QATM: Quality-Aware Template Matching For Deep Learning

Jiaxin Cheng   Yue Wu   Wael Abd-Almageed   Premkumar Natarajan
USC Information Sciences Institute, Marina del Rey, CA, USA
chengjia@{usc/isi}.edu {yue_wu,wamageed,pnataraj}@isi.edu

Abstract

Finding a template in a search image is one of the core problems many computer vision, such as semantic image semantic, image-to-GPS verification etc. *We propose a novel quality-aware template matching method, QATM, which is not only used as a standalone template matching algorithm, but also a trainable layer that can be easily embedded into any deep neural network. Here, our quality can be interpreted as the distinctiveness of matching pairs. Specifically, we assess the quality of a matching pair using soft-ranking among all matching pairs, and thus different matching scenarios such as 1-to-1, 1-to-many, and many-to-many will be all reflected to different values. Our extensive evaluation on classic template matching benchmarks and deep learning tasks demonstrate the effectiveness of QATM. It not only outperforms state-of-the-art template matching methods when used alone, but also largely improves existing deep network solutions.*

1. Introduction and Review

Template matching is one of the most frequently used techniques in computer vision applications, such as video tracking[35, 36, 1, 9], image mosaicing [25, 6], object detection [12, 10, 34], character recognition [29, 2, 21, 5], and 3D reconstruction [22, 23, 16]. Classic template matching methods often use sum-of-squared-differences (SSD) or normalized cross correlation (NCC) to calculate a similarity score between the template and the underlying image. These approaches work well when the transformation between the template and the target search image is simple. However, these methods start to fail when the transformation is complex or non-rigid, which is common in real-life. In addition, other factors, such as occlusions and color shifts, make these methods even more fragile.

Numerous approaches have been proposed to overcome these real-life difficulties applying standard template matching. Dekel et al. [11] introduced the Best-Buddies-Similarity (BBS) measure, which focuses on the nearest-neighbor (NN) matches to exclude potential and bad matches caused by the background pixels. Deformable Diversity Similarity (DDIS) was introduced in [26], which explicitly considers possible template deformation and uses the diversity of NN feature matches between a template and a potential matching region in the search image. Co-occurrence based template matching (CoTM) was introduced in [14] to quantify the dissimilarity between a template and a potential matched region in the search image. These methods indeed improve the performance of template matching. However, these methods cannot be used in deep neural networks (DNN) because of two limitations — (1) using non-differentiable operations, such as thresholding, counting, *etc.* and (2) using operations that are not efficient with DNNs, such as loops and other non-batch operations.

Existing DNN-based methods use simple methods to mimic the functionality of template matching [15, 30, 28, 27, 4], such as computing the tensor dot-product [18][1] between two batch tensors of sizes $B \times H \times W \times L$ and $B \times H' \times W' \times L$ along the feature dimension (*i.e.*, L here), and producing a batch tensor of size $B \times H \times W \times H' \times W'$ containing all pairwise feature dot-product results. Of course, additional operations like max-pooling may also be applied [30, 31, 18, 7].

In this paper, we propose the quality-aware template matching (QATM) method, which can be used as a standalone template matching algorithm, or in a deep neural network as a trainable layer with learnable parameters. It takes the uniqueness of pairs into consideration rather than simply evaluating matching score. QATM is composed of differentiable and batch-friendly operations and, therefore, is efficient during DNN training. More importantly, QATM is inspired by assessing the matching quality of source and target templates, and thus is able handle different matching scenarios including 1-to-1, 1-to-many, many-to-many and no-matching. Among different matching cases, only the 1-to-1 matching is considered to be high quality due to it's more distinctive than 1-to-many and many-to-many cases.

The remainder of paper is organized as follows. Section 2 discusses motivations and introduces QATM . In Section 3, the performance of QATM is studied in classic template

[1] See `numpy.tensordot` and `tensorflow.tensordot`.

matching setting. QATM is evaluated on both semantic image alignment and image-to-GPS verification problems in Section 4. We conclude the paper and discuss future works in Section 5.

2. Quality-Aware Template Matching

2.1. Motivation

In computer vision, regardless of the application, many methods implicitly attempt the solve some variant of following problem — *given an exemplar image (or image patch), find the most similar region(s) of interest in a target image*. Classic template matching [11, 26, 14], constrained template matching [31], image-to-GPS matching [7], and semantic alignment [18, 19, 8, 13] methods all include some sort of template matching, despite differences in the details of each algorithm. Without loss of generality, we will focus the discussion on the fundamental template matching problem, and illustrate applicability to different problem in later sections.

One known issue in most of existing template matching methods is that typically, all pixels (or features) within the template and a candidate window in the target image are taken into account when measuring their similarity[11]. This is undesirable in many cases, for example when the background behind the object of interest changes between the template and the target image. To overcome this issue, the BBS [11] method relies on nearest neighbor (NN) matches between the template and the target, so that it could exclude most of background pixels for matching. On top of BBS, the DDIS [26] method uses the additional deformation information in NN field, to further improve the matching performance.

Unlike previous efforts, we consider five different template matching scenarios, as shown in Table 1, where t and s are patches in the template $\mathbf{T}$ and search $\mathbf{S}$ images, respectively. Specifically, "1-to-1 matching" indicates exact matching, i.e. two matched objects, "1-to-N" and "M-to-1" indicates s or t is a homogeneous or patterned patch causing multiple matches, *e.g.* a sky or a carpet patch, and "M-to-N" indicates many homogeneous/patterned patches both in $\mathbf{S}$ and $\mathbf{T}$. It is important to note that this formulation is completely different from the previous NN based formulation, because even though t and s are nearest neighbors, their actual relationship still can be any of the five cases considered. Among four matching cases, only 1-to-1 matching is considered as high quality. This is due to the fact that in other three matching cases, even though pairs may be highly similar, that matching is less distinctive because of multiple matched candidates. Which turned out lowering the reliability of that pair.

It is clear that the "1-to-1" matching case is the most important, while the "not-matching" is almost useless. It is therefore not difficult to come up the qualitative assessment for each case in the Table 1. As a result, the optimal matched region in $\mathbf{S}$ can be found as the place that maximizes the overall matching quality. We can therefore come up with a quantitative assessment of the matching as shown in Eq. (1)

	Matching Cases				**Not Matching**
	1-to-1	1-to-N	M-to-1	M-to-N	
Quality	High	Medium	Medium	Low	Very Low
QATM (s,t)	1	$1/N$	$1/M$	$1/MN$	$1/\lVert T\rVert\lVert S\rVert \approx 0$

Table 1: Template matching cases and ideal scores.

$$R^* = \arg\max_R \{ \sum_{r\in R} \max \left\{ \text{Quality}(r,t) | t \in \mathbf{T} \right\} \} \tag{1}$$

such that the region R in $\mathbf{S}$ that maximizes the overall matching quality will be the optimally matching region. R is a fixed size candidate window and we used the size of object as window size in the experiment.

2.2. Methodology

To make Eq. (1) applicable to template matching, we need to define Quality(s,t), *i.e.* how to assess the matching quality between (s,t). In the rest of section, we derive the quality-aware template matching (QATM) measure, which is a proxy function of the ideal quality assessment Quality(s,t).

Let f_s and f_t be the feature representation of patch s and t, and $\rho(\cdot)$ is a predefined similarity measure between two patches, *e.g.* cosine similarity. Given a search patch s, we define the likelihood function that a template patch t is matched, as shown in Eq. 2,

$$L(t|s) = \frac{\exp\{\alpha \cdot \rho(f_t, f_s)\}}{\sum_{t'\in\mathbf{T}} \exp\{\alpha \cdot \rho(f_{t'}, f_s)\}} \tag{2}$$

where α is a positive number and will be discussed later. This likelihood function can be interpreted as a soft-ranking of the current patch t compared to all other patches in the template image in terms of matching quality. It can be alternatively considered as a heated-up softmax embedding [38], which is the softmax activation layer with a learnable temperature parameter, *i.e.* α in our context.

In this way, we can define the QATM measure as simple as the product of likelihoods that s is matched in $\mathbf{T}$ and t is matched in $\mathbf{S}$ as shown in Eq. (3).

$$\text{QATM}(s,t) = L(t|s) \cdot L(s|t) \tag{3}$$

Any reasonable similarity measure $\rho(\cdot)$ that gives a high value when f_t and f_s are similar, a low value otherwise could be used. When t and s truly matched, $\rho(f_t, f_s)$ should

Matching Case	$L(s\|t)$	$L(t\|s)$	QATM(s,t)
1-to-1	1	1	1
1-to-N	1	1/N	1/N
M-to-1	1/M	1	1/M
M-to-N	1/M	1/N	1/MN
Not Matching	1/$\|\|\mathbf{S}\|\|$	1/$\|\|\mathbf{T}\|\|$	≈ 0

Table 2: Ideal QATM scores

be larger than those unmatched cases $\rho(f_t, f_{s'})$. Equivalently, this means $\rho(f_t, f_s)$ is the best match and thus the maximum score. This score will ideally be 1, after lifting by α and activating by the softmax function, when appropriate α parameter is selected. Similarly, when t matches N of s patches, we should have N equally high matching scores, indicating $L(s|t) = 1/N$ in the ideal case. Table 2 summarizes the ideal scores of all five cases, and their values match the subjective quality assessment on individual cases shown in Table 1. Once we have the pairwise QATM results between $\mathbf{S}$ and $\mathbf{T}$, the matching quality of an ROI s can be found as shown in Eq. (4)

$$q(s) = \max \left\{ \text{QATM}(s,t) | t \in \mathbf{T} \right\} \tag{4}$$

where $q(\cdot)$ indicates the matching quality function. Eventually, we can find the best matched region R^* which maximizes the overall matching quality as shown in Eq. (5).

$$R^* = \arg\max_{R} \left\{ \sum_{r \in R} q(r) \right\} \tag{5}$$

2.3. QATM As An Algorithmic DNN Layer

Proposed QATM assesses the matching quality in a continuous way. Therefore, its gradients can be easily computed via the chain rule of individual function (all of which can be implemented through either a standard DNN layer *e.g.* `softmax` activation, or basic mathematical operators provided in most of DNN frameworks).

In Alg. 1, we demonstrate how to compute the matching quality map form both $\mathbf{T}$ and $\mathbf{S}$. One can easily implement it into DNN in roughly 30 lines of `Python` code using deep learning librarys such as `Tensorflow` and `Pytorch`. Specifically, we use the *cosine similarity* as an example to assess the raw patch-wise similarity, `tf.einsum`(line 4) computes all patch-wise similarity scores in a batch way. Once QATM(t,s) is computed, we can compute the template matching map for the template image $\mathbf{T}$ and the target search image $\mathbf{S}$, respectively, as shown in lines 9 — 10. As one can see, when the α parameter is not trainable, *i.e.* a fixed value, then the proposed QATM layer degrades to a classic template matching algorithm.

Algorithm 1 Compute QATM and matching quality between two images

1: **Given:** template image I_T and search image I_S, a feature extracting model F, a temperature parameter α. Func$(\cdot|I)$ indicates doing operation along axis of I.
2: $T \leftarrow F(I_T)$
3: $S \leftarrow F(I_S)$
4: $\rho_{st} \leftarrow$ Patch-wiseSimilarity(T, S) ▷ Which can be easily obtained by off-the-shelf functions such as `tensorflow.einsum` or `tensorflow.tensordot`
5: $\rho_{st} \leftarrow \rho_{st} \times \alpha$
6: $L(s|t) \leftarrow$ Softmax$(\rho_{st}|T)$
7: $L(t|s) \leftarrow$ Softmax$(\rho_{st}|S)$
8: $QATM \leftarrow L(s|t) \times L(t|s)$
9: $S_{map} \leftarrow$ Max$(QATM|T)$ ▷ Matching quality score
10: $T_{map} \leftarrow$ Max$(QATM|S)$

2.4. Discussions on α

In this section, we discuss how α should be picked in a direct template matching scenario that does not involve training a DNN. We later show that QTAM can easily be embedded as a trainable layer in DNNs to perform template matching without manual tuning structure according to tasks.

When applying Eq. (2), α serves two purposes — (1) matched patches will have ranking scores as close to 1 as possible, and (2) unmatched patches will have ranking scores as close to 0 as possible. As one can see, as α increases, $L(t|s)^+$, the likelihood of matched cases, will also increase, and will quickly reach its maximum of 1 after some α. However, this does not mean we can easily pick a large enough α, because a very large α will also push $L(t|s)^-$, the likelihood of unmatched cases, to deviate from 0. Therefore, a good α choice can be picked as the one that provides the largest quality discernibility as shown in Eq. (6)

$$\alpha^* = \arg\max_{\alpha > 0} \left\{ L(t|s)^+ - L(t|s)^- \right\}. \tag{6}$$

In practice, it is difficult to manually set α properly without knowing details about the similarity score distributions of both matched and unmatched pairs. If both distributions are known, however, we can simulate both $L(t|s)^+$ and $L(t|s)^-$. Without loss of generality, say there are N patches in $\mathbf{T}$. $L(t|s)$, whether or not (t,s) is the matched pair, can be obtained by simulating one f_t feature and N of f_s feature, or equivalently, by simulating N number of $\rho(f_t, f_s)$ similarity scores according to its definition Eq. (2). The major difference between the matched and unmatched cases is that we need one score from the score distribution of matched pairs and $N-1$ scores from the distribution of

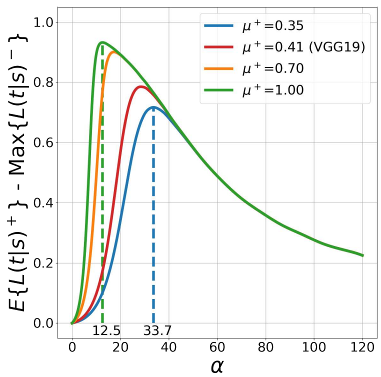


Figure 1: The quality discernibility for varying α.

unmatched pairs for $L(t|s)^+$, while all N scores from the distribution of unmatched pairs for $L(t|s)^-$.

Fig. 1 shows the difference between $\mathbb{E}[L(t|s)^+]$ and $\max\{L(t|s)^-\}$ for different α values, when the genuine and imposter scores follow the normal distribution $\mathcal{N}(\mu^+, 0.01)$ and $\mathcal{N}(0, 0.05)$ for $N = 2200$. As one can see, the difference plot is uni-modal, and the optimal α increases as the mean μ^+ decreases. This figure is more meaningful when the used feature is from a DNN and the used raw similarity measure is the cosine similarity. Zhang et al. [37] provides the theoretical cosine similarity score distribution for unmatched pairs, whose mean is 0 and variance is $1/d$, where d is the feature dimension. Our empirical studies shows that many DNN features attains μ^+ above 0.3, *e.g.* the VGG19 feature. Consequently, a reasonable α for DNN features is roughly in $[12.5, 33.7]$ when cosine similarity is used.

3. QATM Performance in Template Matching

We start with evaluating the proposed QATM performance on the classic template matching problem. Our code is released in the open repository https://github.com/cplusx/QATM.

3.1. Experimental Setup

To find the matched region in the search image $\mathbf{S}$, we compute the matching quality map on $\mathbf{S}$ through the proposed `NeuralNetQATM` layer (without learning α) (see Alg. 1), which takes a search image $I_\mathbf{S}$ and a template image $I_\mathbf{T}$ as inputs. One can therefore find the best matched region R^* in $\mathbf{S}$ using Eq. (5).

We follow the evaluation process given in [24] and use the standard OTB template matching dataset [32], which contains 105 template-image pairs from 35 color videos. We use the 320-d convoluational feature from a pretrained ImageNet-VGG19 network. The standard intersection over union (IoU) and the area-under-curve (AUC) methods are used as evaluation metrics. QTAM is compared against three state-of-the-art methods, BBS [11], DDIS [26] and CoTM [24], plus the classic template matching using SSD and NCC.

3.2. Performance On The Standard OTB Dataset

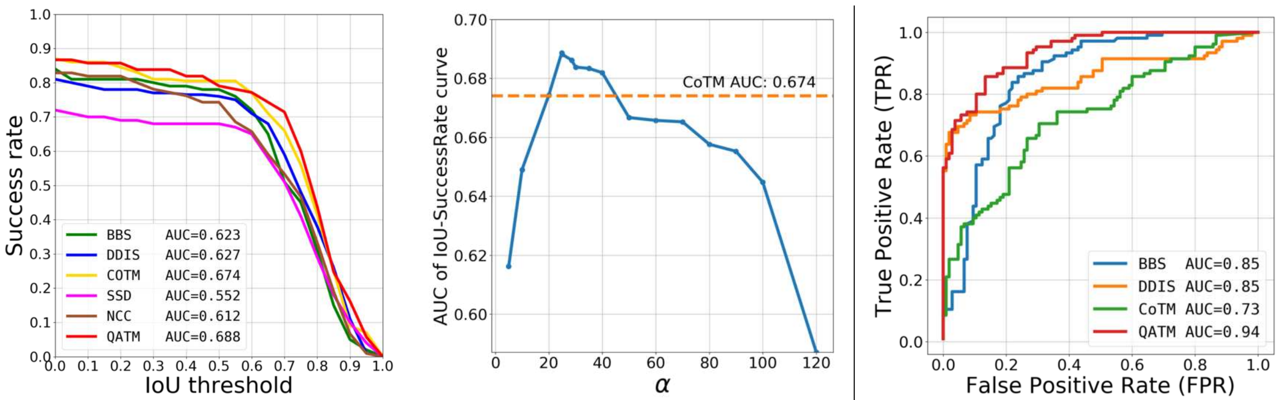
In this experiment, we follows all the experiment settings from [14], and evaluates the proposed QATM method on the standard OTB dataset. The α value is set to 28.4, which is the peak of *VGG*'s curve (see Fig. 1). The QATM performance as well as all baseline method performance are shown in Fig. 2-(a). As one can see, the proposed QATM outperforms state-of-the-art methods and lead the second best (CoTM) by roughly 2% in terms of AUC score, which is clearly a noticeable improvement when comparing to the 1% performance gap between BBS and its successor DDIS.

Since the proposed QATM method has the parameter α, we evaluate the QATM performance under varying α values as shown in Fig. 2-(b). It is clear that the overall QATM performance is not very sensitive to the choice of value when α is around optimal solution. As indicated by the horizontal dash line in Fig. 2-(b), a range of α (rather than a single value) leads to better performance than the state-of-the-art methods. More qualitative results can be found in Fig. 3.

3.3. Performance On The Modified OTB Dataset

One issue in the standard OTB dataset is that it does not contain any negative samples, but we have no idea whether a template of interest exist in a search image in real-applications. We therefore create a modified OTB (MOTB) dataset. Specifically, for each pair search image $\mathbf{S}$ and template $\mathbf{T}$ in OTB, we (1) reuse this pair ($\mathbf{S}$,$\mathbf{T}$) in MOTB as a positive sample and (2) keep $\mathbf{S}$ untouched while replacing $\mathbf{T}$ with a new template $\mathbf{T}'$, where $\mathbf{T}'$ is from a different OTB video, and use this ($\mathbf{S}$,$\mathbf{T}'$) as a negative sample. The negative template $\mathbf{T}'$ is chosen to be the same size as $\mathbf{T}$ and is randomly cropped from a video frame.

The overall goal of this study is to fairly evaluate the template matching performance with the presence of negative samples. For each sample in MOTB, a pair of (template, search image), we feed it to a template matching algorithm and record the average response of the *found* region in a search image. For the proposed QATM method, we again use $\alpha = 28.4$. These responses along with the true labels of each pairs are then used to plot the AUC curves shown in Fig. 2-(c). Intuitively, a good template matching method should give much lower matching scores for a negative sample than for a positive sample, and thus attain a higher AUC score. The proposed QATM method obviously outperform the three state-of-the-art methods by a large margin, which is roughly 9% in terms of AUC score. More importantly, the proposed QATM method clearly attains much higher true positive rate at low false positive rates. This result is not surprising since the proposed QATM is quality aware.



(a) (b) (c)

Figure 2: Template matching performance comparisons. (a) QATM v.s. SOTA methods on the OTB dataset. (b) QATM performance under varying α on the OTB dataset. (c) QATM v.s. SOTA methods on the MOTB dataset.

For example, when a negative template is homogeneous, all methods will find a homogeneous region in the search image since it is the most similar region. The difference is that our approach is quality-aware and thus the matching score of this type will be much lower than that of a positive template, while other methods do not have this feature.

3.4. Discussions

Fig. 3 provides more qualitative results from the proposed QATM method and other state-of-the-art methods. These results confirm the use of QATM, which gives 1-to-1, 1-to-many, and many-to-many matching cases different weights, not only finds more accurate matched regions in the search image, but also reduces the responses in unmatched cases. For example, in the last row, when a nearly homogeneous negative template is given, the proposed QATM method is the only one that tends to give low scores, while others still returns high responses.

Finally, the matching speed also matters. We thus estimate the processing speed (sec/sample) for each method using the entire OTB dataset. All evaluations are based on an `Intel(R) Xeon(R) E5-4627 v2` CPU and a `GeForce GTX 1080 Ti` GPU respectively. Table 3 compares the estimated time complexity of different methods. Though QATM contains relative expensive `softmax` operation, its DNN compatible nature makes GPU processing feasible, which clearly is the fastest method.

Methods	**SSD**	**NCC**	**BBS**	**DDIS**	**CoTM**	**QATM**	
Backend	CPU					CPU	GPU
Average (sec.)	1.1	1.5	15.3	2.6	47.7	27.4	**0.3**
StandDev (sec.)	0.47	0.53	13.10	2.29	18.50	17.80	**0.12**

Table 3: Time complexity comparisons. (Time for feature extraction is excluded)

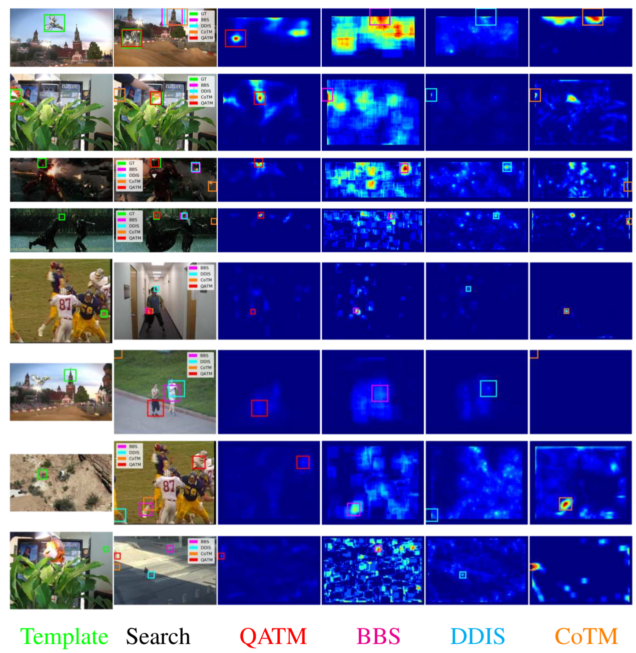


Figure 3: Qualitative template matching performance. Columns from left to right are: the template frame, the target search frame with predicted bounding boxes overlaid (different colors indicate different method), and the response maps of QATM, BBS, DDIS, CoTM, respectively. Rows from top to bottom: the top four are positive samples from OTB, while the bottom four are negative samples from MOTB. Best viewed in color and zoom-in mode.

4. Learable QATM Performance

In this section, we focus on use the proposed QATM as a differentiable layer with learnable parameters in different template matching applications.

4.1. QATM for Image-to-GPS Verification

The image-to-GPS verification (IGV) task attempts to verify whether a given image is taken as the claimed GPS location through visual verification. IGV first uses the claimed location to find a reference panorama image in a third-party database, *e.g*. *Google StreetView*, and then take both the given image and the reference as network inputs to verify visual contents via template matching and produces the verification decision. The major challenges of the IGV task compared to the classic template matching problem are (1) only a small unknown portion visual content in the query image can be verified in the reference image, and (2) the reference image is a panorama, where the potential matching ROI might be distorted.

4.1.1 Baseline and QATM Settings

To understand the QATM performance in the IGV task, we use the baseline method [7] , and repeat its network training, data augmentation, evaluation *etc*., except that we replace its *Bottom-up Pattern Matching* module with the proposed `NeuralNetQATM` layer (blue box in Fig. 4).

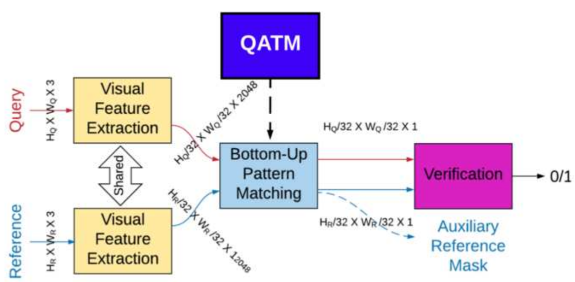


Figure 4: The baseline network architecture from [7], and the QATM version. The dashed arrows indicate the replacement relationship.

The *Bottom-up Pattern Matching* module first computes the cosine similarity between two image features, and then pools out the maximum response only. More precisely, its matching score for a patch s given the template $\mathbf{T}$ relies on Eq. (7),

$$R(s|\mathbf{T}) = \max\{\rho(f_t, f_s)|t \in \mathbf{T}\} \tag{7}$$

while the QATM version relies on Eq. (4).

4.1.2 Performance Comparison

To evaluate QATM performance, we reuse the two dataset used by [7], namely the Shibuya and Wikimedia Common dataset, both of which contain balanced positive and negative samples. Comparison results are listed in Table 4. The proposed QATM solution outperforms the baseline BUMP method on the more difficult Shibuya dataset, while slightly worse on the Wikimedia Common dataset. This is likely attributed to the fact that the *Verification* (see Fig. 4) in the baseline method is proposed to optimize the BUMP performance but not the QATM performance, and thus the advantage of using QATM has not fully transfer to the verification task.

	Wikimedia Common	Shibuya
NetVLAD [3]	0.819 / 0.847	0.634 / 0.638
DELF [17]	0.800 / 0.802	0.607 / 0.621
PlacesCNN [39]	0.656 / 0.654	0.592 / 0.592
BUPM* [7]	**0.864 / 0.886**	0.764 / 0.781
QATM	0.857 / **0.886**	**0.777 / 0.801**

Table 4: Image-to-GPS verification performance comparisons. Performance scores are reported in the (ROC-AUC / Avg. precision) format. (* indicates the baseline network.)

We therefore annotate the matched regions in terms of polygon bounding boxes for the Wikimedia Common dataset for better evaluating the matching performance. These annotations will be released. With the help of these ground truth masks, we are able to fairly compare the proposed QATM and BUMP only on the localization task, which is to predict the matched region in a panorama image. These results are shown in Table 5, and the QATM improves the BUMP localization performance by 21% relatively for both F_1 and IoU measure, respectively. The superiority of QATM for localization can be further confirmed in qualitative results shown in Fig. 5, where the QATM-improved version produces much cleaner response maps than the baseline BUMP method.

Wikimedia Common	F1	IoU
BUPM	0.33	0.24
QATM	**0.40**	**0.29**

Table 5: Localization performance comparisons. Performance scores are averaged over the entire dataset.

4.2. QATM for Semantic Image Alignment

The overall goal for the semantic image alignment (SIA) task is to wrap a given image such that after wrapping it is aligned to a reference image in terms of category-level correspondence. A typical DNN solution for semantic image alignment task takes two input images, one for wrapping and the other for reference, and commonly output a set of parameters for image wrapping. More detailed descriptions about the problem can be found in [18, 19, 13].[234]

[2]https://www.di.ens.fr/willow/research/cnngeometric/
[3]https://www.di.ens.fr/willow/research/weakalign/
[4]https://www.di.ens.fr/willow/research/scnet/

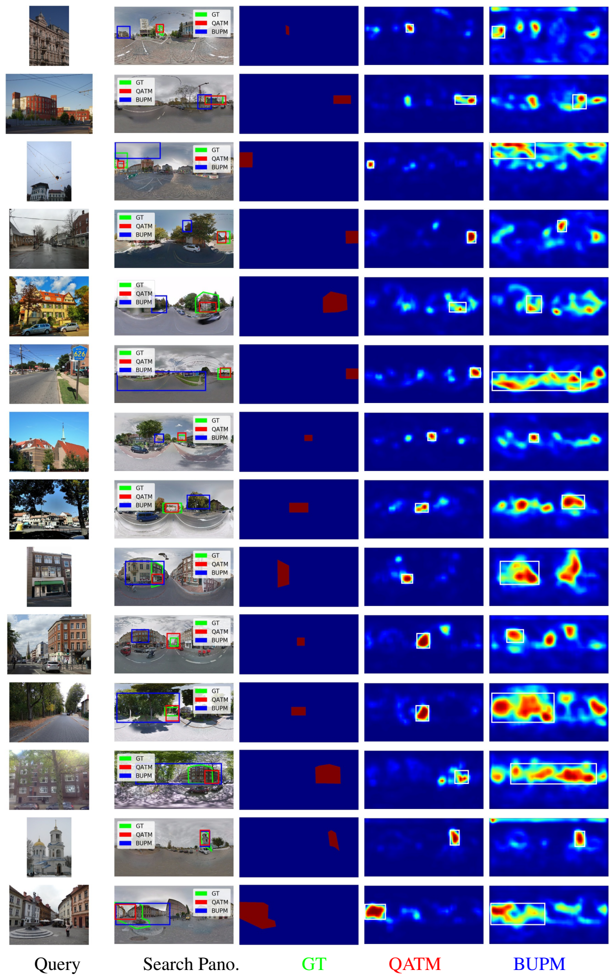


Figure 5: Qualitative image-to-GPS results. Columns from left to right are: the query image, the reference panorama image with predicted bounding boxes overlaid (GT, the proposed QATM, and the baseline BUPM), and the response maps of ground truth mask, QATM-improved, and baseline, respectively.

4.2.1 Baseline and QATM Settings

To understand the QATM performance in the SIA task, we select the baseline method GeoCNN [18], and mimic all the network related settings, including to network architecture, training dataset, loss function, learning rates, *etc*., except that we replace the method's *matching* module (orange box in Fig. 6) with the `NeuralNetQATM` layer (yellow box in Fig. 6).

Unlike in template matching, the SIA task relies on the raw matching scores between all template and search image patches (such that geometric information is implicitly preserved) to regress the wrapping parameters. The *matching* module in [18] is simply computed as the cosine similarity between two patches, *i.e.* $\rho(\mathbf{S},\mathbf{T})$ (see ρ_{st} in line 4 of Alg. 1) and use this tensor as the input for regression. As a result, instead of the matching quality maps, we also make the corresponding change that let the proposed `NeuralNetQATM` produce the raw QATM matching scores, *i.e.* QATM$(\mathbf{S},\mathbf{T})$(see `QATM` in line 8 of Alg. 1).

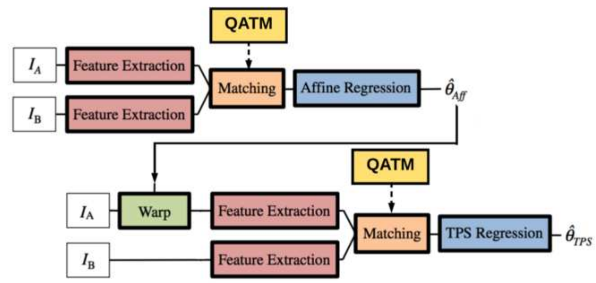


Figure 6: The baseline network architecture from [18], and the QATM version. The dashed arrows indicate the replacement relationship.

4.2.2 Performance Comparisons

To fairly compare SIA performance, we follow the evaluation protocols proposed in [13], which splits the standard PF-PASCAL benchmark into training, validation, and testing subsets with 700, 300, and 300 samples, respectively. The system performance is reported in terms of the percentage of correct key points (PCK) [33, 13], which counts the percentage of key points whose distance to ground truth is under a threshold after being transformed. The threshold is set to $\tau = 0.1$ of image size in the experiment. Table 6 compares different methods on this dataset. The proposed QATM method clearly outperforms all baseline methods, and also is the top-ranking method for 7 out of 20 subclasses. Furthermore, the SCNet [13] uses much more advanced features and matching mechanisms than our baseline GeoCNN method. And [19] used training subset of PF-PASCAL to fine-tune on GeoCNN with a very small learning rate. However, our results confirm that simply replacing the raw matching scores with those quality-aware scores could lead an larger gain than using more a complicated network without fine-tuning on PF-PASCAL subset. A concurrent work [20] adopted a similar idea to re-rank matching score through softmax function as QATM. They reassigned matching score by finding soft mutual nearest neighbour and outperformed QATM when trained on PF-PASCAL subset. More qualitative results can be found in Fig. 7

Class	UCN [8]	SCNet [13]	GeoCNN* [18]	WSup [19]	NC-Net [20]	QATM
plane	64.8	**85.5**	82.4	83.7	-	83.5
bike	58.7	84.4	80.9	**88.0**	-	86.2
bird	42.8	66.3	**85.9**	83.4	-	80.7
boat	59.6	70.8	47.2	58.3	-	**72.2**
bottle	47.0	57.4	57.8	68.8	-	**78.1**
bus	42.2	82.7	83.1	**90.3**	-	87.4
car	61.0	82.3	**92.8**	92.3	-	91.8
cat	45.6	71.6	**86.9**	83.7	-	**86.9**
chair	49.9	**54.3**	43.8	47.4	-	48.8
cow	52.0	**95.8**	91.7	91.7	-	87.5
d.table	48.5	**55.2**	28.1	28.1	-	26.6
dog	49.5	59.5	76.4	76.3	-	**78.7**
horse	53.2	68.6	70.2	77.0	-	**77.9**
m.bike	72.7	75.0	76.6	76.0	-	**79.9**
person	53.0	56.3	68.9	**71.4**	-	69.5
plant	41.4	60.4	65.7	**76.2**	-	73.3
sheep	83.3	60.0	**80.0**	**80.0**	-	**80.0**
sofa	49.0	**73.7**	50.1	59.5	-	51.6
train	**73.0**	66.5	46.3	62.3	-	59.3
tv	66.0	**76.7**	60.6	63.9	-	64.4
Average	55.6	72.2	71.9	75.8	**78.9**	75.9

Table 6: Semantic image alignment performance comparison on PF-PASCAL. (* indicates the baseline network.)

5. Conclusion

We introduced a novel quality-aware template matching method, QTAM. QTAM is inspired by the fact of natural quality differences among different matching cases. It is also designed in such a way that its matching score accurately reflects the relative matching distinctiveness of the current matching pair against others. More importantly, QTAM is differentiable with a learable paramters, and can easily be implemented with existing common deep learning layers. QTAM can be directly embedded into a DNN model to fulfill the template matching goal.

Our extensive experiments show that when used alone, it outperforms the state-of-the-art template matching methods and produces more accurate matching performance, fewer false alarms, and at least 10x speedup with the help of a GPU. When plugged into existing DNN solutions for template matching related tasks, we demonstrated that it could noticeably improve the scores in both the image semantic alignment tasks, and the image-to-GPS verification task.

Acknowledgement This work is based on research sponsored by the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

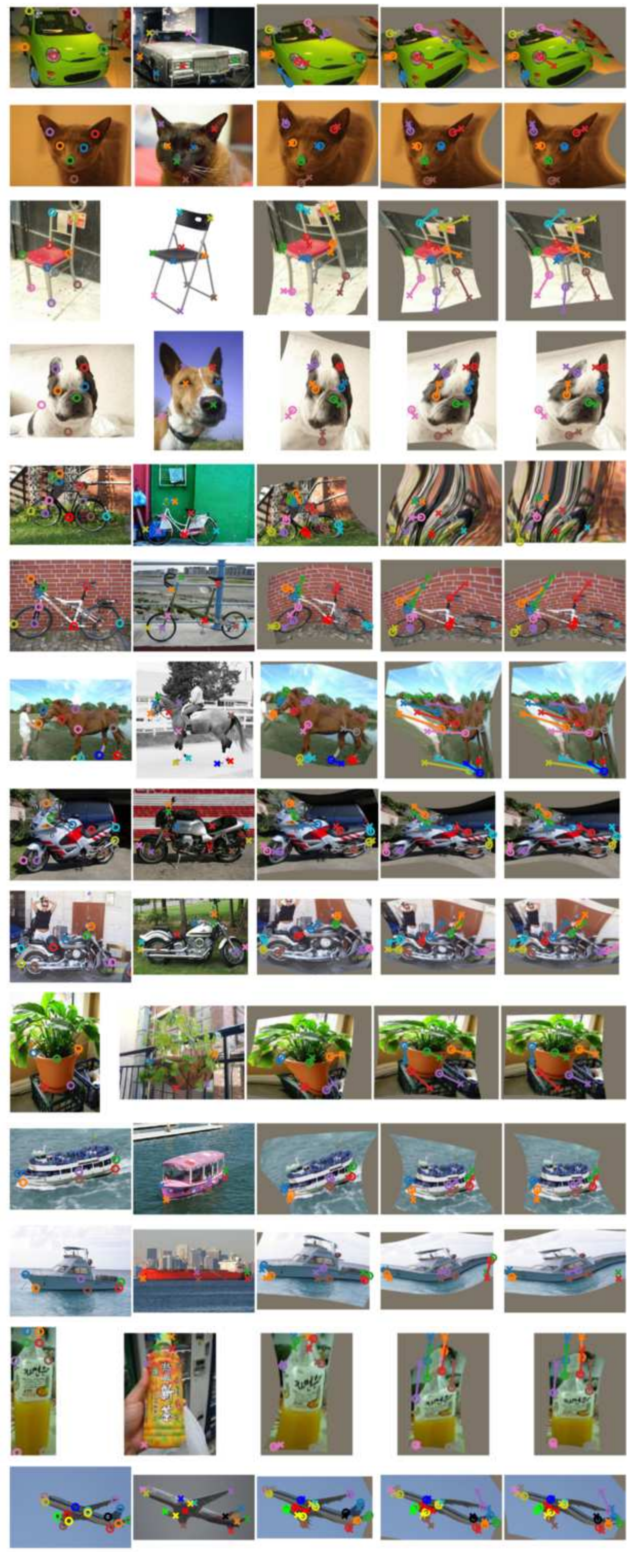


Figure 7: Qualitative results on PF-PASCAL dataset. Columns from left to right represent source image, target image, transform results of QATM, GoeCNN[18] and [19]. Circles and crosses indicate key points on source images and target images.

References

[1] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 798–805. IEEE, 2006.

[2] Alireza Alaei and Mathieu Delalandre. A complete logo detection/recognition system for document images. In *Proceedings of International Workshop on Document Analysis Systems)*, pages 324–328. IEEE, 2014.

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[4] Min Bai, Wenjie Luo, Kaustav Kundu, and Raquel Urtasun. Exploiting semantic information and deep matching for optical flow. In *European Conference on Computer Vision*, pages 154–170. Springer, 2016.

[5] Raluca Boia, Corneliu Florea, Laura Florea, and Radu Dogaru. Logo localization and recognition in natural images using homographic class graphs. *Machine Vision and Applications*, 27(2):287–301, 2016.

[6] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[7] Jiaxing Cheng, Yue Wu, Wael AbdAlmageed, and Prem Natarajan. Image-to-gps verification through a bottom-up pattern matching network. In *Asian Conference on Computer Vision*. Springer, 2018.

[8] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, pages 2414–2422, 2016.

[9] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[10] James Coughlan, Alan Yuille, Camper English, and Dan Snow. Efficient deformable template detection and localization without user initialization. *Computer Vision and Image Understanding*, 78(3):303–319, 2000.

[11] Tali Dekel, Shaul Oron, Michael Rubinstein, Shai Avidan, and William T Freeman. Best-buddies similarity for robust template matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2021–2029, 2015.

[12] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[13] Kai Han, Rafael S. Rezende, Bumsub Ham, Kwan-Yee K. Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, Oct 2017.

[14] Rotal Kat, Roy Jevnisek, and Shai Avidan. Matching pixels using co-occurrence statistics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1751–1759, 2018.

[15] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016.

[16] Abed Malti, Richard Hartley, Adrien Bartoli, and Jae-Hak Kim. Monocular template-based 3d reconstruction of extensible surfaces with local linear elasticity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1522–1529, 2013.

[17] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Largescale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3465, 2017.

[18] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.

[19] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6917–6925, 2018.

[20] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems*, pages 1658–1669, 2018.

[21] Michael Ryan and Novita Hanafiah. An examination of character recognition on id card using template matching approach. *Procedia Computer Science*, 59:520–529, 2015.

[22] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[23] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 519–528. IEEE, 2006.

[24] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[25] Richard Szeliski et al. Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104, 2007.

[26] Itamar Talmi, Roey Mechrez, and Lihi Zelnik-Manor. Template matching with deformable diversity similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1311–1319, 2017.

[27] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision*, pages 100–111. Springer, 2016.

[28] James Thewlis, Shuai Zheng, Philip HS Torr, and Andrea Vedaldi. Fully-trainable deep matching. In *British Machine Vision Conference*, 2016.

[29] Oivind Due Trier, Anil K Jain, Torfinn Taxt, et al. Feature extraction methods for character recognition-a survey. *Pattern Recognition*, 29(4):641–662, 1996.

[30] Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection. In *Proceedings of the ACM on Multimedia Conference*, pages 1480–1502. ACM, 2017.

[31] Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Busternet: Detecting copy-move image forgery with source/target localization. In *European Conference on Computer Vision*, September 2018.

[32] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2411–2418, 2013.

[33] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, 2013.

[34] Alan L Yuille, Peter W Hallinan, and David S Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

[35] Tianzhu Zhang, Kui Jia, Changsheng Xu, Yi Ma, and Narendra Ahuja. Partial occlusion handling for visual tracking via robust part matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1258–1265, 2014.

[36] Tianzhu Zhang, Si Liu, Narendra Ahuja, Ming-Hsuan Yang, and Bernard Ghanem. Robust visual tracking via consistent low-rank sparse learning. *International Journal of Computer Vision*, 111(2):171–190, 2015.

[37] Xu Zhang, X Yu Felix, Sanjiv Kumar, and Shih-Fu Chang. Learning spread-out local feature descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4605–4613, 2017.

[38] Xu Zhang, Felix Xinnan Yu, Svebor Karaman, Wei Zhang, and Shih-Fu Chang. Heated-up softmax embedding. *arXiv preprint arXiv:1809.04157*, 2018.

[39] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.