# Explorify : An Unsupervised Photography Spot Exploration and Recommendation System

Anish Hegde, George Jacob, Dongsuk Lim, Sylvain Marchienne, Christopher Scherban, Shafa-At A Sheikh
Georgia Institute of Technology
{ahegde41,gjacob8,dlim46,smarchienne3,cscherban3,ssheikh9}@gatech.edu

## INTRODUCTION

In recent years, people contribute geo-tagged photos online to share their travel experiences via services like Flickr and Instagram. These geo-tagged photos reveal useful information such as Areas of Interest (AOI), as well as good photographs to take. This paper presents a coherent end-to-end web application for understanding geo-tagged photos based on AOI by providing an interactive map experience for travelers and photographers. Explorify identifies AOI using unique clustering algorithms, and helps recommend places to take good photographs based on an image score and location.

## PROBLEM DEFINITION

**Core User :** Travel and Photography enthusiasts who want to explore and take good photographs.
**User Problem :**

- What are good photographs to take?
- Where exactly can I take these photographs?

**Solution :** Explorify will provide an interactive photograph-driven map experience to find the best spots to take the best photographs, enhancing the way people plan to explore cities.

## SURVEY

The core challenge for us is to extract relevant photography spots using unsupervised clustering. For instance, [1] and [2] use geo located and tagged photographs from Flickr to recommend places that match user preferences and time constraint when they visit a city. They both use the DBSCAN [3] algorithm at the heart of their clustering strategy to extract AOI. Their work validates our assumption that Flickr data can be leveraged to find AOI, and gives leads on how to exploit the associated data for user recommendation. Also, [4] and [5] tackle the problem of AOI extraction from Flickr data and strongly emphasize on how to clean photographs and metadata. Particularly they detail how to filter relevant tags associated to the photographs using TF-IDF [6] to undermine the ubiquitous ordinary tags that carry poor information. They also suggest what should be a preferable photograph to represent an AOI among different candidates.

In addition to the geo location, our method leverage information from the photos themselves and metadata such as tags. Modern convolutional neural networks trained on big image classification challenge such as ImageNet [7] are well suited for extracting a good vector representation. Such large networks trained to predict a large number of common classes (animals, buildings, etc) inherently learn high level features that describe the content of the photos. Deep learning frameworks such as Pytorch [8] provide an easy way to load these large pretrained networks such as VGG16 [9] and ResNet [10].

Related to the photos, [11] proposes a deep ranking model to give an aesthetics score to pictures. Better pictures are predicted with a higher score. This is especially useful to pick the best pictures of a cluster. State-of-the-art work like [12] scores images on a range of 1 - 10. This approach matches human assessment with an 81.5% accuracy.

The tags from Flickr can be noisy or incomplete, solutions exist to help clean and improve them. For instance, [13] managed to reach state-of-the-art performances on image classification using a pretrained convolutional model on social media hashtag prediction. In our context, such a model can be used to predict new tags on the Flickr photographs and provide even more semantic understanding for the clustering model. Also since Flickr is a social platform and data is user sourced, tags might be cleaned. Wordnet Synset from [14] is a solution to figure out if two tags have the same underlying semantic meaning and remove duplicates. Having clean and meaningful tags both help the clustering but also the matching of the user query for a photo spot. The library NLTK [15] provides an implementation. [16] also shows that tags might be subjective and depends on the user, which can cause noise in tag prediction. The authors present a way to make better tag predictions by using a joint probabilistic model that captures user subjectivity. Since photos have to be compared, a vector representation can be used.

Finding areas of interest may lead to another interesting problem: route recommendations. For this, two common approaches are utilized. As found by [17] and [18], Markov chains can provide a generalizable framework to analyze tourist travel patterns. An alternative to this is Associative Rule Mining, first presented by [19].

## PROPOSED METHOD

The aim of Explorify is to provide an interactive photograph-driven map experience to find the best spots to take the best photographs. The project has a 2-pronged approach, which includes back-end logic and front-end visualization. In this section both back-end logic and front-end visualization approaches will be separately discussed.

### Approach : Back-end Logic

The task to determine the best spots and the best photographs is primarily handled by our back-end logic. It is organized as a python package encapsulating the necessary scripts and algorithms to fetch data, perform clustering and scoring of the photos. We detail the different steps for getting the data, inferring clusters and aesthetic photo scores.

### *Data Collection from Flickr*

Photographs are downloaded through the Flickr api using a free registered key. We have written a python module to simplify the process of downloading public photographs by city. The module locally saves images and metadata of the search result into a HDFS5 file. The image metadata contains information like the title, date, description, latitude and longitude, number of views and author-annotated tags. Finally, a feature representation of the image is extracted from an intermediate layer of the pre-trained VGG16 (using Pytorch). We argue that it provides a reliable vectorized embedding of image content.

### *Data Cleaning*

Tags play a crucial role as they can capture semantic information. To integrate them into the clustering process, we clean and vectorize them as a bag of words. For a photo, we translate all its tags into English in case they are not and then use a lemmatizer to reduce tags to a simplified word form (for instance 'cats' to 'cat'). Wordnet Synset is used to remove semantically duplicated tags ('cattle' and 'cow' might be merged). In the end, we use the famous TF-IDF method to convert the sequence of preprocessed tags into a numerical vector. Translation is done with Google Translate API. NLTK and Scikit-Learn are used for lemmatization, Wordnet reduction and vectorization.

### *Unsupervised AOI clustering*

Our goal is to automatically extract photo spots in a city using unsupervised machine learning and available data from Flickr. DBSCAN is the ideal candidate for this task since it has been widely used for AOI tasks in the literature and offers great performances for location-based clustering. In our context, we want clusters to fulfill two requirements: photos in a cluster should be close and they should depict related visual and semantic content. For instance, a cluster associated with the Eiffel Tower should contain photos located around it, with tags related to the tower and, potentially, the surrounding area. For this purpose we developed a framework that can integrate into DBSCAN the three major data information we have: the photos, their location and their tags. We believe the combination of these three feature sets is necessary for consistent photo clusters. We project the photos into a feature space and provide a distance between all pairs under a matrix form. This distance matrix is then fed to DBSCAN which uses it to find neighbors and form the clusters. The mathematical formulation of the distance matrix $D_{ij}$ is given in Equation 1.

$$D_{ij} = \alpha \cdot D_{ij}^{\text{loc}} + \beta \cdot D_{ij}^{\text{img}} + \gamma \cdot D_{ij}^{\text{tag}} \qquad (1)$$

$D_{ij}^{\text{loc}}$ is the geographical distance between photos $i$ and $j$, $D_{ij}^{\text{img}}$ is the cosine dissimilarity between the feature representation of the photos and $D_{ij}^{\text{tag}}$ is the cosine dissimilarity between their bag of words vector. The three matrices are rescaled to $[0,1]$ using min-max normalization. The values $\alpha, \beta, \gamma$ are trade-off weights summing up to one. We use Scikit-Learn implementations of DBSCAN and distance functions (cosine similarity and dissimilarity for computing the distance between photo features vector form).

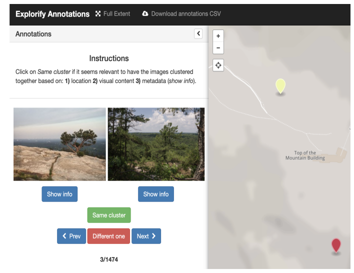



Figure 1: Annotation tool as a javascript application

### *Annotations*

The problem is unsupervised as we don't have information about what the real clusters should be. For validation of the model and the search of optimal hyperparameters we designed an annotation tool, shown in Fig 1.

We asked human annotators (ourselves) to compare two photographs based on their visual content, their location and the metadata such as tags. They have two options: either they click on *same cluster* or *different one*. Clicking the first option creates a label saying that the two photos should be contained into the same cluster, and the second option says they should not. We end up with a csv file having three columns: id of the two photos and the label 1 (same) or 0 (different). Around 250 pairs for each city (Paris and Atlanta) were annotated.

For validation of the DBSCAN clusters, we compute the score metric as follows. We first loop over the annotation files and compute the ratio $s_A$ of pairs correctly clustered. We then loop over the clusters and compute the ratio $s_C$ of pairs inside them that are correctly predicted. The first strategy acts as a recall while the second as a precision. The final score $s$ is the harmonic mean between the ratios that yield a final performance metric in range $[0,1]$.

$$s = 2 \cdot \frac{s_A \cdot s_C}{s_A + s_C} \qquad (2)$$

This metric helps at tuning hyperparameters (DBSCAN's parameters, the weights $\alpha/\beta/\gamma$) with a gridsearch.

### *Photograph Scoring*

Once we have found landmarks and places in a city, it is important to show the places for the best photograph. There are two metrics to help find the best photo: aesthetic assessment and Like-to-View ratio. For the aesthetic score, our approach is based on NIMA architecture [12] for building a neural network to score images on a range of 1 - 10, which are rescaled to a range of 1 - 5 for Explorify to use. This approach matches human assessment with an 81.5% accuracy. While this isn't as good as the 83% result achieved with $MP_{ADA}$[20], idealio's model implementation for NIMA is well supported and compatible with our architecture. Our qualitative survey of the score found notable mismatches (see Experiment 3), but on the whole that scoring fell in line with our perceptions.

In order to mitigate this issue of potential mismatching scores, photographs are subject to client-side evaluation.

On the client-side application, where photographs are ranked and sorted using their aesthetic score, each interaction is an opportunity improvement. Thus, whenever a user views an image, they can optionally give the image a positive, neutral, or negative rating, as detailed in the Front-End Approach. Using this rating, a new score is assigned by providing a weight of .99 to the old score, and a weight of .01 to the new user rating, as follows:

$$\text{new\_score} = .99 * \text{old\_score} + .01 * \begin{cases} 5 & \text{positive} \\ 3 & \text{neutral} \\ 1 & \text{negative} \end{cases} \quad (3)$$

*Data structure*
We use the GeoJSON format for easy use on a map. Each element of the GeoJSON file represents a marker being a photo with its location, all its metadata, its cluster number and its scores. The GeoJSON data structure is stored into a `city.geojson` file sent to the frontend application.

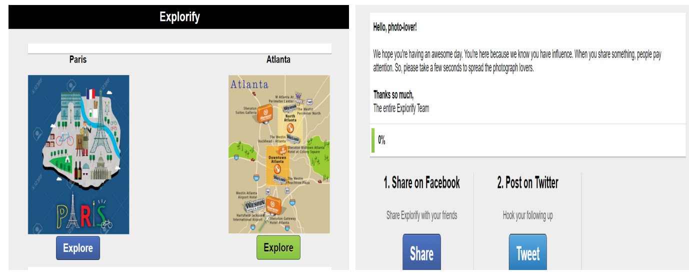
*Scalability and Hosting*
While results for this paper are presented for the Atlanta and Paris flickr image corpora, the clustering and scoring approaches can extend to other cities and regions. Thus to facilitate easy addition of new cities to display, as well as user feedback, we provide a RESTful API to add new cities, modify image properties, and to serve the Explorify Client. To achieve this, the Explorify back-end uses the flask microframework, paired with the pymongo MongoDB driver. MongoDB gives Explorify schema flexibility through its easy to use document store, while flask ensures that the API is simple to modify and deploy.

Explorify is hosted with Heroku, running two workers on the gunicorn WSGI server implementation. Future work can move this API to an auto-scaling solution with AWS Lambda or AWS FarGate.

*Innovation - Back-end Logic*
To the best of our knowledge, we are the first to develop a framework for combining location and other features like image and tags into the input for DBSCAN. Most papers related to the problem only use location for the clustering. We also use state-of-the art deep learning models for the feature representation of photos and the aesthetics quality scoring. We leverage transfer learning from the best models in the different fields for the benefit of our users.

**Approach : Front-end Visualization**
The design and experience are optimized to the user's need of finding the best photographs and best locations to take the photographs. The geographic data structure is encoded in GeoJSON format, read by the web application and is populated on the map. We are building the frontend in Javascript using Mapbox and jQuery libraries. We will also be making calls to the Flickr API to display photographs, and will explore Google Map APIs to redirect users for navigation to specific places.

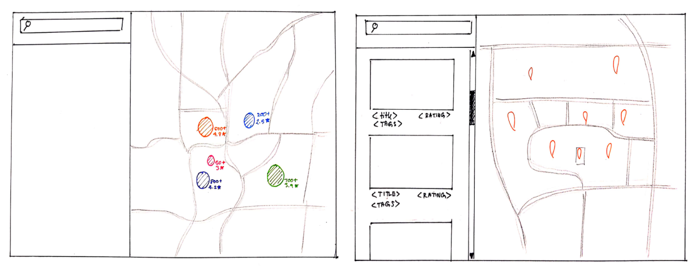
*Landing Page*
A landing page has been designed which will facilitate the user to choose the city of his choice. It has been made quite simple so that the average user can explore the city of his choice without facing any complication. The landing page has two portions. Using upper part of landing page as depicted in figure 2 (a) , the user can use the buttons "Explore" to go to Atlanta or Paris web page. The bottom part of the landing page as depicted in Figure 2 (b) helps the user to share the project/page on the Facebook and twitter.




(a) (b)

Figure 2: Landing Page for the Project. Figure (a) depicts the upper part of landing page helps user to explore Atlanta and Paris. Figure (b) assists the user to share the project on social media.

*Designs and Wireframes*
To service our key features, wireframes of designs and visualizations were designed to address our user needs. The designs were made from a usability as well as feasibility of implementation standpoint. Fig 3 depicts the wireframes.



(a) (b)

Figure 3: Wireframes of user interface

*Read and Visualize data*
Data generated from the back-end in the form of GeoJSON is ingested using our web application. Based on clustering criteria determined by the back-end logic, the clusters and photographs are marked and visualized on the map interface.

*User Interaction: Navigation, Search*
After being able to visualize the clusters on the map, we wanted a user to be able to interact with our interface to gain the information that they need. Key functionalities implemented were the ability for clustering to be responsive to zooming and panning, clicking on clusters revealing photograph locations, clicking of a location marker revealing a pop up with photograph and other details, clicking on a photograph in a side panel navigating to the photograph on the map. Upon clicking a particular location marker, the photograph is displayed in a popup with associated details like the tags and score, as described in the next section. Apart from just being able to explore through navigation, we want the user to

be able to search on our application. We look to implement a search functionality that would search over tags generated for photographs and title. Searching using the search bar narrows down the photographs displayed in the side panel. The images displayed were also provide with the Aesthetic score. The sort functionality lets a user sort images in ascending or descending order based on the aesthetic score.

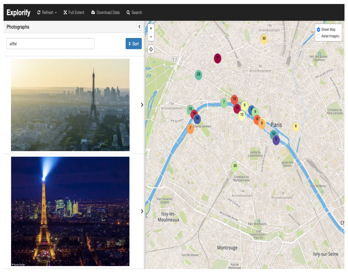



Figure 4: Navigation via user interaction.

*User Interaction: Explore Marker*

When users click markers to explore, Explorify brings up an interactive popup that offers the user a chance to view details, share the image, or see its location on google maps, as shown in Fig 5.

This popup provides an option for user feedback on the image quality. This is vital to ensuring that scores align with overall user preferences, rather than the population of annotators used in dataset collection. Feedback is sent to back-end for processing.

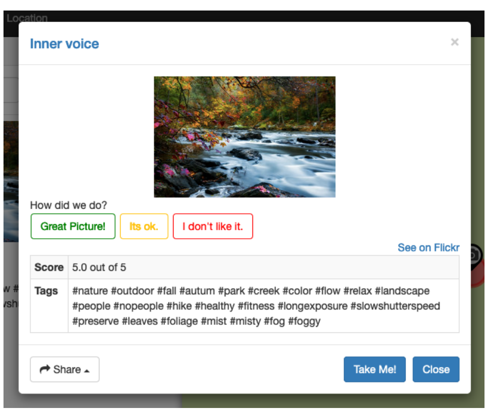
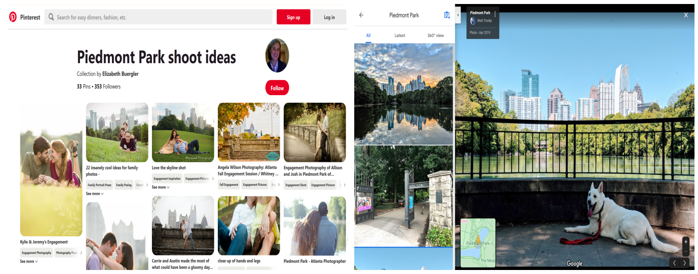



Figure 5: This popup is displayed when exploring an image.

*Innovation*

Current solutions like websites and blogs such as Pinterest Fig 6(a) and Medium are hard to find and do not provide helpful location information to click similar photographs. Map applications like Google Maps Fig 6(b) provide photographs alongside a lot of other cluttered information, but no specific location information. These solutions do not let a user search for particular types of photographs at a location such as 'sunset' or 'skylines'. This is addressed in Explorify through its photograph-driven navigational features and enriched tag data. Existing solutions do not make an effort to score photographs or provide options to sort based on good photographs. With the aesthetic score provided by our application, users have the ability to find high quality photograph ideas thus enhancing the exploring experience.




(a) (b)

Figure 6: Existing Solutions: Pinterest,Google maps

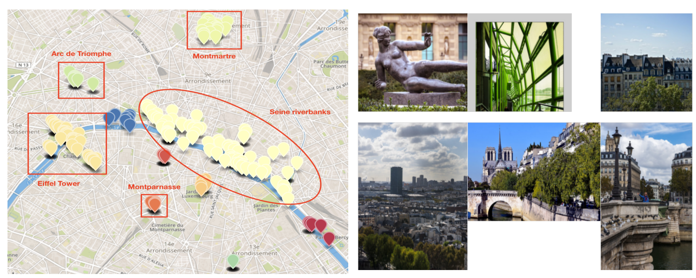
## EXPERIMENTS/EVALUATION

### Experiment 1: Location-based photograph clustering

**Question answered:**Is the data we are using from Flickr of required fidelity and value?

**Approach:** As a baseline, we clustered images only using their location data (latitude, longitude). Physically close photo spots should then be clustered together. We used Paris for experimentation but results can be reproduced to any city.

**Conclusion:**




(a) (b)

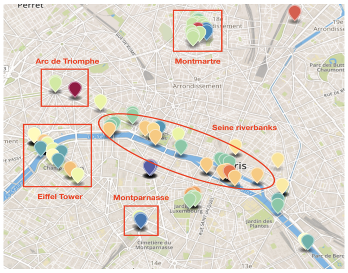
Figure 7: Clustering using location feature

The results for Paris are shown in Fig 7(a). Visually on the map it is very consistent. Famous points of interest for photography are correctly identified, such as the Eiffel Tower area, the Arc de Triomphe and also the famous Seine riverbanks. Note that the granularity of the clustering can be adjusted via the `eps` and `min_samples` parameters. However, looking at the various pictures inside the clusters, we find that they might not always be similar. An example is given in Fig 7(b). In this same cluster corresponding to the Seine riverbanks, we find oddities: indoors and outdoors, landscapes, statues, bridges, buildings. Nevertheless, the location-based clustering provides a very powerful result for identifying key areas in the city.

### Experiment 2: Multi-features photograph clustering

**Question answered:** Do the images and the tags help improve the consistency of clusters?

**Approach:** Now we consider more than the location for the clustering. We feed to DBSCAN a weighted combination of the locations, the photos, and their tags. We used $\alpha = \beta = \gamma = 1/3$, in other words an equal contribution for each component




(a)

(b)

(c)

Figure 8: Clustering using multiple features for the distance between photos.

**Conclusion:** First, the map shown in Fig 8(a) is visually less clear than previously since the location is no longer the only component considered by DBSCAN. Clusters are more spread out. Second, we notice more consistency in the photos within the same cluster. We present in Fig 8(b) the Seine riverbanks to contrast with previously. The cluster seems to represent less diversity of content. We no longer have indoor photos nor statues but instead a majority of views from the river. Lastly, the model is able to cluster photos with the same semantic content even if they are visually diverse. In Fig 8(c), we found a cluster of the Eiffel Tower cropped differently, almost occluded by a statue and in different color sets. This experiment let us believe that the contribution of photos and tags actually help.

We later performed a gridsearch on the values of $\alpha/\beta/\gamma$ and DBSCAN parameters (`eps` and `min_samples`). For each combination we evaluated our custom clustering scoring metric defined in section 4.1.4. The optimal balance of the weights are shown in table 1. As we can see, the combination of all features performs best compared to a single feature configuration.

| $\alpha$ | $\beta$ | $\gamma$ | Score |
|---|---|---|---|
| 0.5 | 0.2 | 0.3 | **0.813** |
| 0.0 | 0.0 | 1.0 | 0.804 |
| 0.0 | 1.0 | 0.0 | 0.669 |
| 1.0 | 0.0 | 0.0 | 0.631 |

Table 1: Clustering scores after a gridsearch on Paris data. $\alpha$ is for the location, $\beta$ for images and $\gamma$ for tags.

### Experiment 3: Aesthetic Scoring Qualitative Assessment

**Question answered:** How well does the aesthetic Scoring match with our qualitative assessment of images?

**Approach:** This work is a validation of the NIMA aesthetic scoring model. We selected images of Atlanta and evaluated whether the relative scoring made sense. Below Fig 9 is a representative sample of 3 "beautiful" images (top row) as well as 3 "unpleasing"(bottom row).

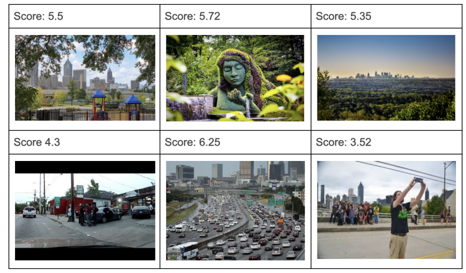



Figure 9

**Conclusion:** Scores typically fall withing a range of 3-7, which is reasonable given that few images are either horrible or beautiful to everyone. On the whole, relative scoring are good, an anomaly of note is high ranking for the image of a highway.

An improvement upon these results is difficult to achieve, given the available annotated datasets, so to mitigate potential score issues, a rating mechanism has been added to the platform.

### Experiment 4: Usability testing of Web Application

**Question answered:** Is the user experience intuitive for our target user? Does information displayed provide value to user of application?

**Approach:** Users were made to interact with and explore the application. They were asked to complete certain tasks to evaluate design decisions and website layout.

**Conclusion:** The layout and designs were intuitive for users as they were able to successfully carry out tasks such as exploring locations on the map, sorting images by score, and using photographs and the search bar to find locations of interest. Users were delighted at the value proposition of being suggested high quality photograph ideas, exact location to get the best views, as well as being able to navigate there. Feedback regarding the size of photographs being displayed was implemented as users preferred to see bigger sizes photographs for better visual experience. The decision to display tag information underneath photographs instead of the photo title was validated as users were not able to derive value from titles which were often generic, unhelpful and non-descriptive.

## CONCLUSION

Using commonly available tools and publicly available data, we were able to build an application to solve a unique problem and provide value to users. Our unique approach of clustering images based on multiple features apart from just location, as well as scoring images to gauge quality elevates the value provided to users. The user interface built was also intuitive and novel compared to existing solutions. Our implementation method is also quite scalable. While we have focused on just showing photographs for only two cities, Atlanta and Paris, the coverage can be scaled to include many more cities in the future.

Thinking ahead in terms of expanding our app to cover a lot more cities, we think there can be some improvements. Currently tuning the hyperparameters for sensible clustering of photographs is manually done because different cities have attractions spread out very differently - Atlanta was quite spaced out whereas Paris was quite dense. We think that factoring in densities of cities can be a novel parameter to explore for sensible clustering of photographs. Another area of improvement is to enable the search bar to be able to search by location and addresses of the exact photograph location.

## DISTRIBUTION OF TEAM EFFORT

The four phases the team set for itself were (1) Data collection and cleaning (2) Building Geoclusters and extracting areas of interest (3) Building interactive WebApp to show map clusters (4) Build search query capabilities.

**Sylvain**: Explored the Flickr API. Designed backend architecture. Clustering algorithm and optimization. Report content for Proposed Method and Experiments.

**Christopher**: Initially worked on the download pipeline and scoring. Implemented different scoring models, integrating "NIMA" image scoring neural network as a part of dataset generation. Chris added user feedback to the client-side application, moved Explorify implementation to Flask framework, moved the data from local JSON to MongoDB, and managed deployment onto Heroku.

**Shafaat**: Designing of landing page, concept of overall back-end logic, reviewed validated the back-end proposed method, user testing, progress and final project report.

**Dongsuk**: Visualized related images based on cluster, finalized the landing page, hashtag visualization, integrated Google Map feature, polished overall UI, enhanced navigational features of clusters in user interface, and reviewed front-end logic. Worked on Final Report.

**Anish**: Populating the photographs and clusters in the web application using javascript and jquery. Integrated Mapbox library and used the inbuilt functions to make the user experience more intuitive. Worked on the side bar and the pop up window that appears when clicked on the photo location. Final Report Template.

**George**: Developed wireframes and designs for visualizations. Navigational aspects of web applications including clustering visualizations using Mapbox, Javascript, jQuery. Search and sort functionality over tag an scores. User testing. Final Report.

## REFERENCES


[1] Imran Memon, Ling Chen, Abdul Majid, Mingqi Lv, Ibrar Hussain, and Gencai Chen. Travel recommendation using geo-tagged photos in social media for tourist. *Wireless Personal Communications*, 80(4):1347–1362, 2015.

[2] Xia Peng and Zhou Huang. A novel popular tourist attraction discovering approach based on geo-tagged social media big data. *ISPRS International Journal of Geo-Information*, 6:216, 07 2017.

[3] Ian H Witten and Eibe Frank. Data mining: practical machine learning tools and techniques with java implementations. *Acm Sigmod Record*, 31(1):76–77, 2002.

[4] Evaggelos Spyrou, Michalis Korakakis, Vasileios Charalampidis, Apostolos Psallas, and Phivos Mylonas. A geo-clustering approach for the detection of areas-of-interest and their underlying semantics. *Algorithms*, 10:35, 03 2017.

[5] Yingjie Hu, Song Gao, Krzysztof Janowicz, Bailang Yu, Wenwen Li, and Sathya Prasad. Extracting and understanding urban areas of interest using geotagged photos. *Computers, Environment and Urban Systems*, 54:240–254, 2015.

[6] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 2004.

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

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

[11] Ning Ma, Alexey Volkov, Aleksandr Livshits, Pawel Pietrusinski, Houdong Hu, and Mark Bolin. An universal image attractiveness ranking framework, 04 2018.

[12] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, Aug 2018.

[13] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 185–201, Cham, 2018. Springer International Publishing.

[14] George Miller, R. Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to wordnet: An on-line lexical database*. 3, 01 1991.

[15] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing, 2009.

[16] Andreas Veit, Maximilian Nickel, Serge Belongie, and Laurens Maaten. Separating self-expression and visual content in hashtag supervision. 11 2017.

[17] Yan-Tao Zheng, Zheng-Jun Zha, and Tat-Seng Chua. Mining travel patterns from geotagged photos. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–18, 2012.

[18] Huy Quan Vu, Gang Li, Rob Law, and Ben Haobin Ye. Exploring the travel behaviors of inbound tourists to hong kong using geotagged photos. *Tourism Management*, 46:222–232, 2015.

[19] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.

[20] Kekai Sheng, Weiming Dong, Chongyang Ma, Xing Mei, Feiyue Huang, and Bao-Gang Hu. Attention-based multi-patch aggregation for image aesthetic assessment. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 879–886, 2018.