Visual Geo-Localization: mapping images to GPS

Testa Mario
Politecnico di Torino
mario.testa@studenti.polito.it

Cambria Paolo
Politecnico di Torino
paolocambria@studenti.polito.it

Ciciriello Giuseppe
Politecnico di Torino
giuseppe.ciciriello@studenti.polito.it

Abstract

Visual Geo-localization (VG) is the task of estimating the position where a given photo was taken by comparing it with a large database of images of known locations. In this work, we investigate the results of CosPlace [1] in the task of Visual Geo-Localization. Their method casts the training as a classification problem and achieves state-of-the-art performances on a wide range of datasets and domain changes. We conducted a thorough ablation study of the loss function by comparing it with SphereFace [2] and ArcFace [3] loss functions. Moreover, we carried out experiments on Domain Adaptation and Model Ensemble showing how a hybrid generative-discriminative approach helps robustness across domains, with special focus to the night one. Finally, we assessed the behavior of our best model in a qualitative way, showing the correct reasoning in the image retrieval task. Dataset, code and trained models are available for research purposes at:
https://github.com/mariotesta-dev/CosPlace-Extended

1. Introduction

Visual geo-localization (VG) is one of the most promising approaches in the field of computer vision and image localization, due to his importance in applications such as autonomous driving [4] or even augmented reality [5]. VG usually consists of a place recognition task: given a query image of a place, its geographical location has to be roughly recognized and retrieved by finding the closest database geo-tagged images, usually with a tolerance of few meters. This task is extremely challenging due to the intrinsic dynamism of public places and different problems must be taken in account: there are a lot of moving objects which determine occlusion, environmental changes, different illuminations during daylight and night time, season changing. Furthermore, most of the recent learning based VG methods focus on recognizing the location of images in a relatively small sized geographical area (*e.g.* a neighborhood), which is not enough for real-world applications which are posed to operate at much larger scale (*e.g.* whole cities).

Non-representative datasets. To achieve a VG task on a wider geographical area, a large representative dataset is required and, as underlined by [1], the majority of current datasets are either too small in the geographical coverage [6–8], or too sparse [9, 10]. Moreover, those datasets split the collected images into disjoint sets for training and inference and this not suits a realistic use case, since the search query might be often an already seen place. For this reason it is recommendable to use the whole dataset to train the model, given also the cost of collecting the images for a consistent dataset.

Training scalability. Having access to a massive amount of data raises the question of how to use it effectively for training. Many of the recent SOTA (state-of-the-art) methods take advantage of contrastive learning [6, 11], which can often depend on contrasting positive to negative examples across the training dataset, a costly operation in terms of computation. CosPlace [1], instead, addresses this limitation by using:

- A new large-scale and dense dataset, called San Francisco eXtra Large (SF-XL), which includes multi-domain queries.
- A highly scalable training method, properly designed to work on large dataset, based on a classification task to produce a model that will later be used to extract descriptors for the retrieval

successfully reaching SOTA results with compact descriptors.

Contributions. In this paper, we want to investigate how the work of [1] can be improved for place recognition tasks, but using a smaller version of SF-XL dataset (called SF-XS)

for the training, due to our limited resources. In particular we focus on:

- Generally improve the recalls on SF-XS and Tokyo-XS (a smaller version of Tokyo 24/7 [12]), even with the help of ensembles to concatenate the descriptors of different models.
- Improve robustness to domain shift by borrowing techniques from the field of domain adaptation [11], and testing the results on a dataset which only contains night images, called Tokyo-Night (a filtered version of Tokyo-XS) and deeply understanding the applied domain adaptation techniques.
- Assess how CosPlace behaves when data quality is scarce (*e.g.* occlusion and blurry photos).

Through some experiments, we demonstrate that our solution exceeds the baseline provided by CosPlace on three different datasets and that the core ideas of the architecture (data augmentation, both domain-driven and non, and subsequent domain adaptation) provide modest increments to the performance.

2. Related Works

Visual geo-localization as image retrieval. Visual geolocalization on large scale is commonly considered as an image retrieval task, in which the correctness is determined by an established tolerance (usually 25 meters) from the query's ground truth position [6, 11]. One of the most representative study in this field is NetVLAD [6], which introduces a VLAD layer, which has parameters learnable with back-propagation, that pools descriptors extracted from a CNN backbone into a fixed image representation.

Visual geo-localization as classification. An alternative approach to visual geo-localization is to consider it a classification problem, as done in [1]. Most of the methods of this kind, divide the geographical area of interest in cells and group the database of images in classes according to their cell, which has a big limitation: nearly identical images may be assigned to different classes due to quantization errors. CosPlace, instead, proposes to train the model only using groups of non-adjacent classes and iterates over them while using CosFace [13] as a scalable loss.

Deep Face Recognition. In deep face recognition (FR) problem under open-set protocol, ideal face features are expected to have smaller maximal intra-class distance than minimal inter-class distance under a suitably chosen metric space. The most widely used classification loss function in the field, softmax loss, is presented as follows:

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \tag{1}$$

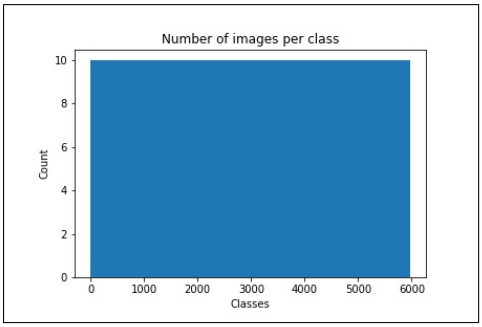
where $x_i \in \mathbb{R}^d$ denotes the deep feature of the i-th sample, belonging to the y_i-th class, W_j denotes the j-th column of the weight W and b_j is the bias term. The batch size and the class number are N and n, respectively. However, the softmax loss function does not explicitly optimise the feature embedding to enforce higher similarity for intraclass samples and diversity for inter-class samples. So, by moving features from an Euclidean space to an angular one and introducing a margin, different methods [2, 3, 13] have shown how learning angularly discriminative features on a hypersphere manifold with an adjustable margin helps to improve accuracy on both verification and identification tasks. The most representative studies in this field are SphereFace [2], CosFace [13] and ArcFace [3] where three different kinds of margin penalty are proposed, e.g. multiplicative angular margin m_1, additive angular margin m_2, and additive cosine m_3, respectively. The following equation combines all the above mentioned margins and can derive the three different losses by turning off the two unused margins for each method:

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}}{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}+\sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}} \tag{2}$$

where we have fixed the bias $b_j = 0$, the logits have been transformed as $W_j^T x_i = ||W_j||\,||x_i||\cos\theta_j$, where θ_j is the angle between the weight W_j and the feature x_i. Following [2, 3, 13], the individual weight is fixed as $||W_j|| = 1$ by $l2$ normalization and the embedding feature $||x_i||$ is fixed as well by $l2$ normalization and re-scaled to s.

Unsupervised domain adaptation. Unsupervised domain adaptation attempts to reduce the shift between the source and target distribution of the data by relying only on labeled source data and unlabeled target data. For example, the source domain can consist of synthetic images and their corresponding pixel-level labels (*e.g.* for semantic segmentation), and the target can be real images with no ground-truth annotations. One approach for unsupervised domain adaptation is to learn domain-invariant features from the data, it was introduced by [14] and is based on a domain discriminator network with a gradient reversal layer (GRL) that forces feature extractor to produce domain-invariant representations. This has been used by AdaGeo-Lite [11] architecture, which combines a domain-driven data augmentation module that uses a non-learned style transfer method (called FDA [15]) producing a pseudo-target labeled dataset, with a network that produces domain-invariant image descriptors by setting up a

(a) Example of images from San Francisco XS dataset: they cover different time of the day, weather conditions, viewpoint and occlusions.



(b) Numbers of images per class in San Francisco XS dataset.

Figure 1. Example of a short caption, which should be centered.

min-max game where the discriminator tries to minimize the domain classification loss given three datasets (source, pseudo-target and a few samples from the target domain), while the feature extractor acts as an adversary to the discriminator.

3. Method

In this section, we present different approaches we have used to try improving CosPlace results. We started from its original implementation and decided to add custom augmentation to help boost its ability to learn embeddings even on sub-optimal picture quality. Then, we moved our focus to domain shift, given that test-images are likely to come from different domains than the source and that the domain shift in the dataset of interest Tokyo-Night is caused by illumination (day/night), we tried adapting the method used by [11] combined to CosPlace in order to create a modular architecture composed of two parts:

- A non-learned domain-driven data augmentation module that transfer the style of the target domain (night) to the source images.
- A network that produces the image descriptors, composed of a CNN, an aggregation layer and domain adaptation module.

At the end, we tried averaging the weights of models trained in the previous steps using Model Soups' approach [16] to improve recalls.

3.1. Datasets

For our work, we have been provided with two datasets to experiment with: San Francisco eXtra Small (**SF-XS**) and Tokyo eXtra Small (**Tokyo-XS**). These datasets are respectively subsets of SF-XL [1] and Tokyo 24/7 [12] purposely reduced for our convenience due to limitations in compute availability. SF-XS contains 59704 images (Fig. 1a) for the train (both queries and database), 16008 for the evaluation (8015 as database, 7993 as queries) and 28191 for the test (27191 as database, 1000 as queries). Tokyo-XS only contains 13086 images for the test (12771 as database, 305 as queries). All these datasets are provided with labels that can be derived from the images' names as $\{east,\ north,\ heading\}$ and specifically for the training dataset, there are 5965 different classes, each one of them containing 10 images, as shown in Fig. 1b. Finally, to understand if the changes applied to our model help in geolocalizing night images, we created a new dataset called **Tokyo-Night**: it is a subset of Tokyo-XS of which only the night images from queries are retained (1 out of 3, so 105 query images in total).

3.2. CosPlace

Our work starts from CosPlace, an innovative approach in the field of visual geo-localization [1]. CosPlace casts the network training as an image classification problem: it partitions the geographical area of interest in oriented cells, representing different classes, using UTM coordinates $\{east, north\}$[1] and orientation/heading $\{heading\}$. The extent of each class in terms of position and heading is defined by two parameters M and α, respectively. Then, it iteratively considers subsets of these cells (called groups G_{uvw}) to train the network. These groups are generated by fixing the minimum spatial separation that two classes of the same group should have, either in terms of translation or orientation. For this reason, they introduced two param-

[1]UTM coordinates are defined by a system used to identify locations on earth in meters, where 1 UTM unit corresponds to 1 meter. They can be extracted from GPS coordinates (i.e., latitude and longitude) and allow approximating a restricted area of the earth's surface on a flat surface.

eters: N controls the minimum number of cells between two classes of the same group, and L is the equivalent for the orientation. Therefore, the training is done sequentially over the groups as:

$$\mathcal{L}_{cosPlace} = \mathcal{L}_{lmcl}(G_{uvw}) \tag{3}$$

where $\mathcal{L}_{lmcl}$ is the Large Margin Cosine Loss as defined in [13], and $u \in \{0, ..., N\}, v \in \{0, ..., N\}, w \in \{0, ..., L\}$ represent the different values of $\{east, north, heading\}$. In our case, due to limitations in compute availability, we could only train each epoch on the same group G, so:

$$\mathcal{L}_{cosPlace} = \mathcal{L}_{lmcl}(G) \tag{4}$$

At validation and test time, we used the model generated not to classify the query, but rather to extract image descriptors as in [13] for a classic retrieval over the database. This allows for the model to be used also on other datasets from unseen geographical areas, like Tokyo-XS and the smaller Tokyo-Night.

3.3. Custom Augmentations

To assess the robustness of CosPlace [1], its capability to learn a good embedding space even on sub-optimal picture quality and try to find augmentations to achieve better generalization, we conducted a study on the augmentation pipeline already present in the original implementation (Color Jitter, RandomResizedCrop, Normalize) that we named **base**. During the training step, we added 4 new augmentations on the images after the default Color Jittering, with a 25% chance of being applied and a fixed seed for reproducibility purposes. Those augmentations consisted of:

- **Gaussian Blur**
  Blurring images can show the robustness of CosPlace on realistic use case in which the provided images are not always clear.
- **Grayscale**
  In the context of place recognition, Grayscale has already been successfully employed in [17] to reduce the effect of illumination variation, which suits also our case.
- **Horizontal Flip**
  The datasets in use contain information on orientation and heading of the images, applying a horizontal flip should prove consistency on the query images.
- **Erasing**
  Randomly erasing an amount of pixels in the picture, can simulate the realistic case of occlusion and usually improves the CNN-based recognition models as [18].

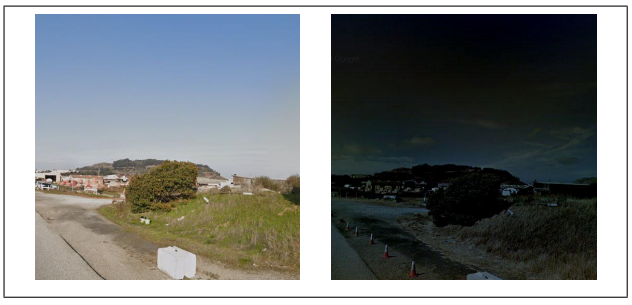


Figure 2. On the left the source image, on the right the pseudo-target image generated with FDA + saturation and temperature convertion.

3.4. FDA and GRL

Domain-Drive Data Augmentation. The purpose of the domain-driven data augmentation (DDDA) module is to find a mapping $D_s \mapsto D_{pt}$ from the source domain to a pseudo-target domain that better approximates the target domain, i.e., $D_{pt} \approx D_t$. This mapping can then be applied to the source dataset X_s to generate a new labeled dataset with pseudo-target images X_{pt}, it is a data augmentation technique and is performed only once, offline. Inspired by [11, 15], we used a DDDA method based on Fourier Domain Adaptation (**FDA**) [15] to generate a pseudo-target dataset X_{pt} given two randomly sampled images x^s and x^t from source and target. First, the low frequency part of the amplitude of x^s is replaced by that of x^t, then the modified spectral representation of the source is mapped back to an image whose content is the same as x^s but will resemble the appearance of a sample from the target distribution. Since the pseudo-target images generated didn't look as "night" as thought, for all the processed images, we reduced their saturation to 70% and converted their temperature towards blue, to give them a colder appearance, as shown in Fig. 2. Afterward, we use both X_s and X_{pt} to train the descriptor extraction network, leading to a more robust model.

Domain Adaptation Module. In order for the retrieval to work well across domains, the embeddings produced by the descriptor extraction network must be domain agnostic, i.e., they do not encode domain-specific information. We achieve this by using a domain discriminator composed of two fully connected layers which receives embeddings from the three domains D_s, D_{pt} and D_t, and its goal is to classify the domain to which they belong. Just before the discriminator, there is a gradient reversal layer (**GRL**) [14] that in the forward pass acts as an identity transform, while in the backward pass multiplies the gradient by -λ, where $\lambda > 0$. The use of this layer effectively sets up a min-max game, where the discriminator tries to minimize the domain classification loss, that is a cross-entropy loss L_{CE}, while the feature extractor learns to produce domain-invariant embeddings, acting as an adversary to the discriminator.

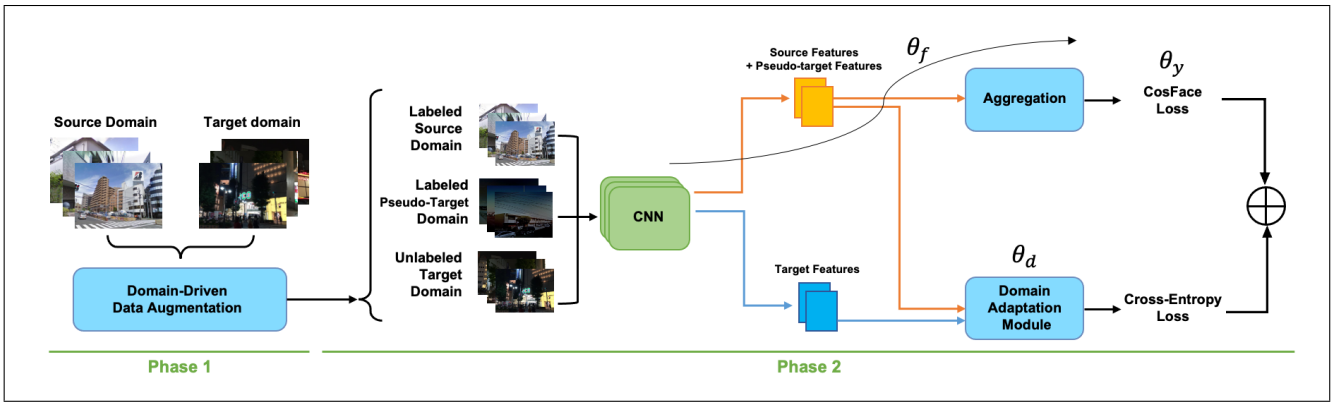


Figure 3. The proposed architecture. First, a domain-driven data augmentation method is used to generate a labeled pseudo-target dataset from the source dataset and just 5 unlabeled target images. Then, the source dataset, pseudo-target dataset, and the 5 unlabeled target images are used to train the network that extracts the image descriptors. This network leverages an aggregation module and a domain adaptation module to provide robustness to shifting.

Architecture details. The idea is that for each input x, we predict its class $y \in Y$ and its domain label $d \in \{0, 1\}$, depending if it is an image from day or night. We assume that the input x is the output of the frozen layers of our CNN backbone, which we call feature vector f. Then, we fine-tune the last layers of the CNN, the upper branch aggregator in Fig. 3 pools f (if it comes from the labeled source) with GeM Pooling to finish the image descriptor extraction, and we denote the parameters of this section with θ_f. Then the descriptor is mapped to its class label probabilities y by CosFace, with parameters θ_y. Finally, the same feature vector f is mapped to the domain label d by the lower branch in Fig. 3 (*domain classifier*), with θ_d as parameters. During the learning stage, we aim to minimize the label prediction loss on the labeled part (i.e. the source part) of the training set, and the parameters of both the feature extractor θ_f and the label predictor θ_y are thus optimized in order to minimize the cross-entropy loss for the source domain samples L_y. At the same time, to make the features f domain-invariant we seek θ_f that maximize the loss of the domain classifier L_d, while simultaneously seeking θ_d that minimize the latter. The trade-off between the two objectives is controlled by a parameter λ. The parameters update are similar to the ones of stochastic gradient descent (SGD) for a feed-forward deep model that comprises feature extractor fed into the label predictor and into the domain classifier, but the difference here is given from λ so a direct implementation of SGD is not possible. This can, instead, be accomplished by a Gradient Reversal Layer (GRL) which during the forward propagation acts as an identity transform and during the backpropagation takes the gradient from the subsequent level, multiplies it by $-\lambda$ and passes it to the preceding layer. At the end of the training phase, the two losses (L_y, L_d) are combined as $L_y + \alpha \cdot L_d$ where $\alpha = 0.1$.

3.5. Model Soup

The conventional way for maximizing model accuracy is to train multiple models with various hyperparameters and pick the one which performs best on a validation set, discarding the rest. A recent proposal [16] revisits the second step of this procedure, suggesting how averaging the weights of multiple models fine-tuned with different hyperparameter configurations often improves accuracy and robustness. This adds very little memory cost, learning time and inference time. There are two ways to combine two (or more) models to create a soup: *greedy* and *uniform*. In this project we implemented the greedy version of the proposed method and then we tested the resulting model on the three datasets of interest.
Considering the parameters of our N models $\{\theta_1, ..., \theta_N\}$ we express their average as follow:

$$\theta_{\mathcal{S}} = \frac{1}{N} \sum_{i=1}^{N} \theta_i \tag{5}$$

The uniform method simply averages all the parameters, but this can lead to worse results. We apply, instead, a greedy algorithm that sequentially adds each model as a possible ingredient and only keeps it if performances improve. This way, it's possible to get a resulting model that is better (or equal) than the best one, according to the following steps:

- We initialize an empty array of ingredients and then we iterate over the models, which are sorted in decreasing order of accuracy on a given validation set.
- In each iteration, we calculate the average between the model current ingredients and the current model parameters (θ_i).

- We calculate the new accuracy on the validation set and if it is greater than or equal to the previous one, the current model parameters are added to the ingredients.
- After all the iterations, the average of the final ingredients set is returned and the model is tested.

4. Experiments

Experiment with the Baseline. We started running some experiments to better understand how the training procedure worked. The backbone used was a ResNet-18 pre-trained on ImageNet with GeM pooling [19] and due to limitations in compute availability (*e.g.* Colab time and GPU limitations), we trained the baseline model for only 3 epochs to finally end up with the results in Tab. 1

	SF-XS(test)	Tokyo-XS	Tokyo-Night
R@1/R@5	**52.2/66.3**	**69.5/84.8**	**49.5/72.4**
R@10	71.8	89.2	79.0
R@20	76.3	92.7	84.4

Table 1. Baseline results, values refer to recall@K, for K={1, 5, 10, 20}

Ablation study when changing the angular loss function. Once we were familiar with the baseline, we started making a few modifications to the model, specifically to the loss function. Standard CosPlace uses the Large Cosine similarity loss [13], so we've tried replacing it with two other alternative cosine based losses: SphereFace [2] and ArcFace [3]. We trained the modified version on SF-XS and the results in Tab. 2 show how CosFace loss performs better than the competition on the biggest dataset (SF-XS), while on smaller ones (Tokyo-XS and Tokyo-Night) SphereFace and ArcFace actually get better recall values. This isn't what we expected at first, but further analysis suggest us that the implementations we've used for the last two losses might converge earlier than the original CosFace, so with a low number of epochs they perform better. It would be interesting to have some trials with the full SF-XL dataset and a greater number of epochs to better understand the behavior in the long run.

Augmentation Pipeline. Due to the high computational requirements, we were only able to conduct this ablation efficiently on SF-XS dataset with a lower amount of iterations per epochs fixed to 5000 and a batch size reduced to 16 instead of respectively 10000 and 32 for the default model. The results reporting recalls are presented in the table 3 above.

The couple of augmentations **{blur, grayscale}** seems the best choice for the augmentation pipeline, since not only the model manages to show robustness to that data augmentation pipeline, but also gets improvements. Further investigations should be performed with different parameters (*e.g.* number of iterations, batch size, augmentation intensity, etc.) to deeply understand their behavior, however it seems that applying grayscale to the model leads to better performances, since it might add invariance to the colors of the scenes changing through the day.

	SF-XS(test)	Tokyo-XS	Tokyo-Night
CosPlace with CosFace	**52.2/66.3**	69.5/84.8	49.5/72.4
CosPlace with SphereFace	49.7/64.2	**70.2/84.8**	**59.0/75.2**
CosPlace with ArcFace	49.7/61.1	69.5/81.6	56.2/64.8

Table 2. Results of our ablation study when changing the angular loss function from CosFace to SphereFace and ArcFace, the values refer to Recall@1/Recall@5.

Augmentations	R@1	R@5	R@10	R@20
Base	43.9%	61.7%	67.0%	71.2%
+{blur}	44.2%	60.4%	67.3%	70.8%
+{grayscale}	45.6%	60.9%	67.4%	70.7%
+{flip}	43.8%	62.0%	68.0%	71.9%
+{erasing}	44.0%	59.8%	65.0%	70.2%
+{grayscale, erasing}	43.4%	59.2%	65.2%	70.1%
+{blur, grayscale}	**45.9%**	61.3%	68.6%	70.9%
+{blur, erasing}	43.8%	60.2%	65.3%	70.4%
+{blur, grayscale, flip, erasing}	43.6%	59.1%	64.8%	70.4%

Table 3. Evaluation of the impact in terms of recall of different augmentation pipelines for CosPlace compared to Base (Color Jitter, RandomResizedCrop, Normalize)

Unsupervised Domain Adaptation with FDA and GRL. To evaluate the capability of our solution to generalize to unseen domains, specifically the night one, we compare the baseline, the unchanged architecture of the baseline trained with a dataset containing the source domain samples and target-domain samples generated with FDA, and the architecture with the domain discriminator module attached (FDA+GRL).

From the results in Tab. 5, we surpass the baseline in all the three datasets by using both FDA and GRL together, reaching a $\sim$ 4% improvement on Tokyo-Night. This confirms that the newly generated model was able to produce "more" domain-invariant features than before, with the downside of increasing the training time. Better can be expected by trying out different values for α or maybe performing a better post processing on the pseudo-

	SF-XS(test)	Tokyo-XS	Tokyo-Night
Baseline	52.2/66.3	69.5/84.8	49.5/72.4
FDA	50.5/65.4	67.0/85.7	47.6/72.4
FDA+GRL	**53.7/66.5**	**70.5/84.8**	**53.3/73.3**
FDA+GRL+g	53.4/65.5	67.6/84.8	48.6/72.4

Table 4. Baseline results compared with FDA only, FDA+GRL (with $\alpha = 0.1$) and FDA+GRL+grayscale.

	SF-XS(test)	Tokyo-XS	Tokyo-Night
Model Soup	**53.7/66.5**	**70.5/84.8**	**53.3/73.3**

Table 5. Baseline results compared with FDA only, FDA+GRL (with $\alpha = 0.1$) and FDA+GRL+grayscale.

target images generated. Finally, adding the most relevant custom augmentation (grayscale with 25% chance to be applied) didn't provide any improvement to the FDA+GRL model, showing that if real colors are available, they will outperform the generalization improvement of grayscale.

Averaging model weights with Greedy soup algorithm. To maximize model accuracy of our solution we pick all the best models trained and then merge them in a single model trying to improve performances without increasing inference time. The greedy soup is constructed by sequentially adding each model as a potential ingredient in the soup, and only keeping the model in the soup if performance on a held out validation set improves. For this purpose we decided to sort the models in a decreasing order of their recall values, in this way the soup have to be certainly better or equal than the best model found. Unfortunately, due to Colab limitations (in terms of batch size and epochs number for the training step) we had a limited number of models to use and we didn't achieve any real improvement on our best model (FDA+GRL) after testing on all the datasets available. This can happen if every time the next model in the list is added, it gets discarded because it doesn't provide any performance increment. This could be one of the approaches to be further explored, in the future, with more advanced tools.

Qualitative Evaluation with Top K-NN. To assess the goodness of the results obtained applying FDA and GRL in particular on Tokyo-Night dataset, we look at the nearest neighbors in the feature space, to understand the trained model's reasoning on different proposed queries. Therefore, we built a top K - Nearest Neighbors visualization for which given a query image, the K closest points in the learning embedding space are retrieved. In order to retrieve the top K = 5 most similar images to a given one, we calculate the Euclidean distance between the target image's embedding and each image's embedding in the dataset, one by one. For our experiment, we used and tested Tokyo-Night as database for the queries retrieval in Fig. 4 in two different ways. The model was trained with both FDA and GRL active, however:

- in the set **A** we didn't apply FDA to the dataset used for the Top K-NN retrieval, just to be sure that our trained model worked as intended with the original set, both on day and night queries. The result was perfect for the first 3 neighbors.
- in the set **B** we applied FDA to the dataset used for the Top K-NN retrieval and used as query the same image treated with FDA (on the left) and from the night domain (on the right). Once again the result was satisfactory and the FDA-treated image was perfectly recognized and integrated in both cases, proving the correct reasoning of our proposed model.

Figure 4. Given a target image, top 5 matching images are shown. Preset A) corresponds to Tokyo-Night dataset without FDA applied. Preset B) corresponds to Tokyo-Night dataset with FDA applied on the target image on the left.

5. Conclusions

In this paper, we proposed different improvements to CosPlace. In order to obtain better performances than the baseline on the night dataset, different from the training one for colors, brightness and city where the pictures were taken, we demonstrated that it is possible to improve domain shifting by applying FDA on the source dataset and using a Domain Adaptation module that forces the feature extractor to produce domain-invariant representations. We have also found that this solution improves the recalls of SF-XS and Tokyo-XS as well, and we assessed our results with a Top K-NN retrieval.
Furthermore, we confirmed our hypothesis of applying grayscale on images to improve illumination invariance.
Finally, we tried boosting recalls of our model with a Greedy Soup approach, but didn't get any enhancement or worsening.

We believe that this work can be still improved, possibly with the use of an attention layer before the aggregator, as done in [11] that, given this study field, on top of our proposals it might show interesting results.

References

[1] G. Berton, C. Masone, and B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 4878–4888. 1, 2, 3, 4

[2] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," 04 2017. 1, 2, 6

[3] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," 01 2018. 1, 2, 6

[4] C. McManus, W. Churchill, W. Maddern, A. D. Stewart, and P. Newman, "Shady dealings: Robust, long-term visual localisation using illumination invariance," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 901–906. 1

[5] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, "Scalable 6-dof localization on mobile devices," 09 2014. 1

[6] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 06 2017. 1, 2

[7] N. Carlevaris-Bianco, A. Ushani, and R. Eustice, "University of michigan north campus long-term vision and lidar dataset," *The International Journal of Robotics Research*, vol. 35, 12 2015. 1

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: the kitti dataset," *The International Journal of Robotics Research*, vol. 32, pp. 1231–1237, 09 2013. 1

[9] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, 11 2016. 1

[10] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, and J. Civera, "Mapillary street-level sequences: A dataset for lifelong place recognition," 06 2020, pp. 2623–2632. 1

[11] V. Paolicelli, G. Berton, F. Montagna, C. Masone, and B. Caputo, "Adaptive-attentive geolocalization from few queries: A hybrid approach," *Frontiers in Computer Science*, vol. 4, 06 2022. 1, 2, 3, 4, 8

[12] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *CVPR*, 2015. 2, 3

[13] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," 01 2018. 2, 4, 6

[14] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," 09 2014. 2, 4

[15] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," 04 2020. 2, 4

[16] M. Wortsman, G. Ilharco, S. Gadre, R. Roelofs, R. Gontijo-Lopes, A. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," 03 2022. 3, 5

[17] A. J. Zetao Chen, Obadiah Lam and M. Milford, "Convolutional neural network-based place recognition," 11 2014. 4

[18] G. K. S. L. Y. Y. Zhun Zhong, Liang Zheng, "Random erasing data augmentation," 08 2017. 4

[19] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, 11 2017. 6