Generation of a region's interesting spots based on social interest
A Nagoya University Internship Report

Magali Philippe

04/01/15 - 06/14/15

Contents

Résumé

L'objet de ce stage de recherche est l'utilisation des réseaux sociaux dans la génération de cartes représentatives de l'intérêt social d'une région.

Contexte

Ce stage a été réalisé au sein du laboratoire MuraseLab, dirigé par Mr. MURASE Hiroshi, au sein de l'université de Nagoya. A la demande de monsieur IDE Ichiro, j'ai été chargée :

1. De réaliser une application de deep learning permettant de détecter le concept représenté par une image. Cette application, en plus d'être utile à ma recherche, devrait être utilisée par toute personne du laboratoire intéressée.
2. D'utiliser cette application pour ma recherche : à savoir la génération automatique de cartes touristiques basées sur les informations trouvées dans les réseaux sociaux.

Approche

Pour réaliser l'application finale, nommée POI map puisque représentant les points d'intérêts d'une région, je me suis d'abord concentrée sur le détecteur de concept.

Une fois le détecteur de concept implémenté, j'ai pu l'utiliser pour générer des cartes touristiques.

L'approche générale que j'ai choisie est de télécharger les images et métadonnées de Flickr, via l'API Flickr qui permet de ne télécharger que les images géotaggées appartenant à une région choisie. A partir des coordonnées, un algorithme de clustering permet de sélectionner les zones d'attention principales. Une sélection des "meilleurs" endroits à visiter, les endroits les plus populaires, est ensuite réalisée. C'est sur ces clusters que sera réalisée la détection de concept, de façon à attribuer un ou plusieurs mots-clés à chaque cluster.

Détection de concept

Pour le détecteur de concept, que j'ai réalisé en C++, je me suis servie du framework Caffe.

Le détecteur de concept peut-être utilisé avec n'importe quel réseau, selon les besoins de recherche. Dans mon cas, j'ai choisi un réseau trouvé dans le Caffe Model Zoo, où la communauté Caffe peut poster ses réseaux après apprentissage de façon à ce que les autres membres économisent du temps et des ressources.

L'apprentissage de ce réseau a été réalisé sur la base de donnée Places205, un ensemble de 205 catégories non pas d'objet, mais d'endroits, tels que "montagne", "désert " ... Pour réaliser cet apprentissage, le réseau de neurone ayant remporté le premier prix de classification lors de la compétition annuelle de reconnaissance d'images ImageNet a été utilisé. Il s'agit de GoogLeNet, un réseau de 22 couches nommé par hommage à LeNet, le premier réseau de neurone à convolutions connu.

Avec ce réseau tel quel, les catégories étaient un peu trop précises pour l'usage voulu. J'ai donc réorganisé ces 205 catégories en 30 catégories, que j'ai pu tester sur un sous-ensemble de la base de donnée SUN. Les résultats obtenus ont été très satisfaisants : 68 % des prédictions contenaient la bonne réponse dans la première proposition, 88% des prédictions contenaient la prédiction correcte parmi les 5 premières propositions.

Mise en place de la POI Map

L'approche choisie pour la POI Map est de télécharger des photos et leurs informations sur la forme de fichiers xml via l'API Flickr, puis de traiter ses photos de façon à obtenir un résumé des informations essentielles sous la forme d'un fichier JSON.

Le traitement consiste à :

1. Parser les informations xml
2. Utiliser un algorithme de clustering sur les coordonnées des images
3. Sélectionner les principaux clusters
4. Faire une reconnaissance d'image sur chaque image des clusters concernés
5. Conclure sur chaque cluster en leur assignant des tags et la précision correspondante

De façon à permettre l'interactivité (sur interface Google Maps), la méthode originale a été légèrement modifiée de façon à ce qu'une partie du traitement se fasse également en Javascript, pour que l'utilisateur puisse changer certains paramètres en temps réel.

Le principe est donc d'enregistrer dans le fichier JSON les informations permettant d'obtenir un maximum de clusters avec un maximum de tags, de façon à ce qu'à la demande de l'utilisateur (ou pour réaliser les réglages standards), il suffise de fusionner certains de ces clusters, et de cacher les tags n'ayant pas une précision assez élevée.

Paramètre temps

Les activités n'étant pas les mêmes à toute heure de la journée, une amélioration a été proposée : l'option temps. Cette option permet à l'utilisateur de sélectionner s'il veut voir les activités du jour, du soir, en semaine ou en weekend. Cela est permis par la création de plusieurs fichiers JSON pour un même lieu, chacun correspondant à des options de temps précises.

En plus de ces options de temps générales, l'idée d'ajouter des graphiques d'heures populaires pour réaliser certaines actions à été évoquée. Nous avons donc ajouter ces graphiques pour le label "manger", permettant à l'utilisateur de connaître les heures auxquelles les gens mangent, pour toute la région couverte, puis par clusters.

Un cas d'utilisation peut être le touriste qui veut manger à 20h, regarde sur le graphique s'il s'agit d'une heure populaire, clique pour voir les clusters concernés, regarde le nombre de photos correspondant à cette heure de dîner dans chaque cluster, puis peut avoir un aperçu de ce qu'il pourrait manger en faisant défiler les photos.

Conclusion

Le détecteur de concept fonctionne comme prévu. A mon départ, deux autres personnes du laboratoire l'utilisaient pour leur recherche.

Pour l'application cartes, elle a été complétée comme prévu. L'interface est facile d'utilisation et permet d'accéder aux informations rapidement. Les tags assignés aux clusters sont cohérents avec les images. La seule limite est le nombre de photos disponibles : Flickr ne permet pas l'accès en temps réel à ses images qu'il faut donc télécharger.

Le stage en général ce sera bien déroulé, et m'aura permis d'apprendre de nouvelles méthodes, plus modernes, de traitement d'images. Cependant les méthodes vues dans ma formation m'auront beaucoup aidée.

Abstract

Nowadays, social networks provide a huge database of various contents, updated everyday. This research uses a photo-sharing social network, Flickr, to generate POI maps. A POI map is a map that represents every cluster of social attention, called point of interest (POI), on a simple user interface.

The generated interface is an interactive visual summary of different areas where :

- The POIs are represented by icons, where those icons give an information on what type of place the POI is. Those icons can be actions (eating, drinking), or scenes (mountain, water).
- The popularity of the POIs can be set by the user, as well as the time range, the size to the clusters, or other parameters.
- Photo miniatures allow the users to have a view of the clusters they are interested in
- The label assignment process does not use any type of metadata. It is realized exclusively with deep learning.

Introduction

When visiting a new place, several resources are available. Most people spend hours, even days, to study an area before visiting it. For popular places, tourist guides are available, but they might be irrelevant as changes happen quickly. Moreover the information provided in these guides only concerns extremely popular places, when some users might be interested in places slightly less popular.

On the other side, internet websites can give additional information about places. However the quantity of available information, scattered everywhere with a quality that varies widely, might make it difficult to find where to go when visiting a new place.

At the same time, social networks are becoming more and more popular. For users it is a good way to share and connect, for researchers it is an ever-changing source of data : textual data (Twitter), images (Instagram, Flickr) or even videos (Youtube) are available everyday in huge quantities. Why not, then, use it to measure the popularity of different places ? If we make the assumption that people are more likely to take photos in places they like, geotagged photos can be a good indicator of places people like to go to.

Thus, we decided to make an interactive map allowing people to get a visual summary of the interesting places of an area: this research follows [4](which uses Panoramio images to find the most popular spots, then classifies each of them in one of the five scene categories “city”, “mountain”, “flatland“, ”water“, ”forest“) with several improvements :

- We added actions(eating, sleeping, shopping, drinking) to the scene categories and extended the number of categories to 30
- As suggested in the article, and because places can be popular for more than one reason, a popular spot can be represented by more than one label(labels visually represent categories)
- Time has been taken into consideration : places where people go during the week are different than places they go to during the weekend
- The map is interactive: users can select options to adjust the popularity, the time, and obtain additional information
- The social network is Flickr : it is well known for the quality of its photographers and seems more popular than Panoramio

The first chapter will detail the purpose of the application and the approach, then a second chapter will focus on image recognition when a third chapter will explain how the map itself was designed and implemented. Finally, the last chapter will be dedicated to the results and conclusion.

Context

The internship took place in MuraseLab : this laboratory is a media laboratory named after Mr. Murase Hiroshi, professor of this laboratory. It belongs to Nagoya University School of Information Science and counts about 30 members.

The research in this lab is divided in three different fields :

1. ITS : the car-related multimedia research, such as pedestrian detection
2. Media : everything related to the research of how to process videos, images, sounds and even texts that can be found on social networks

3. Recognition : this group was dedicated to the research on image recognition, video recognition or speech recognition

Because my research was based on deep learning, I was a member of the recognition group.

During the research, I was supervised by Mr. Ichiro IDE, associate professor. All members of my group could hear about my progress regularly and make suggestions, as we were having meetings every other week.

Chapter 1

Approach

1.1 Purpose

To help tourists find relevant information about a place they are interested in, based on geotagged photos obtained via Flickr, is the main purpose of this research. The key points are the following :

To offer a visual summary of the information obtained by a photo-sharing social network In order to present the basic information about a place, each point of interest will be represented by one or more labels. Those labels can be airport, temple, religious sites(other than temple), beach, art(museum, art galleries), eating, sports (indoor or outdoor), park (or garden), zoo, castle, forest, mountain, playing, building(or house, mansion), bridge, drinking, water(river, lake), desert, market, show(concert), swimming, field, shopping, dancing, tower, amusement park, aquarium, bowling, sleeping or other. Each label will be represented by a main icon, and secondary icons. To each point of interest, will also be associated a list of photos, and a time information.

To let the user interact with this visual summary Because different users are interested by different places, activities, times for different reasons, this research provides an interactive map where users can get a closer look to the POIs they are interested in, in addition to the ability of changing different parameters :

- The popularity of the points of interest (POIs) they want to see : this will be determined by the number of different users that took photos in the corresponding cluster;
- The size of the POIs
- The number of labels displayed for a POI : if the number is small, the few labels that will appear will have a really high relevance, but the user might miss other activities; if the number is too high, however, many activities will be found in every POI, but they might not be relevant at all.
- The time values : weekdays versus weekends and day versus night.

1.2 Similar research and main differences

The increasing popularity of social networks such as `Twitter`, `Instagram` and `Flickr`, where people can share photos with the world encouraged researchers to use these social networks as huge database for tourism, social studies, places representation...

Concerning tourism, similar research as the one mentioned in this report has been conducted. Thus, it is important to keep in mind the important differences about this research :

1. **This research is not about recommending new places to visit.** Diverse articles about travel recommendation have been written, in particular ”A worldwide tourism recommendation system based on geotagged web photos.” [1] that provides the user with a recommended destination to visit after either a photo of the desired scenery or a keyword describing the place of interest has been entered. The research we conducted is not about suggesting new places : it is about providing information on a specific place, after it had been chosen by the user. The main use cases here are :
   — A tourist wants to visit a new place and tries to plan his trip
   — A tourist is interested in several places and wants an efficient way to get a better idea of what he can actually find in those places
   — A user is interested in finding other places in his own area : he may discover hidden places only a few people know (nice sceneries for example)

2. **This research classifies the POIs(points of interests) in predefined categories, allowing the user to only check what matches his interests, whereas other research considers each POI as a unique element** ([2]). Instead of landmarks names or miniatures, we will have instead the different categories they belong to.

3. **Not only the main landmarks are displayed: the user can interactively choose the popularity of places he wants to see**. Whereas some research ([8]) combines `Wikipedia` to the found landmarks, in order to only show places a city is known for, we consider that any place where a lot of different people take photos can be interesting (even shopping areas, restaurant areas), and even places where few people go can be found interesting (an unknown village where it is nice to walk around for example).

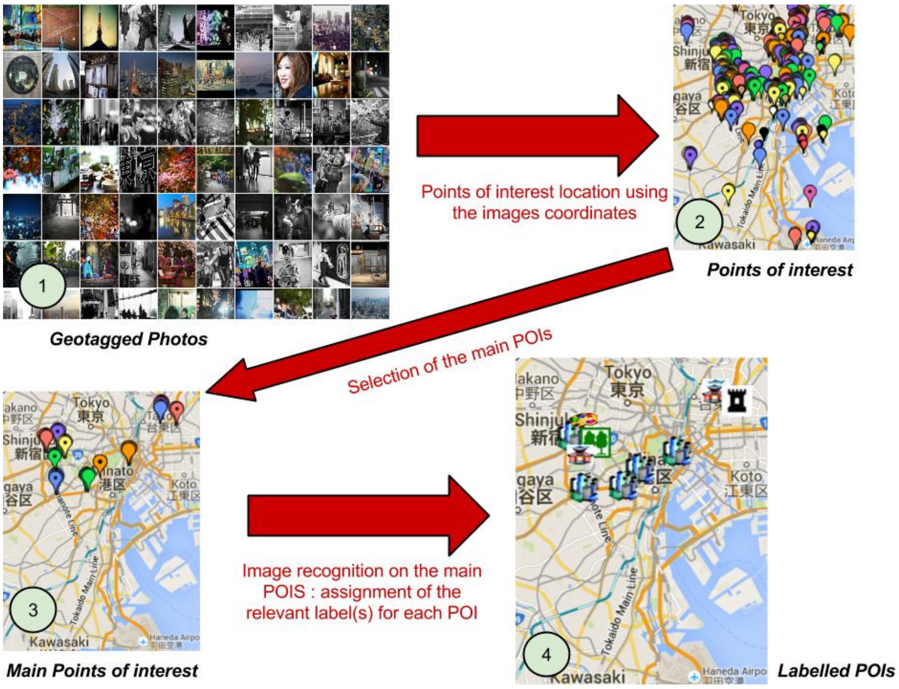
4. **This research does not use the meta-data for the description of the selected places : the results do not depend on the photos owners tags choices or title choices.** Contrarily to previous research ([3][2][8] among others), instead of using photos tags or titles, we use deep learning to recognize the image concept. This can be really useful because some users do not use tags at all, or use irrelevant tags; automatically generated titles are common too.

5. **The generated map is interactive, allowing the user to choose different options and to obtain additional information on the clusters he finds interesting**

1.3 Approach

To provide visual summaries from geotagged photos, we follow three steps :

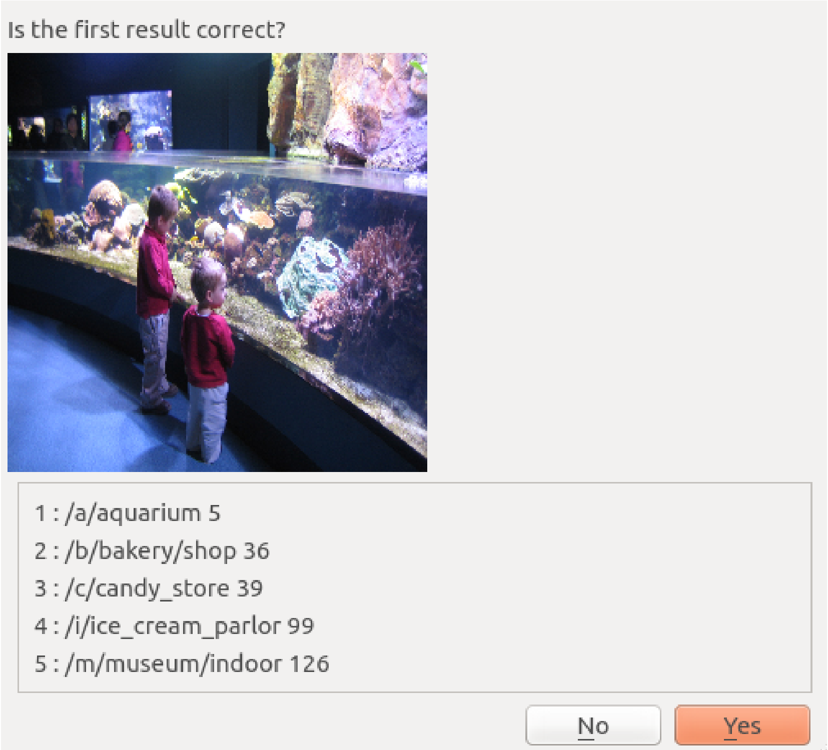
1. From the geotagged photos collected with the Flickr API, create clusters based on the coordinates
2. Among these clusters, select the main POIs
3. With an image recognition algorithm based on deep learning, assign one or several label(s) to each POI


1
Geotagged Photos
Points of interest location using the images coordinates
2
Points of interest
Selection of the main POIs
3
Main Points of interest
Image recognition on the main POIS : assignment of the relevant label(s) for each POI
4
Labelled POIs

First, the image recognition will be detailed. Then, we will explain how the image recognition was used to obtain the final maps.

Chapter 2

Image recognition : How to classify an image so it can be labelled



Is the first result correct?

1 : /a/aquarium 5
2 : /b/bakery/shop 36
3 : /c/candy_store 39
4 : /i/ice_cream_parlor 99
5 : /m/museum/indoor 126

As mentioned in the previous chapter, the labels assignments are completely based on deep learning for image recognition(no meta-data necessary). This chapter will describe :

1. First a presentation of the main concepts necessary to understand how it works,
2. Then a description of how deep learning was used in this research,
3. Finally the image recognition results obtained.

2.1 What is deep learning and what can we do with it?

2.1.1 Object recognition

Object recognition has been a popular research matter in the last decade (ImageNet annual contest for example). It can be used in various fields such as robotic, automatic cars, surveillance ... It can be image classification (for example Google's "personal image search", allowing users to research keywords among their personal photo collection), object detection (locate an object from one image), or segmentation.

The object recognition used to be realized by features recognition, where the features were hand-made. With the use of neural networks, this is not necessary anymore : networks can learn features. Today, every state-of-the-art recognition task is based on Deep Learning.

Deep learning is a great tool for object recognition, but it is also what many speech recognition algorithms stand on today. They can even be used for 3D applications.

2.1.2 Neural Networks

The idea behind neural networks was to study how the human brain recognizes objects. The first experiments started in 1943 when the neurophysiologist Warren McCulloch and the mathematician Walter Pitts modeled a single neuron with electronic circuits. This neuron was processing several inputs and outputting a result depending on their respective weights.

In the 1950's, powerful computers made it possible to simulate a neural network on a larger scale :

- 1955 : IBM launches a study group with the mission of simulating the behaviour of abstract neural networks
- 1959 : ADALINE and MADALINE are developed by Widrow and Hoff. The latter will be the first neural network applied to a real-life problem : eliminate echoes on phone lines.

The first learning machine is designed in 1960 by Cornell : the Perceptron is a linear classifier on top of a simple feature extractor.

In the 1990's, a successful type of neural network revolutionized deep learning : Yann LeCun's CNNs (Convolutional Neural Networks). Since then, CNNs were applied successfully to:

- Face recognition ([7], [12], [10])
- Hand-written digit recognition (on the MNIST database)

— More challenging recognition tasks (ImageNet : since 2012([6]) all the winning teams have used CNNs [9] [11])

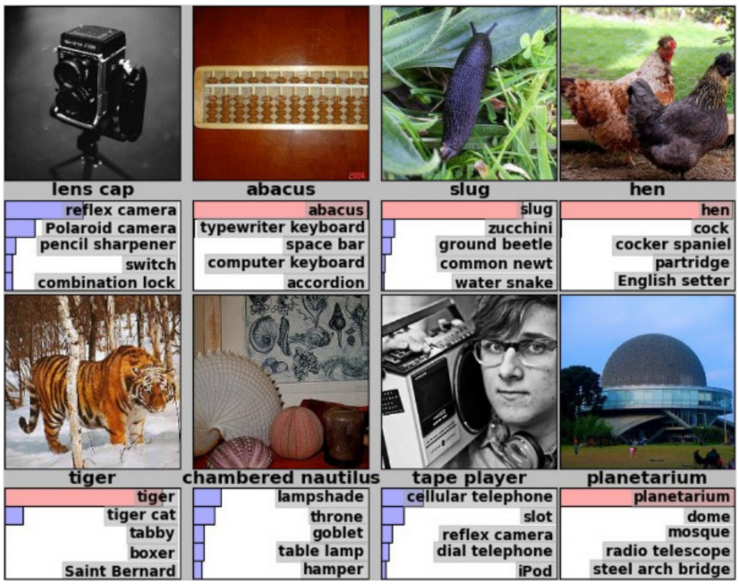


Figure 2.1: ImageNet 2012 : prediction examples

2.1.3 Convolutional neural networks

Compared to other image classification algorithms, convolutional neural networks use relatively little pre-processing. This means that the network is responsible for learning the filters that in traditional algorithms were hand-engineered. The lack of a dependence on prior-knowledge and the existence of difficult to design hand-engineered features is a major advantage for CNNs.

The parameters are learnt by classification error backtracking. For image recognition, a set of images and corresponding labels is used to train the network -learn the parameters at every stage. An average CNN combines, at every stage convolutions, bias, non linearity (ReLU or sigmoid functions), and pooling, to obtain a feature map.

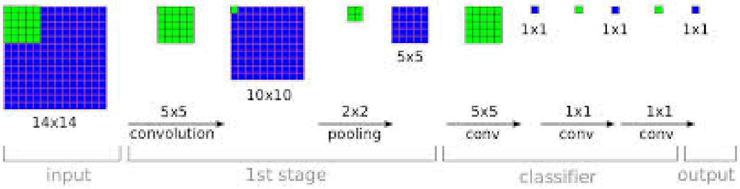


Figure 2.2: Simplified CNN : first stage

Parameters sharing and pooling take advantage of local coherence to learn invariant features. The obtained features at each stage are called feature maps.

As we get higher in the net features get more and more precise, similarly to the brain representation (pixel -> edge -> shape -> object).

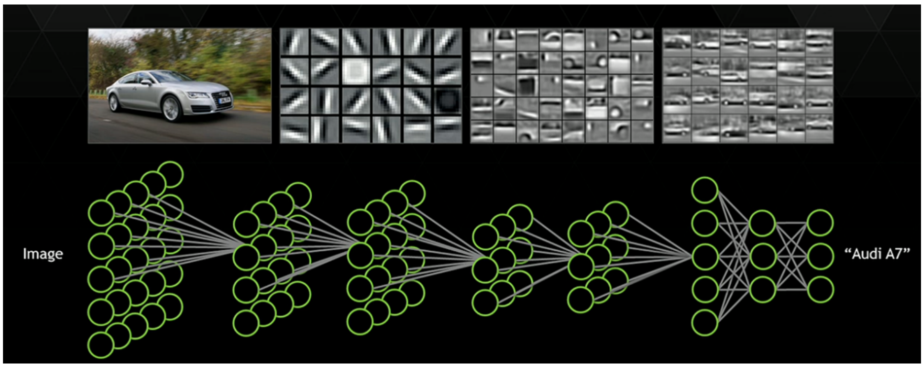


Figure 2.3: Feature maps at different levels

2.2 Application to our problem

Convolutional neural networks (CNNs) being the state of the art for image classification, it appeared natural to use this technique in our research. Three main elements constitute the image recognition we used in our research : the framework, the database and the actual convolutional network.

2.2.1 Framework

With the increasing popularity of deep learning, several frameworks are nowadays available to help programmers. In this research, we used Caffe ([5]), which we ran in C++.

Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors. Yangqing Jia created the project during his PhD at UC Berkeley.

2.2.2 Network choice

ImageNet is the most challenging image recognition classification. We decided for that reason to use the last winning network, which is GoogLeNet (as a tribute to LeNet), a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

It is based on a deep convolutional neural network architecture codenamed ”Inception”, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC 2014). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. More details are in this paper : Going Deeper with Convolutions - Szegedy et al [11].

2.2.3 Database

When visiting a new place, we are interested in scene concepts (mountain, water, park) more than object concepts (grass, boat, dog). It was thus important to find a scene-centered database in order to detect relevant concepts for our application.

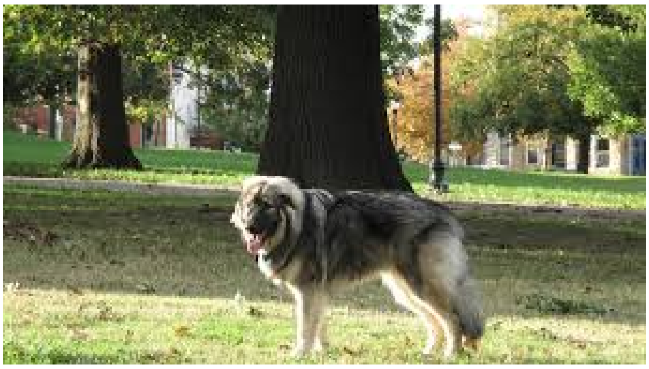


Figure 2.4: Different concepts for one image : dog or park ?

For example, let's consider a photo with a dog playing in a park. A CNN trained on an object-centered database would detect the dog whereas a CNN trained on a scene-centered database would detect the park.

For that reason we chose the scene-centered database Places205, described in this article : Learning Deep Features for Scene Recognition using Places Database - Zhou et al [14].

It appeared that the author trained the GoogLeNet network on this database, and made the pretrained network available in the Caffe Model Zoo. We decided to use this pretrained network, that was particularly relevant to our work. Because 205 categories were not necessary for are application, we changed the output to 30 categories instead.

2.3 Tests and results

Before using the concept detector, a few tests were realize to prove its efficiency. We decided to run the tests on a well-known benchmark, instead of personal data (those tests were also ran, but with way less images).

Another reason we did the tests is because we had to determine whereas it was better to map the labels, or better to fine-tune the network quickly(a proper fine-tune would've taken to much time and resources we didn't have).

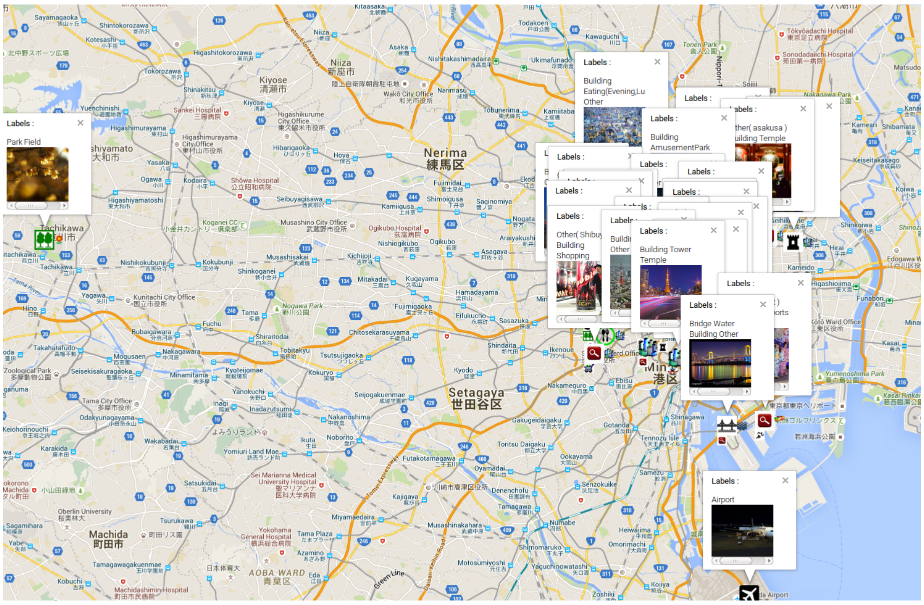
We used the SUN397 Scene benchmark ([13]) to evaluate the accuracy of the predictions. On the 397 categories provided by this database, only a few are interesting for tourists ('mountain', for example, is interesting, whereas 'parking lot' is not so interesting).

A top-k accuracy is defined by the number of images for which the right prediction was among the first k results. This is what the ImageNet contest measures to evaluate the success of a network, this is what we also decided to use.

For the pre-trained network mapped with the new labels, we obtained a top-1 accuracy of 68% and a top-5 accuracy of 88% for the classification of 76 categories of the SUN database into 30 categories. This was about 15% better than what we could realize with the quickly fine-tuned network.

Chapter 3

Application to a set of geotagged images : How to use the image classification



The visual summary we mentioned in the introduction will be called POI map. This name reflects the fact that this map represents the main Points Of Interests of an area.

3.1 POI Map Concept

As a reminder, the POI map (the application of this research), is a map that :

- Uses Flickr photos to understand which areas are interesting to visit/go shopping/go eating etc... It is based on the assumption that people are more likely to take photos in interesting places;
- Represents the most interesting areas (POIs);
- Automatically assigns one or several label(s) to each POI (this is where the image recognition is important)

3.2 POI Map implementation

3.2.1 Clusters selection and label assignments

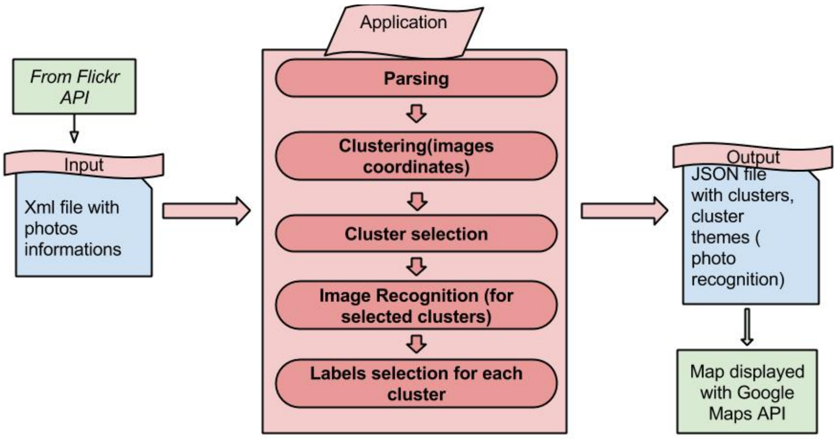

From Flickr API
Input
Xml file with photos informations
Application
Parsing
Clustering(images coordinates)
Cluster selection
Image Recognition (for selected clusters)
Labels selection for each cluster
Output
JSON file with clusters, cluster themes (photo recognition)
Map displayed with Google Maps API

Figure 3.1: Approach

As described in the above figure, the map generation is based on the following operations :

- Parsing of the xml file : The Flickr API allows us to download a folder of images and the corresponding information. We generated an xml file for each package of photos where each photo, identified by its Flickr id, had the following information :
  - Title, tags
  - Author : every user has a unique user id
  - Coordinates
  - Date and time when the photo was taken
- Clustering : in order to cluster the image coordinates, we used the mean shift algorithm

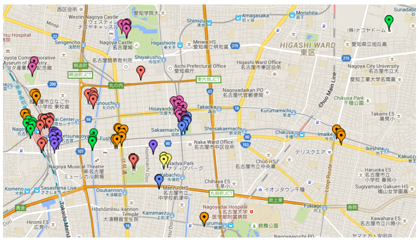


Figure 3.2: Clusters obtained in Nagoya with the mean shift algorithm

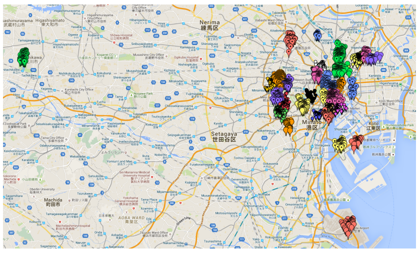


Figure 3.3: Clusters obtained in Tokyo Prefecture with the mean shift algorithm

— Cluster selection : only the most popular clusters are selected. The popularity of a cluster depends on the number of different users that took pictures in the same cluster. Counting the number of different users instead of simply counting the number of photos in a cluster is justified by the fact that some users take much more photos than others. If one user takes 20 photos in one place, for example, it is very likely that this place is less interesting than somewhere where 10 users take only one photo, even though this place will have a lower number of photos.

— Labels assignment (photo recognition and labels selection): For each cluster, the information obtained with the concept detector (deep learning application) is generalized from each photos to the whole cluster

— Generation of a Json file : this is what will be transformed in a visual summary with Javascript and the Google Maps API

Effectuating these operations in this order is what seemed intuitive at first, and this is what will be described in the "first method". However, to improve it, a "second method" was considered, slightly changing the first method in order to obtain an interactive map where the user selects the parameters.

First method

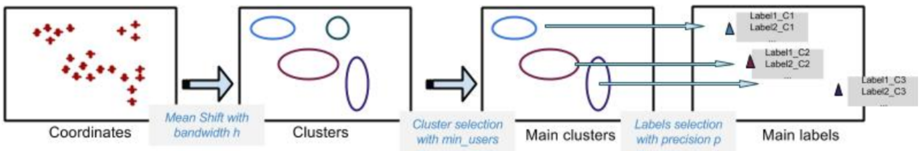


Figure 3.4: First Method

Our first approach was :

1. Cluster the coordinates with the mean shift algorithm, with a preset bandwidth h
2. Choose the main clusters with the minimum number of different users preselected
3. On the selected clusters, apply the image recognition for each photo, then select the relevant clusters. The relevance was measured with this

approach :

- A preset number precision was meant to measure the minimum relevance of the selected labels
- A label had a high precision if the concept detector detected (with a high probability) this label on many photos of the same cluster. For example in a restaurant area, there might be one photo of an aquarium (which you can find in some restaurants) : this is not enough to claim that this area is known for its aquarium
- The precision of a label also depended of the probability given by the concept detector when predicting this specific label for an image. For example, if half of the images of a cluster have the label ”forest” predicted, but each time with a probability of 0.5%, it is very unlikely that there is a forest in the cluster, which is why the precision has to remain low.

4. Generate a Json file with all this collected information : one map per area

Problems with this method :

- The parameters have to be chosen before the generation of the Json file: each Json file corresponds to one choice of parameters. Because we cannot choose the same parameters for every area, the Json generation cannot have default parameters.
  For example a city and a country won't have the same scale : we cannot choose the same bandwidth for both of them.
- The user might be unsatisfied with the parameters choice : the “best parameters” notion is based on a completely personal perception. Different Json files could be generated each time so the user can choose, but it is not possible to generate every possible Json files for each region.

In order to solve these problems, a second approach was chosen, improving the map interactivity, that way improving the adaptability to a specific user.

Second method : how to let the user choose the parameters in real time

In the first approach, the mean shift algorithm bandwidth, the popularity of the displayed clusters, and the minimum precision were chosen before the map visualization. In order to let the user set these parameters in real time, we had to take a different approach.

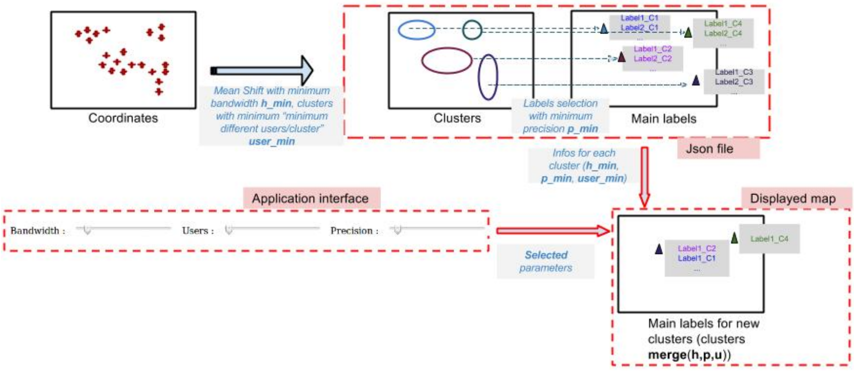

Coordinates
Mean Shift with minimum bandwidth h_min, clusters with minimum "minimum different users/cluster" user_min
Clusters
Labels selection with minimum precision p_min
Main labels
Label1_C1
Label2_C1
...
Label1_C4
Label2_C4
...
Label1_C2
Label2_C2
...
Label1_C3
Label2_C3
...
Json file
Infos for each cluster (h_min, p_min, user_min)
Application interface
Bandwidth :
Users :
Precision :
Selected parameters
Displayed map
Label1_C4
Label1_C2
Label1_C1
...
Main labels for new clusters (clusters merge(h,p,u))

Figure 3.5: Second method

The generated Json file, is made to contain as much information as needed for every settings a user might want :

— `hjson` , the bandwidth, is chosen really small as we have more clusters with a small bandwidth

— `ujson` , the minimum number of different users(different users that took photos) in each POI, is chosen small too, as we want to keep a maximum of clusters in the `json`

— `pjson` , the precision is also chosen small : that means a lot of possible labels will be available in each cluster (with their corresponding precision)

From this Json file, new maps can be calculated with the user preferences :

— `h` , the bandwidth, can be increased by the user. When the user modifies the bandwidth, the mean shift algorithm is ran with the new bandwidth : the considered points will be the centers of the small clusters (remember `hjson` was chosen small), with a weight equal to the number of different users per cluster.

— `u` , the minimum number of different users can be increased : the json file gives the number of different users in each cluster : when the cluster merge (because the selected bandwidth is most likely bigger than `hjson`) the resulting number of users is the sum of all merged clusters users

— `p` , the precision can be increased on the interface : the json knows the precision of each label in each cluster. When n clusters merge, the precision of a label l in the obtained cluster becomes : $P_l = \frac{\sum\limits_{k=1}^{k=n} P_l(k) * u(k)}{\sum\limits_{k=1}^{k=n} u(k)}$

For those who do not want to spend time finding the right settings, default settings are also provided.

3.2.2 Time parameter integration

To add the possibility for the user to choose the time range he is interested in (day vs night and weekdays vs weekends), we generate for each area four files :

— Weekend days

- Weekend nights
- Weekday days
- Weekday nights

Then, depending on the options the user is interested in, the corresponding files are read. For example, if the user wants to check the interesting activities at night, `weekend_nights` and `weekday_nights` will be read. This feature is interesting as people don't do the same activities at night as they do during the day. For example the daytime is more likely to be used to visit parks, castles, landmarks. The night is more likely to be used to go drinking, eating, seeing shows...

However, for some actions, it can be interesting to know the specific hour when this action is most likely to take place. For that reason, we added a feature that gives more information about eating times to the user (this can be extended to any other action, the exact same way).

Here is how we processed :

- For each cluster, when the label `Eating` is selected, a research is done on the photos that were recognized as food photos
- For these photos, the meta-data is processed in order to get a set of every eating hours recognized in the cluster
- When the `JSON` file is generated, a table of the eating hours and their occurrences will be added to the clusters when possible

This way, the user can see the eating hours in all clusters, click to see which clusters are concerned by the hours he/she is interested in (the markers of the concerned clusters will bounce).

For a specific cluster, a graph of the eating hours is also available, to give the user an idea of how representative this cluster is of the eating tendencies of one place.

Chapter 4

Results and comments

The final application generated three maps :

- Midi-Pyrenees
- Tokyo Prefecture
- Nagoya

For each map, about 10 000 photos were downloaded from the Flickr API. To do so, we used the photo-search function with the interestingness-desc sorting option: this means that only the most 10 000 interesting photos were downloaded for each map. The interestingness is measured by a Flickr secret algorithm, originally made to select photos that appear each day on the "Explore" page.

4.1 Labels

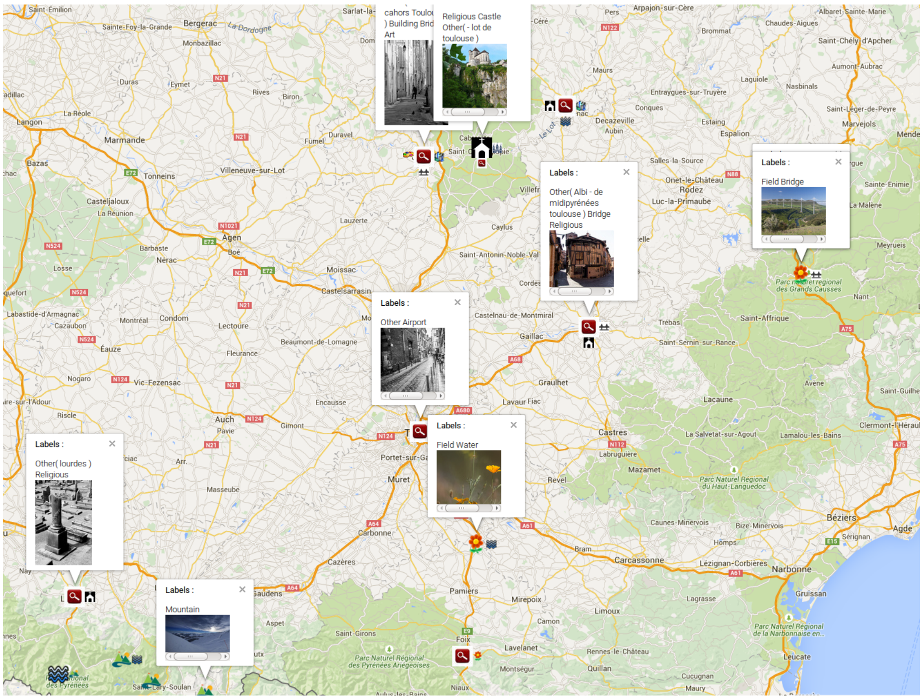
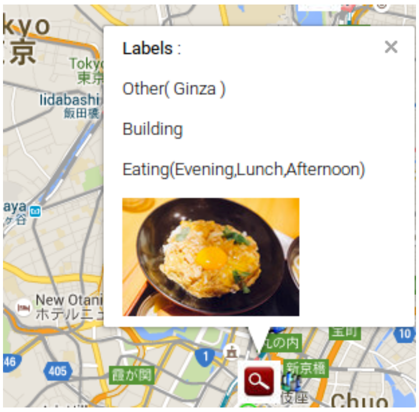


Figure 4.1: Example : Midi Pyrenees

In addition to the icons, the label names are written in the `infowindow` . On this figure you can see the label ”Other”, represented by the search icon. It was first introduced with the hope that the photos metadata (title and tags) could give additional information when the concept represented was not among the concept detector’s concepts. However, because most users don’t provide useful information in the title (often let as the default title) or in the tags (tags tend to be camera brands, adjectives that don’t describe the concept, or city names that are the same for the whole album), this idea was dropped and if the label is ”Other”, it is simply not displayed anymore.



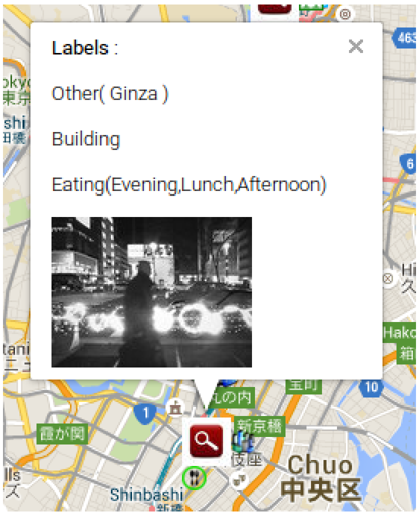
Labels :

Other(Ginza)

Building

Eating(Evening,Lunch,Afternoon)

Figure 4.2: First photo



Labels :

Other(Ginza)

Building

Eating(Evening,Lunch,Afternoon)

Figure 4.3: Second photo

Photo miniatures allow the user to get a closer look of a POI. Because POIs can have several labels, and because several photos provide more information than just one, the user can click on a cluster to see different photos of it. The displayed photos are the ones on which the concept detector predicted the chosen labels.

4.2 Obtained maps

Here are some examples of Midi Pyrenees(France), Nagoya and Tokyo Prefecture :

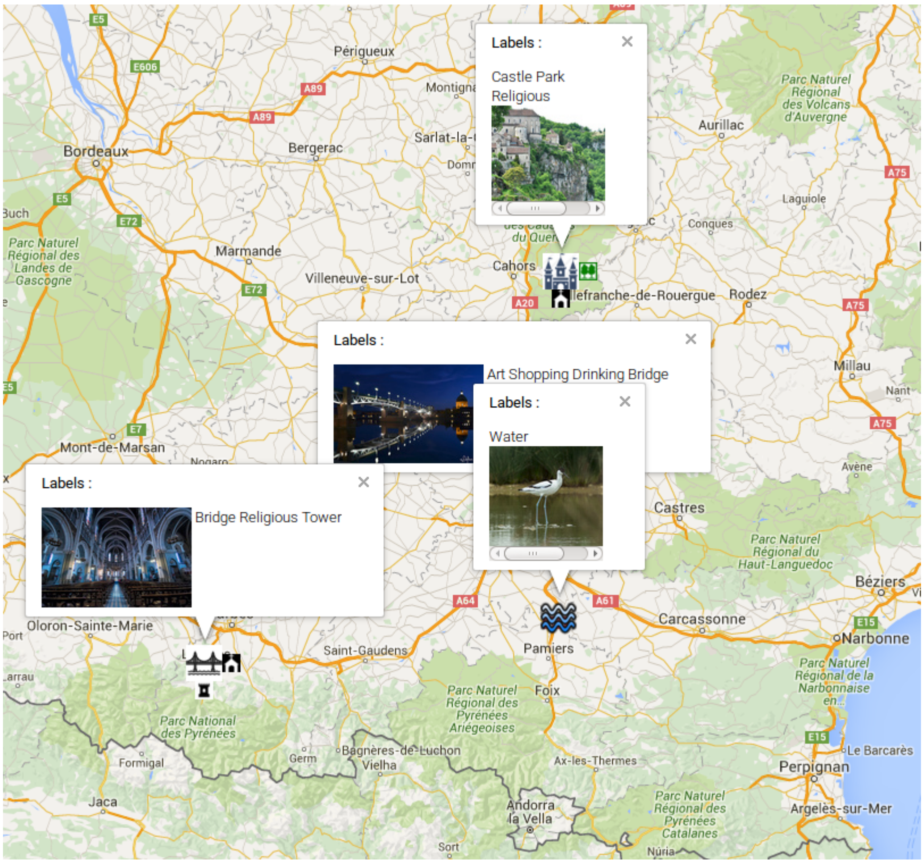


Figure 4.4: Example : Midi Pyrenees

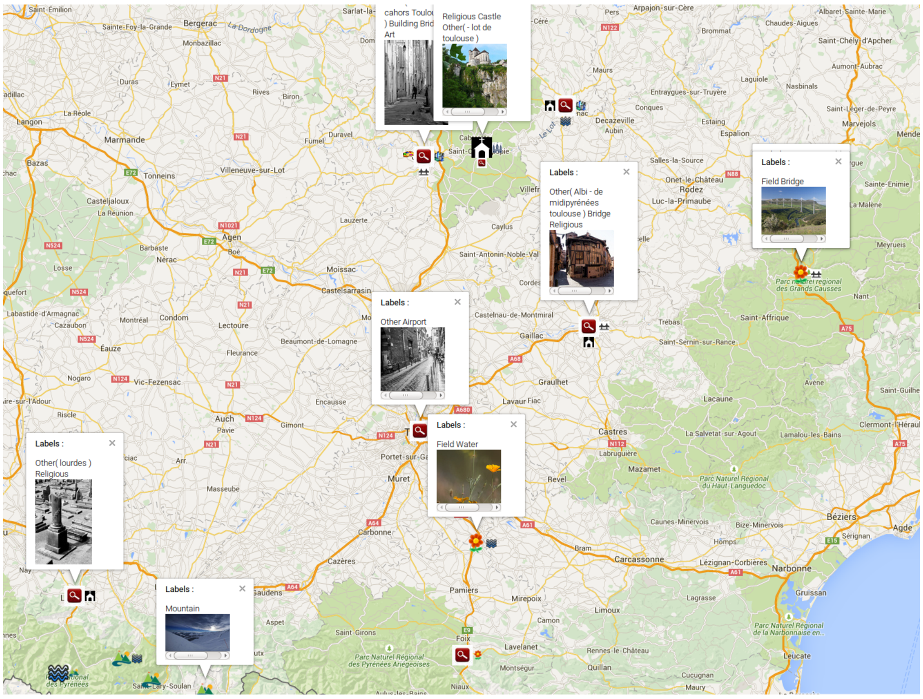
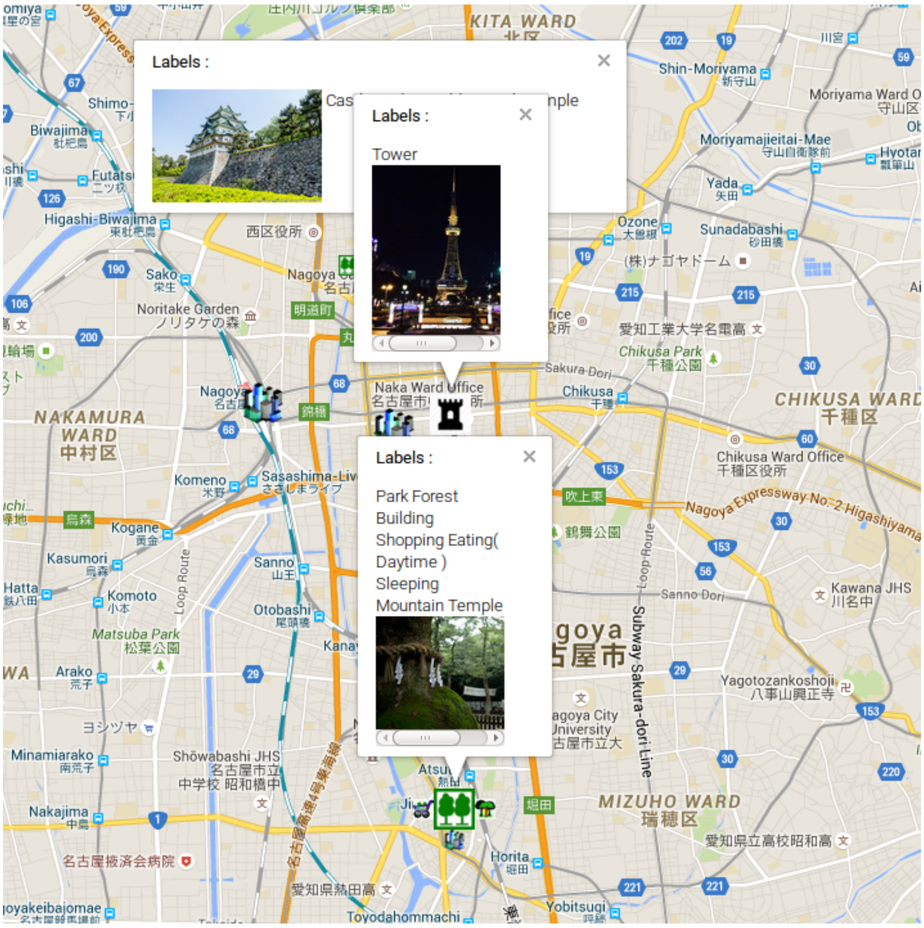

cahors Toulouse) Building Bridge Art
Religious Castle Other(- lot de toulouse)
Labels :
Other(Albi - de midipyrénées toulouse) Bridge Religious
Labels :
Field Bridge
Labels :
Other Airport
Labels :
Field Water
Labels :
Other(lourdes) Religious
Labels :
Mountain

Figure 4.5: Example : Midi Pyrenees, other settings

For Midi Pyrenees, you can see Toulouse, Lourdes, the Pyrenees, Albi, and the Castle. Those are the areas you expect someone from Midi Pyrenees to describe.



Labels :

Cas… …nple

Labels :

Tower

Labels :

Park Forest
Building
Shopping Eating(
Daytime)
Sleeping
Mountain Temple

Figure 4.6: Example : Nagoya

The main attractions of Nagoya are Nagoya castle, Nagoya Station (Meieki), Osu Kannon and Sakae. The map is quite accurate with what we expect to see.

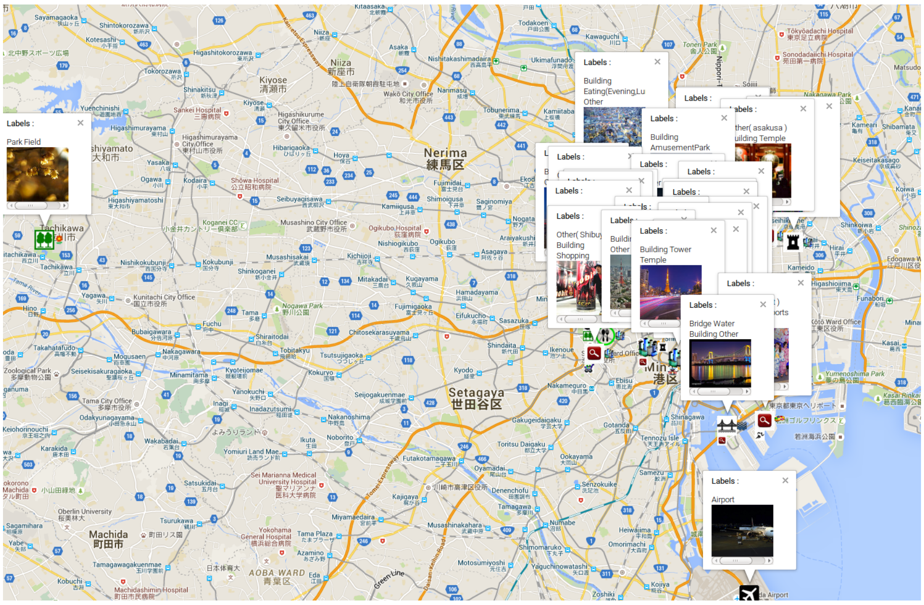
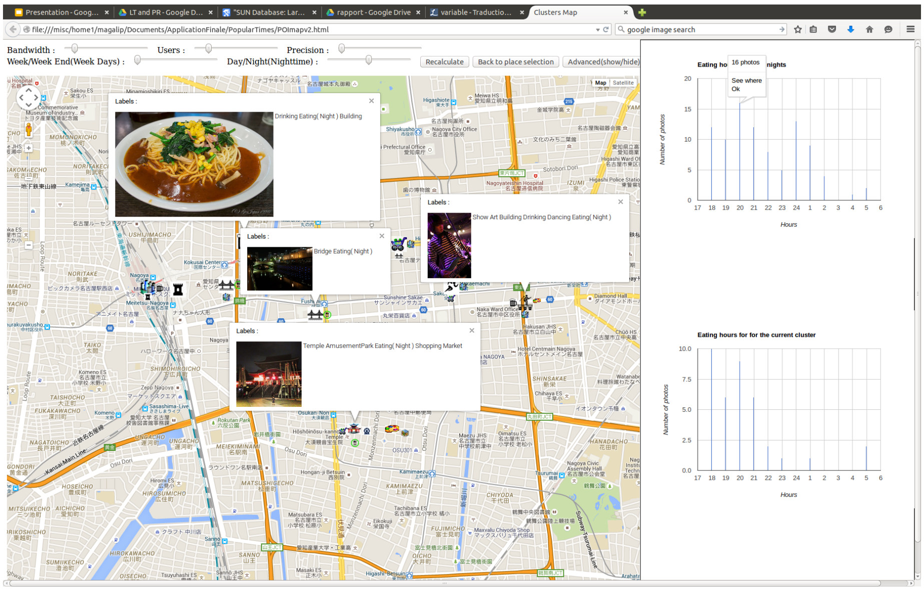


Figure 4.7: Example : Tokyo Prefecture

The size of this prefecture makes it difficult to describe everything but :

- More clusters appear as we get closer to Tokyo
- In Tokyo, we can see the main popular areas : Akihabara, Shinjuku, Asakusa ...
- Tokyo tower and every spot we can see it from are displayed on the map

4.3 Time



file:///misc/home1/magalip/Documents/ApplicationFinale/PopularTimes/POImapv2.html

Bandwidth : Users : Precision :
Week/Week End(Week Days) : Day/Night(Nighttime) :

Labels : Drinking Eating(Night) Building

Labels : Bridge Eating(Night)

Labels : Show Art Building Drinking Dancing Eating(Night)

Labels : Temple AmusementPark Eating(Night) Shopping Market

16 photos
See where
Ok

Figure 4.8: Example : Eating times in Nagoya

You can see on this figure the eating times graphs for Nagoya during weekday nights. The first graph is for all clusters, the second is for the selected cluster.

4.4 Conclusion

This research allows users to get a visual summary on a place they are likely to visit. The results seem relevant : the POIs are where we were expecting them. The labels seem coherent to the photos we can find. This shows the power of deep learning, since no meta data were used for the predictions.

A short survey was conducted on 20 volunteers to judge the interest of such an application. The highlighted benefits of such an application are :

— The interface is easy to use.

— Having actual photos instead of what we would see with Google Street View gives a better understanding of the beauty of a place

— The ability to quickly have an idea of the main activities, with the icons, is really convenient.

— After the obvious interest of visiting a new place, it was also pointed out that this application could allow the discovery of smaller places because a few users took beautiful photos on Flickr.

This research sets the basic elements of how we could use Flickr images to obtain easily an interactive visual summary.

The limits of this application might me the limited number of photos. Flickr doesn't allow an access to its data in real time, which is why every photo has to be downloaded if we want to use it .

An ideal case would be a full access to the whole Flickr image database (or any other social network), to have access to all the images at anytime and to be able to update the maps regularly without having to download the photos.

Chapter 5

Conclusion

5.1 Work results

I was assigned two tasks for this internship :

1. Create a concept detector : an application based on deep learning for image recognition
2. Use this concept detector for the generation of tourist maps

As for the first task, it was a success since the application was finished within the first two months. When I left, two people were using it for their research. I hope it will help other people since this concept detector can be applied to various image classification tasks : in my case scene recognition, but other pre trained networks can be used for object recognition, flower recognition, cars recognition, face recognition ...

The second task seem to be successful too : the map works and is easy to use. The results are relevant according to a personal knowledge about the tested places.

The labels assigned to each cluster match the photos displayed with the interface. The predictions given by the concept detector are extremely accurate, considering the difficulty of some photos.

Everything we wanted to add has been added to the application : there was enough time but not too much for this project.

5.2 Overall conclusion

This internship was a great opportunity to learn about various multimedia research : thanks to the meetings and presentations we had every week, I had the opportunity to understand the research other people were working on. The language was japanese, which made it difficult for me to understand the details. Nevertheless, I was still very interested to hear about what other people were doing.

I also learnt a lot during my own research, since I didn't know a thing about deep learning at the start. This research gave me the opportunity to use different APIs as well, Google Maps, Flickr API for example. I had the opportunity to learn something every week and I am thankful for that reason. This project is by far the most interesting project I ever worked on. Sadly, it wasn't a team work, since I was by myself on that research. However, I knew I could always ask for help.

Bibliography

[1] Liangliang Cao, Jiebo Luo, Andrew Gallagher, Xin Jin, Jiawei Han, and Thomas S Huang. Aworldwide tourism recommendation system based on geotaggedweb photos. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2274–2277. IEEE, 2010.

[2] Wei-Chao Chen, Agathe Battestini, Natasha Gelfand, and Vidya Setlur. Visual summaries of popular landmarks from community photo collections. In *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, pages 1248–1255. IEEE, 2009.

[3] Quan Fang, Jitao Sang, Changsheng Xu, and Ke Lu. Paint the city colorfully: Location visualization from multiple themes. In *Advances in Multimedia Modeling*, pages 92–105. Springer, 2013.

[4] Ichiro Ide, Jiani Wang, Masafumi Noda, Tomokazu Takahashi, Daisuke Deguchi, and Hiroshi Murase. Construction of a local attraction map according to social visual attention. In *Intelligent Interactive Multimedia: Systems and Services*, pages 153–162. Springer, 2012.

[5] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[7] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *Neural Networks, IEEE Transactions on*, 8(1):98–113, 1997.

[8] Adrian Popescu and Gregory Grefenstette. Mining social media to create personalized recommendations for tourist visits. In *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications*, page 37. ACM, 2011.

[9] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[10] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[12] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708. IEEE, 2014.

[13] Jianxiong Xiao, James Hays, Krista Ehinger, Aude Oliva, Antonio Torralba, et al. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010.

[14] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. *NIPS*, 2014.