Improving CosPlace localization on night-time images

Lorenzo D'Amico, Jacopo De Cristofaro, Vittorio Arpino
Politecnico di Torino
Corso Duca degli Abruzzi, 24 - 10129 Torino (TO)
s296273@studenti.polito.it, s302298@studenti.polito.it, s301607@studenti.polito.it

Abstract

This study focuses on improving the performance on locating night-time images using the current state-of-the-art visual geo-localization model CosPlace. CosPlace is able to achieve high performance and scalability but is limited by domain shift. To overcome this limitation, this research proposes a data augmentation procedure to synthetize night time images starting from day-light ones and a domain adaptation module that leverages adversarial learning. The study also explores other losses and the finetuning on different pretrained models to see how it affects the outcome. The results show that these methods significantly improve the accuracy of night-time image location without drastically reducing the performance of day-light image location. Code is available for research purposes here.

1. Introduction

The Visual geo-localization (VG) task has the objective to find the approximate geographical location in which a photo was taken. Achieving this goal typically involves using an image retrieval approach, where the query image to be located is compared against a geo-tagged database [2]. A very used approach is using CNNs to project images into an embedding space to obtain and use the similarity between the query image and the database ones. CosPlace [3] is a method that leverages classification to be able to train the model on large datasets avoiding the computational weight of contrastive learning, that is commonly used for the VG task. It obtains state-of-the-art performance on VG but does not address the loss of performance caused by domain shift.

Domain shift Photos of the same place can be very different. Specifically the same place can be photographed using different devices, but also under varying weather conditions and times of the day, leading to significant changes in the characteristics of the image. As a result, the diversity of images arising from different times, weather and cameras results in the emergence of visual domain shifts, which pose a significant challenge in developing robust visual geo-localization models.

Contributions In this study, we addressed this limitation specifically for night-time images by proposing some changes of the Cosplace Model:

(a) A **Data Augmentation** procedure that tries to approximate night time images starting from day light images, in order for the model to better learn how to extract features that are characteristic of that domain.

(b) Introducing a **Domain Adaptation** module that leverages the idea of using the Gradient Reversal Layer [7], which allows the network to learn features that are invariant with respect to the shift between the source domain (day-light images) and the target domain (night-time images). We also tried to overcome some limitations of this framework through Asymmetric Adversarial Domain Adaption [18] .

In an attempt to enhance the overall performance of the model, we also conducted the following tests:

(c) The use of the **Model Soup** ensemble technique to try to obtain better performance by combining many models into a single one, with no additional costs at inference time.

(d) Testing other losses and fine-tuning networks pretrained on different datasets to investigate the role of the pre-train in our task.

We show that some of these methods were successful in improving the accuracy of night-time image location, surpassing the performance of current default CosPlace model. Notably, our methods achieved this improvement without significantly reducing the performance of day-light image location.

2. Related Works

In this section, we will explore related works that have been published previously on this topic or served as inspiration for our study.

Visual geo-localization VG is usually seen as a image retrieval problem, and is usually approached with the use of contrastive learning through a triplet loss. One approach

is to use a backbone followed by an aggregation module, one particularly noteworthy example is NetVLAD [2]. The baseline used to start our project is CosPlace [3] that uses classification task as a proxy to extract descriptors to use for retrieval. The authors present a new method for partitioning an image database into classes using the UTM coordinates and orientation of each image. They then build groups of nonadjacent classes, called CosPlace groups. Each class belongs to exactly one group, and images of two different classes within the same group are sufficiently separated in order to not show the same scene. The training is achieved by iterating on every group, that can be considerend a dataset on its own. Our study extends this work by focusing on enhancing the performance specifically for night-time images.

Domain adaptation Unsupervised domain adaptation endeavors to address data distribution shift using labeled source data and unlabeled target data. A common approach to the problem is to transfer the image style, using GANs [9] or autoencoders [14]. Another way to proceed is to learn domain-invariant features from the data. Following this last approach we took inspiration from Adageo [4], where the authors proposed a method to perform localization of images using queries and database that do not belong to the same visual domain. They deployed a domain adaptation module in order to produce domain agnostic embeddings and an attention module to focus on the most important parts of the image for the retrieval task. The results obtained in Adageo were therefore an inspiration to introduce a domain adaptation module ourselves to obtain domain invariant embeddings.

Model ensemble There are different ways to maximize the model accuracy and, generally, it can be done by combining multiple models trained on the same data, leveraging the strengths of each one [1] [5]. Using the method presented in [17], we tried the uniform soup methodology, where we average the weights of different models exploiting the fact that fine-tuned models optimized independently from the same pre-trained initialization lie in the same basin of the error landscape.

Day-to-Night Image Synthesis The synthesys of night images starting from day-light images is a challenging task that, like the domain adaptation one, is usually approached using generative approaches [10] [20]. These methods, however, require a large amount of good quality night time images to train additional heavy models. A method that does not require any training nor any night-time images is presented in [12]. Although their objective was to train nightmode ISP networks, their main problem matched ours, that was to find a large dataset of night-time images. They developed a non-learnable method to synthetize night-time images starting from day ones. They start from RAW images and process them to obtain a pseudo night-time image. Taking inspiration from them, we developed our own method to syntethize night images that is different and more applicable since it is applied to jpeg images that are easier to gather.

3. Method

In this section we present our solution to improve the robustness of the network to domain shifts. The extension that we propose aims to improve the recall on the night-images dataset, while trying to keep stable the recalls on the other ones. The main extension consists of two components:

- A domain-driven data augmentation module, that tries to transfer the style of the night domain images to the day ones.
- A domain adaptation module to produce domain-agnostic embeddings.

The main idea is to use the data augmentation module to build a synthetic nocturnal version of the training set, that can be used during training to perform domain adaptation. The full pipeline can be observed in figure 1. The proposed extensions, as we will discuss later, can work only if there are available non-annoted night images. Due to the lack of these images in the train dataset, we used the test queries to implements the two aformentioned modules. To address this problem we looked for other solutions that do not require using the datasets directly, these include:

- The usage of different Angular loss functions, specifically the Sphereface Loss [11] and the Arcface Loss [6]
- fine-tuning models pre-trained on different datasets tailored to our task.
- An ensemble of various models.

Formally, we will denote the labeled source dataset as $\mathbf{X}^s = \{(x_i^s, y_i^s\}_{i=1}^{n^s}$, where x_i^s is an image coming from the source domain D^s, y_i^s is the geo-tag and n^s is the number of source images available; while the target dataset $\mathbf{X}^t = \{(x_j^t)\}_{j=1}^{n^t}$ similarly is composed of the night-time test queries deprived of their geo-tags during the training process. We will now proceed to describe every part of the process.

3.1. Fourier Domain Adaptation

The reasons behind the domain-driven data augmentation module, formally, is to find an effective mapping $D^s \mapsto D^{pt}$, where D^{pt} is the pseudo target domain that describe well the real target domain D^t, i.e. $D^{pt} \approx D^t$. Taking insipiration from [4], we generated a preliminary pseudo target dataset X^f by applying the Fourier Domain Adaptation [19]. By employing this transformation, we can remove environmental changes manifested as statistical differences in low-level factors across domains, without requiring an extra pre-training step for the backbone. Describing the process formally: let $\mathcal{F}^A, \mathcal{F}^P : \mathbb{R}^{3\times H\times W} \to \mathbb{R}^{3\times H\times W}$ be the amplitude and phase functions of the Fourier transform $\mathcal{F}$

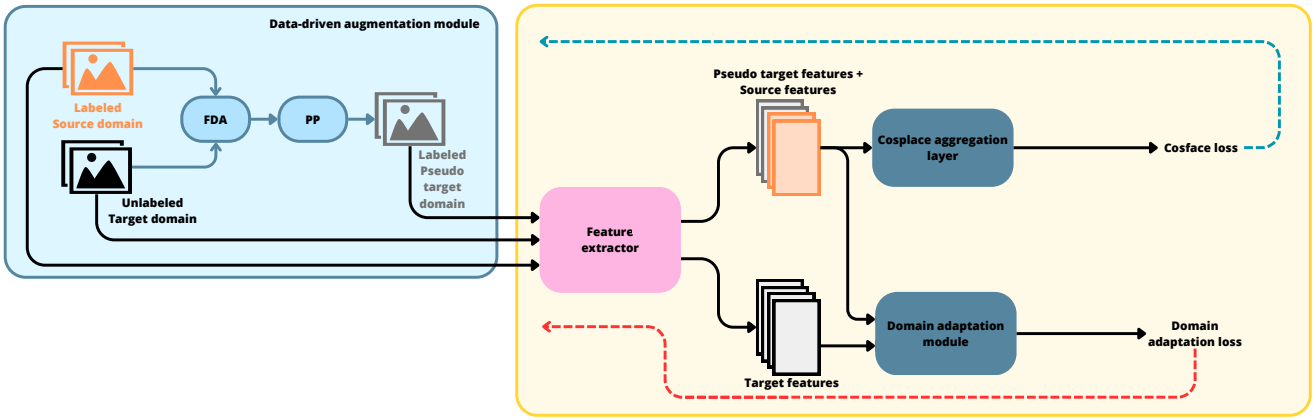


Figure 1. Training pipeline of our model. On the left there is the Data-driven augmentation module, from which we gather all the images used for train. On the right there is the model. Target domain images features are extracted and sent to the domain adaptation module with the other ones, labeled binary. On the CosPlace aggregation layer only the labeled images are sent. The dotted backward arrows represent the backpropagation process.

that take as input an RGB image. We denote with M_β a mask whose value are all zero except for the center region defined by an hyperparameter $\beta \in (0, 1)$:

$$M_\beta(h, w) = 1_{(h,w)\in[-\beta H:\beta H,-\beta W:\beta W]} \quad (1)$$

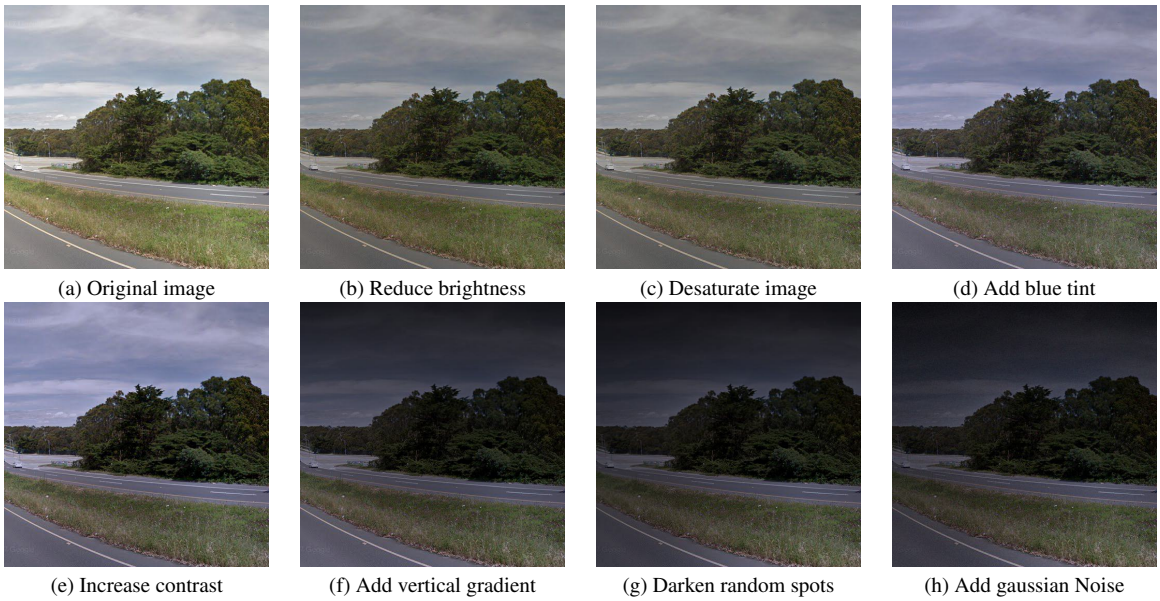
where the center of image is assumed to be in the position $(0, 0)$. Then, given two randomly sampled image, taken from sf-xs and from tokyo-night, i.e. $x^s \sim D^s$ and $x^t \sim D^t$, this step is formalized as:

$$x^f = \mathcal{F}^{-1}([M_\beta \circ \mathcal{F}^A(x^t) + (1 - M_\beta) \circ \mathcal{F}^A(x^s), \mathcal{F}^P(x^s)]) \quad (2)$$

where the frequency part of the amplitude of the source image $\mathcal{F}^A(x^s)$ is replaced by that of the target image x^t, then keeping untoched the phase component of the source image, the modified spectral representation of it is mapped back into the image space trough the inverse of the Fourier transform $\mathcal{F}^{-1}$. The result x^f has the content of x^s, but will resemble the appearance of the sample x^t.

3.2. Night time image synthesis

To provide further data augmentation, we developed a way to manipulate the day-light images of the training dataset to appear as night images. The process took inspiration from the one in [12], that had the same aim, however the transformations we used are different since they were not directly applicable to *jpeg* images. The application of this process can be observed in Figure 2. In particular the transformation pipeline is:
(a) Take the original image x_i^s
(b) Reduce the overall brightness of the image
(c) Desaturate the image, since night time images have usually a lower saturation than day time images.
(d) Adding a slight blue tint, by increasing the blue channel value.
(e) Increase contrast, in fact nighttime images typically have higher contrast compared to day-light images due to the presence of intense lights, such as streetlights, which produce bright regions in the image.
(f) Apply a vertical gradient that goes from black to transparent. This was done to darken the bright sky in day-light images by applying a vertical gradient that transitions from black to transparent, simulating the dark sky of night-time images.
(g) Darken random spots of the image by leaving invariant some circular, blurred spots, to account the fact that in nighttime images there are lights that brighten only part of the image, leaving the rest unilluminated
(h) Add gaussian noise. Noise is commonly observed in nighttime due to low light conditions and limitations of imaging sensors.
The process will be called PP (Post Process) and be doneted as P from now on. Applying P to the whole dataset, we can obtain a *Pseudo target dataset* $X^{ppt} = \{(P(x_i^s), y_i^s\}_{i=1}^{n^s}$. For the purpose of the study, we noticed that the FDA approach was good at simulating lamppost and billboards lighting, while PP created a night look. So in the end the pseudo target dataset we ended up creating is $\mathbf{X}^{pt} = \{(P(x_i^f), y_i^s\}_{i=1}^{n^s}$. where x_i^f is the image obtained by applying FDA to the source image x_i^s. The process can be seen in Figure 3. By synthesizing night images, we were able to create additional samples that were not present in the original dataset to try to get better results on night-time queries without having any real night-time images to train on.



(a) Original image (b) Reduce brightness (c) Desaturate image (d) Add blue tint

(e) Increase contrast (f) Add vertical gradient (g) Darken random spots (h) Add gaussian Noise

Figure 2. Night time image synthesis steps

(a) Original image (b) FDA applied (c) PP applied

Figure 3. Pseudo night images pipeline with FDA and PP

3.3. Gradient Reversal Layer

Taking inspiration from [7], we introduce on top of the image features extracted by the backbone, a new head that acts as a domain classifier. The architecture of the domain discriminator is taken from [4], which is composed by two fully connected layers and how the name suggests it is in charge of classifying the domain of the embeddings. The unsupervised domain adaptation is achieved by introducing between the extractor and the domain classifier a Gradient Reversal Layer that acts as an identity function during the forward step while in the backward one multiplies the gradient by the hyperpameter $-\alpha$ that controls the importance of the transferability. Describing formally, denoting the backbone G_f and the label predictor G_y the optimization of the main branch can be described as:

$$\min_{G_f, G_y} \mathcal{L}_{CE}(\mathbf{X}^s, \mathbf{X}^{pt}) - \alpha \mathcal{L}_{DA}(\mathbf{X}^s, \mathbf{X}^{pt}, \mathbf{X}^t) \quad (3)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss and $\mathcal{L}_{DA}$ is the loss of the discriminator, which in this case is described by a binary cross entropy loss in which the labels reflects the belonging domain. The use of this new resulting loss set up an "adversarial" effect, in which the backbone will be trained to maximize the domain loss while the discriminator to minimize it. In the end, it will be able to produce features that are discriminative and at same time domain-invariant, improving the performances against target domain test sets.

3.4. Asymmetric Adversarial Domain Adaptation

As have been highlighted by [18], even if the symmetric Domain Adaptation Training improves the transferability across domains, it sacrifices the discriminability in the target one. This kind of traning pushes the domains as close as possible, and in the worst case the backbone will generate features that are not discriminative for the target domain samples, as long it is able to fool the discriminator. This happens since this symmetric training involves both the domains in the adversarial features learning. Fixing the source domain and only moving the target domain to approach it, the feature discriminability is preserved. For this purpose, the domain adaptation module described before can be replaced by an autoencoder (AE) that acts as domain classifier with margin m, which is a hyper-parameter. The optimization proposed by this framework can be formally described as:

$$\min_{G_f, G_y} \mathcal{L}_{CE}(\mathbf{X}^s, \mathbf{X}^{pt}) + \gamma \mathcal{L}_{AE}(\mathbf{X}^{pt}, \mathbf{X}^t) \quad (4)$$

$$\min_{G_a} \mathcal{L}_{AE}(\mathbf{X}^s) + \max(0, m - \mathcal{L}_{AE}(\mathbf{X}^{pt}, \mathbf{X}^t)) \quad (5)$$

where γ acts as the λ of the GRL framework, G_a is the autoencoder and $\mathcal{L}_{AE}$ is its loss. This approach embodies only the $\mathcal{L}_{AE}(\mathbf{X}^{pt}, \mathbf{X}^{t})$ term into the training of G_f, pushing the target domain towards the source one while keeping the latter fixed. On the contrary the training of G_a will push the domains away from a margin m.

3.5. Model Soups

Model soups [17] is a non-traditional ensemble technique by which a final model is generated by averaging the weights of multiple models that share the same architecture at inference time in order to improve accuracy and robustness. Lets consider a neural network $f(x, \theta)$ with input data x and parameters θ. For hyperparameters configurations $h_1, ..., h_k$ let $\theta_1, ..., \theta_k$ be the weights obtained after training the model. The model soups technique instead produce a model $f(x, \theta_S)$ with:

$$\theta_S = \frac{1}{k} \sum_{i=1}^{k} \theta_i \tag{6}$$

The above equation describe, to be precise, the *uniform soup* recipe of the technique by which the final model is constructed by averaging *all* the models. Although Model Soups is intended to be used to "ensemble" models that share the same architecture in every phase of the machine learning pipeline, it is possible to observe that more generally the only constraint that the fused models need to satisfy is to have only at inference time the same layers described by θ_i. Thanks to this observations we are able to merge various models that are not trained using different hyperparameters configuration h_i, but that have different training-exlusive modules at training time like the domain classifier head. For this reason, we souped up models trained with different loss functions in order "to average" the benefits provided by them.

4. Datasets

We conduct experiments on three different VG datasets.
sf-xs is the dataset that was used for training and is a subset of San Francisco eXtra Large [3]. The images in this dataset are created from Google StreetView's equirectangular panoramas and contain 6 DoF information. The training set is composed of 59650 images divided equally in 5965 classes and, coming from this dataset, there is also a test set composed of a database of 27191 images and 1000 queries. San Francisco eXtra large was reduced because of the limited capability of the computing power we had available. A validation set is also present with 8015 database images and 7993 queries.
tokyo-xs is a subset of Tokyo 24/7 [15]. This contains images taken with smartphones that contains images taken at different time of the day, proving to be an invaluable asset for our study on generalization. This dataset was used to test the model and contains 12771 database images and 315 queries.
tokyo-night is a subset of tokyo-xs, built by keeping the database as it is and extracting from the queries only the ones that were taken at night. This was used as a test set to see how the models would perform specifically on night images and is composed of 105 queries.

5. Experiments

5.1. Training Setup

The computational resources at our disposal were modest, which resulted in us conducting the tests utilizing only the resnet18 [8] backbone pretrained on IMAGENET1K_V1 [13], finetuning from the third convolutional layer and freezing the rest. As mentioned before, we also tested the same backbone pretrained on Places365 [21] and GLDv2 [16]. The former is a scene recognition dataset with 1.8 million training samples, the latter is a dataset which contains images annotated with labels representing human-made and natural landmarks. The tests were executed over a span of three epochs, each consisting of 10,000 iterations, with a batch size of 32. The optimization process was performed using the Adam optimizer, with a learning rate of 10^{-5}. With regards to symmetric domain adaptation, given the fact that there were only two domains: *day and night*, we chose cross-entropy as loss function. The evaluation metric adopted for the evaluation is the *Recall@N* with N equal to (1, 5, 10), that is the percentage of queries for which at least one of the first N predictions is withing 25 meters distance from the query.

5.2. Ablation on the Angular loss function

We conducted an ablation study to see how different angular margin penalty-based losses perform for our task. Two other losses were considered: **SphereFace loss** and **ArcFace loss**. As we can observe from table 1, cosface outperforms the other two losses on sf-xs. This is probably due to the fact that cosface is better able to minimize the intra-class covariance, that is necessary for our retrieval task since each CosPlace class should contain the same scene. For the same reason Arcface performs worse since it relaxes the intra-class constraint of forcing all samples close to the corresponding positive centers. Regarding tokyo-xs they all perform similarly, but we can observe a big difference on tokyo-night. In fact, Sphereface obtains better results on this dataset. This is probably because Cosface and Arcface use a constant margin that does not depend on the value of the angle, but this doesn't allow generalization if the training is only on day images, when night images arrive the model finds it hard to project them in the correct place in the space.

Loss function	sf-xs	tokyo-xs	tokyo-night
Cosplace Default	**52.7** / **65.5** / **70.8**	70.2 / 84.1 / **89.5**	52.4 / 70.5 / **80.0**
Arcface (s = 64, m = 0.5)	48.4 / 61.1 / 66.0	70.2 / **84.4** / 87.9	51.4 / **74.3** / **80.0**
Sphereface (s = 30, m = 1.5)	50.4 / 63.7 / 68.6	**71.7** / **84.4** / 87.9	**56.2** / 70.5 / 77.1

Table 1. R@1 / R@5 / R@10 obtained using Cosface, Arcface and Sphereface loss.

Augmentation Type	sf-xs	tokyo-xs	tokyo-night
Cosplace Default (colorJitter)	52.7 / 65.5 / 70.8	70.2 / 84.1 / **89.5**	52.4 / 70.5 / 80.0
No colorJitter	**56.9** / 68.0 / 73.1	65.7 / 81.6 / 84.8	38.1 / 60.0 / 65.7
Lower brightness	56.7 / 68.6 / 73.0	67.0 / 82.2 / 87.3	41.9 / 62.9 / 71.4
Lower contrast	56.4 / **69.6** / **74.0**	69.2 / 82.2 / 88.6	42.9 / 61.9 / 74.3
Lower saturation	56.1 / 69.4 / 73.7	66.0 / 80.6 / 86.3	38.1 / 59.0 / 70.5
Lower BCS	56.4 / 69.5 / **74.0**	67.9 / 82.2 / 86.0	41.9 / 61.9 / 68.6
PP	55.4 / 68.2 / 71.8	70.5 / 83.8 / 87.0	55.2 / 68.6 / 73.3
FDA	56.0 / 65.8 / 71.3	68.6 / 83.2 / 89.2	52.4 / 67.2 / 78.1
PP (colorJitter)	51.7 / 66.8 / 70.8	69.5 / 83.8 / 87.6	54.3 / **74.3** / **81.9**
FDA (colorJitter)	51.3 / 65.2 / 69.8	**71.7** / 83.8 / 88.3	53.3 / 69.5 / 77.1
FDA on Test	52.1 / 66.6 / 70.6	70.2 / **84.4** / 88.6	**56.2** / 71.4 / 81.0
FDA+PP	51.0 / 65.9 / 71.0	69.8 / 84.1 / 88.6	55.2 / 71.4 / 81.0

Table 2. R@1 / R@5 / R@10 for different augmentation types. the use of color jitter is here explicitly specified because it was removed for the other experiments to have a fair comparison of the different methods. BCS stands for the combination of brightness, contrast and saturation applied sequentially to an image.

5.3. Data-driven augmentation

In Table 2 we reported the results of different augmentation techniques used to show the effectiveness of our method. The comparisons are done with CosPlace default.

Pytorch transformations The comparison was conducted by using the default Pytorch transformations for brightness, contrast and saturation. We can see that just using the basic transformations, the performance on sf-xs is increased but has poor performance on tokyo-night, and this is expected since the model lacks of generalization. This also shows that default Pytorch transformations are not enough to generate pseudo night images. Utilizing color jitter does not contribute to enhancing the performance either.

Our augmentation We can immediately notice that our results for sf-xs and tokyo-xs are generally inferior when compared to the baseline, however, this was anticipated as the primary focus of our research was to improve results for tokyo-night. This was achieved by synthesizing night-images for training, which in turn resulted in fewer day images observed during training. Good results are obtained with our method **PP**, that increases the performance on the R@1 on all the datasets. Applying only **FDA** increases the performance on sf-xs, probably because the lack of color jitter decreases generalization, but is able to maintain stable results on tokyo-night, showing that even though we have no benefit, it still gives some information to better recognize night images than the pytorch transformations. We can see that **PP coupled with color jitter** is able to obtain better performance on tokyo-night, showing again that our method is effective at synthetizing night-time images. **FDA with color jitter** show a little improvement on tokyo-xs and tokyo-night. It is likely that the FDA can assist in synthesizing nighttime images, but it doesn't provide significant additional help beyond what color jitter already accomplishes with its regularization effect. On the other hand, **FDA applied on the test set** actually gave us better results than the others on tokyo-night, proving that FDA is better able to transform night-time images to day-light ones than viceversa. The results on sf-xs and tokyo-xs remained basically unchanged because transferring the style of day images to other day images doesn't seem to affect the images too much. Optimal performance on tokyo-night were obtained with **FDA + PP**. We think that this happens because FDA used to go from day to night is able to transfer light information (given by streetlights or billboards) to the images, that are then darkened by the PP procedure to appear more nocturnal (see Figure 3). This allows us to get better night synthetic images and, therefore, better results.

5.4. Domain Adaptation

The results of our study on domain adaptation without data augmentation are shown in table 3.

Domain adaptation Looking at the two adversarial domain adaptation techniques **GRL** and **AE**, we can see that they improve performances on tokyo-night, remaining stable on

Model	sf-xs	tokyo-xs	tokyo-night
Cosplace default	52.7 / 65.5 / 70.8	70.2 / 84.1 / 89.5	52.4 / 70.5 / 80.0
GRL	52.7 / **68.0** / **73.0**	71.7 / 86.7 / 91.1,	56.2 / **78.1** / 84.8
AE (m = 0.5, γ = 0.01)	**53.6** / 66.5 / **73.0**	**74.0** / 86.7 / 91.1	**57.1** / 77.1 / **85.7**
AE (m = 5, γ = 0.01)	**53.6** / 66.5 / **73.0**	**74.0** / 86.7 / 91.1	**57.1** / 77.1 / **85.7**
AE (m = 5, γ = 0.2)	52.6 / 66.7 / 72.0	73.3 / **87.0** / **92.1**	**57.1** / **78.1** / **85.7**

Table 3. R@1 / R@5 / R@10 for different models using the domain adaptation techniques without data augmentation.

Model	sf-xs	tokyo-xs	tokyo-night
Cosplace default	**52.7** / **65.5** / **70.8**	70.2 / 84.1 / 89.5	52.4 / 70.5 / 80.0
FDA+PP+GRL (α = 0.01, β = 0.01)	50.4 / 63.9 / 69.0	71.4 / 86.0 / 89.8	59.0 / 79.0 / 83.8
FDA+PP+GRL (α = 0.1, β = 0.01)	50.9 / 63.4 / 69.1	73.3 / 85.4 / 88.6	61.0 / 81.0 / 83.8
FDA+PP+GRL (α = 1, β = 0.01)	47.2 / 62.5 / 67.7	58.1 / 71.4 / 78.1	**68.3** / **82.9** / 87.3
FDA+PP+GRL (α = 0.1, β = 0.09)	46.5 / 61.8 / 68.3	71.4 / 84.8, / 89.5	64.8 / 74.3 / 81.0
FDA+PP+GRL (α = 0.1, β = 0.05)	48.4 / 63.3 / 69.4	73.7 / 87.0 / 90.5	61.9 / 78.1 / 84.8
FDA+PP+AE (m = 0.5, γ = 0.01)	49.7 / 64.1 / 68.6	70.5 / 85.1 / 90.5	58.1 / 75.2 / 83.8
FDA+PP+AE (m = 0.5, γ = 0.02)	48.7 / 63.4 / 69.6	72.1 / 86.3 / 90.5	60.0 / 80.0 / **87.6**
FDA+PP+AE (m = 0.5, γ = 0.2)	51.8 / 65.0 / 69.5	**74.9** / **87.3** / **91.1**	61.9 / 78.1 / 84.4
FDA+PP+AE (m = 5, γ = 0.2)	51.7 / 64.8 / 68.9	71.4 / 84.8 / 89.5	57.1 / 76.2 / 82.9

Table 4. R@1 / R@5 / R@10 for different models using the pseudo target images for domain adaptation.

the other two datasets, regardless the hyperparameters chosen, probably because of the few real target images available, showing the need of a data augmentation module to increase night images. The employing of the adversarial domain adaptation training, through the use of the GRL or the AE, improves the performances of the model on tokyo-night as expected. Comparing the latter to the first, we can observe a very small improvements on this test-set which means that the asymmetric adversarial domain adaptation is already capable to provide meaningful features for night queries. However, probably, using a bigger test-set we would appreciate more the benefits of the asymmetric training. The usage of the AE framework help the model to generalize better also on not night-time queries by fixing the source domain,which lead the model to generate more discriminative descriptors. As a result of that we have a modest improvements against sf-xs and tokyo-xs. Given these promising results, what was left was to use a model that uses all these techniques all together.

5.5. Domain adaptation using pseudo target images

We now present in table 4 the results for the domain adaptation achieved using pseudo images.

FDA + PP + GRL Considering a β value of 0.01 for FDA, it can be seen that this approach allows us to get great results on tokyo-night without damaging the performance on the other 2 datasets using α = 0.1. This improvement can be attributed to the increased diversity in the training data and the GRL framework that allows the the backbone to generate domain-invariant embeddings. Regarding other values of the hyperparameter α, we can observe that $\alpha = 0.01$ gives us worse performance on tokyo-night, this is because it gives less importance to the domain adaptation module, that in return obtains better performance on sf-xs. However, increasing it too much ($\alpha = 1$), we're able to obtain great results on tokyo-night, but at a huge cost for the performance on the other two datasets. Also, training the network on pseudo target datasets with higher values of β results in the dataset incorporating excessive information about the target domain. This leads to significant improvements in performance on the test set consisting of images from that particular domain. However, there is a considerable drop in performance on the other two datasets. As stated in [19], choosing a wider mask, the syntethized images will approach more to the target one at the cost of introducing visible artifacts.

FDA+PP+AE Ascertained that $\beta = 0.01$ is the best choice for FDA, we also used it for the asymmetric domain adaptation and we can notice that results are similar to the usage of the GRL on tokyo-night but better on the other two datasets for the same reasons that were discussed before in 5.4 about the usage of AE without pseudo target images. It can be interesting to notice that bigger γ has positive effects on sf-xs compared to smaller γ. This counter intuitive behaviour is the result of the stronger effect of the asymmetrical domain adaptation that is able to introduce some regularization by focusing on more discriminant features, thereby compensating for the introduction of pseudo target images.

Dataset Used	sf-xs	tokyo-xs	tokyo-night
Cosplace Default (IMAGENET1K)	52.7 / 65.5 / 70.8	**70.2** / **84.1** / 89.5	**52.4** / **70.5** / 80.0
Places365	**53.6** / **66.8** / **71.9**	69.8 / **84.1** / **90.5**	48.6 / **70.5** / **82.9**
GLDv2	51.5 / 65.4 / 69.7	66.3 / 81.3 / 87.0	44.8 / 64.8 / 74.3

Table 5. R@1 / R@5 / R@10 for the backbone pre-trained on different datasets.

Uniform Soup	sf-xs	tokyo-xs	tokyo-night
Cosplace Default	**52.7** / **65.5** / **70.8**	70.2 / 84.1 / **89.5**	52.4 / 70.5 / 80.0
FDA+PP+GRL (α = 0.1, β = 0.01)	50.9 / 63.4 / 69.1	**73.3** / **85.4** / 88.6	**61.0** / **81.0** / **83.8**
Angular losses	51.6 / 65.1 / 70.1	68.9 / 82.5 / 87.9	52.4 / 71.4 / 77.1
FDA+PP+GRL α soup	49.8 / 64.7 / 69.8	72.1 / 84.1 / 89.2	57.1 / 78.1 / **83.8**
Mixed	51.4 / 65.3 / 70.3	70.8 / 84.1 / 88.9	55.2 / 75.2 / 81.9

Table 6. R@1 / R@5 / R@10 obtained using Uniform Soup. Angular losses is a soup composed of models trained with Cosface, Sphereface, Arcface. FDA+PP+GRL α soup is obtained using the different values of α. Mixed is obtained by mixing the different models we trained using data augmentation with and without domain adaptation.

5.6. Ablation on Places365 and GLDv2

We have conducted our experiments also with a pre-trained backbone on Places365 Standard and Gldv2 because they're mainly used for image retrieval task. As we can see from table 5, on sf-xs the performance with pre-training on Places365 is slightly higher than ImageNet1K, while the performance on GLDv2 is generally lower than that of the other two neural networks. In particular, if we consider the Geolocation Task, the Places365 dataset is beneficial because it contains urban scenes, which can help in the classification of the geolocalization context. Furthermore, the Places365 dataset was specifically designed for scene image classification and contains a large variety of images, which could lead to better generalization compared to other datasets such as GLDv2 or IMAGENET. However on Tokyo-night Places365 performs worse, probably because IMAGENET contains images depicting a wide range of different objects, leading to better generalization and so performing better when the domain changes. GLDv2 performs worse than the other two in both datasets, we think this happens because the urban landmarks present in the dataset depict architectures that are very different from the ones you observe usually on the streets.

5.7. Model Soups

We conducted experiments using the uniform soup recipe, using for each run "ingredients" that share some characteristics (same architectures with different hyperparameters or optimized for a similar "objective"), the results are shown in table 6. The use of this ensemble technique did not produce better results compared to the best performing ingredient used, on the contrary the performances worsen a little against all the test sets considered. This is due to the fact that this protocol, does not aggregate the decision of the elementary models (i.e. through a voting scheme), but linearly combines the weights of them as one. This leads the final encoder to a state in which, likely, the parameters set is not a local minimum for the loss function. Although the final models may not be remarkable, their performances closely align with their respective baselines (since the models are trained similarly, they likely show similar parameter sets).

6. Conclusions

In this work we have proposed several methods to improve the performance of CosPlace on night images, trying to keep the results on day images unchanged. We introduced a non-learnable data augmentation process to synthesize labeled pseudo-night images, composed by two independent phases in which only the first requires a small number of night images. Furthermore, we used two distinct types of domain adaptation methods, namely RevGrad-DANN and AADA, in order to build models robust to domain shift. By combining these techniques, the model was able to surpass the performance of the baseline on night-time queries. Additionally, it managed to maintain relatively stable performance on daylight queries. We also studied the effect of fine-tuning backbones pre-trained on Places365 and GLDv2, which produced interesting results. Different loss functions, specifically ArcFace and SphereFace, were investigated to determine if they could contribute to improve the results compared to the baseline and "Model Soups" was tested to create ensemble models. While not all of these techniques resulted in performance improvements, they provided valuable insights for further exploration. We hope that the proposed strategies and the achieved results will inspire and drive further research and advancements in this rapidly evolving field.

References

[1] Kamal M Ali and Michael J Pazzani. Error reduction through learning multiple descriptions. *Machine learning*, 24:173–202, 1996. 2

[2] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition, 2016. 1, 2

[3] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4878–4888, June 2022. 1, 2, 5

[4] Gabriele Moreno Berton, Valerio Paolicelli, Carlo Masone, and Barbara Caputo. Adaptive-attentive geolocalization from few queries: a hybrid approach. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, jan 2021. 2, 4

[5] Leo Breiman and Philip Spector. Submodel selection and evaluation in regression. the x-random case. *International statistical review/revue internationale de Statistique*, pages 291–319, 1992. 2

[6] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5962–5979, oct 2022. 2

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation, 2015. 1, 4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 5

[9] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018. 2

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. 2

[11] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition, 2018. 2

[12] Abhijith Punnappurath, Abdullah Abuolaim, Abdelrahman Abdelhamed, Alex Levinshtein, and Michael S. Brown. Day-to-night image synthesis for training nighttime neural isps, 2022. 2, 3

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[14] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi. Vigan: Missing view imputation with generative adversarial networks, 2017. 2

[15] Akihiko Torii, Relja Arandjelović, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1808–1817, 2015. 5

[16] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *Proc. CVPR*, 2020. 5

[17] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022. 2, 5

[18] Jianfei Yang, Han Zou, Yuxun Zhou, Zhaoyang Zeng, and Lihua Xie. Mind the discriminability: Asymmetric adversarial domain adaptation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 589–606, Cham, 2020. Springer International Publishing. 1, 4

[19] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation, 2020. 2, 7

[20] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation, 2018. 2

[21] 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. 5