# Interpretable Deep Visual Place Recognition

Xiangwei Shi

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

# Interpretable Deep Visual Place Recognition

by

**Xiangwei Shi**

in partial fulfillment of the requirements for the degree of

**Master of Science**
in Computer Science

at the Delft University of Technology,
to be defended publicly on Friday August 31, 2018 at 16:00 PM.

| | | |
|---|---|---|
| Student number: | 4614909 | |
| Supervisor: | Dr. J. C. van Gemert | |
| Thesis committee: | Prof. dr. ir. M. J. T. Reinders, | TU Delft, chair |
| | Dr. J. C. van Gemert, | TU Delft, supervisor |
| | Dr. S. Khademi, | TU Delft, daily supervisor |
| | Prof. dr. ing. C. M. Hein, | TU Delft |

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft Delft University of Technology

# Preface

This MSc thesis records the exploratory results of my MSc graduation project on the topic of interpreting visual place recognition neural networks. The main content of this report is the scientific paper. This paper proposes a framework to interpret visual place recognition and presents several methods on validating the performance. The rest of this thesis is supplemental document that contains an introduction of motivation and research questions, a brief review of preliminary scientific background and supplementary results of experiments.

This work could not be possible without valuable contribution of several people. First of all, I would like to thank my supervisor Dr. Jan van Gemert who is kind and supportive. He consistently provides guidance and keeps my project in a right direction. Furthermore, I am very grateful to have Dr. Seyran Khademi as my daily supervisor. Her daily supervision and support are significant and appreciated. I also would like to acknowledge Prof. dr. ir. Marcel Reinders as the chair of my graduation committee and Prof. dr. ing. Carola Hein as a member of my graduation committee. Besides, I also want to show the appreciation to my friends and people who helped and supported me during last two years of my master period. Last but not least, the support from my family has always been appreciated.

*Xiangwei Shi*
*Delft, August 2018*

# Interpretable Deep Visual Place Recognition

Xiangwei Shi
Student
Delft University of Technology
X.Shi-3@student.tudelft.nl

Seyran Khademi
Supervisor
Delft University of Technology
S.Khademi@tudelft.nl

Jan van Gemert
Supervisor
Delft University of Technology
J.C.vanGemert@tudelft.nl

## Abstract

*We propose a framework to interpret deep convolutional models for visual place classification. Given a deep place classification model, our proposed method produces visual explanations and saliency maps that reveal the understanding of images by the model. To evaluate the interpretability, t-SNE algorithm is used for mapping and visualization of these latent visual explanations. Moreover, we use pretrained semantic segmentation networks to label all objects appearing in the visual explanations for our discriminative models. This work has two main goals. The first one is to investigate the consistency of visual explanations by different models. The second goal is to investigate whether visual explanations are meaningful and interpretable or not in an unsupervised manner. We find that varying the CNN architecture leads to variations in the discriminative visual explanations, but these visual explanations are interpretable.*

## 1. Introduction

Deep neural networks, including convolutional neural networks (CNNs) have already achieved great performance and breakthroughs in many computer vision fields, such as image classification [10, 11, 25], object detection [5, 6, 9, 18, 19], semantic segmentation [2, 8, 12, 14], place recognition [1, 27, 29, 30] and so on. Deep learning models are considered as black boxes, and human observers cannot know the reason why decisions are made or why models fail because of the lack of intepretability.

To interpret the deep convolutional models, [23] proposed a visualization technique, gradient-weighted class activation mapping (Grad-CAM), to generate visual explanations based on [35], which is called class activation mapping (CAM) and identifies the discriminative regions that expose the implicit attention of CNN models. Both methods project back the weights of specific class of the output layer on to the convolutional feature maps. Their methods succeed in interpreting most deep learning models for, such as, image classification, image caption and visual question answering. The visual explanations generated by their methods are always with labels, which makes their methods for interpreting CNN models in a supervised manner. For example, given a CNN model for classifying the images of cars and houses, the labels (car & house) for images should remain the same as discriminative visual explanations. On the contrary, how to interpret the other deep learning models, of which the discriminative visual explanations do not share the same labels, has not been solved. The deep convolutional model for visual place recognition belong to the latter models.

Visual place recognition is a well-defined but still challenging problem [15], which is to recognize the place where the query image was taken or return another image from database of which the location is closed to the query image. This kind of task can be accomplished by image classification [29, 30], image matching [27] and image retrieval [1]. Taking image classification as instance, there is a deep visual place recognition model that can classify the images of Tokyo and Pittsburgh. If the visual explanations generated by interpreting methods are meaningful, they are supposed to be buildings, pathway, vegetation or characters in signs, of which the labels are no longer the ones of images (Tokyo & Pittsburgh).

Visual place recognition is not only useful to find out where the image was taken, but also helpful for humanity studies, architecture and even biology. The latent information in images reveals not only the locations. Human recognize places from images by find information and objects

that are associating with location, such as signs with specific characters, famous spots, historic sites, regional animals and vegetation and so on. For urban images, buildings contain much useful information of location, such as the architectural style. Doersch et al. [4] attempt to find out geo-informative elements of Paris and other cities in Europe. To find the architectural stylistic patterns from visual place recognition may be useful to architects and urban historians.

To interpret the CNN models for visual place recognition, we firstly train place classification models that are able to classify images from Tokyo and Pittsburgh. Next we use the visualization technique in [23] to generate the visual explanations and descriptors for each image by forwarding the images through deep visual place recognition model. The pipeline of generating visual explantaions is shown in Fig 1. At last, we apply three different methods to test the interpretability of place classification model. (1) We apply t-distributed stochastic neighbor embedding technique (t-SNE) [16] to cluster the descriptors of visual explanations in 2-dimensional space. By replacing 2-dimensional points with visual explanations, we can visually test the interpretability of deep visual place recognition models. (2) We apply pre-trained semantic segmentation to label all objects appearing in the visual explanations, and the distributions of all objects reflects the visual explanations, which can be used for interpreting visual explanations. (3) We manually annotate one or two discriminative objects in visual explanations.

There are two main aspects of contributions in our work, one of which is that we try to interpret several deep visual place recognition models and investigate the consistency of visual explanations learned by different models. The other one is that we investigate the interpretability of visual explanations in an unsupervised manner.

This paper will be structured as follows. We present related work in Sec.2. And we explain our proposed framework on how to interpret the deep visual place recognition models in Sec.3. The details of our experiments and discussion is presented in Sec.4 and two research questions of 'consistency' and 'interpretability' are also answered in this section. The conclusion of our work will be presented in Sec.5. Appendix presents visualization results in the last part of this paper.

## 2. Related Work

Our work is based on the related work of visualization of CNN models, visual place recognition, nonlinear dimensionality reduction technique and interpretability assessment.

### 2.1. Visualization of CNN models

There are many previous works focusing on how to interpret deep CNNs, one of which is to visualize what convolutional filters have learned [23, 24, 26, 33, 34, 35, 38].

Grün et al. [7] provide a taxonomy to classifiy and compare the methods for visualizing learned features in CNNs. The first kind of generalized methods is input modification methods. Zeiler et al. [33] and Zhou et al. [34] cover a portion of the input images and detect the importance of features in the input space. Grey squares are used to occlude different portions of the input images in [33]. Zhou et al. [34] replace grey squares with random colored squares. Although these methods can find out where is important in input images and what have learned by convolutional filters, they are not straightforward regarding finding importance and locality.

The second kind of visualization methods is to use the network structure itself. Zeiler et al. [33] propose Deconvolutional Network (deconvnet) to pass activities of feature maps back to the input images. [24] uses class-specific scores through backpropagation to generate saliency maps. And then Springenberg et al. [26] present guided backpropagation by combining both deconvnet and backpropagation methods and learn what the intermediate and lower layers learn. Clearly, this second kind of methods is straightforward on finding what the convolutional layers learn and able to generate fine-grained visualization. However, some of these methods [26, 33] are not class-discriminative and generate similar maps for different classes. And the other methods [24] visualize CNN model overall instead of visualizing input images.

To visualize the model for input images and make it class-discriminative, Zhou et al. [35] present class activation mapping (CAM) to visualize class-discriminative features on input images. Based on the work of Zhou et al. [35], Selvaraju et al. [23] present gradient-weighted class activation mapping (Grad-CAM) to visualize any activation of feature maps in the last layer by using gradients generated. Grad-CAM in [23] is model-agnostic and generates visual explanations for each image learned by CNN models. One limitation of Grad-CAM is that it cannot visualize the feature maps of intermediate convolutional layers. However, in our work, we use Grad-CAM to generate visual explanations for deep visual place recognition models because it produces class-discriminative visualization and does not need to alter CNN models.

### 2.2. Visual place recognition

Visual place recognition is one of challenging topics in computer vision and robotics communities, the task of which is to recognize the place or location of a given query image accurately and efficiently. Recently convolutional models have already achieved state-of-the-art performance

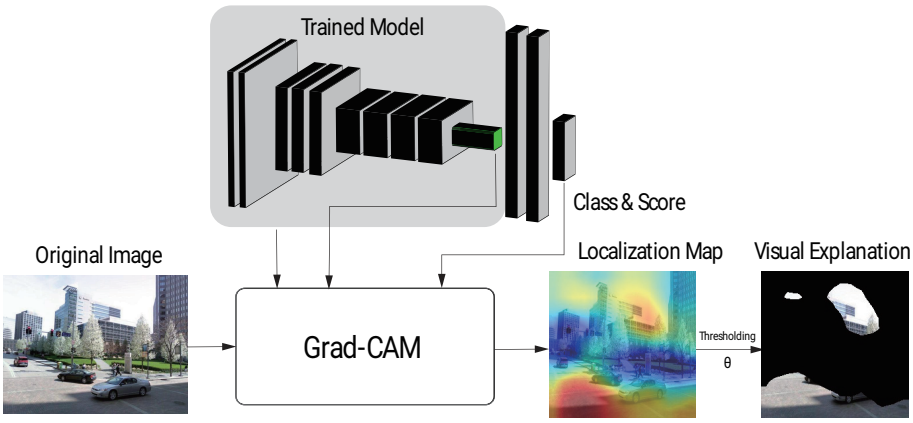


Figure 1. Pipe-line of generating visual explanations with Grad-CAM

on visual place recognition task. Weyand et al. [30] and Vo et al. [29] accomplish place recognition task by image classification, where geo-tagged images are classified into multi-scale geographic cells of the earth's surface, which are partitioned by S2 cells [30] and by setting thresholds on the number of images in a cell and physical area [29]. Tian et al. [27] match the buildings detected by CNN models from street view images and bird's eye view images. Arandjelovic et al. [1] provide a new CNN architecture in an end-to-end manner for place recognition, which return an image from database of which the location is close to the one of the query image.

Although image classification, image matching and image retrieval accomplish the visual place recognition task through different methods with CNNs, little work has investigated to find interpretable visual explanations for visual place recognition CNN models. Doersch et al. [4] use machine learning methods to find local interpretable and geo-informative features for Paris and other cities. However, the method in [4] cannot be used to interpret deep visual place recognition CNN models.

### 2.3. Nonlinear dimensionality reduction technique

Compared with traditional linear dimensionality reduction techniques, such as principal components analysis (PCA), nonlinear techniques have the ability to deal with complex nonlinear data, in particular for real-world data [28]. Clearly, the limitation of linear dimensionality reduction techniques like PCA cannot capture nonlinear relationships defined by higher than the second order statistics [32]. Kernel PCA is an extension to nonlinear PCA [22], which applies nonlinear mapping and kernel functions on PCA. One weakness of kernel PCA is how to select a proper kernel. Roweis et al. [21] propose locally linear embedding (LLE) that reconstructs each data point from its neighbors in high dimensional space. One major disadvantage of LLE is it is less accurate in preserving global structure because it attempts to preserve local properties of data. T-distributed stochastic neighbor embedding (t-SNE), as a clustering technique, clusters high-dimensional data by preserving nearest-neighbors in low dimensional space. T-SNE is also a nonlinear dimensionality reduction technique for high-dimensional data, which lie on several different, but related, low-dimensional manifolds in particular [16]. Therefore, t-SNE is suitable for visually exhibiting images of objects from different classes and taken from multiple viewpoints, such as visual explanations for visual place recognition.

### 2.4. Interpretability assessment

How to interpret intelligent models has always been a problem, which concerns whether and when the users trust the models[13, 20]. Lipton identifies the notions of interpretability for machine learning models in [13]. How to assess the trust in any classification model is introduced in [20]. Motivated by [13, 20], our framework is evaluated by human evaluation eventually like [34], where human evaluation determines whether the individual units behave as object detector in a scenes classification network.

The most relevant work to ours is the network dissection approach presented by Bau et al. in [3]. To evaluate the interpretability, Bau et al. apply binary segmentation task to every visual concept from the dataset and then compare the labels from segmentation with human annotation. However, the labels of visual concepts are provided and compared with the ones interpreted by network dissection, which is in a supervised manner.

To sum up, we introduce a framework to interpret deep visual place classification models based on CNNs. This framework generates class-discriminative visual explanations first for any CNN models. Because of the lack of labels for visual explanations, we apply t-SNE to cluster and map the visual explanations, which is also for investigating whether the visual explanations are meaningful or not. At

last, we validate the interpretability of visual explanations with semantic segmentation and manual annotations.

# 3. Methodology

## 3.1. Research questions

The main goal of this study is to interpret deep visual place recognition models based on CNNs, which is a black box and difficult to understand the insides. To this aim, we study the following research questions:

**Question 1: Are the features learned by different visual place recognition models consistent?** Visual place recognition task recently is accomplished by CNNs. The convolutional models in most of deep visual place recognition models are model-agnostic. To test the consistency of features learned by different visual place recognition models is an easy attempt to interpret place recognition models. We train four different place classification models in advance, which are VGG11 [25], ResNet18 [10] and two shallow networks built by ourselves.

**Question 2: Are the features learned by visual place recognition models interpretable?** Interpreting deep visual place recognition models is different from interpreting CNNs for other tasks, such as image classification. Because of the lack of labels of visual explanations, how to interpret visual place recognition models is the major problem through our work.

To answer these questions, we generate visual explanations from four different CNN models that classify images from two places. We compare these visual explanations for each image. After, we generate descriptors for all images, which are high-dimensional. We apply t-SNE on these descriptors to visualize the visual explanations. To validate the maps after t-SNE, we use semantic segmentation and human annotation to label the visual explanations.

## 3.2. Generating the Visual Explanations

The most common method for interpreting convolutional models is to visulaize what features the convolutional layers have learned. To interpret visual place recognition models, we apply Grad-CAM proposed in [23] to generate class-discriminative visual explanations. Grad-CAM utilizes the gradient information that flows into the last convolutional layer of the CNN models and generates the class-discriminative localization maps. These class-discriminative localization maps are obtained by a linear combination of weighted forward activation maps:

$$L^c_{\text{gc}} = ReLU(\sum_k \alpha^c_k A^k). \tag{1}$$

The class-specific weights $\alpha^c_k$ are calculated by the scores $y^c$ for class $c$ and feature maps $A^k$ ($k$ represents the $k$-th feature map):

$$\alpha^c_k = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A^k_{ij}}, \tag{2}$$

where $Z$ is the size of filters of convolutional layers. Both equations are proposed in [23]. These localization maps reveal class-discriminative significance for each image. To visualize the significance, the localization maps are converted into saliency maps. Afterwards, class-discriminative visual explanations for each image can be generated from original image by thresholding localization maps with value as $\theta$. One can see the pipeline in Fig 1.

Grad-CAM is able to generate visual explanations from CNN-based models, which makes it model-agnostic. Therefore, we apply Grad-CAM firstly to generate class-discriminative localization maps and visual explanations from four different visual place recognition CNN models, VGG11, ResNet18 and two shallow models, for each image.

## 3.3. Extracting Descriptors

Our goal is to interpret what the visual explanations are after getting localization maps and visual explanations for each image, but there are several problems in interpreting visual explanations. First of all, we do not have the true labels of objects appearing in visual explanations, which are not the same as the labels of images. Next we need to extract descriptors of visual explanations. Because of the irregular shape of visual explanations and black areas around visual explanations in original images, it is difficult to extract descriptors only from the visual explanations.

Instead of extracting descriptors only from the visual explanations of single image, we extract descriptors from the whole image by passing single image through convolutional layers of our place classification models. To emphasize the visual explanation of each image, we propose two methods to extract descriptors.

- We use the localization maps $L^c_{\text{gc}}$ as descriptors of images, where
$$D_{cam} = L^c_{\text{gc}}. \tag{3}$$
These localization maps have the same size of the feature maps after the last convolutional layer. At last, the localization maps are reshaped into high-dimensional vectors.

- We take the sum of the products obtained by multiplying the localization maps $L^c_{\text{gc}}$ and feature maps $A^k$ element-wise, as in 4,
$$D_{short} = \sum_k (L^c_{\text{gc}} \circ A^k). \tag{4}$$
As above, we reshape the descriptors $D$ into high-dimensional vectors.

### 3.4. Clustering

We lack the labels of visual explanations, therefore, we apply clustering method to find whether the visual explanations are interpretable and meaningful or not. T-distributed stochastic neighbor embedding [16] is not only a clustering method, but also a dimensionality reduction method that is able to present clustering results in two or three-dimensional space.

According to results, applying t-SNE on the high-dimensional descriptors is not a good option. Therefore, we reduce the dimensionality of the descriptors by principal component analysis (PCA) before t-SNE. We extract dimensionality of descriptors in two situations. The first one is that we extract the first 50 dominating features. The other situation is that the sum of the variance of the features that we extract is greater than 90%. Table 1 shows the numbers of dimensionality of descriptors of four different models after two situations of PCA.

Table 1. Number of dimensionality of descriptors of four different models after two situations of PCA.

| | Original | PCA(50) | PCA(90%) |
|---|---|---|---|
| VGG11 | 196 | 50 | 17 |
| ResNet18 | 49 | 49 | 11 |
| Simple | 324 | 50 | 40 |
| Simpler | 1764 | 50 | 91 |

### 3.5. Evaluation Method

Through our work, there exists a problem, which is we do not have the labels of the objects appearing in visual explanations that we use to interpret the visual place recognition models. Although we cluster the descriptors with t-SNE, we still need other methods to evaluate the clustering results and to interpret the visual explanations. Because no work has been done for this situation, we try to evaluate the results in two different ways.

We first use pre-trained semantic segmentation [31, 36, 37] to get the labels of the objects and the distribution of all objects appearing in the visual explanations for each image. The labels of objects and distribution from semantic segmentation can be used not only for evaluating the results of t-SNE, but also to interpret the visual explanations. Besides, we manually annotate the labels of a portion of images. These annotations can be regarded as the ground truth and used to evaluate t-SNE and segmentation.

## 4. Experiments and Discussion

### 4.1. Datasets

We train four different place classification networks with two different datasets, which are **Pittsburgh** and **Tokyo 24/7** introduced in [1].

- **Tokyo 24/7** This dataset contains 76k database images and 315 query images. Each place in the query images is captured at different times of day, and for the same place in the database, 12 images from different directions are taken. We only use database images that are taken in daytime and divide them into training, validation and test datasets with the following proportions as 6:2:2.There are 15204 test images in total.
- **Pittsburgh** This dataset contains 250k database images from Google Street View and 24k query images from Google Street View at different times. To avoid unbalanced dataset, we only use 76k images from database images. For each place, 24 images are captured from 12 different directions and 2 different angles. Training, validation and test datasets are structured by the same proportions as Tokyo 24/7.

### 4.2. Experimental Setup

#### 4.2.1 Visual place recognition CNN models

We train four different CNN models to classify images taken from two different places, which are VGG11 [25] and ResNet18 [10] and the other two are shallow networks. The configurations of two shallow networks, Simple and Simpler, are shown below, in Table 2. In this table, 'convN×N' represents convolutional layer with a N×N filter, and each convolutional layer is followed by ReLU activation function. The number after hyphen represents the number of channels in the corresponding feature map, and the numbers in the brackets is the size of filter in max pooling layer. The size of input images (224×224×3) remains the same, which makes the visual explanations comparable among these four different models.

Table 2. Configurations of two shallow networks.

| Simple | Simpler |
|---|---|
| Input images:224×224×3(RGB) | |
| conv5×5-20 | conv9×9-20 |
| max pooling(2×2) | |
| conv7×7-64 | conv9×9-64 |
| max pooling(2×2) | |
| conv5×5-96 | conv9×9-96 |
| max pooling(2×2) | |
| conv7×7-128 | |
| max pooling(2×2) | |
| fully connected-4096 | |
| fully connected-100 | |
| fully connected-number of classes:2 | |

#### 4.2.2 Training details

These four models are trained with the same training images. The loss function is cross-entropy function, and Adam

optimizer is selected. The initial learning rate is set as 0.0001, and after ever 10 epochs learning rate is multiplied by 0.1. The $\beta$s of Adam optimizer are set as 0.9 and 0.999. And the parameter to improve numerical stability $\epsilon$ is $1\times10^{-8}$. The test error rates of four models are shown in Table 3 below.

Table 3. Test error rates of four models

| Name | VGG11 | ResNet18 | Simple | Simpler |
|---|---|---|---|---|
| Test error rate | 0.0002 | 0.0004 | 0.0069 | 0.0182 |

### 4.3. Performance analysis

#### 4.3.1 Experiment 1: Comparing Visual Explanations From Four Place Recognition Models

To investigate the consistency of visual explanations generated from different visual place recognition models, we have trained four different models, VGG11, ResNet18, Simple and Simpler. We use Grad-CAM to generate the localization maps as saliency maps. The saliency maps reflect the important regions in the images that are dominating place recognition models to classify the images. Fig 2 shows some examples that different visual place recognition models generate dissimilar localization maps.

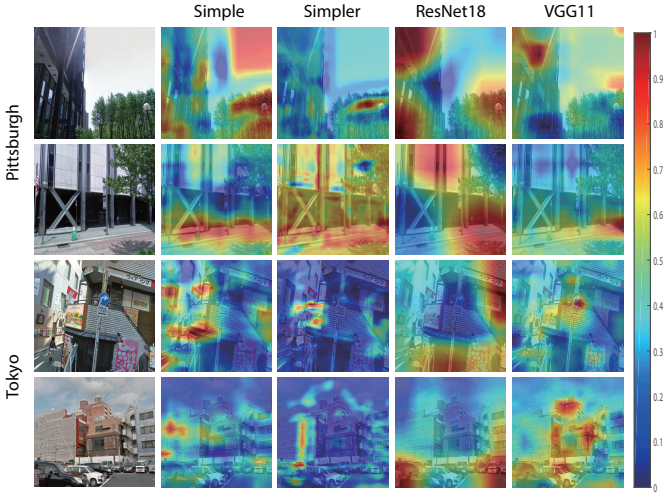


Figure 2. Different visual place recognition models generate dissimilar localization maps. The first two rows present two test images taken from Pittsburgh and the last two rows show the test images taken from Tokyo. From the second column to the fifth column, the class-discriminative localization maps (saliency maps) are shown for Simple, Simpler, ResNet18 and VGG11 place recognition models, respectively.

Although the examples in Fig 2 have already shown that four place recognition models classify the images based on different visual explanations, it is hard to conclude that there is no consistency of visual explanations for place recognition models. Therefore, we compare the localization maps for all images among four models. For each image and each model, we can get a localization map $L^c_{\text{gc}}$, which can be regarded as a vector. To compare the difference between any two models, we calculate the average residual (AR) for single image between two models:

$$AR = \frac{\left|L^c_{\text{gc},m_1} - L^c_{\text{gc},m_2}\right|}{H \times W}, \tag{5}$$

where H and W are the height and width of localization maps, and m represents place classification model. For single image, we can calculate 6 average residuals between any two models out of four. We compute all average residuals for all test images, and distributions of the average residuals between any two models are shown below in Fig 3. If two models learn the same features from the same image, the average residual will be 0, and if they learn totally different features from the same image, the average residual will be 1. Therefore, large value of the average residuals mean that the localization maps of the same image for different place recognition models are dissimilar to a large extent, and vice versa. From Fig 3, we can see that most of the average residuals range from 0.2 to 0.6, and the average residuals of most of images between any two place recognition models are located around 0.4 except the last one (Simple and Simpler, 0.3). Therefore, we can conclude that there is little consistency of visual explanations for different visual place recognition models.

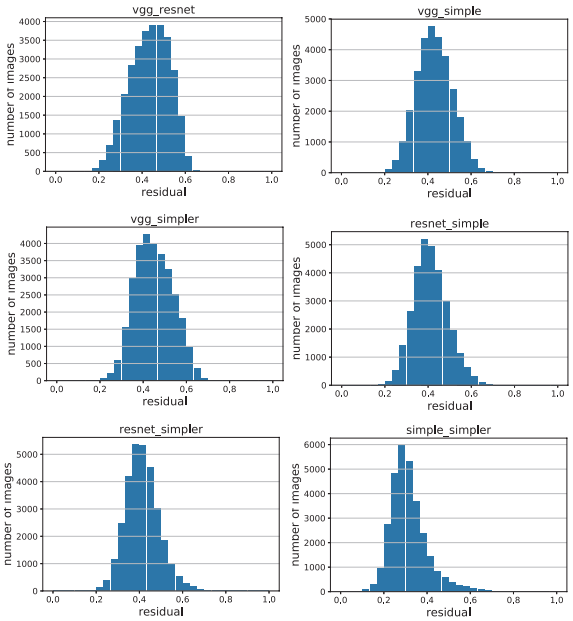


Figure 3. Distributions of average residuals between different models. The horizontal axis represents the average residuals between two place recognition models, and the vertical axis represents the number of images. The scale of average residuals is from 0 to 1.

#### 4.3.2 Experiment 2: Clustering Visual Explanations With t-SNE

Because of the lack of true labels of the objects appearing in visual explanations, we need to use unsupervised method to cluster the descriptors extracted from visual explanations. We firstly use principal component analysis (PCA) to reduce the dimensionality of the descriptors of all test images generated by Grad-CAM and visual place recognition models. Then we apply t-SNE as clustering method to cluster the descriptors.

Since there is no need to compare the visual explanations of different place recognition for investigating the interpretability, from experiment 2, we take VGG11 as example to show the results. Fig 4 shows the scatter plots of results after t-SNE clustering.

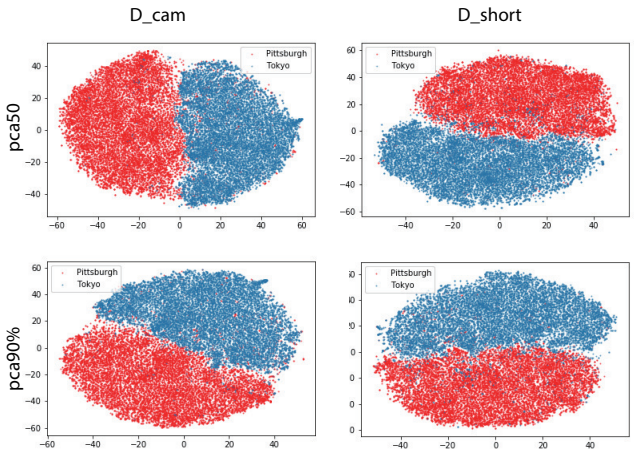


Figure 4. Scatter plots of t-SNE results for VGG11 model with different kinds of descriptors and different dimensionality of descriptors after PCA.

From Fig 4, it can be seen that most of the visual explanations of Pittsburgh images and Tokyo images are separable regardless of the methods of extracting descriptors of visual explanations, and the clustering results with different numbers of dimensionality after PCA are similar with respect to the same descriptor extraction method. However, the scatter plots cannot visually exhibit the visual explanations. Next we replace the points with visual explanations generated from original images by thresholding saliency maps by $\theta = 0.6$. Since the number of test images is considerable, it is impossible to visually exhibit all visual explanations. We randomly select around 500 visual explanations to exhibit with 50 dimensionalities of descriptors after PCA, as shown in Fig 5.

Based on the results shown in Fig 5, we can find that different kinds of descriptors lead to different t-SNE results and similar visual explanations gather together regardless of the methods of descriptors extraction. For $D_{cam}$, we can see that the visual explanations of wall are gathering at the top right corner, with the visual explanations of sky and vegetation gathering at the top left corner, the ones of pathways located in the bottom and the ones of buildings located in the middle of left image of Fig 5. On the other hand, for $D_{short}$, it can be seen that the visual explanations of wall are gathering in the middle of right region, with the explanations of vegetation located in the bottom, the ones of pathways gathering at the top, the ones of sky gathering in the middle of left region and the ones of buildings surrounded by pathways, sky, walls and vegetation.

Although we can interpret a portion of the visual explanations generated by visual place recognition models based on visually exhibiting clustering results of t-SNE, we still cannot answer the second research question without further evaluation or validation, because we lack the true labels of the visual explanations and analyzing the results of clustering is subjective.

#### 4.3.3 Experiment 3: Evaluating Results with Semantic Segmentation and Human Annotation

In this experiment, we apply semantic segmentation and human annotation to evaluate the clustering results after t-SNE and investigate the interpretability of visual explanations.

**Semantic segmentation**

Because of the lack of the labels of visual explanations, we need to get the labels of these visual explanations to investigate whether they are interpretable and meaningful or not. Besides the shapes of visual explanations are irregular and there are black areas around the visual explanations, it is difficult to label all the objects and find the most dominating one. Therefore, the first method to figure out what objects are in the visual explanations is using semantic segmentation. In this experiment, we utilize one semantic segmentation model trained on MIT ADE20K scene parsing dataset [31, 36, 37]. This semantic segmentation model is built on ResNet50 with pyramid pooling, bi-linear upsample and deep supervision trick and able to classify 150 different kinds of objects. Fig 6 shows an example of using segmentation model. The colors of segmentation result represent different labels of objects.

To avoid missing any information of visual explanations, we use segmentation model to classify the objects appearing in visual explanations and record the distribution of all objects by pixel instead of the only object with the most number of pixels. For different datasets, we average the number of pixels of 150 objects. Fig 7 shows the histograms of several objects appearing in visual explanations that are generated by thresholding with 2 different values ($\theta$) for Pittsburgh and Tokyo datasets, as shown in Fig 1, respectively. Only if the number of average pixels is larger than 100, the object will be selected.

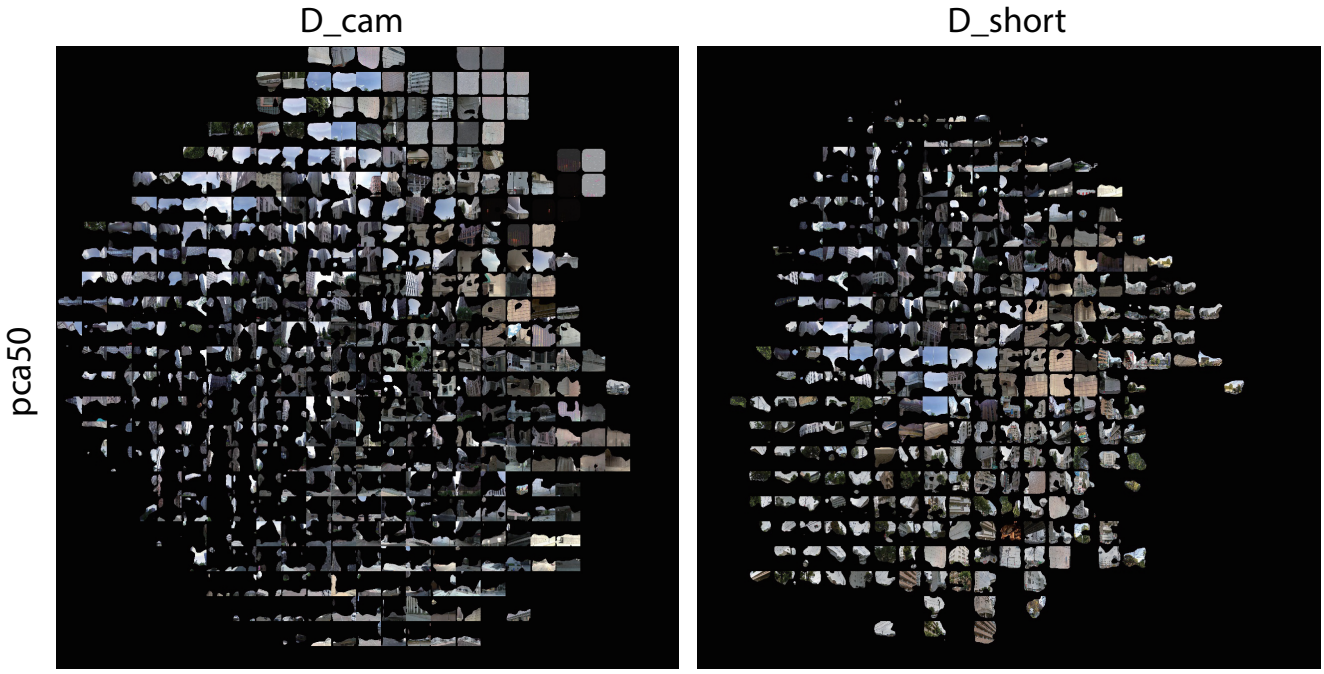


Figure 5. t-SNE results exhibiting visual explanations by VGG11. The descriptors are reduced to 50 dimensionalities with PCA, and the visual explanations are generated from original images by thresholding saliency maps by $\theta = 0.6$.

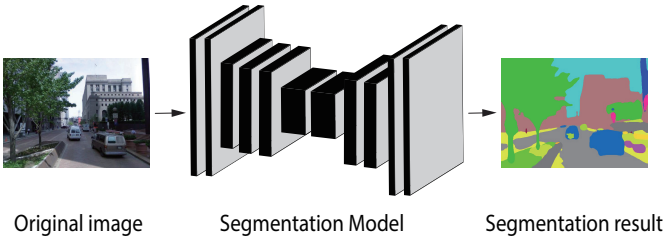


Figure 6. Example of using segmentation model.

From Fig 7 it can be seen that except building and sky, skyscraper, floor and earth (ground) are discriminative objects appearing in visual explanations of Pittsburgh. On the other hand, plant, fence and signboard are different discriminative objects for the visual explanations of Tokyo. Building and sky are the common discriminative attributes in both Pittsburgh and Tokyo datasets. We have shown that the frequent objects are appearing as the most discriminative elements to classify a place in our setting. This indicates that the visually discriminative clues for our VGG11 place classification model are consistent and meaningful. The results would have been difficult to interpret if the importance of different objects were equally distributed, which is not our case here.

To evaluate the results of t-SNE with segmentation, we should show the scatter plots with the distribution of visual explanations in each image. However, the distribution cannot be plotted. Instead, we replace the histogram with the top object, shown in Fig 8.

From Fig 8, it can be seen clearly that two different objects gather at different locations, hence it confirms the clustering results of t-SNE, which means the results of t-SNE can reflect the interpretability of objects in visual explanations according to the segmentation labels.

**Manual annotation.**

Although the result of semantic segmentation reveals the fact that the visual explanations are interpretable, we still need to evaluate the results of segmentation because the segmentation model that we used is not trained specially by our datasets. To get the labels of the objects appearing in the visual explanations, we manually annotate the visual explanations of 1000 images (500 of Pittsburgh, 500 of Tokyo). Instead of recognize all objects appearing in visual explanation, we annotate two dominating objects in a single image at most. Fig 9 shows examples of human annotation. For both datasets, we always select the main object appearing in the visual explanations. If there are two main objects appearing in the visual explanations with similar areas, we select these two objects. No more objects will be selected by human annotation.

Firstly, we use the manual annotation to evaluate the results of t-SNE, as in Fig 10. We can see that images labeled as building and images labeled as road gather at different areas, which is consistent with the scatter plots in Fig 8. In other words, t-SNE clusters the visual explanations and

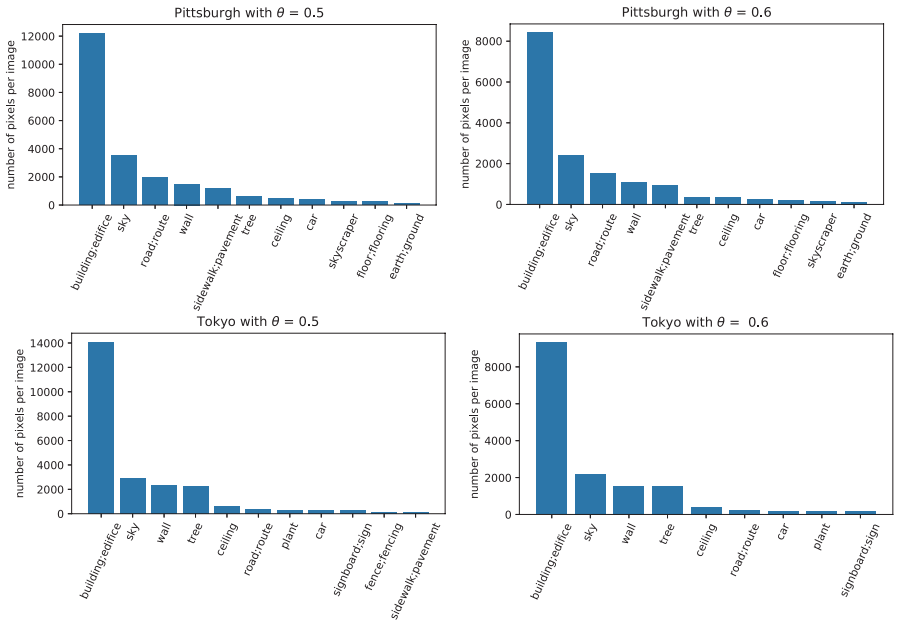


Figure 7. Histograms of top objects appearing in visual explanations of different datasets with different thresholds.

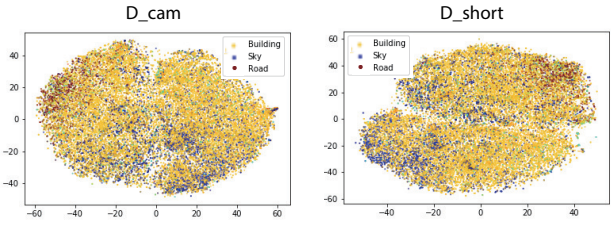


Figure 8. Scatter plots of t-SNE results with the top one segmentation label.


Figure 9. Example of human annotation.

the results reflect the dominating objects appearing in visual explanations.

Secondly, we compare the manual annotations with the labels of segmentation. Because of the same problem that we cannot show or compare the histograms of objects by segmentation with one or two dominating objects by human annotation. Therefore, we need to select one or two labels from the segmentation histograms for each image. For all images, we select the object with the largest number of pixels in one image. And we set another threshold to select the second object. If the ratio between the sum number of pixels of the top two objects and the number of total pixels of all objects appearing in the visual explanations is larger than this threshold for selecting a second object, then we also select the second object. And we compare the labels of visual explanations of 1000 images between segmentation labels and human annotation in two different situations, as in Fig 11 and 12.

The first situation is that the labels of segmentation are regarded as correct only if all the selected labels are the same as human annotations for every single image, as in Fig 11. And the second situation is that when one of two selected labels is consistent with human annotations, it can be considered as correct, as in Fig 12. Fig 11 shows that the accuracy of segmentation in different datasets with different thresholding values ($\theta$). The accuracy improves as threshold for selecting a second object rising. When the threshold is set as 100%, the accuracy of segmentation results is the largest, around 0.6. On the contrary, Fig 12 shows that the

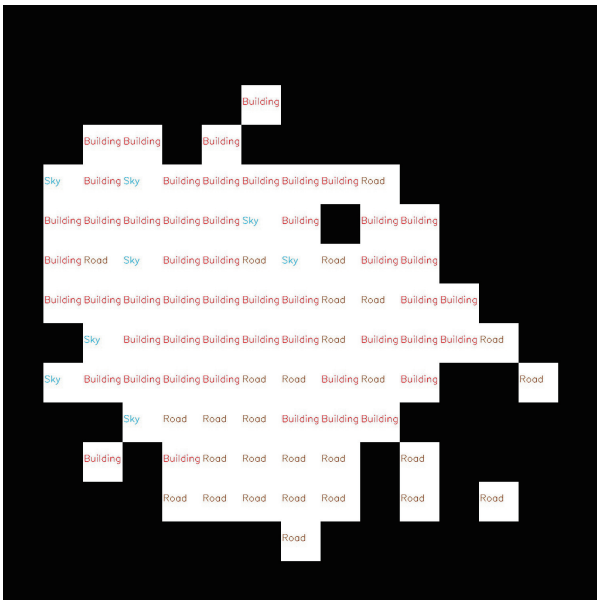


Figure 10. t-SNE results with manual annotation labels with $D_{cam}$.

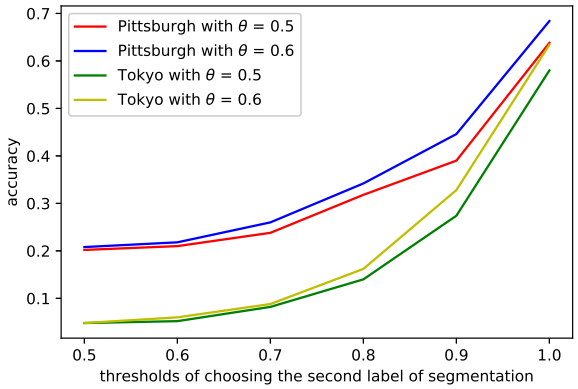


Figure 11. Comparison between segmentation and human annotation only when both selected labels are consistent with human annotations is correct. The x-axis represents the thresholds of selecting the second label of segmentation, and the y-axis represents the accuracy of segmentation with human annotation as ground truth.

accuracy decreases as the threshold for selecting a second object rising. We can also find that the accuracies of Pittsburgh images are larger than the ones of Tokyo images in both situations. One possible reason could be the categories of objects appearing in Tokyo images are more complicated than the ones appearing in Pittsburgh images. Instead of labelling objects as delicate classes, human annotate objects as generalized classes.

Comparing Fig 11 and Fig 12, we can easily find that the trends of lines are totally different. In the first situation, only if the second selected label of the object appearing in

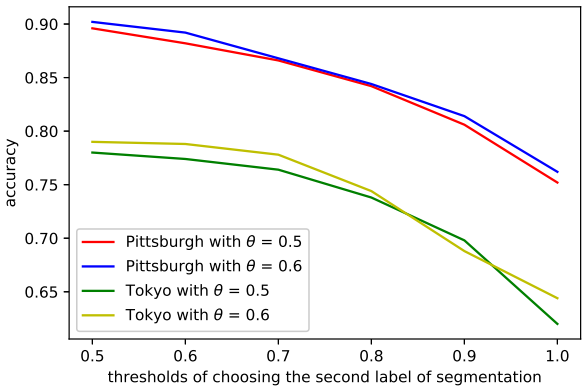


Figure 12. Comparison between segmentation and human annotation when one of two selected labels are consistent with human annotations is correct. The x-axis represents the thresholds of selecting the second label of segmentation, and the y-axis represents the accuracy of segmentation with human annotation as ground truth.

the visual explanations is consistent with human annotation, it can be considered as correct. When the threshold for selecting the second label is large, it is difficult to select a second label of object. The number of selected labels of objects are more likely to be one, and the selected objects are more likely to be consistent with human annotations. Therefore, in Fig 11 the accuracies go up. On the contrary, in the second situation, if one selected label of object is the same with human annotation, it will be considered as correct. When the threshold for selecting the second label is small, it is easy to select a second label of object, which provides more options to compare with human annotations. It is more likely to be consistent with human annotations. Therefore, in Fig 12 the accuracies go down.

## 4.4. Application

From the previous experiments, we find that different visual place recognition models recognize the same image based on different visual explanations. The visual explanations are meaningful and interpretable. Our presented framework also can be used to explore the urban symbols and architecture styles for different cities. Here we discuss the directions.

### 4.4.1 Exploring urban symbols

Our framework can used for exploring urban symbols. Urban symbolism is one of major research interests for sociologists and historians [17]. A lot of sociological and historical materials from all over the world are related to urban symbolism, such as street patterns, 'sign' language and street names. Our framework generates visual explanations

between different cities. By comparing visual explanations of different cities, unique discriminative objects can be regarded as urban symbols. Fig 13 shows the examples of urban symbols of Pittsburgh and Tokyo.

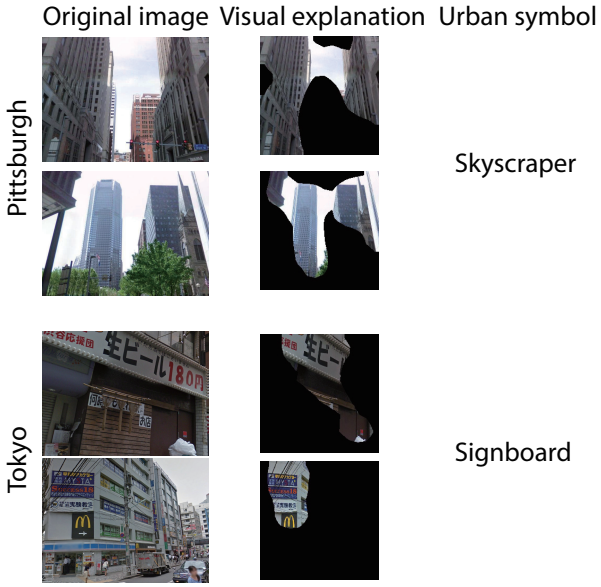


Figure 13. Urban symbols of Pittsburgh and Tokyo.

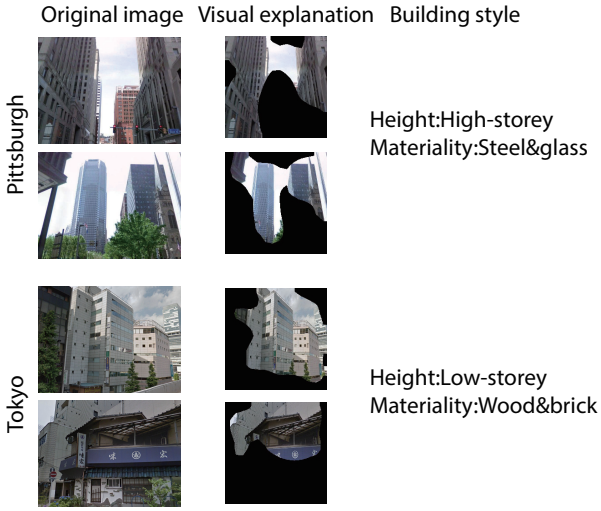


Figure 14. Architecture styles of Pittsburgh and Tokyo.

When we enlarge our datasets with more images of more places, urban symbols can be generalized for each city. Given the city location and global positioning system (GPS) coordinates of images, we can easily display the visual explanations with urban symbols on a map. Besides, historians can also use our framework help their researches. With images taken at the same city at different time, historians can find out urban symbols of the same city at different time.

#### 4.4.2 Exploring architecture styles

Except exploring urban symbols, our framework can also explore the architecture styles. For images taken in urban cities, buildings are common, and the semantic information is meaningful. By comparing the buildings from different cities, city-specific buildings can be found. Our framework generates a lot of visual explanations that are buildings. These buildings can be considered as city-specific buildings. Fig 14 compares the building styles of Tokyo and Pittsburgh. We can see that the buildings in Pittsburgh are always high-storey and mostly made of steel and glass. On the contrary, the buildings in Tokyo are low-storey and mostly made of wood and bricks.

This application is helpful for architecture researches. Except exploring architecture styles, architectural historians can use our framework to find out how the architecture styles of the same city along time.

## 5. Conclusion

In this work, we present a framework that interprets deep visual place recognition models. We investigate the consistency of visual explanations by comparing different visual place recognition models that classify images into different places. Because of the lack of true labels of objects appearing in the visual explanations, we apply t-SNE as clustering algorithm to cluster the visual explanations and use semantic segmentation and human annotation to evaluate the results and to investigate the interpretability of the visual explanations. We conclude that different visual place recognition models recognize the place by different visual explanations of the same image, however, these visual explanations are interpretable. Besides, we also discuss the potential usage of our framework, which is helpful for architecture researches and architectural historians.

## References


[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[3] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. *arXiv preprint arXiv:1704.05796*, 2017.

[4] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012.

[5] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[7] F. Grün, C. Rupprecht, N. Navab, and F. Tombari. A taxonomy and library for visualizing learned features in convolutional neural networks. *arXiv preprint arXiv:1606.07757*, 2016.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[13] Z. C. Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[14] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[15] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[16] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[17] P. Nas. *Urban symbolism*, volume 8. Brill, 1993.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[19] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[20] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[21] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[22] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[23] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via Gradient-Based Localization. In *ICCV*, pages 618–626, 2017.

[24] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[27] Y. Tian, C. Chen, and M. Shah. Cross-view image matching for geo-localization in urban environments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1998–2006, 2017.

[28] L. Van Der Maaten, E. Postma, and J. Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71, 2009.

[29] N. Vo, N. Jacobs, and J. Hays. Revisiting IM2GPS in the deep learning era. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2640–2649. IEEE, 2017.

[30] T. Weyand, I. Kostrikov, and J. Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016.

[31] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified Perceptual Parsing for Scene Understanding. *arXiv preprint*, 2018.

[32] H. Yin. Nonlinear dimensionality reduction and data visualization: a review. *International Journal of Automation and Computing*, 4(3):294–303, 2007.

[33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[34] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

[35] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[36] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016.

[37] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene Parsing through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[38] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

# Contents

# 1 Introduction

This work aims to interpret the deep visual place recognition models based on convolutional neural networks. We present a framework to investigate the consistency of visual explanations that are important for decisions for different models and whether these visual explanations are interpretable. This chapter provides an overview of background and motivation of this work. Also, the research questions of this work are presented in this chapter.

## 1.1. Background

Deep neural networks, especially convolutional neural networks (CNNs) have already become one powerful tool in many computer vision fields due to great performance and breakthroughs, such as image classification [1–3], object detection [4–8], semantic segmentation [9–12], place recognition [13–16] and so on. However, there is one major problem with deep neural networks, which is that deep neural networks are considered as black boxes. People who use deep convolutional neural networks will not know the reason why decisions are made or why models fail because of the lack of interpretability, which will result in that people doubt and distrust deep neural networks. Therefore, interpreting deep neural networks is significant.

Because of the importance of interpreting deep neural networks, convolutional neural networks in particular, many researches have already attempted to interpret them. To interpret deep neural networks, first of all, we should know what is interpretabililty. Lipton identifies the notions of interpretability for machine learning models in [17]. Ribeiro et al. introduce several methods on how to assess the trust in any classification model in [18]. Next, we should know how to interpret deep convolutional neural networks. One common method is to visualize what convolutional filters have learned. Grün et al. [19] provide a taxonomy to classifiy and compare the methods for visualizing learned features in CNNs. The first kind of methods is to modify the input images. Zeiler et al. [20] and Zhou et al. [21] occlude the input images and detect the importance and locality of features in the input space. The difference between these two methods is that Zeiler et al. use mono colored grey squares, while Zhou et al. use random squares and generate discrepancy map for each image. Although these methods can find out where is important in input images and what have learned by convolutional filters, they are not straightforward regarding finding importance and locality. The second kind of methods in [19] is to use the network structure itself to visualize. Zeiler et al. is the first to present this kind of method in [20], where Deconvolutional Network (deconvnet) is proposed to map activities from feature maps back to the input pixel space through convolutional neural networks. Simonyan et al. provide a variation of deconvnet in [22], which uses class score derivative through backpropagation to generate saliency maps. Springenberg et al. [23] present guided backpropagation by combining both deconvnet and backpropagation methods and get insight into the intermediate and lower learned layers. Compared with the first kind of methods, the second kind of methods is straightforward on finding what the convolutional layers learn and able to generate fine-grained visualization. There are several other methods that are able to generate class-discriminative visual explanations for individual input image. Zhou et al. present Class Activation Mapping (CAM) to visualize class-discriminative features on input images in

[24]. [25] proposes gradient-weighted class activation mapping (Grad-CAM) to visualize any activation in the last layer by using gradients generated when input images flow into the last convolutional layer of CNNs. Grad-CAM in [25] is workable for any CNN-based models and generates visual explanations for each image learned by CNN models.

Those methods mentioned above are tested and broadly used for interpreting image classification, object detection and semantic segmentation. Little work has done on interpreting convolutional neural networks for visual place recognition. Visual place recognition is one of challenging topics in computer vision and robotics communities. The task of visual place recognition can be summarized as to recognize the place where the query image was taken or to return another image from given database, the location of which is closed to the query image. Recently, this task can be accomplished by convolutional neural networks based models, such image classification models [15, 16], image matching models [14] and image retrieval models [13].

## 1.2. Motivation

Many efforts have been made to interpret CNNs models, such as image classification, object detection and semantic segmentation. Doersch et al. [26] use machine learning methods to find local interpretable and geo-informative features for Paris and other cities. Although this work aims to interpret place recognition model, the model is based on machine learning instead of deep neural networks. However, it is difficult to those methods mentioned above to interpret visual place recognition models directly. One major problem is we lack the true labels of visual explanations for each image. Taking image classification as instance, there is a deep visual place recognition model that can classify the images of Tokyo and Amsterdam. If the visual explanations generated by interpreting methods are meaningful, they are supposed to be buildings, pathway, vegetation or characters in signs, of which the labels are no longer the ones of images (Tokyo & Amsterdam). In this scenario, interpreting visual place recognition models should be in unsupervised manner.

To solve this unsupervised problem, the first solution should be clustering method. Since the visual explanations are parts of the images that are high-dimensional, there is another problem to cluster them. High-dimensional data is always sparse, and most clustering methods are based on the distance between each other. Because of the sparse high-dimensional data, the distances are very similar. How to cluster the visual explanations is the second problem. Therefore, interpreting deep visual place recognition models is challenging and useful.

This work aims to interpret deep visual place recognition models based on convolutional neural networks, and a framework is proposed. The pipeline of our framework is shown in Fig 1.1.

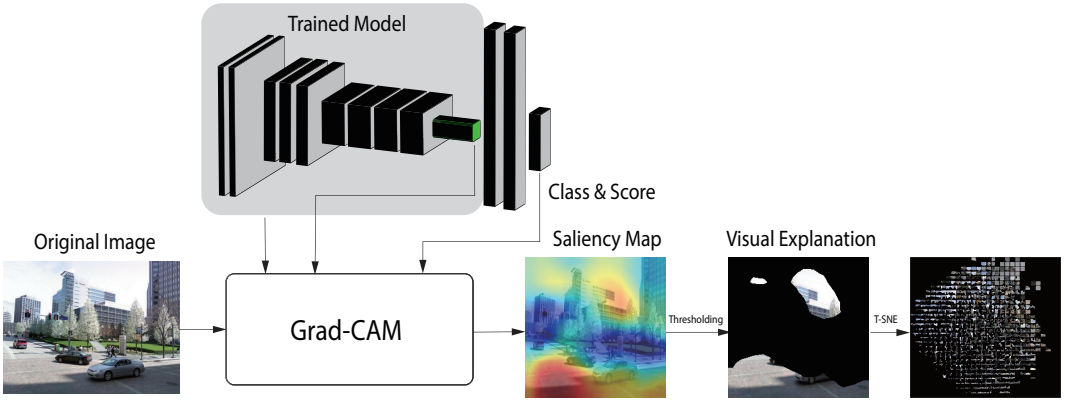


Figure 1.1: Pipe-line of visualizing place recognition models with Grad-CAM

## 1.3. Research questions

The main goal of this study is to interpret deep visual place recognition models based on CNNs, which is a black box and difficult to understand the insides. To this aim, the following research questions are proposed,

- Are the features learned by different visual place recognition models consistent?
  Because most of the deep visual place recognition models based on convolutional neural networks are model-agnostic, which means convolutional layers can be replaced, is there consistency of features learned by different convolutional models? It is an easy attempt to interpret visual place recognition models.
- Are the features learned by visual place recognition models interpretable?
  Because of the lack of labels of visual explanations, how to interpret visual place recognition models is the major problem through our work.

In summary, this work has two main aspects of contributions, one of which is that we try to interpret several deep visual place recognition models and investigate the consistency of visual explanations learned by different convolutional models.The other one is that we investigate the interpretability of visual explanations for all images in an unsupervised manner.

## 1.4. Structure

This chapter presents the introduction of this work. And the rest of this thesis is organized as follows. The second chapter presents preliminary scientific background of theoretical knowledge of deep neural networks and algorithms. The third chapter proposes supplementary results of experiments.

# 2

# Preliminary Background

This chapter presents the preliminary scientific background which is necessary for getting insight and deep understanding of this work. The purpose of this work is to interpret deep visual place recognition neural networks models. To provide the preliminary knowledge, this chapter presents the basic idea of image processing and a brief introduction of neural networks and convolutional neural networks.

## 2.1. Basic idea of image processing

Before we use images to accomplish complex tasks, such as images classification and image matching, we need to use image processing methods on images in advance. One major aspect of image processing is image filtering.

### 2.1.1. Convolution

Before introducing image filtering, it is necessary to know what convolution is. Convolution is a mathematical operation that computes the amount of overlap of one function as it is shifted overanother function. For continuous functions, the convolution of two functions can be written in 2.1. And for discrete functions, it can be written in 2.2. In computer vision, we consider images as matrices of integer values, which are discrete.

$$f * g = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau \tag{2.1}$$

$$f * g = \sum_{-\infty}^{\infty} f(m)g(n-m) \tag{2.2}$$

### 2.1.2. Image filtering and filters

The main purposes of image filtering are reducing noise of images and extracting useful features and knowledge from images, which can achieved by taking convolutions between images and filters, as in 2.3.

$$G(i,j) = \sum_{u=-k}^{k}\sum_{v=-k}^{k} H(u,v)F(i-u,j-v), \tag{2.3}$$

where $F$ represents filter and $H$ represents image. For example, Fig 2.1 shows an image with noisy points. The simplest method to remove the noisy points is to replace the noisy pixels by neighbourhood average. Therefore, we can use an averaging filter over the image. Different kinds of filters are used for different purposes, such as blurring, sharpening, edge detection and so on. Fig 2.2 shows an example of image filtering. Among all kinds of filter, Gaussian filter is commonly used, which is because Gaussian is the only function that does not introduce artifacts [27].

Figure 2.1: Image with noisy points

Figure 2.2: An example of image filtering. The left image is original image. The right image is generated after convolving edge detecting filter with original image.

## 2.2. Introduction of convolutional neural networks

### 2.2.1. Artificial neural networks

Artificial neural networks (ANNs) were created based on the inspiration of biological neural networks in the brain in [28], like Fig 2.3. ANNs consists of a collection of simple interconnected computational units called neurons, one major feature of which is that they are adaptive and solving problems by learning from examples [29]. Fig 2.4 presents the structure of a pair of connected neurons. From Fig 2.4, it can be seen that neurons are connected with previous neurons and the connections between two neurons associate with different weights. Besides, for one single neuron, it first computes the weighted sum of all inputs and takes a threshold over the weighted sum. And then the output of one neuron can be calculated after applying a non-linear activation function and passed to next neuron. The standard structure of feed forward ANNs consists of one input layer, one or many hidden layers

and one output layer, which only allow signals to travel from input to output. Each layer consists of one or many neurons.

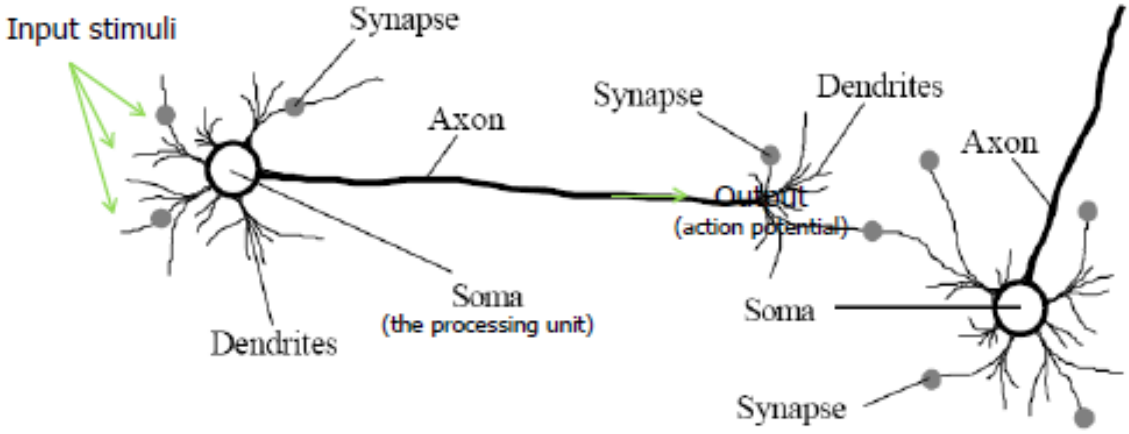


Figure 2.3: Biological neural networks in the brain

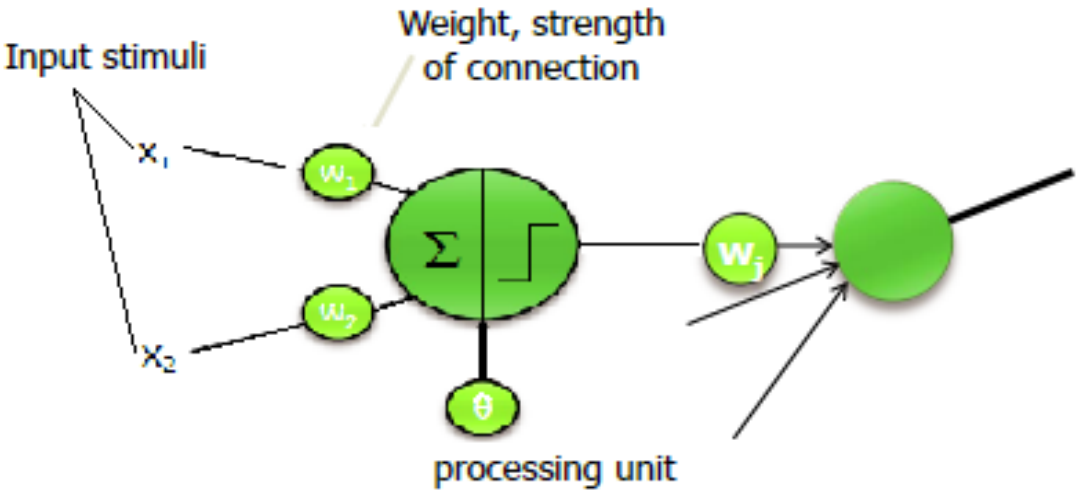


Figure 2.4: Structure of interconnected neurons

The activation function has the biggest impact on behaviour and performance of the ANN [29]. The major task of activation function is to map the outputs of neurons to a bounded interval such as $[0, \infty)$. One commonly used and successful activation function is rectified linear unit (ReLU), which is defined as the positive part of the argument, as 2.4.

$$f(x) = \max(0, x) \tag{2.4}$$

ANNs solve problems by learning from examples. In the training process, the weights associating two neurons are updated so as to minimize the error between predictions by ANNs and true labels of

training data. The error is defined as loss function,

$$E = \sum_{n=1}^{N} G(y_p - y_n), \tag{2.5}$$

where $y_p$ represents the prediction of input data $X_n$, $y_n$ represents the true label of input data $X_n$ and $G$ is an operator.

To minimize the total error or the loss, we can use optimization method. The gradients are along the directions of steepest descent. During the training process, the weights are updated after each epoch by:

$$w_i = w_i - \alpha \nabla E(w_i), i = 1, 2, 3, \ldots, n \tag{2.6}$$

$$\nabla E(w_i) = \frac{\partial E}{\partial w_i}, \tag{2.7}$$

where $\alpha$ is the learning rate that influences the speed and quality of learning.

To update the weights and minimize the error, one classical algorithm is backpropagation. Backpropagation can be summarized as two phases.

- **Forward propagation:** (a) propagating forward through the whole network to generate the output (b) computing the loss (c) propagating the output activations back through the network to genernate the difference between the targeted and actual outputs of all output and hidden neurons
- **Backward propagation:** (a) multiplying the difference and input activation of each output and hidden neurons to find the gradient of the weight (b) updating the weight by 2.6

The input of regular neural networks are not supporting high-dimensional data, such as image data. One problem of using regular neural networks for image data is that it will neglect the spatial information hidden in the image. Another problem is curse of dimensionality. Therefore, there is another kind of neural networks that is suitable for image inputs.

### 2.2.2. Convolutional neural networks

Most of convolutional neural networks consist of three different types of layers, which are convolutional layers, pooling layers and fully connected layers. Fig 2.5 shows an example of convolutional neural networks.

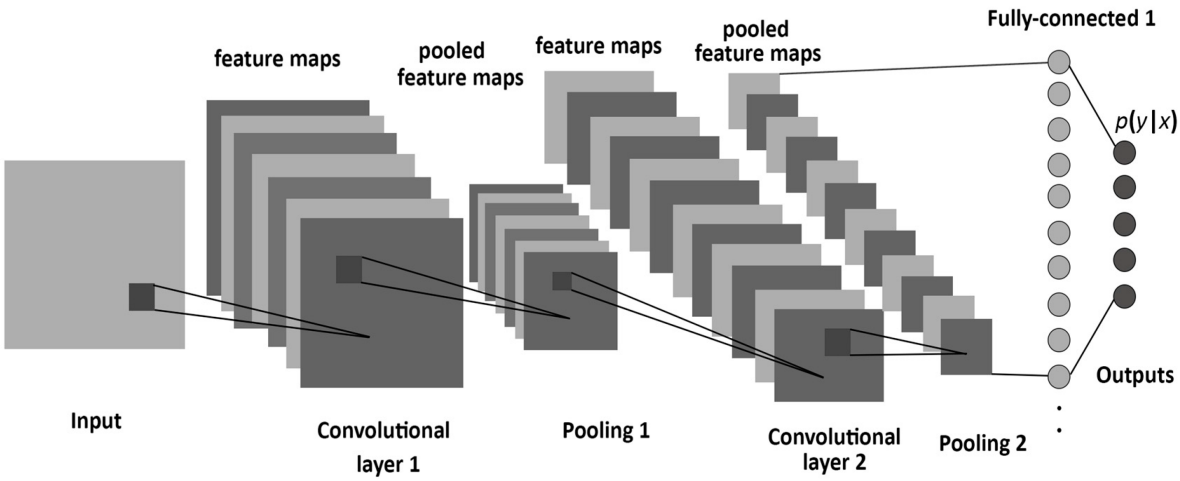


Figure 2.5: Example of convolutional neural networks [30].

### Convolutional layer

Convolutional layer is the most important part of convolutional neural networks. The idea of convolutional layer is based on image filtering. For each channel of each convolutional layer, there is a filter sliding over the whole image. The output of single convolutional layer can be computed by 2.3. There are many parameters can influence the output of convolutional layers, and these parameters are called hyperparameters. The first one is the size of the convolutional filter. The number of filters is another hyperparameter that is determined by the number of output channels. Stride represents how many pixels are skipped when the filters are sliding over the image. Padding means that number zeros are added around the input image. Padding equal to one means there is one zero circle along the border of input image.

### Pooling layer

There are two main purposes of pooling layer. The first one is to reduce the size of the output of convolutional layer. Another one is to prevent overfitting. There are two commonly-used kinds of operations in pooling layer, which are max operation and average operation. Similar with image filtering, we use a pooling window sliding over the output of convolutional layer. Instead of convolution operation, max operation takes the maximum value out of all values appearing in the pooling window as output. Average operation calculates the average of all values appearing in the pooling window and takes the average as output.

## Fully connected layer

Fully connected layers are the final part of convolutional neural networks, which are similar to regular neural networks with neurons in different layers interconnected with each other. The convolutional layers and pooling layers extract useful features and information out of input images and fully connected layers make different decisions based on the tasks.

# 3

# Supplementary Results

## Datasets

We train four different place classification networks with two different dataset, **Pittsburgh** and **Tokyo 24/7** introduced in [13]. Here we show some examples of these two datasets, as shown below in Fig 3.1 and Fig 3.2.

## Results of t-SNE

Because of the number of pages of scientific paper and the size of image, we cannot show more visual explanations in the result of t-SNE. Fig 3.3 and Fig 3.4 show supplementary t-SNE results with more visual explanations generated by VGG11.

## Results of segmentation

Fig 3.5 presents several examples of segmentation results.

Figure 3.1: Example of Pittsburgh images. These 24 images were taken from the same spot with 12 different directions and 2 angles.


Figure 3.2: Example of Tokyo images. These 12 images were taken from the same spot with 12 different directions.

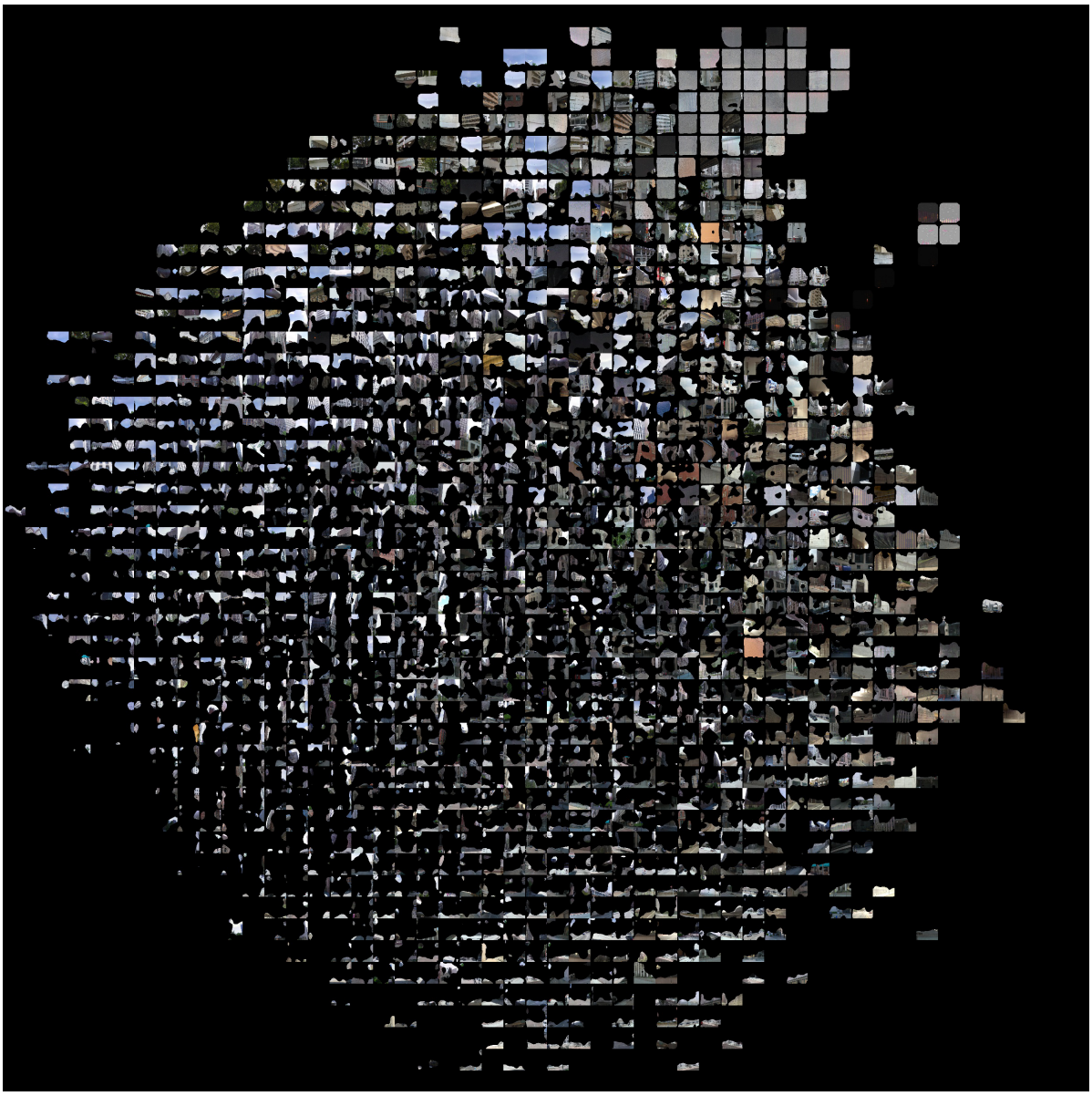

Figure 3.3: Example of t-SNE results with more visual explanations by $D_{cam}$.

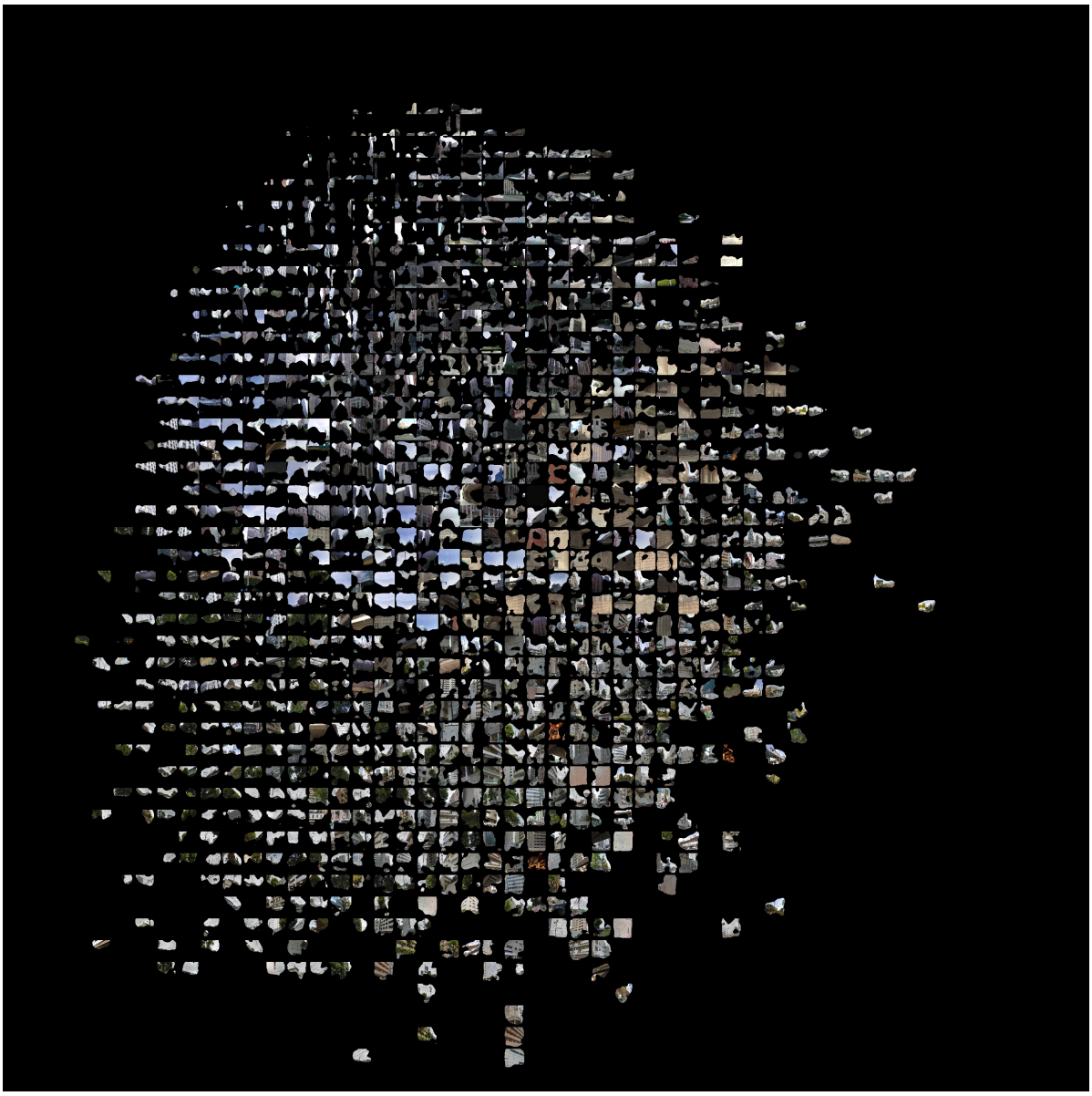

Figure 3.4: Example of t-SNE results with more visual explanations by $D_{short}$.

Figure 3.5: Example of segmentation results. Different colors represent different objects.

# Bibliography


[1] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 770–778.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks,* in *Advances in neural information processing systems* (2012) pp. 1097–1105.

[3] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition,* arXiv preprint arXiv:1409.1556 (2014).

[4] R. Girshick, *Fast r-cnn,* in *Proceedings of the IEEE international conference on computer vision* (2015) pp. 1440–1448.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014) pp. 580–587.

[6] K. He, X. Zhang, S. Ren, and J. Sun, *Spatial pyramid pooling in deep convolutional networks for visual recognition,* in *European conference on computer vision* (Springer, 2014) pp. 346–361.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 779–788.

[8] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks,* in *Advances in neural information processing systems* (2015) pp. 91–99.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, *Segnet: A deep convolutional encoder-decoder architecture for image segmentation,* arXiv preprint arXiv:1511.00561 (2015).

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask r-cnn,* in *Computer Vision (ICCV), 2017 IEEE International Conference on* (IEEE, 2017) pp. 2980–2988.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context,* in *European conference on computer vision* (Springer, 2014) pp. 740–755.

[12] J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation,* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[13] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, *Netvlad: Cnn architecture for weakly supervised place recognition,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 5297–5307.

[14] Y. Tian, C. Chen, and M. Shah, *Cross-view image matching for geo-localization in urban environments,* in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017) pp. 1998–2006.

[15] N. Vo, N. Jacobs, and J. Hays, *Revisiting im2gps in the deep learning era,* in *Computer Vision (ICCV), 2017 IEEE International Conference on* (IEEE, 2017) pp. 2640–2649.

[16] T. Weyand, I. Kostrikov, and J. Philbin, *Planet-photo geolocation with convolutional neural networks,* in *European Conference on Computer Vision* (Springer, 2016) pp. 37–55.

[17] Z. C. Lipton, *The mythos of model interpretability,* arXiv preprint arXiv:1606.03490 (2016).

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, *Why should i trust you?: Explaining the predictions of any classifier,* in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (ACM, 2016) pp. 1135–1144.

[19] F. Grün, C. Rupprecht, N. Navab, and F. Tombari, *A taxonomy and library for visualizing learned features in convolutional neural networks,* arXiv preprint arXiv:1606.07757 (2016).

[20] M. D. Zeiler and R. Fergus, *Visualizing and understanding convolutional networks,* in *European conference on computer vision* (Springer, 2014) pp. 818–833.

[21] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, *Object detectors emerge in deep scene cnns,* arXiv preprint arXiv:1412.6856 (2014).

[22] K. Simonyan, A. Vedaldi, and A. Zisserman, *Deep inside convolutional networks: Visualising image classification models and saliency maps,* arXiv preprint arXiv:1312.6034 (2013).

[23] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, *Striving for simplicity: The all convolutional net,* arXiv preprint arXiv:1412.6806 (2014).

[24] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, *Learning deep features for discriminative localization,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 2921–2929.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, *Grad-cam: Visual explanations from deep networks via gradient-based localization.* in *ICCV* (2017) pp. 618–626.

[26] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros, *What makes paris look like paris?* ACM Transactions on Graphics **31** (2012).

[27] J. J. Koenderink, *The structure of images,* Biological cybernetics **50**, 363 (1984).

[28] W. S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity,* The bulletin of mathematical biophysics **5**, 115 (1943).

[29] X. He and S. Xu, *Artificial neural networks,* Process Neural Networks: Theory and Applications , 20 (2010).

[30] S. Albelwi and A. Mahmood, *A framework for designing the architectures of deep convolutional neural networks,* Entropy **19**, 242 (2017).