

Night-to-Day Image Translation for Retrieval-based Localization

Asha Anoosheh¹, Torsten Sattler², Radu Timofte¹, Marc Pollefeys^{2,3}, Luc Van Gool^{1,4}

Abstract—Visual localization is a key step in many robotics pipelines, allowing the robot to approximately determine its position and orientation in the world. An efficient and scalable approach to visual localization is to use image retrieval techniques. These approaches identify the image most similar to a query photo in a database of geo-tagged images and approximate the query’s pose via the pose of the retrieved database image. However, image retrieval across drastically different illumination conditions, e.g. day and night, is still a problem with unsatisfactory results, even in this age of powerful neural models. This is due to a lack of a suitably diverse dataset with true correspondences to perform end-to-end learning. A recent class of neural models allows for realistic translation of images among visual domains with relatively little training data and, most importantly, without ground-truth pairings.

In this paper, we explore the task of accurately localizing images captured from two traversals of the same area in both day and night. We propose ToDayGAN – a modified image-translation model to alter nighttime driving images to a more-useful daytime representation. We then compare the daytime and translated-night images to obtain a pose estimate for the night image using the known 6-DOF position of the closest day image. Our approach improves localization performance by over 250% compared the current state-of-the-art, in the context of standard metrics in multiple categories.

I. INTRODUCTION

Many tasks such as autonomous vehicular navigation and mixed-reality revolve around keeping track of the source of visual sensing, the camera, in its surroundings. The problem of being able to notice a previously-observed spot is known as place recognition, and it is often intertwined with the related problem of localization: keeping track of one’s position with respect to the previous position and the surrounding environment. Place recognition can aid or even serve as the basis for localization itself.

One way of performing place-recognition is directly comparing traversal images against images captured during a potentially-different traversal. Between comparisons, viewing conditions such as weather and lighting can change considerably. Proper comparison should ideally match places correctly regardless of differing conditions. Yet in practice, this method is still hampered by shifts between the domains of the images used for querying and those used for reference. Closing this gap should lead to easier and more accurate comparison among images. This paper focuses on the problem of image comparison across contrasting visual conditions for the purpose of visual localization.

¹Computer Vision Lab, Department of Information Technology & Electrical Engineering, ETH Zürich

²Department of Computer Science, ETH Zürich

³Microsoft, Switzerland

⁴ESAT, KU Leuven

Modern learning-based methods such as deep neural networks should theoretically be well-suited for tackling this type of problem. But the issue holding them back currently appears to be lack of appropriate training data. Training such a model for this task directly would require hundreds of thousands (or more) of images taken from many different positions, with multiple images taken at each position under diverse conditions. This is difficult to gather or generate automatically.

Instead of gathering more data, we exploit advantages of a recent class of neural-net models that perform a task called image-to-image translation. This refers to the idea of changing the visual properties of images from one domain to appear as if it came from another, with domains defined by collections of data alone. The process is conveniently unsupervised: rather than requiring tuples of images depicting the same place under different conditions, one just needs collections of any images taken under the same conditions. Based on these collections, one can train a model that translates between the different conditions. In addition, these models can train properly with as little as ~ 500 points of data per domain, unlike most neural-network-based tasks which demand tens-of-thousands to millions.

Image translation began as altering the characteristics of an image between perceived styles for artistic and/or entertainment purposes. With projects such as [1], [5], [12], [22] breaking ground, it was now possible to perform high-quality image translation. Soon thereafter, the idea was used to aid other learning tasks [8], [12], [15]–[17]. These show being able to attain high-quality representations of images in the appearance of other domains proved useful for tasks containing a shift in data domain; irrelevant source-domain-specific information is discarded and helpful target-domain details can be filled in.

This paper explores different methods for tackling the problem of similarity between images captured from car-mounted cameras, specifically for applications in autonomous driving. Our method involves performing image translation from source (night) to target domain (day) and feeding the output to an existing image-comparison tool. Using a fixed representation for comparing the images allows us to decouple the problem into domain adaptation and image matching, where we only need to focus on the former. The number of parameters and degrees of freedom is thus reduced. Additionally, the objective of the translation model is modified so the output not only contains the visually-perceptible qualities of the target domain but also the properties that the image-comparison tool relies on most heavily. We present the only image-translation model

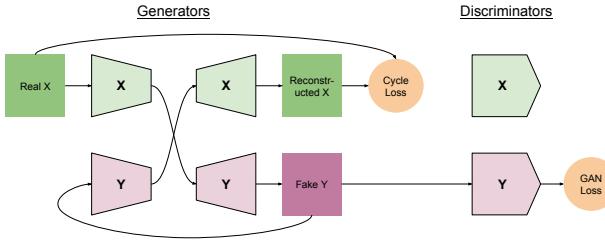


Fig. 1: CycleGAN/ComboGAN generator training pass for domain direction $X \rightarrow Y$. This pass is repeated symmetrically for direction $Y \rightarrow X$ as well.

which specializes discriminators to directly improve a task. It is also the first instance of applying image translation to the problem of retrieval-based localization. This technique greatly outperforms all current state-of-the-art methods in image-based localization.

II. RELATED WORK

A. Image-to-Image Translation

Many tasks in computer vision can be thought of as translation problems where an input image a is to be translated from domain A to b in another domain B . Instead of sampling from a vector distribution to generate images as with regular Generative Adversarial Networks (GANs) [7], translation approaches produce an output conditioned on a given input.

Introduced by Zhu *et al.*, CycleGAN [22] extends this framework to unsupervised image-to-image translation, meaning no alignment of image pairs is necessary. It relies on GANs, a class of neural networks proven to be excellent for capturing the rich distributions of natural images [7]. Adversarial training involves a generator and discriminator network, updated in alternating steps, allowing both to gradually improve alongside each other; first D is trained to distinguish between one or more pairs of real and generated samples, and then the generator is trained to fool D with generated samples. CycleGAN consists of two pairs of generator and discriminator nets, (G_A, D_A) and (G_B, D_B) , where the translators between domains A and B are $G_A : A \rightarrow B$ and $G_B : B \rightarrow A$. D_A is trained to discriminate between real images a and translated images $G_B(b)$, while D_B is trained to discriminate between images b and $G_A(a)$. The system is trained using both an adversarial loss and a cycle consistency loss (see Figure 1). The Cycle consistency loss is a way to regularize the highly unconstrained problem of translating an image unidirectionally, by encouraging mappings G_A and G_B to be inverses such that $G_B(G_A(a)) \approx a$ and $G_A(G_B(b)) \approx b$. The full CycleGAN objective is expressed:

$$\mathcal{L}_{GAN}(G, D_B, A, B) = \mathbb{E}_b[(D_B(b) - 1)^2] + \mathbb{E}_a[D_B(G(a))^2] \quad (1)$$

$$\mathcal{L}_{cycle}(G, F) = \mathbb{E}_a[||F(G(a)) - a||_1] + \mathbb{E}_b[||G(F(b)) - b||_1] \quad (2)$$

$$\begin{aligned} \mathcal{L}(G, F, D_A, D_B) = & \lambda \mathcal{L}_{cycle}(G, F) \\ & + \mathcal{L}_{GAN}(G, D_B, A, B) + \mathcal{L}_{GAN}(F, D_A, B, A) . \end{aligned} \quad (3)$$

Subsequent works improved upon this foundation. Liu *et al.* used a variational-autoencoder loss formulation to encourage a shared feature space for the model they dub UNIT [12]. Ignatov *et al.* performed one-way translation for image enhancement, using two discriminators per domain - one for color and the other for texture - rather than just one [9]. And ComboGAN [1] allowed for n -domain translation, solving the exponential scaling problem in the number of domains.

B. Place Recognition and Localization

Place recognition refers to the task of identifying a real world location from images of said place - essentially location classification. Ideally this process should be invariant to various image and world properties such as camera position, orientation, weather conditions, etc. Visual localization is the process of identifying the camera location (and sometimes orientation) either relative to a local or global map. One way of achieving this is via image-retrieval: finding the most similar image with a known pose to a unknown query. In such a case, invariance to camera orientation/position are not desired, as these are critical to calculating pose as accurately as possible.

A widely-used traditional tool used in place recognition and/or image retrieval is the VLAD descriptor [11]. The descriptor is a (typically rather low-dimensional) vector intended to serve as a description or featurization of the image as a whole. A visual-word vocabulary is built from a diverse dataset, extracting D -dimensional descriptors from affine-invariant detections - and clustered into k centers. Afterwards for a given query image with n local descriptors, the residual from each descriptor to its nearest cluster center is calculated and summed per cluster, resulting in k D -dimensional aggregate vectors. These are then concatenated and normalized to form a unit-length, $k \times D$ -dimensional VLAD descriptor. VLAD vectors across images can now be compared directly (using the original Euclidean distance metric involved) to quantify similarity.

DenseVLAD, a modification of the VLAD procedure, was formalized by Torii *et al.* [21] to improve place recognition as a subprocedure for camera pose estimation. Instead of using a detector to choose where to extract features, 128-dimensional RootSIFT descriptors [3] (an adjustment to the classic SIFT [13]) are simply extracted in regular grid throughout the image. This removes the issue of inconsistent detectors affecting the process altogether. The next steps follow the standard VLAD clustering with k clusters to produce a 128 k -dimensional VLAD descriptor per image, which is then down-projected via PCA. SIFT descriptors are a modified binning of gradients surrounding a point of interest. First a $N \times N$ grid (typically $N = 16$) of x-&-y gradients I_x, I_y are computed around a pixel, followed by their magnitude and orientation ($\sqrt{I_x^2 + I_y^2}$ and $\text{atan2}(I_y, I_x)$, respectively). Each smaller $n \times n$ block (typically $n = \sqrt{N}$) within the grid is grouped into a histogram of orientations weighted by magnitude. After clipping and normalization, we get $N n \times n$

spatial histograms to describe the image region. DenseVLAD extracts multiple versions of these descriptors per image using different bin sizes n for the SIFT histograms, in order to describe the image at multiple feature scales.

In 2015, NetVLAD [2] reformulated the VLAD process into a neural-network framework. The idea was to maintain the clustering and unit vector principles while allowing for a differentiable, trainable architecture. It consists of a feature-extraction portion, which could be early layers from a pre-trained convolutional network, and a “VLAD-Core” module which performs soft-clustering to output a VLAD-like vector.

C. Image Translation for Visual Localization

Domain shift, or equivalently, dataset bias, is an often encountered problem in methods that learn from data. This occurs when the a model built from data is applied to data that differs in some characteristics. Even slight differences in data can cause large variation in outputs, as long as the model is not robust to this new domain. Domain Adaptation is the practice of mitigating the effects of domain shifts. Image-based Localization among differing visual conditions is undoubtedly prone to this issue.

A recent concurrent publication during the development of this paper by Porav *et al.* [16] sought to use CycleGAN to approach almost the same problem as ours: effectively comparing images under different illumination by visually translating one to another domain and then comparing - for example changing nighttime images to day prior to feature-matching. They add an additional cycle losses on top of the original CycleGAN setup: a constraint that reconstructed images should have identical second-derivative and Haar response stacks. This is because the localization procedure employed utilizes SURF [4] features, which relies on second-derivatives for detection and Haar responses for detectors. The model was tested on localizing a night sequence to a daytime sequence from the Oxford RobotCar dataset [14], synthesizing daytime images from the night ones to match features against the real daytime images. Moreover, this project is closely related to our main approach where we translate images to improve descriptor-matching afterwards. Whereas their method enforces a descriptor-aiding feature loss on the input and cyclically-reconstructed image in hopes of these features staying present in the intermediate translated image, our main approach in this paper enforces a similar type of loss directly on the initial translated output. Their project also used a subset of data from the same dataset as ours, albeit trained on a different set of cameras with different orientation and intrinsics. Additionally it was evaluated on the related - but not directly comparable - task of visual-odometry on sequential data with synchronized starting positions. We attempt to adapt the process to our task, though we later determine no fair comparison can be made due to various factors.

III. OUR METHOD - TODAYGAN

Our approach (ToDayGAN) to tackle day-night localization between a daytime reference set with known poses and a

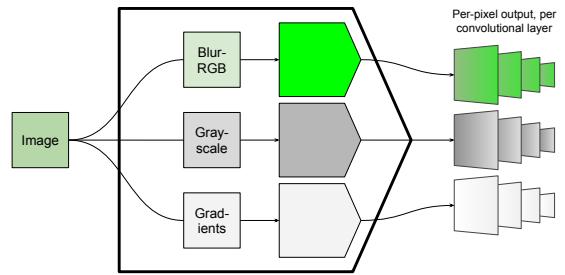


Fig. 2: Each discriminator in our ToDayGAN is comprised of three network clones that operate separately on the blurred-RGB, Grayscale, and xy-Gradients of its input image. Each discriminator outputs a decision at each convolutional layer to cover different receptive-field sizes.

nighttime query set is as follows. First, an image translation model is trained to translate between day and night image domains. Second, the night-to-day direction is used to transform nighttime images into a daytime representation. Both the translated images and the reference images are then fed to an existing featurization process to obtain feature vectors per image. Nearest-neighbor search then gives us a closest matching reference image per query. Then the pose of the query image is approximated as the existing pose of the closest daytime image.

Our image-featurization tool used is DenseVLAD [21]. As detailed in Section II-B, DenseVLAD is an improved take on the VLAD image description technique for describing and comparing image data. By densely extracting descriptors from images and forgoing the detection stage, DenseVLAD is more robust to strong appearance changes than standard VLAD is. As shown in Sattler *et al.*’s localization survey [19], DenseVLAD still outperforms other, more modern methods in terms of generalization on day-night image matching.

Meanwhile our image-translation model is built using the image-translation model ComboGAN [1] as the base. The generator networks are identical to the networks used in CycleGAN, yet each is divided in half, the frontal halves being encoders and the latter halves decoders. For the case of two domains, ComboGAN’s structure and training procedure are identical to CycleGAN’s; thus it’s irrelevant which one is used as the starting point (though using ComboGAN means the model can automatically serve more than two domains if need be). Instead of just using ComboGAN as is, we tailor its setup to fit our problem. ComboGAN’s discriminators, specifically, are modified from that originally presented, resulting in noticeable improvement on the localization task performed using the translated images. Refer to Table VI for network architectures.

Taking notes from [9], we replicate the discriminators to specialize on different forms of the input. In the final version of ToDayGAN, each domain’s discriminator (see Figure 2) becomes *three* copies of the network, expanding on insights from WESPE. One takes the luminance (grayscale) of the input image, one takes the RGB image blurred by a 5×5 3σ Gaussian kernel (exactly as in [9]), and the last takes the horizontal/vertical gradients of the image. These three

discriminators can now separately focus on texture, color, and gradients. The losses are averaged in the end equally. As shown by our experiments, with the addition of each discriminator comes a significant performance boost.

The third discriminator is a novel contribution that serves to emulate the process done when creating SIFT descriptors. The default DenseVLAD implementation from [21] initially grayscales the input image and downsamples with factor $2 \times$ by throwing out every other pixel. Then it creates histograms of magnitude-weighted gradient orientations after computing gradients via convolution with a $[-1 \ 0 \ 1]$ kernel for x -direction gradients and its transpose for the y direction. Therefore our model uses a 1×1 stride-2 convolution to obtain the downsampled image and convolve it with the two filters to obtain the same $x-y$ gradients for the discriminator in a differentiable manner. As opposed to [16], we use this discriminator to create matching-relevant features in the translated version that were nonexistent in the original, whereas they simply preserve the relevant features from the original images in their cyclic reconstructions.

One would think attempting to emulate the full DenseVLAD process even further by feeding magnitudes and orientations of gradients to discriminators would be more fitting for task adaptation. Instead, doing this resulted in poorer localization accuracies - about half of those of using gradients alone - in our case.

In addition, the discriminator now outputs a label/decision after each downsampling layer (barring the very first). Being able to discriminate images at multiple scales encourages consistency in both low- and high-level image statistics, rather than just at the final arbitrary receptive-field size. The outputs for the final loss are weighed linearly, ascending toward the last layer, as the complexity and power of the network's predictions increase with depth. This can be seen as the n outputs (in ascending layer order) weighed by $[1, 2, \dots, n]$ then summed and divided by $n \sum_i^n i$ to average out. To our knowledge, this multi-scale concept is not yet implemented in related works.

Lastly, a recent discriminator loss format, which we dub the Relativistic Loss, was introduced [10], which alters the discriminator loss formulation by only requiring it to determine whether an input is more real relative to a fake, rather than to determine realness in an absolute manner. The motivation behind this is to stabilize training overall by preventing the discriminator becoming too powerful in relation to the generator. Equation (4) defines ComboGAN's least-squares GAN loss from Equation (1) newly adapted to the Relativistic Loss formulation.

$$\mathcal{L}_{GAN}(G_A, D_B, A, B) = \begin{cases} \mathbb{E}_a \mathbb{E}_b [(D_B(b) - D_B(G_A(a)))^2] \\ \quad \text{from Discriminator perspective} \\ \mathbb{E}_a \mathbb{E}_b [(D_B(G_A(a)) - D_B(b) - 1)^2] \\ \quad \text{from Generator perspective} \end{cases} \quad (4)$$

TABLE I: Details of RobotCar and RobotCar Seasons dataset used.

Condition	Purpose	Recorded	# triplets
overcast	reference & training	28 Nov 2014	6,954
night	training	27 Feb & 01 Sep 2015	6,666
night	query	10 Dec 2014	438
night-rain	query	17 Dec 2014	440

IV. EXPERIMENTAL SETUP

A. Dataset

Our source of images for training and evaluation is the Oxford RobotCar dataset [14]. It contains multiple video sequences of the same 10km route captured from an autonomous vehicle in Oxford, England. Three Point Grey Grasshopper2 cameras were mounted on the left, right, and rear of the vehicle, and the traversals were taken over the course of a year, providing variation in lighting, time-of-day, and weather. Though it resulted in over 20 million, 1024×1024 -resolution images total, only a subset of certain traversals are used for this project. The image sets contain corresponding left, right, and rear views, meaning there exist three images per triplet.

We use the RobotCar Seasons variant [19] for evaluation, which provides accurate camera poses for a set of reference and query images: A subset of 6,954 camera triplets of the original RobotCar dataset, which we refer to as Day ("overcast-reference" in [19]), is used as a reference. Another set of 438 triplets, which are captured at night are used as a query set ("night" in [19]). We also use a second query set of 440 images captured during night during the rain ("night-rain" in [19]), whose only purpose is to examine the transferability of our technique (trained without any rain) to a visually-different domain. Finally, another set of 6,666 nighttime triplets, not included in [19] and used only for training, is randomly sampled from three other traversals of the RobotCar dataset directly. We call these three nighttime datasets Night-query, Night-rain, and Night-train.

These datasets are subsets of much larger original video sequences; they have been sub-sampled to reduce sizes to manageable levels. A 3d map was constructed from a vehicle's traversal so the pose of each image can be estimated [19], and then an image was sampled every meter. For the daytime images, using the vehicle's inertial navigation system plus 3d visual tracking is sufficient to build a map, but LIDAR data (also captured by the same vehicle) was necessary to obtain ground-truth poses. We utilize these same calculated poses.

As side-view images are only used to boost data count during some of the training runs, side-views for the query-Nighttime remain unused, and the correspondence (or lack of) for the three views is irrelevant for the purpose of this experiment. As the problem is formulated below, the same Daytime images are available during both training and inference, so the same images are used for both, meanwhile the night images are independent in all stages. Exact details of each dataset can be found in Table I and sample photos from each in Figure 4.

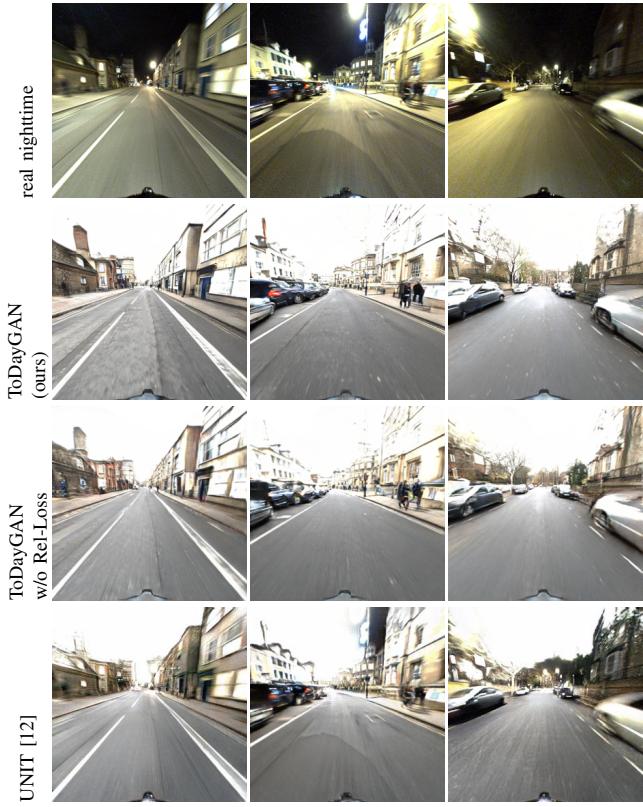


Fig. 3: Top to bottom row: Example real night images before translation, synthetic day images using the best-performing model, synthetic day images using same model but without the Relativistic Loss, synthetic day images produced by UNIT.

B. Training Setup

The following apply to all three types of our models. Images, unless mentioned that left and right viewpoint images were used from the RobotCar dataset, were trained on rear views only. And unless stated otherwise, images in our trials were scaled to 286×286 size and randomly cropped to 256×256 for training. If stated that a 512×512 resolution is used, training crops are of size 384×384 (due to memory constraints). Memory also restricts training on resolutions higher than 512×512 . Inference is always on the pre-crop size because of fully-convolutional architecture allowing for arbitrary input widths. Batches are not used (only one image per input) and random image flipping (left-right) is enabled. Training is run for 40 epochs. Learning rates begin at $2e-4$ for generators and $1e-4$ for discriminators, decreasing linearly to zero during the second half of training epochs.

DenseVLAD uses the $k = 128$ pretrained cluster centers provided by the original project [21]. The VLAD vectors are projected down to 4096 dimensions via PCA prior to comparisons. We also keep the default SIFT extraction scales used in DenseVLAD, at $n \in \{4, 6, 8, 10\}$.

C. Evaluation Protocol & Baselines

Following the evaluation protocol of [19], we report the percentage of query images whose predicted 6-DOF poses is within three error tolerance thresholds: 5-meter and 10-degree, 0.5-meter and 5-degree, and 0.25-meter and 2-degree.

TABLE II: DenseVLAD night query results for direct raw-image baselines, images histogram-equalized beforehand, and those translated with UNIT.

Preprocess	Resolution	Threshold Accuracy (%)		
		5m 10°	0.5m 5°	0.25m 2°
Direct	286	13.9	2.9	0.4
Direct	512	10.2	1.8	0.2
Direct	1024	19.9	3.4	0.9
Hist-Eq night only	1024	23.7	2.5	0.7
Hist-Eq night & day	1024	16.7	2.7	0.4
UNIT [12]	286	17.1	0.4	0.0

Table II lists the results obtained by running DenseVLAD matching on the Daytime and Nighttime images (including Night-rain) directly to benchmark what we believe according to [19] to be the best known solution until now. We include additional results when performing histogram-equalization of images prior to matching; we try this both on the query images only and also on both query and reference. Results are also compared against the out-of-the-box image translation method UNIT [12]. Note that UNIT is not tailored to any task other than perceptive quality of images translated.

Lastly, we compare the best results obtained using our methods with the state-of-the-art methods found in [19]. These include structure-based localization techniques ActiveSearch [18] and CSL [20], in addition to the image-based FAB-MAP [6] and NetVLAD [2]. On the night query set, structure based methods performed very poorly; meanwhile, image-based approaches attained the highest accuracies for the same criteria. DenseVLAD was the best of all methods, followed by NetVLAD.

V. RESULTS

Table III shows the localization rates gathered over various configurations of our own model to determine the effect of certain modifications and variables, and sample visuals from the process can be seen in Figure 3. These include a version of the model with only one discriminator per domain but with all three input features (color, texture, and gradients) concatenated along the channel dimension as a single larger input. This can signal whether separate models are needed, or if simply having these features pre-extracted is the key.

The “Discriminators” column contains up to three letters representing the types of discriminators used for that trial. “C” stands for Color, “L” for Luminance, and “G” for Gradients. Note that when “C” is not used in conjunction with an “L”, the RGB image remains un-blurred when input to the color-discriminator. Likewise, when “L” is not used in conjunction with a “C”, the entire model is run in grayscale, for obvious reasons. “C+L+G” represents unification as a single discriminator with the inputs concatenated. And lastly a trial was performed where the gradient discriminator received as input the magnitude and orientation of gradients in place of the gradients themselves, denoted by “M”. This is closer to the actual DenseVLAD process and should *theoretically* be better suited as task adaptation.

“Rel.-Loss” in Table III means the discriminator loss is the Relativistic Loss mentioned in Section III. “L/R” indicates whether the left and right camera images were used to enlarge the training set. And “Dual-Eval.” refers

TABLE III: Ablation study for the proposed method.

Resolution	Discriminators	Rel.-loss?	Dual-Eval.?	L/R?	Threshold Accuracy (%)		
					5m 10°	0.5m 5°	0.25m 2°
286	C				12.1	1.8	0.2
286	C, G				28.9	6.4	0.9
286	L, G				9.1	2.3	0.4
286	C, L				16.2	2.5	0.6
286	C+L+G				18.7	2.5	0.4
286	C, L, M				15.0	1.8	0.4
286	C, L, G				30.8	5.9	0.6
286	C, L, G	✓			36.0	7.0	1.3
512	C, L, G	✓			44.9	7.3	1.5
512	C, L, G	✓	✓		47.5	8.2	1.5
512	C, L, G	✓	✓	✓	52.9	9.1	1.1

TABLE IV: NetVLAD night query results after ToDayGAN translation; only for a subset of models from Table III to check generalization potential.

Resolution	Discriminators	Rel.-loss?	Dual-Eval.?	L/R?	Threshold Accuracy (%)		
					5m 10°	0.5m 5°	0.25m 2°
286	C, L				12.7	1.5	0.9
286	C, L, G				30.5	4.7	0.9
512	C, L, G	✓	✓	✓	45.6	6.4	1.3

to our Dual-Evaluation procedure added as a finishing-touch enhancement to our models. For this, a horizontally-flipped version of each query image is fed to the network and then re-flipped for DenseVLAD featurization. Distances are calculated between these and the references as well, and the nearest neighbor is the minimum of these and the original unflipped distances. As the network is not invariant to left-right mirroring, this produces two similar yet different “opinions,” which boosts accuracies in our case.

VI. DISCUSSION

A. Comparing results with baselines

Our best model using the large training set with left/right images, 512-resolution, three discriminators, relativistic loss, and the flipped-image dual evaluation attains a gain of 2.65x on the 5m/10° threshold over the best DenseVLAD result. And the 0.5m/5° category also sees a proportional boost from 3.4% to 9.1%. While our 0.25m/2° result also is marginally lower than when not using left/right images, they all seem to be low enough ($\leq 1.5\%$) that their values aren’t meaningful.

Testing the best-performing ToDayGAN model directly on the secondary query set of Night-Rain images also works very well, implying our model is robust to the appearance shift. While the relative increases aren’t as high as the original Night-query’s, the absolute accuracies are nearly identical - even higher for the two stricter thresholds.

UNIT also performed decently in the 5m/10° threshold, better than the comparable vanilla ComboGAN, but nearly zero for the other two thresholds. UNIT’s variational encoding structure tends to result in blurry images, as shown in Figure 3. Lack of low-level detail in the image appears to completely impair its localization at finer scales, but it still contains higher-level details sufficient for coarse scales.

As an additional test, we ran NetVLAD some of the resulting images as well (see Table IV) in order to check the generalization potential of the images for a different comparison method. It turns out to have about the same improvement boost as opposed to directly using NetVLAD (see Table V), suggesting both DenseVLAD and NetVLAD largely rely on the same image characteristics (mostly gradients).

We mentioned in Section II-C that we attempted to compare our method with [16]. Due to lack of source code,

TABLE V: Top overall results for both Night and Night-Rain categories in comparison to other state-of-the-art methods (taken from [19]).

Method	Night			Night-Rain		
	5m 10°	0.5m 5°	0.25m 2°	5m 10°	0.5m 5°	0.25m 2°
FAB-MAP [6]	0.0	0.0	0.0	0.0	0.0	0.0
ActiveSearch [18]	3.4	1.1	0.5	5.2	3.0	1.4
CSL [20]	5.2	0.9	0.2	9.1	4.3	0.9
NetVLAD [2]	15.5	1.8	0.2	16.4	2.7	0.5
DenseVLAD [21]	19.9	3.4	0.9	25.5	5.6	1.1
ToDayGAN (ours)	52.9	9.1	1.5	47.9	12.5	3.2

we could not train their model on our data. Since their method optimizes for SURF features, an approximation of SIFT, it should be suited for DenseVLAD-based localization. The authors did manage to infer our query images on their existing model (trained for a different camera type/view), yet localizing with it performed worse than the naive baseline. So we determined no fair comparison could be made due to the difference in training sets and camera intrinsics.

B. Impact of modifications to ComboGAN

The first sector of Table III ablates the different discriminator combinations to evaluate their contribution to the task. We can see adding the gradient discriminator just about doubles accuracies, while adding the luminance discriminator improves things, though to a much lesser degree. Performing the pipeline in RGB rather than just grayscale (which would seem to be easier for the networks) regularizes and improves the process invaluable. The use of a combined discriminator is considerably inferior to independent ones. Additionally, note the use of gradient magnitude/orientation as discriminator features unexpectedly behaves much worse in practice compared to just the gradients themselves. There is no obvious explanation, but it points to neural nets having more difficulty dealing with the concepts of gradient angles. Meanwhile the second sector of Table III shows the effects of the non-discriminator factors. The Relativistic-Discriminator loss, Dual-Evaluation procedure, and use of left/right images to increase training set size all improve results.

VII. CONCLUSION & FUTURE WORK

In this paper, we have introduced a visual localization system based on image-to-image translations. Our results show that our approach significantly outperforms previous work on the challenging task of localizing nighttime queries against a set of daytime images.

One of the corollaries that can be deduced from our ablation experiments is the partitioning of features for discriminators in a generative-adversarial setup. We find using discriminators tasked with different aspects of a single input image perform better in terms of encouraging the presence of those aspects in generated outputs.

Future work in the field of generative models, in general, can borrow from this very idea. Generated image quality can potentially be improved by using multiple discriminators, each focusing on different image features. Furthermore, these features need not be handcrafted; one could potentially enforce an orthogonality constraint of sorts on the initial features extracted by each discriminator to ensure each concentrates a different aspect of the same input.

APPENDIX

Code for this project is publicly available at <https://github.com/AAnoosheh/ToDayGAN>

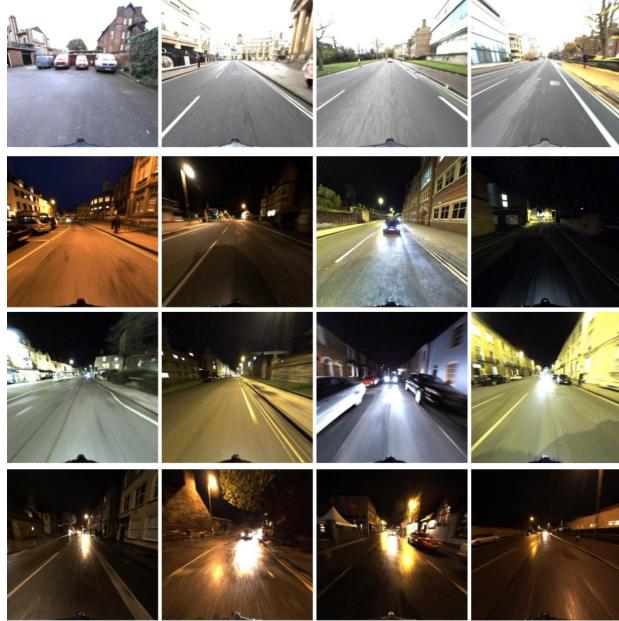


Fig. 4: Top to bottom row: Sample images from the Daytime set, Night-training set, Night-query set, and Night-Rain query set.

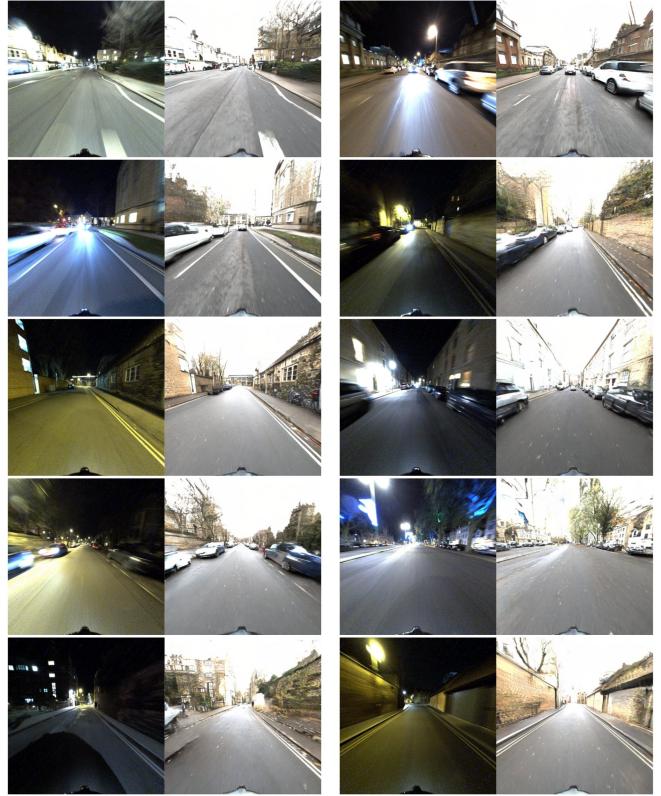


Fig. 5: Additional visuals for night-to-day translation using ToDayGAN.

TABLE VI: Layer specifications for ComboGAN Generator (Encoder + Decoder) and Discriminator. We use the following abbreviations for brevity: N=Neurons, K=Kernel size, S=Stride size. The transposed convolutional layer is denoted by DCONV. The residual basic block is denoted as RESBLK. (Table taken from [1])

Layer #	Encoders
1	CONV-(N64,K7,S1), InstanceNorm, PReLU
2	CONV-(N128,K3,S2), InstanceNorm, PReLU
3	CONV-(N256,K3,S2), InstanceNorm, PReLU
4	RESBLK-(N256,K3,S1), InstanceNorm, PReLU
5	RESBLK-(N256,K3,S1), InstanceNorm, PReLU
6	RESBLK-(N256,K3,S1), InstanceNorm, PReLU
7	RESBLK-(N256,K3,S1), InstanceNorm, PReLU
Layer #	Decoders
1	RESBLK-(N256,K3,S1), InstanceNorm, PReLU
2	RESBLK-(N256,K3,S1), InstanceNorm, PReLU
3	RESBLK-(N256,K3,S1), InstanceNorm, PReLU
4	RESBLK-(N256,K3,S1), InstanceNorm, PReLU
5	RESBLK-(N256,K3,S1), InstanceNorm, PReLU
6	DCONV-(N128,K4,S2), InstanceNorm, PReLU
7	DCONV-(N64,K4,S2), InstanceNorm, PReLU
8	CONV-(N3,K7,S1), Tanh
Layer #	Discriminators
1	CONV-(N64,K4,S2), PReLU
2	CONV-(N128,K4,S2), InstanceNorm, PReLU
3	CONV-(N256,K4,S2), InstanceNorm, PReLU
4	CONV-(N512,K4,S2), InstanceNorm, PReLU
5	CONV-(N256,K4,S1), InstanceNorm, PReLU
6	CONV-(N1,K4,S1)

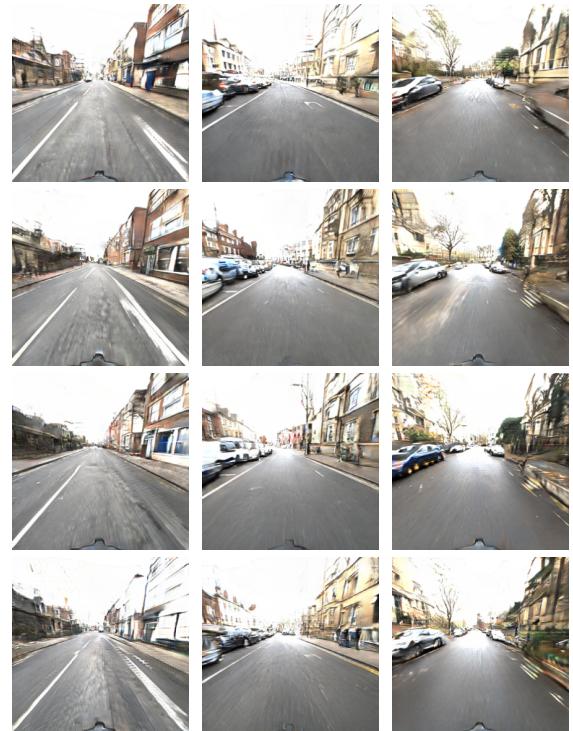


Fig. 6: Daytime translation of the real night images from Figure 3 using different discriminator setups (286 × 286 resolution). Using same notation as in Table III, from top to bottom row: C, CL, CLG, and C+L+G.

REFERENCES

- [1] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
- [2] 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.
- [3] Relja Arandjelovic. Three things everyone should know to improve object retrieval. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 2911–2918, Washington, DC, USA, 2012. IEEE Computer Society.
- [4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.
- [5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [6] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.
- [7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press.
- [8] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. *ArXiv e-prints*, November 2017.
- [9] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. WESPE: weakly supervised photo enhancer for digital cameras. *CoRR*, abs/1709.01118, 2017.
- [10] A. Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. *ArXiv e-prints*, July 2018.
- [11] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, June 2010.
- [12] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 700–708, 2017.
- [13] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.
- [14] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.
- [15] Zak Murez, Soheil Kolouri, David J. Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. *CoRR*, abs/1712.00479, 2017.
- [16] Horia Porav, Will Maddern, and Paul Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. *CoRR*, abs/1803.03341, 2018.
- [17] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser-Nam Lim, and Rama Chellappa. Unsupervised domain adaptation for semantic segmentation with gans. *CoRR*, abs/1711.06969, 2017.
- [18] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, Sept 2017.
- [19] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [20] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1455–1461, July 2017.
- [21] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015.
- [22] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.