, we introduce an orientation normalization network. We demonstrate our contributions on two new high-resolution overhead image datasets. Our method significantly outperforms strong baselines on cross-time geolocalization and is shown to exhibit promising orientation invariance.

1. Introduction

While there has been a fair amount of work on locating ground level imagery [1, 10, 17, 21, 29, 32], there has been little work on the overhead case [6]. However, we believe this is an increasingly important problem due to the ease with which anyone can capture overhead imagery using drones and share it online. While location information typically accompanies traditional overhead imagery, such as from satellite and aerial platforms, location information is often missing or unreliable for drone imagery. It might become lost as the imagery is distributed or deliberately obscured. Our focus on overhead imagery geolocalization is thus timely and important.

This paper focuses on the problem of geolocating overhead imagery captured from satellite, aerial, or drone platforms. By geolocating we mean assigning geographic coordinates such as latitude and longitude values. We allow the "search region" to be large and so this is a difficult problem. As shown in Figure 1, we formulate the problem as matching a query image to tiled wide-area reference imagery with known location. We address two fundamental challenges: the query and reference imagery 1) might have been taken at difference times, and 2) might be oriented differently.

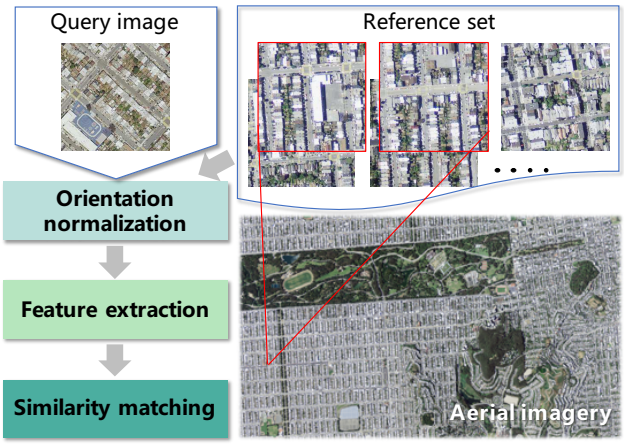


Figure 1: We perform overhead image geolocalization by matching a query image to reference imagery with known location. We address two fundamental challenges: cross-time matching and orientation-invariant matching. Compare the query image with the reference set above.

We exploit recent advances in deep learning, particularly convolutional neural networks (CNNs), to perform the image matching. We show that, as expected, global image features extracted using the fully connected (*fc*) layers are not appropriate due to query-reference tile misalignment and so we instead derive local features from the locality preserving feature maps of the convolutional (*conv*) layers. We show these deep local features significantly outperform traditional local features, such as Scale Invariant Feature Transform (SIFT) features [18], when the query and reference images are from different dates. We next develop a Siamese network to explicitly learn time-invariant features to make our approach even more robust to changes in season, illumination, and sensors, and to changes in what is on the ground. Finally, we tackle the real but challenging problem of when the query and reference imagery are ori-

ented differently (*e.g*., not both pointing north). For this, we develop an Orientation Normalization Network (ONN) that rotates the query and reference imagery to the same canonical orientation. We demonstrate our methods on two new high-resolution overhead image datasets.

The key and novel contributions of our work include:

- We perform large-scale overhead geolocalization via image matching using learned time-invariant deep local features.
- We propose an Orientation Normalization Network to account for when the query and reference imagery are oriented differently.
- We introduce two high-resolution overhead image datasets which will be made publicly available for other researchers.

2. Related Work

Image geolocalization. Estimating the geographic location of an image has been of interest to the computer vision community for some time [1, 14, 15, 16, 22, 23, 24, 28, 30]. However, the focus has been mostly on geolocating *ground level imagery* which is a related but different problem than ours, and has a different set of challenges. Ours is one of the first works to focus on overhead image geolocalization.

Ground level imagery has been geolocated by matching it to maps in geographic information systems (GIS) [3], to other ground level imagery with known location [1, 10, 17, 21, 29, 32], to overhead imagery [2, 11, 15, 16, 26, 28, 30, 33], or to combinations of this reference data [14].

Geolocating a ground level query image by matching it to ground level reference imagery is limited to regions where reference imagery is available such as in urban areas [19, 20, 27] or along roads. It also typically assumes the query and reference images are both oriented with the sky at the top. We instead can geolocate overhead imagery from anywhere and which might be oriented differently from our reference imagery.

The fundamental challenge to geolocating a ground level query image by matching it to overhead imagery is the difference in perspective and so most of the work on this problem focuses on cross-view matching [15, 16, 24]. For example, Shi *et al*. propose a novel Cross-View Feature Transport (CVFT) layer to facilitate feature alignment between ground and aerial domains [24]. In contrast, our query and reference imagery are taken from the same viewpoint and so we face a different set of challenges.

We also expect to be able to geolocate overhead imagery more accurately than ground level imagery.

We know of only one other work on geolocating overhead imagery [6]. It also uses an image matching framework but uses traditional local features and does not address the cross-time or orientation-invariant cases. We include it as one of our baselines.

Orientation alignment. The concept of orientation is very different for overhead images than for images taken at ground level. Most ground level images have a canonical orientation [7]. Street view images and the like typically have the ground at the bottom and sky at the top. Most objects have a canonical orientation when viewed from the side which then dictates the canonical orientation of the image [28]. In fact, researchers have exploited this fact to learn better representations in an unsupervised manner [8]. In contrast, overhead imagery typically does not have a canonical orientation. While most overhead imagery is oriented so that north points up, this has nothing to do with the content of the image and cannot be derived from it in the general case. There has been work on classifying rotation agnostic images [7] in the ImageNet dataset [5] by splitting the image representation into rotation related and unrelated parts. This, however, produces global features which are not appropriate for our problem.

CNNs are inherently limited in their ability to model geometric transformations due to the fixed geometric structure of their constituent modules [4]. Modules have been proposed that enable spatial manipulation, including rotation, of data within the networks [4, 12, 31]. We utilize one such module, Spatial Transformer Networks [12], in our Orientation Normalization Network below.

Image retrieval. Our matching framework has many similarities with image retrieval methods. We distinguish it, though, from the following two main image retrieval paradigms. Similarity-based image retrieval methods seek to retrieve similar images and not necessarily images of the same scene. Retrieving similar images is not sufficient to geolocate overhead imagery since many locations might look very similar from above. Indeed, our results show that even when our matching framework fails, it still retrieves similar images. We need our matching to be more discriminating (yet still allow for differences due to time and orientation). Image retrieval has been used to geolocate ground-level imagery by matching it against ground-level images of the same scene. This is the approach taken by Radenovic *et al*. [21] using a method called fine-tuning image retrieval (FITR). These approaches tend to use global features though, which, as we will demonstrate, are not effective for our problem. We include FITR as one of our baselines.

3. Methodology

We formulate overhead image geolocalization as an image matching problem in which a query image is matched to wide-area reference imagery with known location. We assume the search area is covered by one contiguous reference image even though, in reality, it will be a registered mosaic of large but individually acquired images. In order to localize the matching, we partition the contiguous reference image into tiles the same size as the query (this also enables easy parallelization). Our problem thus reduces to

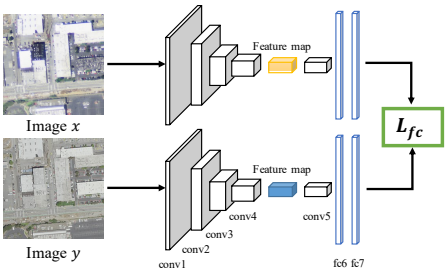


Figure 2: Siamese network for learning features for cross-time matching. The positive training samples are co-located images from different times.

finding a good representation $F(.)$ for overhead imagery so that given a query image q, we are able to find at least one spatially overlapping tile r from reference set R by computing the distance between $F(q)$ and $F(r)$.

Several things make this challenging. First, the query image is randomly located and thus not aligned with any of the reference tiles. The query image overlaps several of the reference tiles but by varying amounts and so we have to be able to perform matching based on this varying overlap. Second, the query and reference imagery might have been taken at different times, for example, when geolocating current drone imagery using archived satellite imagery. And, third, they might not have the same orientation. We describe our novel technical contributions to overcome these challenges in the following.

3.1. Deep Local Features

CNNs have proven effective at mapping images to powerful and often semantically rich feature vectors [13, 34]. Most work utilizes global features extracted from the fully connected layers including the work mentioned above on geolocating ground level imagery by matching against ground or aerial images [1, 15, 16, 21, 24, 28]. However, since our query and reference tiles only overlap, using global features to perform the matching is unlikely to be effective. Our results below demonstrate this.

We instead extract deep local features from the *conv* layers since locality is preserved in the feature maps. We split these feature maps along the channel dimension to produce a set of deep local features. Specifically, given an image x, we apply a trained CNN to compute a *conv* layer output $F(x)$ of size $H \times W \times C$, where $H \times W$ are the spatial dimensions of the feature map and C is the number of channels. $F(x)$ is then split into a set of $H \cdot W$ vectors of length C. We denote these features as $\mathbf{p}_x^i$ where x is the image and i is the feature number which is in the range $(1, H \cdot W)$. Each image x, either query or reference, is thus represented by the set of deep local features $S_x = \{\mathbf{p}_x^i\}_{i=1}^{H \cdot W}$.

Matching between a query image and a set of reference tiles is then performed by finding, for each of the query's features, the nearest neighbor in feature space among all the features of the all reference tiles. Each nearest neighbor

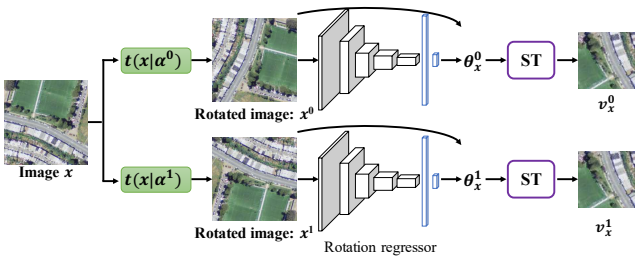


Figure 3: Orientation normalization network (ONN) which learns a rotation regressor to transform differently oriented images of the same location to the same orientation. ST is a spatial transformer layer.

match votes for a reference tile and the votes are accumulated over all the query image's features to rank the reference tiles. Specifically, given a query image q with local features $\mathbf{p}_q^i$ and a set of reference tiles $r \in R$, each with local features $\mathbf{p}_r^i$, for each $\mathbf{p}_q^i$, we use the Euclidean distance to find the nearest neighbor:

$$\mathbf{p}_r^j = \arg \min_{r \in R, j=1,\ldots,H \cdot W} \|\mathbf{p}_q^i - \mathbf{p}_r^j\|_2. \quad (1)$$

This will result in a vote for reference tile r. We then rank the reference tiles in order of decreasing votes and pick the *top one* as the match for query image q. That is, we use only the best match among all the reference tiles to geolocate the query tile even though it overlaps multiple reference tiles. (See Figure 6.)

We first investigate deep local features extracted using a VGG16 network [25] trained on the ImageNet dataset [5]. These features are not specific to overhead image matching nor are they invariant to potential time differences between the query and reference images. One of our key technical contributions therefore is a Siamese network which learns improved deep local features specific to overhead imagery and for cross-time matching.

3.2. Siamese Network for Cross-Time Matching

Our proposed Siamese network is shown in Figure 2. It consists of two embedding CNNs that share weights. During training, the network is presented with either a pair of images from the same geographic location but taken at different times (positive examples) or a pair of images from different locations (negative examples). Positive examples are shown in Figure 5. The goal of the Siamese network is to learn a feature representation (non-linear embedding) $g(.)$ such that images from different locations are far apart in feature space while images from the same location are close *even if they are from different times*. This is done by training the network to minimize a contrastive loss [9]

$$L_{fc} = \frac{1}{2} l D^2 + \frac{1}{2}(1-l) \max\left(0, \left(m - D^2\right)\right), \quad (2)$$

where $l \in \{0, 1\}$ is the label indicating whether the input pair x, y is from the same location ($l = 1$) or not ($l = 0$), D^2 is the squared distance between $g(x)$ and $g(y)$, and m is the margin parameter that omits the penalty if the distance between images from different locations is too large.

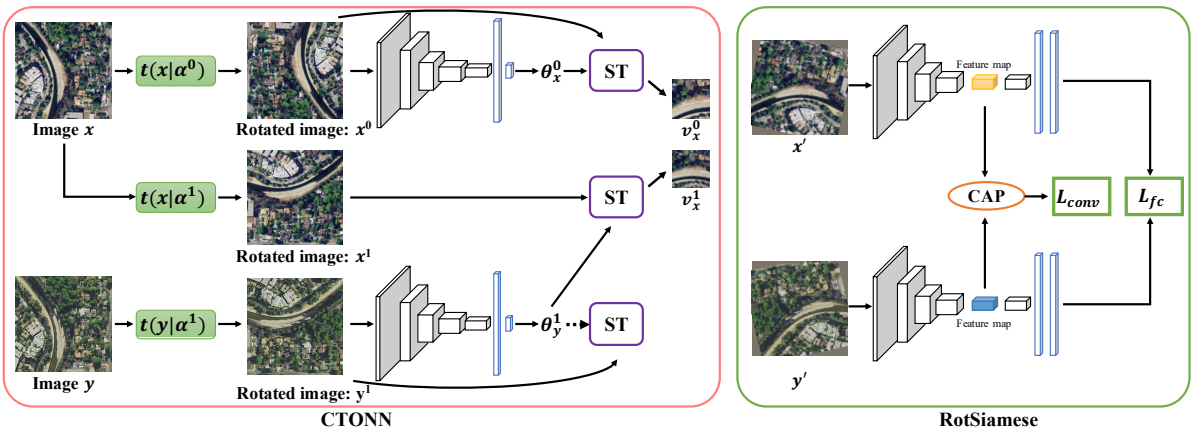


Figure 4: Our cross-time orientation normalization network (CTONN) learns a regressor that can orient images at different orientations and from different times to the same canonical orientation. Our RotSiamese network provides rotation invariance to deal with the noisy output of the CTONN at inference time.

Structurally, our Siamese network consists of two pre-trained VGG16 networks which we modify for fine-tuning on our overhead imagery. We remove the last fully connected layer $fc8$ and use the 4096-dim feature from $fc7$ to compute the Euclidean distance between $g(x)$ and $g(y)$. We investigate deep local features extracted from conv1 to conv4 of the trained embedding network in our experiments.

3.3. Orientation Normalization Network

A fundamental challenge to performing overhead image geolocalization through image matching is that the query and reference imagery typically do not have the same orientation. While the reference imagery is usually oriented northwards, the orientation of the query image is generally arbitrary and unknown. Further, unlike ground level imagery, which has a standard orientation (such as the sky is up) that can be estimated and exploited by works like [8], there is no such standard orientation that can be estimated from overhead imagery.

Therefore, instead of trying to reorient the query image to a standard orientation so that it matches the reference imagery, we instead reorient both the query and reference to the same, potentially arbitrary direction. We seek a framework that can estimate such scene-specific canonical orientations.

Figure 3 shows our framework for learning a network that can be used to normalize the orientation of overhead images. This framework takes as input differently rotated versions of an overhead image and learns a rotation regressor that aligns the images. Specifically, we define a set of K discrete rotation transformations $T = t(.|\alpha^k)_{k=1}^K$, where $t(.|\alpha^k)$ is the operator that applies to image x the rotation transformation with angle α^k that yields the rotated image $x^k = t(x|\alpha^k)$. The α^k are evenly sampled from $0°$ to $360°$ depending on K. We investigate the choice of K in the experiments.

The goal of the rotation regressor in Figure 3 is to predict angles θ_x^0 and θ_x^1 such that when input image x^0 is rotated by θ_x^0, it has the same orientation as input image x^1 rotated by θ_x^1. If image x^0 was derived by rotating x by α^0 and image x^1 was derived by rotating x by α^1 , then the rotation regressor can be learned by minimizing the loss function L_θ

$$L_\theta = \left|\left(\alpha^0 + \theta_x^0\right) - \left(\alpha^1 + \theta_x^1\right)\right|, \tag{3}$$

where θ_x^k is the predicted angle for the rotated image x^k. (Note that the rotation regressor has no knowledge of α^0 and α^1.)

However, this objective alone leads to a trivial solution which predicts $\theta = 0$ regardless of the input. So, we modify the network to also compare the normalized images during training, that is the similarity of image x^0 rotated by θ_x^0 and image x^1 rotated by θ_x^1. We do this by inserting a spatial transformer (ST) layer [12] to produce images v_x^0 and v_x^1 (see Figure 3). Here, $v_x^k = ST\left[x^k|(\theta_x^k, rr)\right]$, where v_x^k denotes the transformed image whose input is x^k with the predicted rotation angle θ_x^k, and rr denotes the reduced ratio to crop the center of the rotated image in order to avoid introducing blank regions in the corners.

The rotation regressor is then learned by minimizing the joint loss function

$$L = \left|\left(\alpha^0 + \theta_x^0\right) - \left(\alpha^1 + \theta_x^1\right)\right| + \lambda_v \left|v_x^0 - v_x^1\right|, \tag{4}$$

which includes the L1 loss between images v_x^0 and v_x^1. The weighting parameter λ_v is chosen empirically.

We implement the rotation regressor using a VGG16 convolutional backbone followed by a two-layer regressor module to produce the angle θ. A scaled Tanh activation layer is appended to the regression model to constrain θ to meaningful values.

3.4. Cross-Time Orientation-Invariant Matching

We are only able to train our Orientation Normalization Network (ONN) using differently oriented images from the

Figure 5: Co-located cross-time training pairs. Top: 2012. Bottom: 2014.

same time due to the sensitivity of the L1 loss to time-related differences. The learned rotation regressor is effective for normalizing images from the same time but has difficulties with images from different times. We therefore develop the cross-time ONN (CTONN) framework shown in Figure 4 left. This network now takes co-located image pairs x and y from different times and separately applies the random rotations α^0 and α^1 to produce x^0 and y^1. The rotation regressor is applied to predict θ_x^0 and θ_y^1 from these images. The spatial transformer module now applies rotation θ_y^1 to x^1 instead of y^1 to produce v_x^1, where x^1 is x rotated by θ_y^1 (the amount y was rotated to get y^1). v_x^0 and v_x^1 are now from the same year and can be compared using L1. The loss function for training CTONN becomes

$$L = \| \left(\alpha^0 + \theta_x^0\right) - \left(\alpha^1 + \theta_y^1\right) \| + \lambda_v \left| v_x^0 - v_x^1 \right|. \quad (5)$$

The proposed CTONN results in a rotation regressor that is more effective for normalizing the orientations of images from different times. However, the normalized images are still not aligned well enough for our cross-time feature extractor, which was trained using images with the same orientation. We therefore need to make our cross-time feature extractor more robust to these slight misalignments.

We develop the second Siamese network shown in Figure 4 right to make our feature embedding network more orientation invariant. We refer to this network as RotSiamese (RotSia for short). This network learns to extract deep local features that are more orientation invariant through 1) the addition of another loss term, and through 2) data augmentation. Specifically, the input images (same location different time) are separately rotated by small random angles sampled from a limited range $(-\phi, \phi)$ before being fed into the embedding network. This data augmentation alone does not result in improved orientation invariance as the loss function computed on the global features is not sensitive to slight differences in orientation. We therefore modify the loss to also compare the convolutional feature maps. We perform average pooling along the channel dimension (CAP) of the $conv$ layer that we use for deep feature extraction and compare these averages.

Specifically, given two rotated images x' and y', feature maps $F(x')$ and $F(y')$ are extracted from the $conv$ layer. The pooled feature maps $CAP(F(x'))$ and $CAP(F(y'))$ are then flattened and compared using the Euclidean distance: $D^2_{conv} = \|CAP(F(x')) - CAP(F(y'))\|$. This is then incorporated into a contrastive loss L_{conv}

$$L_{conv} = \frac{1}{2} l D^2_{conv} + \frac{1}{2}(1-l)\max\left(0, \left(m - D^2_{conv}\right)\right). \quad (6)$$

Finally, the overall objective of the RotSiamese network is the weighted sum of this loss and the original one

$$L = L_{fc} + \lambda_c L_{conv}. \quad (7)$$

The weighting parameter λ_c is chosen empirically. Note that the CTONN and RotSiamese networks are trained separately but training them together in an end-to-end manner could be future work.

Dataset	2012		2014	
	SF	LA	SF	LA
Query	800	900	800	900
Reference	5569	6525	5569	6525

Table 1: The number of 256×256 pixel tiles in our dataset.

3.5. Geolocalization pipeline

Again, we perform geolocalization by matching the features of the query image to the features of the reference tiles and pick the top match through voting. When the query and reference tiles are from different times and have different orientations, we first perform orientation normalization on each tile separately using our trained CTONN and then extract deep local features using the feature embedding from the trained RotSiamese network. Figure 1 illustrates this pipeline. Note that the features can be pre-computed offline for all the reference tiles.

4. Experiments

In this section, we first introduce two new high-resolution overhead image datasets and describe our implementation details. We then we demonstrate our results on the cross-time, orientation-invariant overhead image geolocalization problem with comparison to strong baselines.

4.1. Dataset

We use high-resolution aerial imagery from the National Agriculture Imagery Program (NAIP) for our experiments. The images have a ground sample distance (GSD) of one meter (spatial resolution is 1m/pixel) and measure approximately $6k \times 7k$ pixels. We download eight pairs of spatially contiguous NAIP images from the San Francisco area and nine pairs from Los Angeles area. Each pair of images consists of co-located images but taken at different times, one in 2012 and the other in 2014. These pairs thus form our cross-time dataset. The reference datasets are constructed by partitioning the NAIP images into non-overlapping tiles measuring 256×256 pixels. The query images are not aligned with these reference tiles but are randomly extracted from the NAIP images and also measure 256×256 pixels. Table 1 summarizes the dataset. We will make our dataset publicly available.

Figure 6: A sample query in red and its ground truth in yellow. Geolocalization is successful if the top ranked image in the matched reference set overlaps the query image.

During training, the Siamese networks and the cross-time ONN require pairs of co-located images from different times. We thus construct a training set of $12k$ pairs for SF and $13.5k$ pairs for LA. Examples of training pairs are shown in Figure 5. The negative training samples are cross-time images from different pairs to ensure they are not co-located.

4.2. Implementation Details

Siamese networks We fine-tune the Siamese models using an Adam optimizer with a batch size 24. We set the initial learning rate to 10^{-4} for the fc layers and to 10^{-5} for other layers. The learning rate is decayed by 0.1 every 30 epochs. For the RotSiamese network, we set λ_c to 1 and ϕ to either $10°$ or $20°$.

Orientation normalization networks For training the rotation regressor, we use an Adam optimizer with a batch size 24 and an initial learning rate of 2×10^{-5}. We decrease the learning rate by a factor of 10 every 30 epochs. The last Tanh function is scaled by a factor of 1.5π. For ONN training, we set reduce radio rr to $150/224$ and λ_v to 1. For CTONN training, we set rr to $28/224$ and λ_v to 0.1. We experiment with different sets of rotation transformations for training the rotation regressor (parameter K in Section 3.3). We consider sets of size 4, 8, and 36 corresponding to multiples of $90°$, $45°$, and $10°$ respectively. In order to avoid introducing blank regions into the corners of the rotated images, we rotate images of size 370×370 pixels and then extract images of size 256×256 from the center.

Evaluation metrics We consider the geolocalization to be correct if *the top ranked reference tile overlaps the query image*. As shown in Figure 6, the ground truth for the red query image is the four reference tiles in yellow since picking any of these tiles would geolocate the query. Using only the top ranked image corresponds to top-1 accuracy which is quite strict. In practice, the top-n ranked images could be marked as candidates and the user could easily make the final selection manually. This would greatly increase performance with modest manual effort. In the case of a correct geolocalization using our method, we assume that image registration could be used to determine the exact location of the query image (such as the geographic coordinates of its corners) using the overlapping reference tiles.

The accuracy for a set of queries is the percentage of successful searches for that set. This is the metric that we report below.

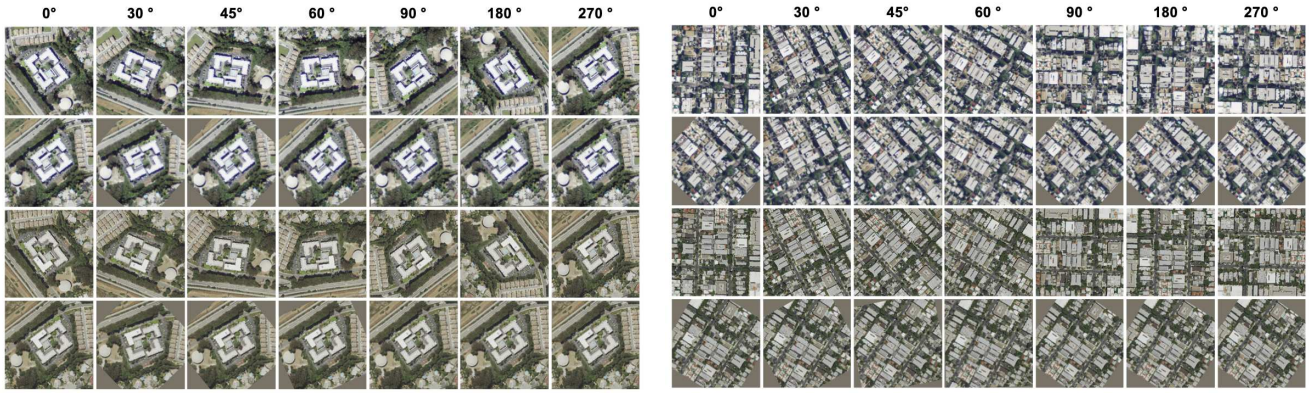
Model	conv3	conv4	conv5	fc6	fc7
Same-time	100	100	91.63	64.00	65.25
Cross-time	76.50	70.63	31.50	19.13	18.50

Table 2: Results of performing geolocalization in the SF dataset using features extracted from various layers of a VGG16 network trained on ImageNet. Top: the query and reference images are from the same time; Bottom: they are from different times.

4.3. Results

Global vs. local CNN features We first compare global versus local features extracted using a VGG16 model trained on the ImageNet dataset. Table 2 compares the performance of deep global features extracted from the *fc* layer and deep local features extracted from various *conv* layers. (See Section 3.1 for details on how these features are extracted.) These results are for the SF dataset and from when the query and reference tiles have the same orientation. The top row corresponds to when the query and reference tiles are from the same year and the bottom row to when they are from different years (cross-time). We draw three conclusions. First, as expected, the local features significantly outperform the global features due to the query and ground truth reference tiles only overlapping (see Figure 6). Second, the deep local features extracted from *conv3* or *conv4* significantly outperform those from *conv5* especially in the cross-time case. This indicates the features in the final *conv* have possibly become too specialized. (When we train our VGG16 networks on the overhead imagery, the features from *conv4* turn out to be optimal so that is what is used in the experiments below.) Finally, the performance is significantly worse in the cross-time case indicating these features possess limited time-invariance.

Cross-time features and baselines Table 3 compares our cross-time features trained using the Siamese network to several baselines: NetVLAD [1], fine-tuning image retrieval (FTIR) [21], and SIFT [6]. We also copy the results from the global (VGG_fc) and local (VGG_conv) features extracted using the VGG16 network trained on the ImageNet dataset. The query and reference tiles again have the same orientation. NetVLAD and FTIR are global features and so again perform poorly. Matching using the local SIFT features is also done through voting. While the SIFT features work well in the same-year case, they per-



(a) Successful examples (b) Failed examples

Figure 7: (a) Successful and (b) failed orientation normalization examples. The first and third rows contain co-located images from years 2012 and 2014 at various orientations. The second and fourth rows contain the normalized images.

Method	SF		LA	
	Same year	Cross year	Same year	Cross year
VGG_fc [25]	65.25	18.50	56.89	7.33
NetVLAD [1]	63.63	30.00	65.22	14.00
FTIR [21]	57.11	38.75	55.22	32.33
SIFT [6]	99.75	61.00	100	39.22
VGG_conv [25]	100	70.63	100	47.44
Siamese	100	**82.50**	99.89	**74.11**

Table 3: Comparison of our cross-time features (Siamese) with several baselines.

form poorly in the cross-year case, indicating they too possess limited time-invariance. Our cross-time deep local features trained using the Siamese network (Section 3.2) are shown to outperform all other approaches especially in the cross-year case. The improvement over VGG_conv in the cross-year case in particular demonstrates the effectiveness of our cross-time Siamese training framework.

ONN: orientation invariance Table 4 shows the results when the query and reference tiles are oriented differently. The columns indicate the difference in orientation between the query and reference: $90°$, $180°$, $270°$, or an arbitrary angle randomly sampled from $(0°, 360°)$. The rows indicate different configurations: no ONN corresponds to no orientation alignment and the other rows indicates the sizes of the sets of rotations the ONN is trained with (K in Section 3.3). For example, in the 4 classes cases, the ONN is trained with images at four rotations: $0°$, $90°$, $180°$, and $270°$.

We first focus on the same-time case (top of Table 4). The no ONN results show just how difficult geolocalization becomes when the query and reference are not oriented the same. Our proposed ONN is shown to significantly improve performance. In particular, the ONN trained with 36 difference rotations achieves around 90% accuracy even in the difficult case of the query having an arbitrary rotation from $0°$to $360°$. For the fixed rotation cases ($90°$, $180°$, $270°$), performance decreases with an increase in training rotation classes. This is because these fixed rotations occur in the training set more often when there are fewer classes.

We now focus on the cross-time case (bottom of Table 4). Incorporating the ONN still improves the performance over no ONN but not by as much, especially in the arbitrary orientation case. This demonstrates that the ONN has difficulty normalizing the orientation of images from different times since that is not what it is trained on.

Figure 7 visually illustrates this. The images on the left show the successful normalization of images from different years. The first and third rows show co-located images from different years rotated by varying amounts. These are the inputs to the ONN. The second and fourth rows show the normalized images. These images are similarly oriented both within and between years. In contrast, the images on the right of Figure 7 show a failure case. Here, the normalized images on the second and fourth rows are misaligned by $180°$. This is a difficult case, though, even for humans.

CTONN: cross-time orientation invariance Table 5 shows the results from our CTONN and RotSiamese frameworks when the query and reference are from different times and the query has arbitrary orientation. Remember that the CTONN is trained using differently oriented images from different years and RotSiamese incorporates additional orientation invariance to deal with the noisy CTONN output. (Please refer back to Section 3.4 for details.) Sia corresponds to the original Siamese network (Section 3.2) and RotSia(20) and RotSia(10) correspond to the RotSiamese networks with $\phi = 20$ and $\phi = 10$ (Section 3.4).

The results in Table 5 demonstrate that CTONN improves the orientation normalization for images from different times and that the RotSiamese framework learns features that are more orientation-invariant. The best results are achieved by combining these two improvements.

5. Discussion

Limitations: We note that our orientation normalization framework only works if the normalized images can be

	Test set	SF				LA			
	Rotations	90°	180°	270°	(0°, 360°)	90°	180°	270°	(0°, 360°)
Same time	no ONN	6.50	35.38	6.75	24.25	4.33	27.89	4.11	18.11
	4 classes ONN	99.75	99.63	99.75	69.13	99.56	98.67	99.33	70.89
	8 classes ONN	94.63	94.75	95.13	88.75	97.67	97.78	97.56	96.00
	36 classes ONN	90.38	90.63	90.38	89.25	90.22	91.67	91.67	90.44
Cross time	no ONN	4.75	19.5	3.63	11.63	2.11	14.11	1.22	7.89
	4 classes ONN	79.63	78.13	81.13	35.38	24.78	24.78	26.33	12.56
	8 classes ONN	35.73	36.00	37.38	27.75	14.44	14.33	14.33	12.89
	36 classes ONN	32.5	34.88	32.00	30.13	23.89	26.56	26.11	19.33

Table 4: Geolocalization results for when the query and reference have different orientations. See the text for details.

Classes	ONN	CTONN	Sia	RotSia(20)	RotSia(10)	SF	LA
4	✓		✓			35.38	12.56
	✓			✓		43.5	14.22
		✓		✓		**46.88**	21.89
		✓			✓	40.25	**22.89**
8	✓		✓			27.75	12.89
	✓			✓		31.00	13.33
		✓		✓		**64.88**	37.33
		✓			✓	56.38	**43.33**
36	✓		✓			30.13	19.33
	✓			✓		30.63	20.33
		✓		✓		**62.00**	37.56
		✓			✓	54.75	**45.00**

Table 5: Cross-time and arbitrary query orientation results. See the text for details.

matched. It cannot improve over the case where the query and reference have the same orientation.

Our performance will of course depend on the content of the query. If the query is not distinctive, such as a homogeneous image of water, forest, or even dense housing, our framework will likely fail due to there being tiles in the reference which, particularly in the cross-year case, are more similar to the query than the ground truth. But, any image-based approach would fail in this case. In the supplementary materials, we provide examples of the types of scenes that our method succeeds and fails on. We note, though, that even when we fail to geolocate the query images, the top matches are visually and semantically very similar. This again emphasizes that performing effective similarity-based image retrieval is not sufficient for our problem.

Our approach currently uses co-located image pairs from the query and reference datasets when training the cross-time and orientation-invariant components. Such pairs will not always be available and so this is another limitation.

Finally, our framework will fail when we do not have reference imagery for the query location. But, high-resolution overhead imagery is available for most if not all of the Earth.

Scalability to partial overlap Finally, we explore how sensitive our approach is to the overlap between the query and the reference tiles. Figure 8 shows the performance as a function of % overlap. *Success here means that a ground truth tile that overlaps the query by a certain amount is in the top n matches where n is the number of ground truth tiles.* (The ground truth tiles are those that overlap the query.) Note, though, that our geolocalization framework only requires that *one* of the ground truth tiles is the top match, not that all of the ground truth tiles are in the top matches. Since we assume a set of contiguous references tiles, as the overlap between the query and any one ground truth tile decreases, the overlap with another ground truth tile necessarily increases (see Figure 6). At least one reference tile overlaps with the query image more than 25%.

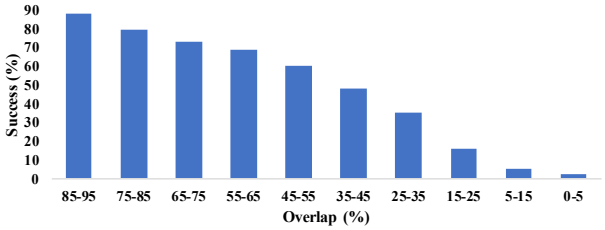


Figure 8: Performance versus % overlap between the query and ground truth tiles. See the text for details.

Figure 8 shows that, as expected, the ability of our matching framework to retrieve a ground truth tile decreases as the overlap decreases. The features are only so local due to the spatial entanglement of the convolutional maps.

6. Conclusion

We perform large-scale overhead image geolocalization by matching a query image to wide-area reference imagery with known location. We demonstrate that local features, particularly those extracted using CNNs, are more effective than global features due to the partial overlap of the query and reference tiles. We develop several technical innovations to deal with the real but challenging cases of when the query and reference are from different times and when the query has an arbitrary orientation. We demonstrate the effectiveness of these innovations on two large datasets of high-resolution aerial imagery.

7. Acknowledgments

This work was funded in part by a National Science Foundation grant, #IIS-1747535. We gratefully acknowledge the support of NVIDIA Corporation through the donation of the GPU card used in this work.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition". In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[2] M. Bansal, K. Daniilidis, and H. Sawhney. Ultra-wide Baseline Facade Matching for Geo-localization. In *European Conference on Computer Vision*, 2012.

[3] F. Castaldo, A. R. Zamir, R. Angst, F. A. N. Palmieri, and S. Savarese. Semantic Cross-View Matching. In *IEEE International Conference on Computer Vision Workshop*, 2015.

[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable Convolutional Networks. In *IEEE International Conference on Computer Vision*, 2017.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[6] M. Divecha and S. Newsam. Large-scale Geolocalization of Overhead Imagery. In *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2016.

[7] Z. Feng, C. Xu, and D. Tao. Self-Supervised Representation Learning by Rotation Feature Decoupling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[8] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning Representations*, 2018.

[9] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[10] J. Hays and A. A. Efros. IM2GPS: Estimating Geographic Information from a Single Image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[11] S. Hu, M. Feng, R. M. H. Nguyen, and G. Hee Lee. CVM-Net: Cross-View Matching Network for Image-Based Ground-to-Aerial Geo-Localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, 2015.

[13] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. In *Nature*, 2015.

[14] T.-Y. Lin, S. Belongie, and J. Hays. Cross-View Image Geolocalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[15] T.-Y. Lin, Y. Cui, S. J. Belongie, and J. Hays. Learning Deep Representations for Ground-to-aerial Geolocalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[16] L. Liu and H. Li. Lending Orientation to Neural Networks for Cross-view Geo-localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[17] L. Liu, H. Li, and Y. Dai. Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map. In *2017 IEEE International Conference on Computer Vision*, 2017.

[18] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision*, 2004.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[21] F. Radenović, G. Tolias, and O. Chum. Fine-Tuning CNN Image Retrieval with No Human Annotation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[22] T. Sattler, B. Leibe, and L. Kobbelt. Fast Image-based Localization Using Direct 2D-to-3D Matching. In *International Conference on Computer Vision*, 2011.

[23] Q. Shan, C. Wu, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz. Accurate Geo-Registration by Ground-to-Aerial Image Matching. In *International Conference on 3D Vision*, 2014.

[24] Y. Shi, X. Y. Yu, L. Liu, T. Zhang, and H. Li. Optimal Feature Transport for Cross-View Image Geo-Localization. In *ArXiv*, 2019.

[25] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *IEEE International Conference on Learning Representations*, 2015.

[26] Y. Tian, C. Chen, and M. Shah. Cross-View Image Matching for Geo-Localization in Urban Environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[27] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[28] N. N. Vo and J. Hays. Localizing and Orienting Street Views Using Overhead Imagery. In *European Conference on Computer Vision*, 2016.

[29] T. Weyand, I. Kostrikov, and J. Philbin. PlaNet - Photo Geolocation with Convolutional Neural Networks. In *European Conference on Computer Vision*, 2016.

[30] S. Workman, R. Souvenir, and N. Jacobs. Wide-Area Image Geolocalization with Aerial Reference Imagery. In *IEEE International Conference on Computer Vision*, 2015.

[31] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Harmonic Networks: Deep Translation and Rotation Equivariance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7168–7177, 2016.

[32] A. R. Zamir and M. Shah. Accurate Image Localization Based on Google Maps Street View. In *European Conference on Computer Vision*, 2010.

[33] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs. Predicting Ground-Level Scene Layout from Aerial Imagery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[34] Y. Zhu, X. Deng, and S. D. Newsam. Fine-Grained Land Use Classification at the City Scale Using Ground-Level Images. In *IEEE Transactions on Multimedia*, 2018.