

M.SC. ENGG. THESIS

# Determining the aesthetic rating and weather of a location from Flickr photos and metadata

by

Ch. Md. Rakin Haider

Submitted to

Department of Computer Science and Engineering

in partial fulfillment of the requirements for the degree of  
Master of Science in Computer Science and Engineering



Department of Computer Science and Engineering


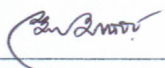

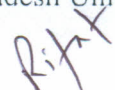

Bangladesh University of Engineering and Technology (BUET)

Dhaka 1000

January 2019

The thesis titled "Determining the aesthetic rating and weather of a location from Flickr photos and metadata", submitted by Ch. Md. Rakin Haider, Roll No. **0416052013 P**, Session April 2016, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on 29th January, 2019.

## Board of Examiners

1.   
\_\_\_\_\_  
Dr. Mohammed Eunus Ali  
Professor  
Department of Computer Science and Engineering  
Bangladesh University of Engineering and Technology, Dhaka.  
Chairman  
(Supervisor)
2.   
\_\_\_\_\_  
Dr. Md. Mostofa Akbar  
Head and Professor  
Department of Computer Science and Engineering  
Bangladesh University of Engineering and Technology, Dhaka.  
Member  
(Ex-Officio)
3.   
\_\_\_\_\_  
Dr. M. Sohel Rahman  
Professor  
Department of Computer Science and Engineering  
Bangladesh University of Engineering and Technology, Dhaka.  
Member
4.   
\_\_\_\_\_  
Dr. Rifat Shahriyar  
Assistant Professor  
Department of Computer Science and Engineering  
Bangladesh University of Engineering and Technology, Dhaka.  
Member
5.   
\_\_\_\_\_  
Dr. Salekul Islam  
Professor and Head  
Department of Computer Science and Engineering (CSE)  
United International University (UIU), Dhaka.  
Member  
(External)

## Candidate's Declaration

This is hereby declared that the work titled "Determining the aesthetic rating and weather of a location from Flickr photos and metadata" is the outcome of research carried out by me under the supervision of Dr. Mohammed Eunus Ali, in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

*CH. MD. Rakin Haider*

---

Ch. Md. Rakin Haider

Candidate

# Acknowledgment

Foremost, I am thankful to the Almighty for his blessings for the successful completion of my thesis. I would like to express my heartiest gratitude, profound indebtedness, and deep respect to my supervisor, Dr. Mohammed Eunos Ali, Professor, Dept. of CSE, BUET, Dhaka, Bangladesh, for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advices throughout the study was of great help in completing this thesis.

I would also want to thank the members of my thesis committee for their valuable suggestions. I thank Dr. Md. Mostofa Akbar, Dr. M. Sohel Rahman, Dr. Rifat Shahriyar, and specially the external member Dr. Salekul Islam.

I am especially grateful to Department of Computer Science and Engineering (CSE) of Bangladesh University of Engineering and Technology (BUET) for providing their support during the thesis work. My sincere thanks goes to CSE Office staffs for providing logistic support to me to successfully complete the thesis work.

Finally, I would like to thank my family, my friends, and all of those who supported me with their appreciable assistance, patience, and suggestions during the course of my thesis.

# Abstract

The last decade has witnessed an unprecedented rise in the popularity of content sharing networks such as Flickr and Twitter. Shared photos are usually accompanied by metadata such as geo-location, timestamp and tags. These photos contain a digital representation of locations and they convey human behavior patterns, photo trails and tour summaries. The availability of such geographic information in the form of multimedia contents has given rise to interesting applications such as recommendation system, point-of-interest mining and tour planning system, user gender and home location prediction system and event recommendation systems. In this work, we have proposed a method to determine the aesthetic rating of a location and weather condition of an image from social metadata of Flickr photos and content analysis of Flickr images respectively.

The aesthetic rating of a location is the evaluation of aesthetic quality of that location. Tourists, artists and urban planners often seek to rate each location by their aesthetics. Popular recommendation websites such as *TripAdvisor* generate a relative rating of the locations to provide recommendations to its users about possible locations to visit. However, such rankings are highly dependent on user contributions. Therefore, in this work, we have proposed a method to generate aesthetic rating of a location using social metadata of user captured and shared Flickr photos. A number of empirical features have been defined and computed from the social metadata of the Flickr photos available at each location. Using these features numerous classifiers have been trained and our classifiers have been able to achieve notable accuracy.

On a different note, weather condition detection and tracking are in practice for a long time. However, most weather detection technologies rely highly on powerful hardware technologies and expertise of weather specialists. With the development of computer vision technologies

several attempts have been taken to recognize weather conditions from images. In this work, we leveraged the availability of user tagged images in Flickr to generate a image dataset with weather condition annotations. Using the dataset we have proposed, deep convolution network based solutions to detect weather conditions of a location from user tagged Flickr images.

We conduct comprehensive empirical analysis to investigate the performance of our proposed algorithms. We have gathered social metadata of Flickr photos of the locations of two major tourist destinations i.e. Rome and Paris. Our classifiers obtained about 80% accuracy in correctly predicting the aesthetic ratings of locations in Rome and achieved about 71% accuracy on Paris dataset. On the other hand we have considered four weather conditions in our weather detection task, namely, sunny, cloudy, rainy and snowy. We have trained several neural networks by varying hyper-parameters. Additionally, we have also applied transfer learning with popular pre-trained neural networks such as VGG16, InceptionV3, InceptionResnetV2 etc. Our classifiers have reported as much as 60% accuracy on our scrapped dataset.

# Contents

<i>Board of Examiners</i>	1
<i>Candidate's Declaration</i>	2
<i>Acknowledgment</i>	3
<i>Abstract</i>	4
<b>Contents</b>	<b>6</b>
<b>List of Figures</b>	<b>8</b>
<b>List of Tables</b>	<b>11</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Applications . . . . .	3
1.2 Research Objectives . . . . .	4
1.3 Overview of Methodology . . . . .	5
1.3.1 Aesthetic Rating Prediction . . . . .	5
1.3.2 Weather Condition Recognition . . . . .	6
1.4 Contributions . . . . .	6
1.5 Organization . . . . .	7
<b>2 State of The Art</b>	<b>9</b>
2.1 Flickr Data Mining . . . . .	9

2.2	Assigning Aesthetic Scores to Locations . . . . .	11
2.3	Weather Condition Detection From Images . . . . .	12
<b>3</b>	<b>Problem Formulation</b>	<b>14</b>
3.1	Aesthetic Rating Prediction . . . . .	14
3.2	Weather Condition Recognition . . . . .	15
<b>4</b>	<b>Aesthetic Rating Prediction</b>	<b>17</b>
4.1	Available Datasets . . . . .	17
4.2	Dataset Generation . . . . .	19
4.3	Feature Extraction . . . . .	21
4.4	Classifiers . . . . .	23
4.5	Oversampling . . . . .	32
4.6	Ensemble Learning . . . . .	35
4.7	Summary . . . . .	36
<b>5</b>	<b>Weather Condition Detection</b>	<b>39</b>
5.1	Available Datasets . . . . .	39
5.2	Dataset Generation . . . . .	41
5.3	Background . . . . .	42
5.3.1	Convolution Neural Network(ConvNet) . . . . .	42
5.3.2	Transfer Learning and Fine Tuning . . . . .	43
5.4	Training Classifiers . . . . .	45
5.4.1	Hyper-parameter Tuning . . . . .	47
5.5	Transfer Learning . . . . .	47
5.6	Summary . . . . .	59
<b>6</b>	<b>Conclusion</b>	<b>68</b>
	<b>Bibliography</b>	<b>70</b>



# List of Figures

4.1	Number of locations in each class of Rome dataset . . . . .	21
4.2	Number of locations in each class of Paris dataset . . . . .	22
4.3	Number of locations in each category of Rome dataset . . . . .	23
4.4	Number of locations in each category of Paris dataset . . . . .	24
4.5	Distributions of attributes in Rome Dataset . . . . .	24
4.6	Distributions of attributes in Rome Dataset . . . . .	25
4.7	Distributions of attributes in Rome Dataset . . . . .	26
4.8	Distributions of attributes in Rome Dataset . . . . .	27
4.9	Distributions of attributes in Paris Dataset . . . . .	27
4.10	Distributions of attributes in Paris Dataset . . . . .	28
4.11	Distributions of attributes in Paris Dataset . . . . .	28
4.12	Distributions of attributes in Paris Dataset . . . . .	29
4.13	Number of locations in each class of oversampled Rome dataset . . . . .	33
4.14	Number of locations in each class of oversampled Paris dataset . . . . .	34
5.1	VGG-16 Architecture . . . . .	44
5.2	Inception module from [1] . . . . .	44
5.3	InceptionV3 architecture in Tensorflow . . . . .	45
5.4	Alexnet Architecture as presented in [2] . . . . .	46
5.5	Learning curves for AlexNet, learning rate = 0.01, epochs = 50, batch size = 20 and different optimizers. . . . .	48

5.6	Learning curves for AlexNet with ADAM, epochs = 50, batch size = 20 and different learning rates. . . . .	49
5.7	Learning curves for AlexNet with ADAM, learning rate = 0.01, epochs = 50 and different batch sizes. . . . .	50
5.8	Learning curves for AlexNet with ADAM, learning rate = 0.01, batch size = 20 and different number of epochs . . . . .	51
5.9	Learning curves for VGG, learning rate = 0.01, epochs = 500, batch size = 20 and different optimizers. . . . .	53
5.10	Learning curves for Inception-ResnetV2, learning rate = 0.01, epochs = 500, batch size = 20 and different optimizers. . . . .	54
5.11	Learning curves for InceptionV3, learning rate = 0.01, epochs = 500, batch size = 20 and different optimizers. . . . .	55
5.12	Learning curves for VGG with ADAM, learning rate = 0.01, epochs = 2000, batch size = 20 . . . . .	57
5.13	Learning curves for Inception-ResnetV2 with ADAM, learning rate = 0.01, epochs = 1000, batch size = 20 . . . . .	57
5.14	Learning curves for InceptionV3 with ADAM, learning rate = 0.01, epochs = 1000, batch size = 20 . . . . .	59
5.15	Learning curves for VGG with ADAM, learning rate = 0.01, epochs = 500 and different batch sizes. . . . .	60
5.16	Learning curves for Inception-ResnetV2 with ADAM, learning rate = 0.01, epochs = 500 and different batch sizes. . . . .	61
5.17	Learning curves for InceptionV3 with ADAM, learning rate = 0.01, epochs = 500 and different batch sizes. . . . .	62
5.18	Learning curves for VGG with ADAM, epochs = 500, batch size = 20 and different learning rates. . . . .	63
5.19	Learning curves for Inception-ResnetV2 with ADAM, epochs = 500, batch size = 20 and different learning rates. . . . .	64

5.20 Learning curves for InceptionV3 with ADAM, epochs = 500, batch size = 20 and  
different learning rates. . . . . 65

# List of Tables

4.1	List of derived features. . . . .	23
4.2	Merit of each attribute obtained from different attribute selection algorithms . . .	25
4.3	Merit of each attribute obtained from different attribute selection algorithms . . .	26
4.4	Accuracy, Precision and Recall of the classifiers trained on Rome Dataset . . . .	31
4.5	Accuracy, Precision and Recall of the classifiers trained on Paris Dataset . . . . .	32
4.6	Accuracy, Precision and Recall of the classifiers trained on oversampled Rome Dataset . . . . .	35
4.7	Accuracy, Precision and Recall of the classifiers trained on oversampled Paris Dataset . . . . .	36
4.8	Ensembled Accuracy, Precision and Recall of the classifiers trained on oversampled Rome Dataset . . . . .	37
4.9	Ensembled Accuracy, Precision and Recall of the classifiers trained on oversampled Paris Dataset . . . . .	38
5.1	AlexNet hyper-parameters and their values . . . . .	47
5.2	Hyper-parameters used in transfer learning and their values . . . . .	56
5.3	Classifier Accuracy for various Optimizer of the classifier VGG . . . . .	56
5.4	Classifier Accuracy for various Optimizer of the classifier Inception-ResnetV2 . .	58
5.5	Classifier Accuracy for various Optimizer of the classifier InceptionV3 . . . . .	58
5.6	Classifier Accuracy for various batch size of the classifier VGG . . . . .	59
5.7	Classifier Accuracy for various batch size of the classifier Inception-ResnetV2 . .	66
5.8	Classifier Accuracy for various batch size of the classifier InceptionV3 . . . . .	66

5.9	Classifier Accuracy for various learning rate of the classifier VGG . . . . .	67
5.10	Classifier Accuracy for various learning rate of the classifier Inception-ResnetV2	67
5.11	Classifier Accuracy for various learning rate of the classifier InceptionV3 . . . .	67

# Chapter 1

## Introduction

In recent era, the proliferation of smart phones and the availability of the Internet have resulted in an unprecedented rise in the popularity of content sharing sites such as Flickr, Instagram, Youtube, Pinterest, etc. One of the most popular form of user generated multimedia content is images or photos. Since capturing high-quality images have been facilitated by the increase of camera availability in smart phones, more and more users are encouraged to capture and share the photos of their surroundings. Therefore, popular content sharing sites such as Flickr, Instagram, Twitter etc. have been able to accumulate a large number of user generated photos. In Flickr, alone there are about 8 billion existing photos [3, 4] and it is reported that Flickr receives 3.5 million uploads every day [3]. Most of these content sharing services support storage of various meta-data associated with each image. For example, each Flickr photos contains meta-data such as time of the photo taken, time of the photo uploaded, properties of the camera with which the photo is captured, textual description of the photo etc. In addition to that most hand-held camera devices are now GPS-enabled. Therefore, most of the uploaded images are gps-tagged. Finally, these content sharing networks doesn't work only as a storage facility for the users to upload their images, but it also provides social media like features to them so that they can interact with each other and their contents. For example, in Flickr a user can view other's uploaded images, add these images to his favorite, provide his opinions about the image as a comment, follow other users' activity, form groups to share photos of similar category etc. These interactions are also tracked with some additional meta-data such as number of views,

number of time an image is added to the favorite, number of comments etc.

This large amount of geo-tagged photos gives us a digital representation of a specific geo-location. It also contains patterns of human behavior, photo trails, transit time between areas of a city etc. Researchers have realized that appropriate studies on Flickr images and their meta-data can lead to unveiling newer insights. Among the fields of Flickr research, the most popular ones are multimedia content retrieval, scene understanding, touristic travel applications, point-of-interest mining, human activity mining etc. Several other aspects such as political analysis, event detection, photo attractiveness assessment and landmark summarization are also explored in previous studies.

In this work, we concentrate our focus on a task related to point-of-interest and a scene understanding task. Our first problem is predicting aesthetic rating of a location from social metadata of Flickr images. With the surge of travel recommendation services such as TripAdvisor, Expedia etc, location ratings based on their aesthetics have become commonly available. We want to exploit Flickr data to generate or predict aesthetic rating of locations. Several studies have been conducted to measure the scenic beauty of a geo-location. *Nirmala et al.* [5] and *Bergen et al.* [6] attempted to estimate scenic beauty of forestry using images. The scenic beauty estimation (SBE) method of [6] was later used to assess visual beauty of landscapes [7]. Studies have also been conducted to assess aesthetic beauty of waterscapes using SBE. In [8] the authors had gathered crowd-sourced SBE of woodland landscapes and performed correlation study with landscape image properties to validate their study. *Quercia et al.* [9] proposed a method to find out a scenic path through a city. Their work covers both crowd-sourcing approach as well as automated approach using Flickr metadata. Although their automated approach uses Flickr social meta-data such as photo density, number of views etc., the validation of their result depends largely on crowd-sourced scenic scores. Additionally, the Flickr-generated routes failed to achieve higher scenic scores from most of their voters. Finally, travel recommendation sites like TripAdvisor have developed a method to generate these rating through crowd-sourced tourist reviews.

On the other hand, our second problem is understanding weather conditions from single Flickr image. Weather detection from fixed viewpoint images has been studied for quite a long

time [10, 11]. The pioneer work [12] on weather detection from dynamic viewpoint images worked with only two weather conditions. Their work has been extended in the works of [13–15]. However, they had to employ human helpers for data labeling. In our problem, we are trying to exploit textual tags of Flickr photos without any human curation to develop a method to detect weather conditions from images.

The rest of the chapter is organized as follows. Section 1.1 briefly discuss about the motivation of our proposed work with its application. Section 1.2 outlines the objectives of our research. Section 1.3 projects our research challenges and solution overview. Then Section 1.4 highlights the contributions of our thesis. Finally, an organization of the remaining chapters are given in Section 1.5.

## 1.1 Motivation and Applications

Nowadays availability of GPS-enabled camera phones has increased sharing of geo-tagged photos. Popular content sharing sites such as Flickr, Instagram, Twitter have already accumulated a large number of user generated photos. In Flickr, alone there are about 8 billion existing photos and it is reported that Flickr receives 3.5 million uploads every day. Each of these photos contains meta-data such as gps-location, time of capturing the photo, number of users viewed, number of favorites etc. This large amount of geo-tagged photos not only provides a digital representation of the specific geo-location but also contains patterns of human behavior, photo trails, transit time between areas of a city etc. So, appropriate mining of such geo-tagged contents can result in designing a set of newer applications.

In this work, we have addressed two problems based on Flickr data mining. First, we have proposed a system that can predict the aesthetic score of a location using Flickr social meta-data. Here, Aesthetic rating of a location is the relative rating of a location considering the scenic beauty of that location.

The second problem of our work, is determining the weather condition of a scenario available in geo-tagged Flickr images with the help of its textual tags. In this work, we have concentrated our focus on only four weather conditions, namely, sunny, rainy, cloudy, snowy.



Aesthetic ranking of a location can be of great value towards tour planning, tourism business etc. Tourists can pick the places they want to visit according to their aesthetic ratings. Similarly, automated travel recommendation systems can be developed keeping the aesthetic ratings in mind. On the other hand, weather recognition plays a vital role in many applications in our day to day life. Current technologies of weather condition reporting rely highly on expensive sensor network and human expertise. To sustain such a system huge amount of resources are required to be engaged in it. However, detecting weather conditions from user tagged images offers us a cheaper alternative. Apart from developing a cost effecting method for weather tracking and reporting, detecting weather conditions from images can also be used in advanced applications such as self-driving cars and intelligent weather-based recommendation systems such as weather-based restaurant recommendation.

## 1.2 Research Objectives

From previous discussion we have identified these following objectives of our study:

- Introducing a novel approach to predict aesthetic ratings and weather of locations from photo capturing, sharing and users' interaction patterns.
- Developing a classification model to predict aesthetic ratings of a location from Flickr social metadata.
- Proposing a deep learning based method to determine weather conditions at a certain time of a location from Flickr photos.
- Handling imbalanced dataset to improve accuracy.
- Simulating the proposed approaches and performing extensive experiments on empirically built real-world datasets to evaluate our proposed solution and applying performance improvement techniques.

The possible outcomes of our study are as follows:

- A new classifier to predict the aesthetic rating of a place from Flickr social metadata which will reduce the dependency of crowd-sourced data to comment about a location's aesthetic rating.
- A novel approach to determine the weather of a location from the photos uploaded at a given time.

## 1.3 Overview of Methodology

As discussed earlier, we are focusing on solving two Flickr data mining problems in this thesis work. We refer to them as Aesthetic Rating Prediction problem and Weather Condition Detection problem. Under this section we have discussed about the challenges we faced, and the solution we proposed to handle these challenges.

### 1.3.1 Aesthetic Rating Prediction

In this work, we have assumed that each of the geo-locations belongs to one of several aesthetic classes. Each member of an aesthetic class has the same aesthetic rating. So, the problem has been reduced to correctly predicting the appropriate aesthetic class of a geo-location using the geographical and social metadata available in Flickr. To perform our desired classification, we propose to build numerous decision tree based classifiers. A major challenge in this task is setting up the ground truth for the training phase. Although the attractiveness of a location is subject to individual's taste and view, we can assume that the huge number of reviews and ratings [16], as reported by multiple sources, accumulated by various travel recommendation websites such as TripAdvisor [17], Expedia [18] etc. can reflect the true scenic value of a location from the perspective of majority of users. Another challenge that we had to face was that the popular available Flickr datasets do not focus on the social attributes of the shared images. As a prerequisite of classification step, we had to scrap data from both Flickr and a well-known web-site named TripAdvisor, to accumulate the dataset used in this work. Later, we derived 11 empirical features for each geo-location. We have trained several classifiers and to perform vali-

dation we have used 10-fold cross validation. Finally, we faced performance issues due to dataset imbalance. We handled this challenge by applying state-of-the-art oversampling technique. To improve classification accuracy we have applied ensemble method on each of the classifiers.

### 1.3.2 Weather Condition Recognition

The weather condition recognition task is also considered as a multi-class classification task. Here, it is assumed that there are only four weather conditions. They are sunny, cloudy, rainy and snowy. Similar, to the previous task most of the available image datasets do not contain any weather annotation. More importantly, most weather dataset are built with the help of human helpers and experts. However, our problem wants to focus on uncurated image tags available in Flickr. So, an image dataset is accumulated using the textual weather tags of Flickr photos. Then, we trained a small and less computationally-intensive convolution neural network. To find out the best accuracy we have performed hyper parameter tuning and reported the accuracy for each hyper parameter combination. We have also used popular pre-trained neural networks which were trained on Imagenet [19] and fine-tuned their weights so that the pre-trained models adapt to our classification task.

## 1.4 Contributions

We make the following contributions as listed below:

- i. a. First, we model the aesthetic rating prediction task as a multiclass classification problem. The required assumption for such modeling is that each of the locations is a member of an equally aesthetically rated class. Given a location our classifier will be able to predict the class in which it belongs.
- b. In order to train a classifier, we build a dataset empirically containing the locations of two tourist destination, namely, Rome and Paris. The dataset is tailor made to contain ground truth about aesthetic ratings.

- c. After extracting relevant features multiple variants of decision tree and other classifiers are trained and their performances are analyzed. Additional steps are taken to remove performance deficiency due to imbalanced dataset.
  - d. Finally, ensemble learning technique such as Bagging and Boosting are applied to improve classifier performance.
- ii. a. Second, we develop a method to detect weather of a location from user uploaded Flickr photos. In order to handle this problem, we propose a deep neural network based solution. We assume that weather can be divided into categories such as sunny, rainy etc. We consider each of these categories as an individual class and our classifier should be able to predict in which class a photo belongs to.
- b. In order to obtain the desired classifier, we gather a set of photos from Flickr API taken at a certain location during a specified range of time. We label each photo with historical data obtained from available weather datasets.
- c. Using these photos we train a convolution neural network and analyze classifier performances.
- iii. To validate the effectiveness and accuracy of our proposed methods, we performed extensive experiments using real world data sets by varying different parameters, such as the number of hidden layers in neural network, number of nodes in each layer etc. We will measure the efficiency of each classifier in terms of accuracy, precision and recall.

## 1.5 Organization

Now we outline the organization of this report. First, we discuss some previous works related to our problems in Section 2. Then, we formulate the problem in Section 3. As discussed above, our problem is divided into two parts. We propose solutions to both the *Aesthetic Rating Prediction* problem and the *Weather Condition Recognition* from image problem. In Section 4, we describe the solution by explaining how the dataset for our task is collected, how the related features were derived and the intuition behind each feature, how aesthetic rating classification is

---

done, the issues that was solved to achieve higher classification accuracy and the experimental results of our solution. The solution for weather condition recognition problem is discussed in Section 5. Finally, we make some concluding remarks in Section 6.

# Chapter 2

## State of The Art

The related works we found in literature can be divided into three major categories. The first group of studies as discussed in Section 2.1 focuses on mining Flickr data to extract various patterns such as popular routes, point-of-interest etc. Section 2.2 discusses about the second track of studies which concentrates on measuring aesthetic properties of a location from images. Finally, our literature review also found several related works on weather condition recognition from images. We have discussed such works in Section 2.3.

### 2.1 Flickr Data Mining

The unprecedented increase in the amount of data in the form of images, videos, texts or meta-data accumulated by various social content sharing sites such Flickr, Twitter, Instagram has attracted researchers to conduct data mining studies on them. Flickr alone has accumulated about 8 billion photos each of which has geo-tags, textual tags, timestamps etc. associated with them. Besides an image is also a representation of the surroundings. So appropriate mining can lead to revelation of many seemingly hidden information. Since a large portion of multimedia contents is of touristic nature, research works often focus on finding out tourist behavior, points of interest, occuerd events and fetivals etc. A series of studies [20–23] have been performed to facilitate route planning and route recommendation. In [20] the authors have classified frequent photo trips obtained from Flickr and also detected people’s trip patterns such as duration of stay

at a location, sequence of visited locations throughout a city etc. In [21], the authors have attempted to answer temporal queries such as amount of time spent in a specific point of interest, duration of journey between two geo-locations etc. Popescu et al. [24] have proposed methods to discover user trips from Flickr metadata, find out trip characteristics and classify whether an image contains interior or exterior view. Several others works [25–28] have also focused on recommending personalized tours. Although route recommendation have been studied at length, Quercia et al. [9] argued that a popular route may not be the most pleasant route for a tourist. They proposed a way of recommending emotionally pleasing routes. They considered three characteristics of routes, i.e., beauty, quietness and happy. They divided the space into a grid of equally spaced cells and assigned each cell with a happiness score obtained from crowd-sourced results. Later they presented that Flickr metadata can be used to avoid crowd-sourcing. However their experimental results show that Flickr-generated route based on their approach is not considered as scenic route by the users. Apart from works on route recommendations, point-of-interests (POIs) and event detection from Flickr data have been studied extensively. Ling et al. [29] and Nitta et al. [30] have proposed solutions to find out popular events. They [29, 30] exploited textual tags, temporal information and geo-tags. However, the former suffers performance issue in case of nonperiodic events. In [31], the authors have proposed method to generate POIs with the help of wikipedia. Other works involving POI detection are [4, 32]. In [33] a machine learning based approach is proposed to predict the popularity of a Flickr image using both image qualities and social attributes. They have used three layer of features. The first layer consists of features generated from colors of the image. As low and high level computer vision features they have used gist, texture, color-patches, gradient and deep learning based object detection filters. Since popularity of an image can be easily quantified by the number of views or number of shares of an image, their classification can easily obtain labeled data to train classifiers. Apart from these, numerous studies [34, 35] have been conducted on content based image retrieval. It should be noted that semantic contents of an image doesn't change with users' perspectives. Most of the Flickr data mining studies where focused on either analyzing the contents of the photos or performing analysis of meta-data such as geo-tags, temporal information, user tags and automatically generated tags etc. Another set of meta-data is the social meta-data of Flickr photos. As social

meta-data we can consider the number of views of each photo, number of people who have added the photo as favorite, number of comments in the photo, content of each comment, popularity of user etc. In other words, the statistics that is generated due to users' interaction with the photo or the owner of the photo can be called social meta-data. In spite of these meta-data being largely available are mostly overlooked by the recent studies.

## 2.2 Assigning Aesthetic Scores to Locations

Visual attractiveness is a highly subjective concept that varies from person to person. But researchers have been trying to quantify beauty measures. The authors of [36] have conducted a survey oriented research to find out scenic properties of landscapes. In [5, 6] the authors have attempted to measure scenic beauty of images in the context of forests. Their goal was to develop an automated system to measure scenic beauty of forestry images. They have used color histogram and edge detection to find out scenic beauty estimation(SBE) of images. Bulut et al. [7] have extended the SBE approach to determine landscape and waterscape beauty scores. A significant drawback of such approaches is that the experiments are conducted on specially taken photos such as photos taken from satellite or cameras placed on special places of a forest etc. to carry out their study.

Another similar work of measuring aesthetic scores of images is [37] where they have defined a machine learning based approach to classify images based on their aesthetic qualities as well as to assign each image a scoring based on its aesthetics. One major drawback of their approach was the use of Photo.net as there source of images, since Photo.net doesn't provide any API to obtain their photos. They used Photo.net because it offers an aesthetic and originality rating of each image and thus making the generation of labeled data easier. As their features, they have considered color tones and saturation, object segment of images, exposure of light and colorfulness, golden ratio approximation, wavelet-based textures, region composition, size and aspect ratios, shape convexity etc.

Few meta-heuristic based works [38, 39] on finding out scenic paths have also been carried out. These works consider the work of [9] as there baseline scoring technique and preform



optimization on arc-orienting problem. But these solutions imposes a budget constraints on finding scenic routes. Another recent work [40] suggests a way to recommend tours based on user interests from his/her visit history

## 2.3 Weather Condition Detection From Images

Although weather condition detection from images can have an vital role in designing numerous applications, studies in this area are still limited. Primitive works usually focused on detecting weather conditions from static images obtained from surveillance cameras, on vehicle cameras etc. In [10, 11], the authors focused on weather detection from vehicle-mounted camera images. However, research works have also been conducted on detecting weather conditions from images that are not captured from fixed-point cameras. One of the first work on detecting weather conditions from dynamic images is conducted by [12]. In this work, they focused on detection of only two types of weather conditions, namely, sunny and cloudy. They have designed features to detect 5 major weather cues. They are sky, shadow, reflection, contrast and haze. Based on these weather cues they have derived feature sets and considering the presence or absence of these weather cues an algorithm is proposed that uses collaborative filtering. The work in [13] designed features for sunny, rainy, snowy and haze weathers and applied multiple kernel learning method to detect multiple weather. Similar work has been performed in [14]. Since neural networks have been performing exceptionally well in image recognition and classification, some neural network based works have also been performed. Such as [15] has applied AlexNet to perform the two (sunny, cloudy) weather condition classification of images and demonstrated better performance of neural networks in weather detection. Finally, [41] have proposed using extra segmentation masks of weather cues to achieve better performance in detecting weather conditions.

However, each of these works on detecting weather conditions from images had to use some image dataset to work on. The authors of [12] have gathered a rather small set of images from the SUN dataset [42], LaBelme dataset [43] and Flickr. However, to label each photos they had to employ helpers who collected and labeled about 10K unambiguous images. Similar ap-

proach was followed by the authors of [13]. The dataset in [12] was later adopted by the authors of [15,41]. To address the issues of unavailability of large datasets and the requirement of manual assistance to provide weather condition ground truths, the authors in [44] have created a dataset from Flickr and provided weather condition labels on each of them with the help of a popular web-based weather platform named Weather Underground. To ensure the collection of unambiguous images they have filtered their gathered images by using sky-region detection and outdoor image detection methods.

However, in this work we have addressed the issue of detecting weather conditions from user generated images of Flickr and the weather condition labels provided by the content uploaders. Although similar to previous works, the higher level of ambiguity in user labels and the number of weather conditions in consideration makes our task more challenging.

# Chapter 3

## Problem Formulation

The goal of this work is to design two intelligent systems for aesthetic rating prediction and weather condition recognition task. Aesthetic rating prediction as discussed in Section 3.1 can be defined as building a classifier to predict aesthetic class of a location with the help of social metadata of Flickr images and location aesthetic rating ground truths from TripAdvisor. On the other hand, Section 3.2 discusses about our problem of building a system to detect weather conditions of an image with the help of Flickr images and its wild tags. .

### 3.1 Aesthetic Rating Prediction

In order to understand this problem and the solutions, we first need to understand how we have defined the aesthetic rating of a location. Aesthetic rating of a location can be defined as follows.

***Aesthetic Rating:** Aesthetic rating of a location is the relative ranking of the locations by comparing their aesthetic beauty. In others words, a location, which appears to be more aesthetic to a visitor, receives an aesthetic rating that is higher than that of a location which appears to be less aesthetic to the visitor.*

Although the attractiveness of a location is subject to individual's taste and viewpoint, we assume that if a huge number of reviews and ratings can be collected from visitors then we can acquire an estimation about the true scenic value of a location. Several travel recommen-

dation services such as TripAdvisor [16] have reported to have accumulated significant amount of reviews and ratings. They provide an estimation of the aesthetic rating of the location by considering user ratings. In this work, we have considered TripAdvisor as the source of our ground truth. TripAdvisor provides aesthetic rating of each location where the aesthetic ratings are discrete values ranging from 0-5 with interval of 0.5.

***Aesthetic Class:** In this work, we define aesthetic class, as a set of locations that have the same aesthetic rating. Therefore, according to TripAdvisor there are 11 aesthetic classes and each location belongs to one and only one aesthetic class.*

To proceed with our problem formulation, we need the definition of social metadata of Flickr images. The definition of social metadata according to our work is as follow.

***Social Metadata of Flickr Images:** Flickr is a well-known content sharing site. However, Flickr can also act as a social network where users can interact with each other and with their contents. These interactions are viewing each others photos, putting a comment under a photo and adding other user's photos as favorite. In this work, the metadata which are generated as a resultant of Flickr user interaction among themselves, are named social metadata of Flickr Images.*

So, the aesthetic rating prediction problem can be defined as the problem of training a classifier, such that given a geo-location, the classifier can predict the aesthetic class of that geo-location by exploiting tailor-made features obtained from Flickr social metadata and the ground truths obtained from TripAdvisor.

## 3.2 Weather Condition Recognition

The goal of this problem is to construct computational models to estimate weather conditions from single Flickr images and Flickr tags in the wild. We first define the concept of *Wild Tags*.

***Wild Tags:** The idea of wild tags was first introduced in the works of [45]. Wild tags are those tags of an image that are directly provided by the user who uploaded that image to some photo-sharing services. In this problem, we use these wild tags without any subsequent manual filtering or curation of the tags.*

Flickr is a large source of photos with wild tags. However, every photo doesn't contain a *wild weather tag*. The definition of *wild weather tag* is as follows,

**Wild Weather Tags:** *Wild weather tags are those wild tags of an image that are used to describe the weather condition captured in the image. In this problem we consider only four such tags. They are sunny, rainy, snowy, and cloudy.*

In this problem, we consider four major weather conditions corresponding to each wild weather tag under consideration. There are a tremendous amount of user tagged photos with the above mentioned weather conditions in several photo-sharing services. For this problem, we are considering only the photos shared in Flickr. That means these images are captured under different viewpoints, in various places, during different portions of the day and with different devices. In short, in this problem we are considering images captured in the wild. Therefore, we can define this problem as a classification task where we aim to train classifiers to accurately detect one of 4 weather conditions from Flickr photos captured in the wild (i.e. dynamic viewpoints) and their associated wild weather tags.

# Chapter 4

## Aesthetic Rating Prediction

Our first problem is to predict the aesthetic rating of a location from the social metadata of Flickr images available at that location. In order to develop our solution, we have accumulated two datasets that contain information of representative locations from Rome, Italy and Paris, France respectively. We have derived 11 features empirically. Then, we trained several classifiers and analyzed their performances on predicting aesthetic rating of a location using the derived features from social metadata of Flickr images. We have also applied state-of-the-art techniques to improve classifier performances. In this section, we first provide a short description of the available datasets related to our work in Section 4.1. Then we discuss about the process through which we have gathered our dataset in Section 4.2. In Section 4.4, we have provided short introduction about the classifiers we have used in this work and their performance on our dataset. Finally, in Section 4.5 we have discussed about oversampling method, necessity to apply oversampling in our problem and the classifier performances after oversampling our datasets. Finally, we performed ensemble learning methods to improve our classifiers and reported their accuracies with ensemble learning in Section 4.6.

### 4.1 Available Datasets

Flickr is a well-known multimedia content hosting web-site . Flickr provides its users facilities to upload and manage their images and videos while at the same time allows others to

view shared contents. According to [3] [4], there are about 90 million Flickr users and more than 14 billion shared images in Flickr. Apart from storing the raw images and videos, Flickr also manages a number of metadata with each image. Each image may be associated with a title, the time-stamp of capturing the photo, textual tags, optional description of the image provided by the user, it's geo-location, EXIF metadata (the properties of the device with which the photo is captured) and so on. In addition to that, Flickr also tracks its users' interaction with the shared photos and captures these behavior as number of view of an image, number of users who have added the photo as a favorite, number of comments in the photo and the text of those actual comments. After its introduction in 2004, Flickr have drawn the interest of photographers, artist and common people. The overwhelming availability of dynamically generated humanistic contents has drawn the attention of researcher community as well.

There are multiple available datasets related to Flickr that can be considered while choosing our dataset. The first one of them is the YFCC100m dataset published by Yahoo Webscope [46]. This dataset contains a list of photos and videos, which are compiled from data available on Yahoo! Flickr. The dataset is divided into three parts. The main part of the dataset contains information about 100 million flickr photos. It contains photo or video identifier, photo/video hash, user information, date in which the photo was taken, upload date, title of the photo, description, user tags (comma-separated), location information, specifications of the device by which the photo or video was captured and URLs of the photo or video. YFCC100m also includes machine tags and human readable place information for every photo. An alternative of YFCC100m dataset is the Multimedia Commons Repository(MMC). The differences between YFCC100m and MMC are the supplemental material to YFCC100m that the MMC offers. MMC offers audio, visual and motion features such as LIRE, GIST, SWIFT that are often used by multimedia researchers. There are several other sources such as MIRFLICKR [47], Flickr API [48] etc., through which one can access large number of Flickr photos and their related attributes.

## 4.2 Dataset Generation

In order to perform the desired task of classification we need a dataset that provides social metadata such as number of views, number of favorites, number of comments etc. of Flickr photos as well as ground truths with respect to the aesthetic scores of geo-locations. From the discussion about YFCC100m and MMC, we can observe that none of them include social metadata of the photos. Moreover, none of these datasets include any ground truth about a place being aesthetically desirable. To facilitate our classification task we decided to build a new dataset using Flickr social metadata and aesthetic ratings of geo-locations. In this process we have used Flickr API, to gather Flickr image metadata, and TripAdvisor, to gather aesthetic rating ground truths. Therefore, a short description about Flickr API and TripAdvisor is given in this section before moving on to the description of the dataset generation process.

### Flickr API

To enable easier and flexible retrieval of Flickr images, Flickr developers have offered an advanced Application Programming Interface(API) [48]. It enables programmers to rely on HyperText Markup Language(HTML) and Hyper Text Transfer Protocol(HTTP) to access Flickr photos. As a result, Flickr has gained popularity among the researcher community. According to Flickr API's documentation, Flickr API is a set of callable methods. If one wants to perform an action using Flickr API, they need to follow a calling convention and send a request to Flickr's endpoint specifying the method and its arguments. If the method, arguments and calling convention are in accord, then the caller will receive a response in one of the supported formats. Otherwise, the response will contain an error code to lead the programmer to designing a proper API call.

### TripAdvisor

TripAdvisor is an American travel and restaurant recommendation website founded in the year 2000. It has gathered user reviews of a large number of restaurants and hotels. TripAdvisor users can view hotel and restaurant reviews to plan their trip. It also helps its users to book



accommodations during their trip. Since most of the contents of TripAdvisor are gathered from its users, TripAdvisor provides its services for free to everyone. However, they earn their revenue from advertising and hotel booking facilities. In this work, we have used the location rating available from TripAdvisor. We call these ratings the aesthetic ratings of locations. TripAdvisor computes these location rating from the user reviews. They consider three properties of user reviews while calculating the rating of a location. They are quantity, quality and recency of the reviews.

In order to appropriately model photo distribution around a city or country, we tried to gather location names and ratings from TripAdvisor.com. Since TripAdvisor does not provide any API for research purposes, we used HTML parser to get data from Trip Advisor. From Trip Advisor we have retrieved ratings, and number of reviews for around 1200 attractions in Rome, Italy and 1200 attractions in Paris, France. After removing duplicates and attractions such as tours, restaurants and hotels we were left with approximately 850 and 650 locations for consideration respectively. Despite the provided ratings being on a scale of 0-5 with an interval of 0.5, we observed from the data, that most of the ratings associated with top attractions in Rome lies in the range 2.5-5 and within 2-5 in Paris. For each location, we retrieved latitude and longitude using Google Places API. Then for each geo-location we fetched metadata of all the images that lie within a circle of 100m radius surrounding that particular location from Flickr API. We gathered social metadata of 6 million Flickr images captured at 850 locations of Rome dataset. At the same time, we scrapped social metadata of 4 million Flickr photos taken at 650 locations of Paris dataset. Thus we set up two datasets where each location is labeled as one of 7 classes i.e. classes with aesthetic scores 2,2.5,3,3.5,4,4.5,5. Let each aesthetic class be named as  $C_i$  where  $i$  is the corresponding aesthetic score. Figure 4.1 and 4.2 show the number of locations in each class in our dataset.

The locations in our dataset can be divided into several categories according to TripAdvisor. Among the locations in Rome there are points-of-interests, museums, church and cathedrals, historic sites, castles, gardens, parks, neighborhoods etc. Similarly, the locations of Paris are also categorized by TripAdvisor. Figure 4.3 and 4.4 show the distributions of locations in different categories in our dataset.

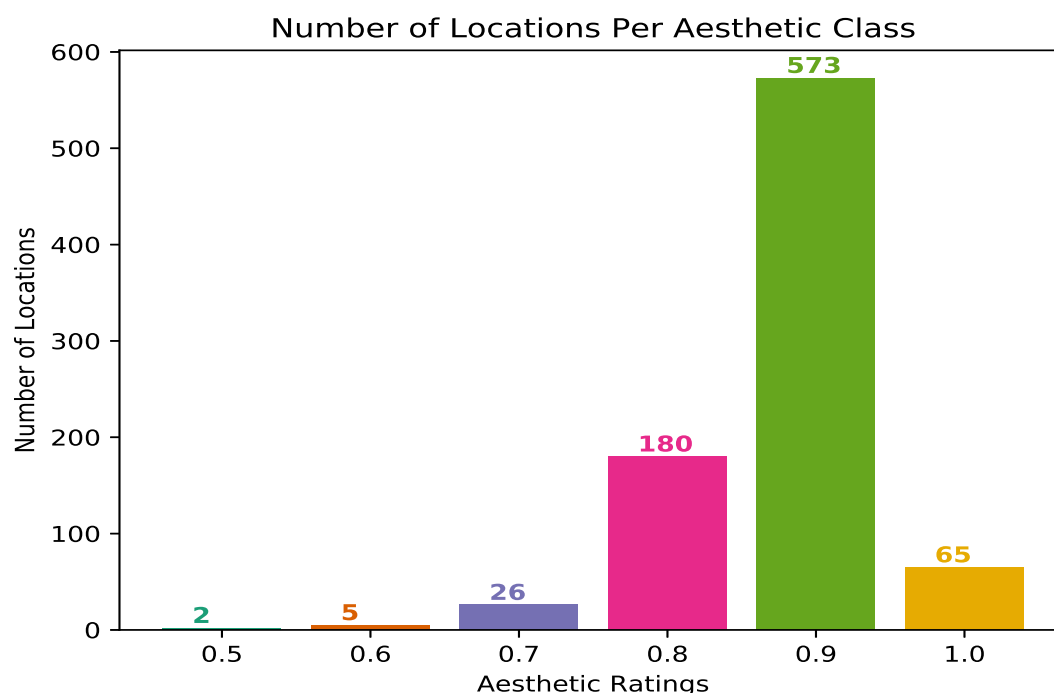


Figure 4.1: Number of locations in each class of Rome dataset

### 4.3 Feature Extraction

The social meta-data related to each photo that are directly available from Flickr are number of times the photo is viewed, number of people who have added the photo as favorite, and number of comments on the photos. We have extracted some aggregate features using these meta-data for each location. The major intuitions behind our features are,

- A place with more aesthetic beauty encourages more users to take photos and upload them. It results in a higher density of photos at that place.
- An aesthetically beautiful place is more likely to draw tourists and the number of distinct users uploading photos at a place will increase.
- People usually searches for photos of beautiful locations and views them.
- The more beautiful a place is, it is more likely that people will capture better photos which results in higher number of people adding them to favorites.

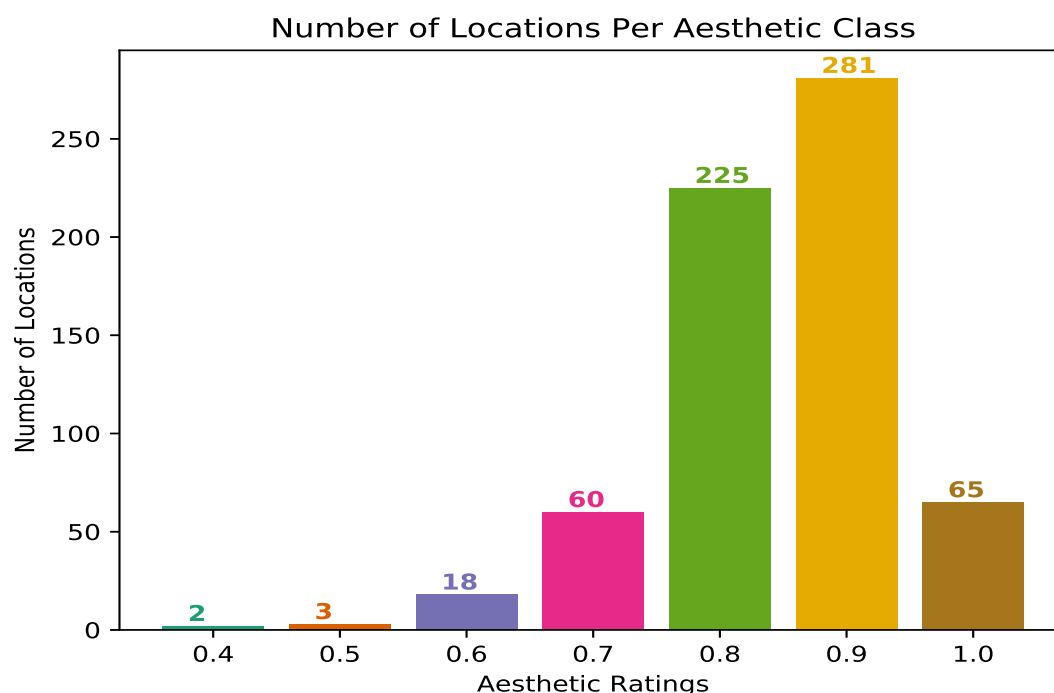


Figure 4.2: Number of locations in each class of Paris dataset

- Higher number of views should lead to higher number of favorites and comments.
- Many professional photographers often capture famous photos even at less popular places. Which may lead to higher deviation from the average number of views and average number of favorites of the photos at that position. But photos taken at beautiful and popular places usually depend on the overall scenario rather than the photographers skills.

Keeping these points in mind we have generated the following features for every location. We have also plotted the distributions of each feature for each aesthetic rating of Rome dataset in Figures 4.5,4.6 4.7 and 4.8. Similarly, the distributions of each feature for each aesthetic rating of Paris dataset are plotted in Figures 4.9, 4.10, 4.11 and 4.12.

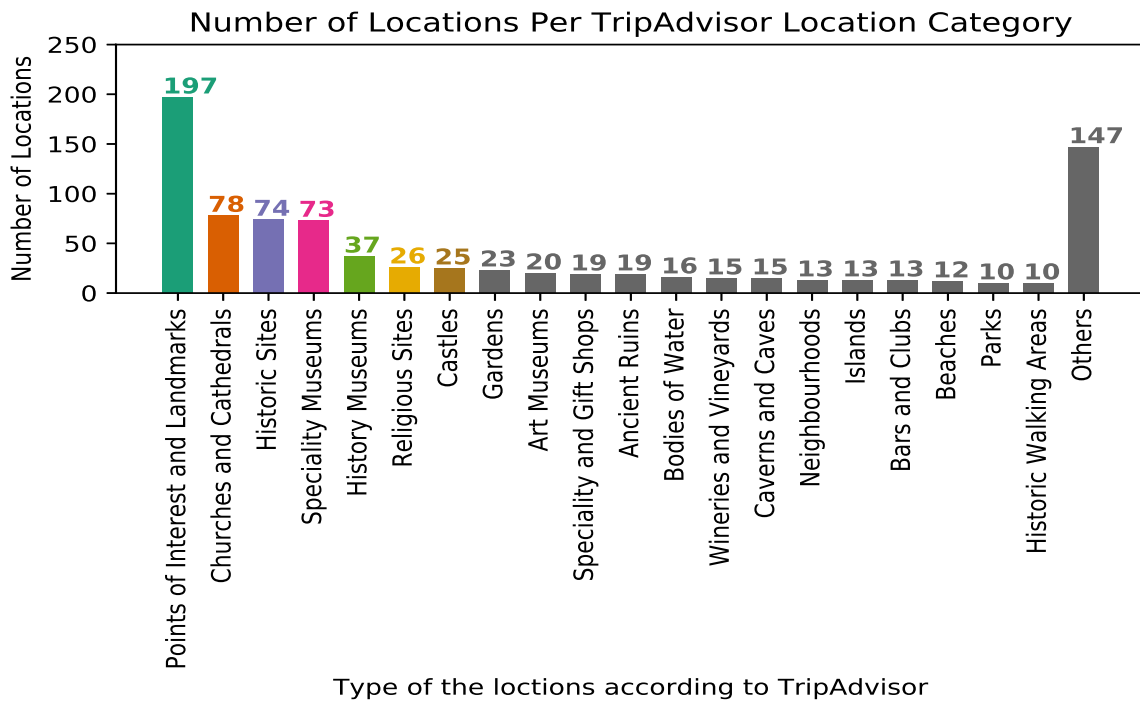


Figure 4.3: Number of locations in each category of Rome dataset

Table 4.1: List of derived features.

Photo density	Total number of views	Total number of favorites
Total number of comments	Average views per photo	Average favorites per photo
Average comments per photo	Ratio of number of favorites to number of views	Ratio of number of comments to number of views
Distinct user count per location	Maximum number of photo per user	

## 4.4 Classifiers

In order to perform the classification, we have trained various types of classifiers and compared their performances. Among the variations of decision tree, we have used J48, REPTree and Random Forrest. We have also trained Naive Bayesian classifier, KNN and several ANNs. In this section, a brief description about these classifiers is provided before reporting the accuracy, precision and recall of each classifier.

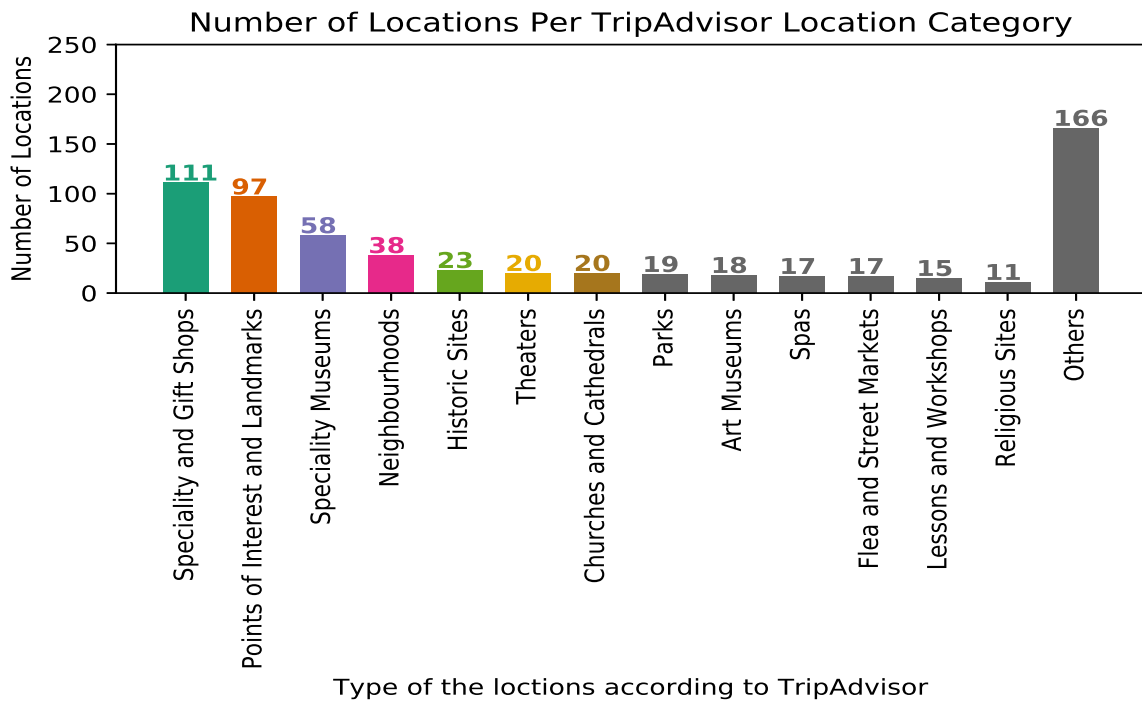
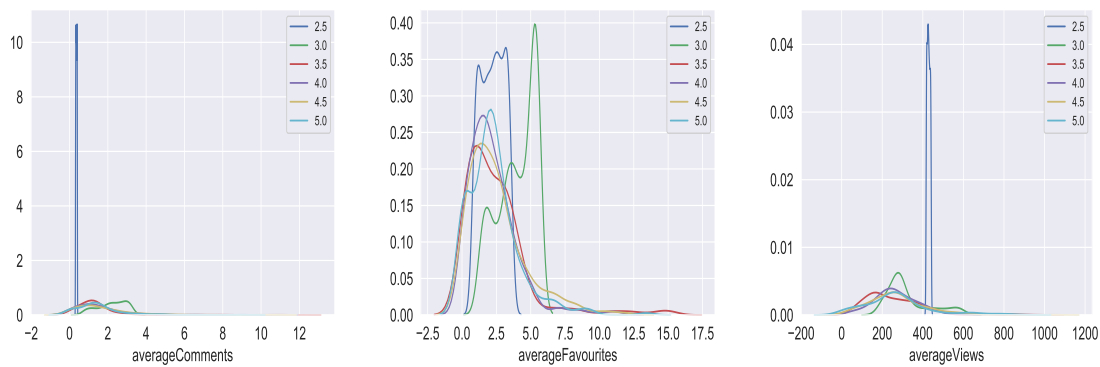


Figure 4.4: Number of locations in each category of Paris dataset



(a) Distribution of *averageComments* attribute in Rome dataset  
 (b) Distribution of *averageFavourites* attribute in Rome dataset  
 (c) Distribution of *averageViews* attribute in Rome dataset

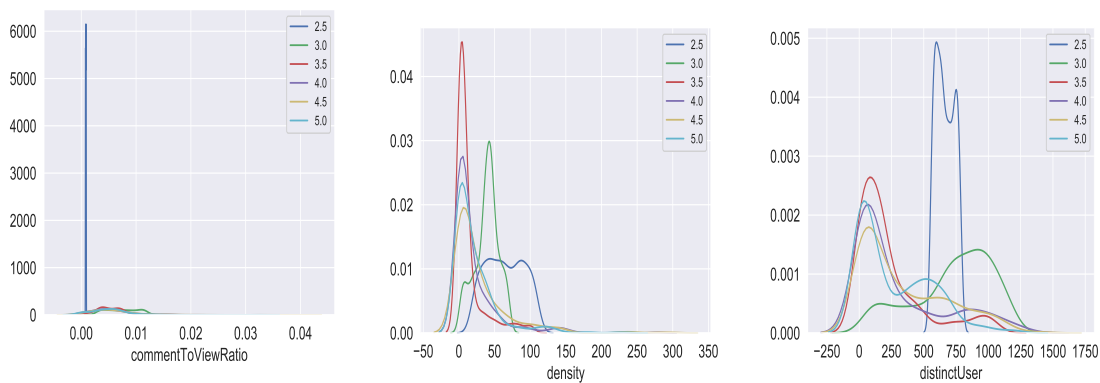
Figure 4.5: Distributions of attributes in Rome Dataset

### Decision Tree Learning

Decision Tree is a tree-like graph which is used to make a decision from an observation. In machine learning, decision tree learning is a supervised learning method to find the tree that

Table 4.2: Merit of each attribute obtained from different attribute selection algorithms

Name of the attribute	Greedy Stepwise with CFS Subset Eval	Ranker with Correlation Attribute Eval	Ranker with Info Gain Attribute Eval	Average
density	0.99	0.42	0.79	0.47
totalViews	0.95	0.57	0.74	0.51
totalFavourites	0.94	0.31	0.65	0.42
totalComments	0.97	0.45	0.64	0.47
averageViews	0.86	0.67	0.94	0.51
averageFavourites	0.98	0.41	0.49	0.46
averageComments	0.98	0.38	1.00	0.45
favouriteToViewRatio	1.00	0.65	0.54	0.55
commentToViewRatio	0.74	0.62	0.96	0.45
distinctUser	1.00	1.00	0.78	0.67
maxPerUser	0.93	0.32	0.84	0.41



(a) Distribution of *commentToViewRatio* attribute in Rome dataset

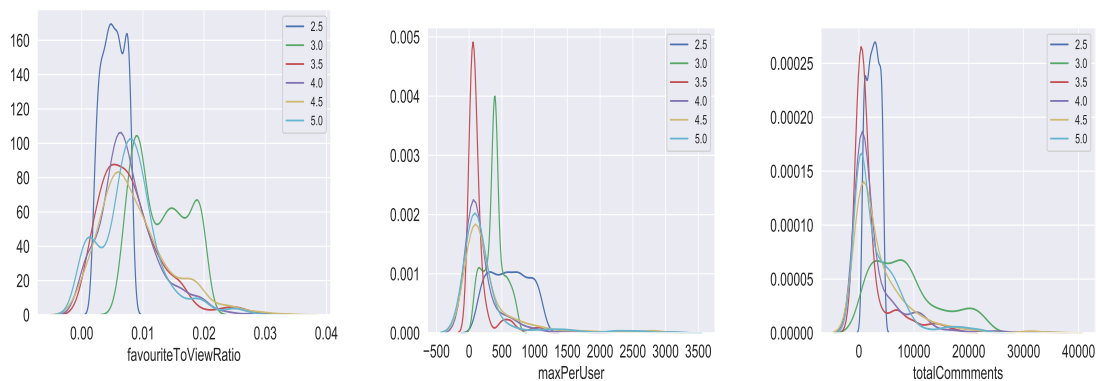
(b) Distribution of *density* attribute in Rome dataset

(c) Distribution of *distinctUser* attribute in Rome dataset

Figure 4.6: Distributions of attributes in Rome Dataset

Table 4.3: Merit of each attribute obtained from different attribute selection algorithms

Name of the attribute	Greedy Stepwise with CFS Subset Eval	Ranker with Correlation Attribute Eval	Ranker with Info Gain Attribute Eval	Average
density	0.99	0.83	0.73	0.61
totalViews	1.00	0.81	0.77	0.60
totalFavourites	0.75	0.70	1.00	0.48
totalComments	0.95	0.80	0.72	0.59
averageViews	0.93	0.58	0.91	0.50
averageFavourites	1.00	0.69	0.67	0.56
averageComments	1.00	0.73	0.41	0.58
favouriteToViewRatio	1.00	0.90	0.52	0.63
commentToViewRatio	0.97	0.65	0.50	0.54
distinctUser	1.00	1.00	0.70	0.67
maxPerUser	0.98	0.68	0.71	0.55

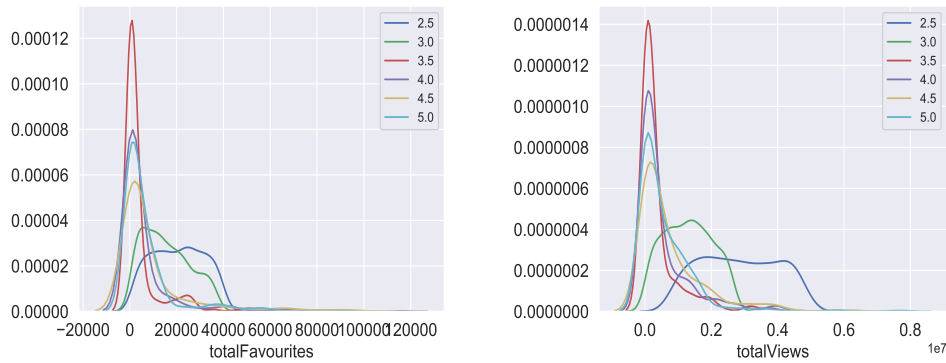


(a) Distribution of *favouriteToViewRatio* attribute in Rome dataset

(b) Distribution of *maxPerUser* attribute in Rome dataset

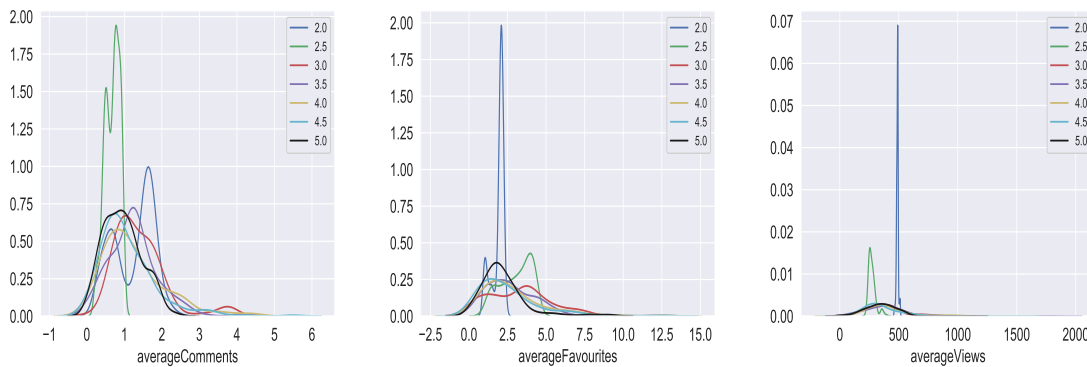
(c) Distribution of *totalComments* attribute in Rome dataset

Figure 4.7: Distributions of attributes in Rome Dataset



(a) Number of locations in each category of Paris dataset (b) Number of locations in each category of Paris dataset

Figure 4.8: Distributions of attributes in Rome Dataset

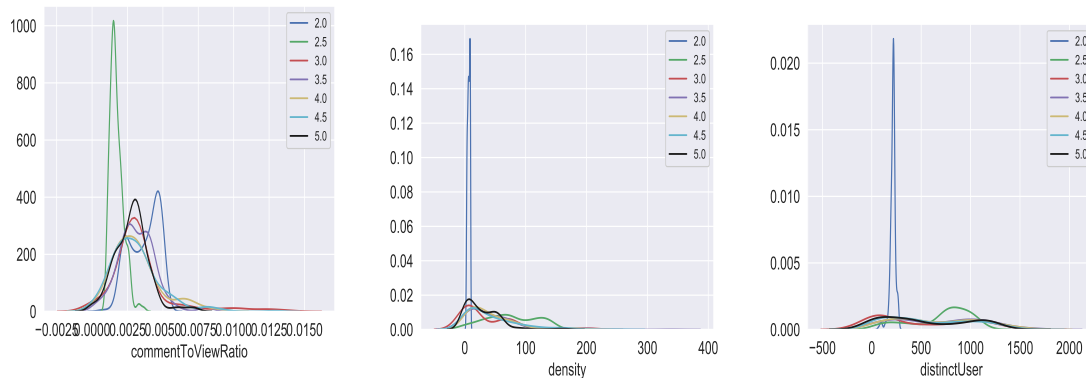


(a) Distribution of *averageComments* attribute in Paris dataset (b) Distribution of *averageFavourites* attribute in Paris dataset (c) Distribution of *averageViews* attribute in Paris dataset

Figure 4.9: Distributions of attributes in Paris Dataset

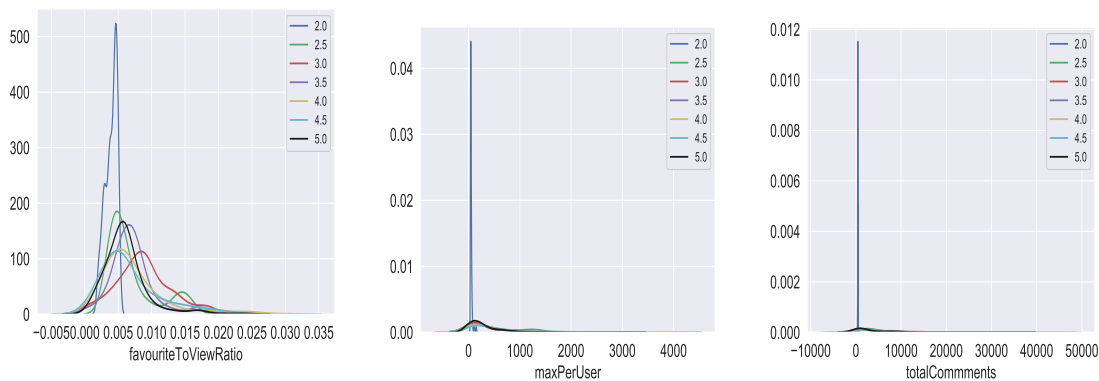
most appropriately represents the hypothesis in the dataset. Decision trees can be either classification tree or regression tree. A decision tree where the leaves or target nodes can take discrete values are called classification tree. In the tree model of a classification tree, the leaf nodes represent class labels and the branches represent conjunction of features that lead to that class label. Now, given an observation, if the branches of a decision tree are followed, a class label for that observation is found. In this work, we have used the following decision tree learning algorithms.





(a) Distribution of *commentToViewRatio* attribute in Paris dataset (b) Distribution of *density* attribute in Paris dataset (c) Distribution of *distinctUser* attribute in Paris dataset

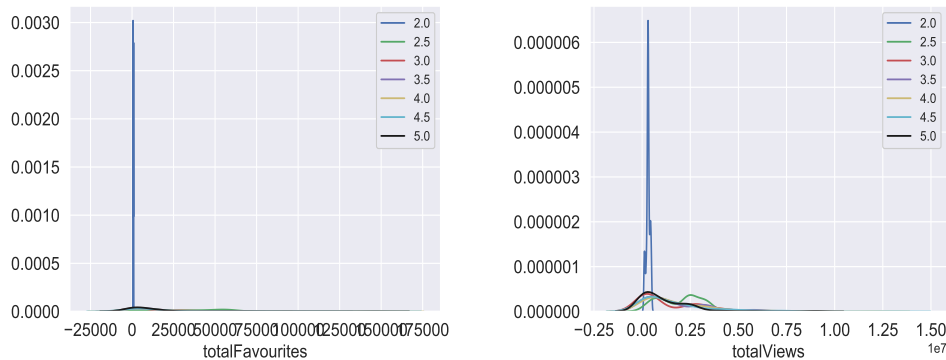
Figure 4.10: Distributions of attributes in Paris Dataset



(a) Distribution of *favouriteToViewRatio* attribute in Paris dataset (b) Distribution of *maxPerUser* attribute in Paris dataset (c) Distribution of *totalComments* attribute in Paris dataset

Figure 4.11: Distributions of attributes in Paris Dataset

- **J48** J48 is an implementation of *Iterative Dichotomiser 3* (ID3) algorithm in Weka [49]. At the beginning of ID3 algorithm it is assumed that there is only one node in the decision tree and it corresponds to the entire dataset. ID3 considers a decision node  $n$  in the decision tree and calculates the information gain corresponding splitting the samples with respect to the values of each attribute which was unused in all of the parent nodes. Then, it splits the subset of data corresponding to node  $n$  by the attribute which has the smallest information



(a) Number of locations in each category of Paris dataset (b) Number of locations in each category of Paris dataset

Figure 4.12: Distributions of attributes in Paris Dataset

gain. It also adds child decision nodes and each child node correspond to one subset of the the data. The label of the child decision node is set equal to the majority of the members in the subset of data. The algorithm then recursively continues for each child node and finally stops when all the attributes are used or when a certain condition in satisfied.

- **REPTree** REPTree is the implementation of another decision tree algorithm C4.5(an improvement of ID3). C4.5 can handle both continuous and discrete attributes, missing attribute values, differing cost of attributes and also performs pruning. The pruning methods are used to keep the decision tree smaller so that it prevents overfitting and generalize better. In REPTree implementation reduced error pruning techniques is used. This pruning technique starts at the leaf and replaces each node and the subtree at that node with its most popular class. If prediction accuracy is not much affected then this change is kept.
- **Random Forest** Random Forest is an algorithm that trains multiple decision trees on random portions of the dataset and then merges them to create a better classifier. To generate a random portion of dataset random forest selects a random subspace of features rather than selecting a random subset of samples.

### Naive Bayes

Naive Bayesian classifier is a generative classifier model. In this method it is assumed that the features are independent to each other and bayesian probabilistic model is used to calculate probability estimates. Given a sample  $x = \{x_1, x_2, \dots, x_n\}$  where  $x_i$  is the  $i$ th feature of  $x$  and a set of classes  $C_i$  where  $i \in [1, 2, \dots, n]$  naive bayesian classifier tries to generate probability estimates of  $P(C_i|x)$  and then classifies  $x$  in the class for which  $P(C_i|x)$  is maximum.

### K-Nearest Neighbour(KNN)

k-nearest neighbour algorithm is an instance based or lazy learning algorithm. That means it defers any computation until the actual classification task is required. When a sample is given, KNN tries to find out its nearest neighbours based on some distance metrics. Among the distance metrics, most common distance metric is *Euclidean distance metric*. According to Euclidean distance metric, distance between two samples  $x_1$  and  $x_2$  is the square root of the summation of squared differences of each feature of  $x_1$  and  $x_2$ .

### Artificial Neural Network(ANN)

Artificial Neural Network is a computing system inspired by the biological neural networks. In this method, there are usually a number of hidden layers between input layer and output layer. Each layer has multiple nodes and each node of a layer is usually connected with nodes of next layer. Each connection has a weight associated with it. The output of each node is usually the value obtained by passing the linear combination of outputs of the nodes of previous layer and their associated weights, through an activation function. In order to perform classification tasks with neural networks, the weights are trained by feeding the input values through the network and propagating the gradient of loss function backwards to update the weights.

In order to assess the performances of our classifiers we used accuracy, precision and recall metrics. Since, ours is a multi-class classification problem we need to compute the average accuracy, precision and recall of all the classes. According to [50], there are two methods to compute average of performance metrics in multi-class classification problems. They are macro-averaging

and micro-averaging. In this work, we have applied macro-averaging method to evaluate classifier performances. Table 4.4 and Table 4.5 demonstrate the accuracy, precision and recall of each classifier trained on Rome and Paris dataset respectively.

Table 4.4: Accuracy, Precision and Recall of the classifiers trained on Rome Dataset

Classifier	Accuracy	Precision	Recall
J48	64.39	NaN	17.67
K-Nearest Neighbour	51.59	NaN	18.15
Naive Bayes	14.57	NaN	19.82
Random Tree	49.00	NaN	18.42
REPTree	66.51	NaN	16.78
Neural Network(5*5*5*5, Learning Rate: 0.1, Iterations: 1000)	20.51	NaN	16.62
Neural Network(5*5*5, Learning Rate: 0.1, Iterations: 1000)	67.33	NaN	16.67
Neural Network(5*5*5, Learning Rate: 0.1, Iterations: 5000)	66.98	NaN	16.77
Neural Network(5*5*5, Learning Rate: 0.2, Iterations: 1000)	67.33	NaN	16.67
Neural Network(5*5*5, Learning Rate: 0.3, Iterations: 1000)	67.33	NaN	16.67
Neural Network(5*5*5, Learning Rate: 0.5, Iterations: 1000)	66.98	NaN	16.58

Table 4.4 shows that among the classifiers J48, REPTree and neural networks demonstrates promising performance. However, we can notice that the precision metric of each classifier is not defined. In macro-averaging scheme, when a classifier doesn't predict any sample as a member of a specific class then precision for that class is undefined. As a result the macro-average of the precision of that classifier becomes undefined. Analyzing the confusion matrices of our classifiers it is observed that the classifiers are biased towards majority classes. This is due to the fact that the ratio between the number of instances in majority class and the number

Table 4.5: Accuracy, Precision and Recall of the classifiers trained on Paris Dataset

Classifier	Accuracy	Precision	Recall
J48	36.09	NaN	18.65
K-Nearest Neighbour	37.00	20.28	20.61
Naive Bayes	10.86	13.96	10.87
Random Tree	36.09	19.87	20.04
REPTree	41.44	NaN	16.04
Neural Network(5*5*5*5, Learning Rate: 0.1, Iterations: 1000)	42.97	NaN	14.29
Neural Network(5*5*5, Learning Rate: 0.1, Iterations: 1000)	42.97	NaN	14.29
Neural Network(5*5*5, Learning Rate: 0.1, Iterations: 5000)	41.28	NaN	14.58
Neural Network(5*5*5, Learning Rate: 0.2, Iterations: 1000)	40.21	NaN	13.72
Neural Network(5*5*5, Learning Rate: 0.3, Iterations: 1000)	39.76	NaN	13.65
Neural Network(5*5*5, Learning Rate: 0.5, Iterations: 1000)	37.61	NaN	13.47

of instances in minority class is too high in the dataset. Therefore, the trained classifiers tend to be biased towards majority classes. Similar results can be observed from Table 4.5. Although the classifiers do not perform well (about 40% accuracy on average), k-NN and Random Tree outperforms other with respect to precision.

## 4.5 Oversampling

After analyzing the distribution of locations in each aesthetic class  $C_i$ , it was discovered that the distribution of locations among several aesthetic class is quite skewed. To be more precise, most locations have a moderate aesthetic rating. Very few locations received maximum aesthetic rating from the users. Similarly, only a few locations were reported to have very low aesthetic rating. Therefore, training a classifier on such a skewed dataset resulted in an undesired bias in the classifiers. In order to handle imbalanced dataset, we can apply either oversampling or undersampling. Undersampling is mostly performed in cases where the number of instances in the

training set is too high. On the other hand, oversampling is the technique of reducing unwanted bias from a dataset by generating new samples. Oversampling techniques are usually used when there are few instances in the training set. Oversampling and undersampling techniques are mainly used when the dataset under consideration is imbalanced and the classifiers are suffering from overfitting by learning the skewed distribution. In this problem, undersampling is not appropriate since in that case only a few instances will be left in the training set. Therefore, we have applied state-of-the-art oversampling technique, SMOTE [51], to generate synthetic instances of the minority classes. To illustrate how SMOTE works, let us consider a sample  $s$  in a dataset. To oversample, we first find out the  $k$ -nearest neighbors of  $s$  in the feature space and take the vector  $v$  between one of the  $k$  neighbors and  $s$ . We then multiply  $v$  by a random real number between 0 to 1 and add the resultant vector to  $s$  to get the new synthetic instance. Using SMOTE we have generated synthetic instances for the aesthetic classes that have very few instances. Thus we prepared a somewhat balanced dataset. Figures 4.13 and 4.14, report the number of instances of each class, after performing oversampling on each dataset.

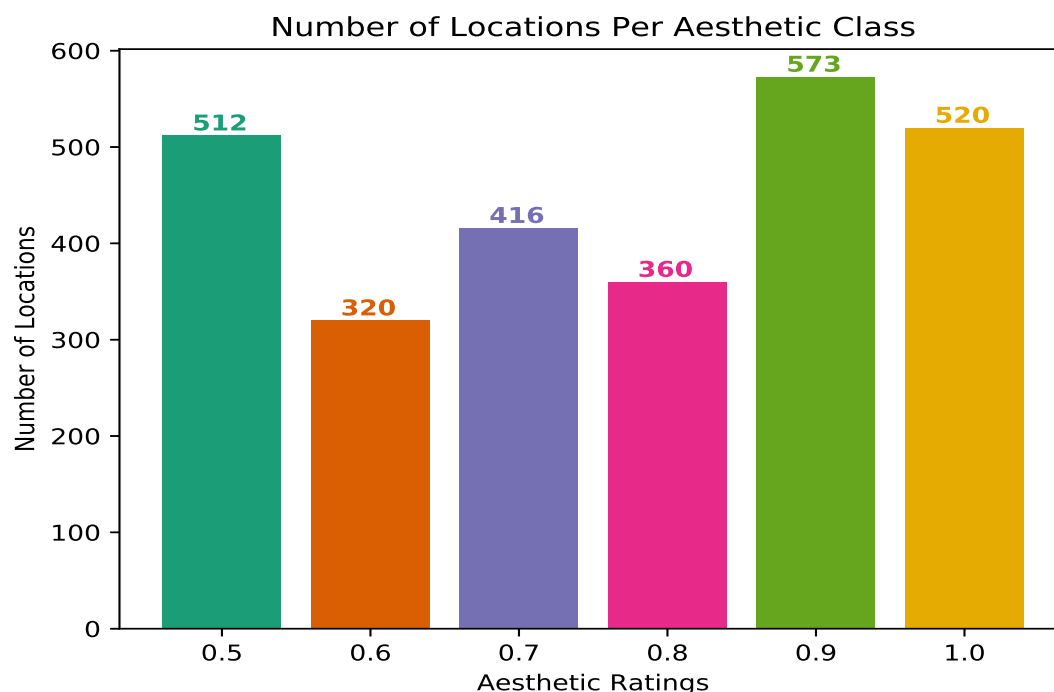


Figure 4.13: Number of locations in each class of oversampled Rome dataset

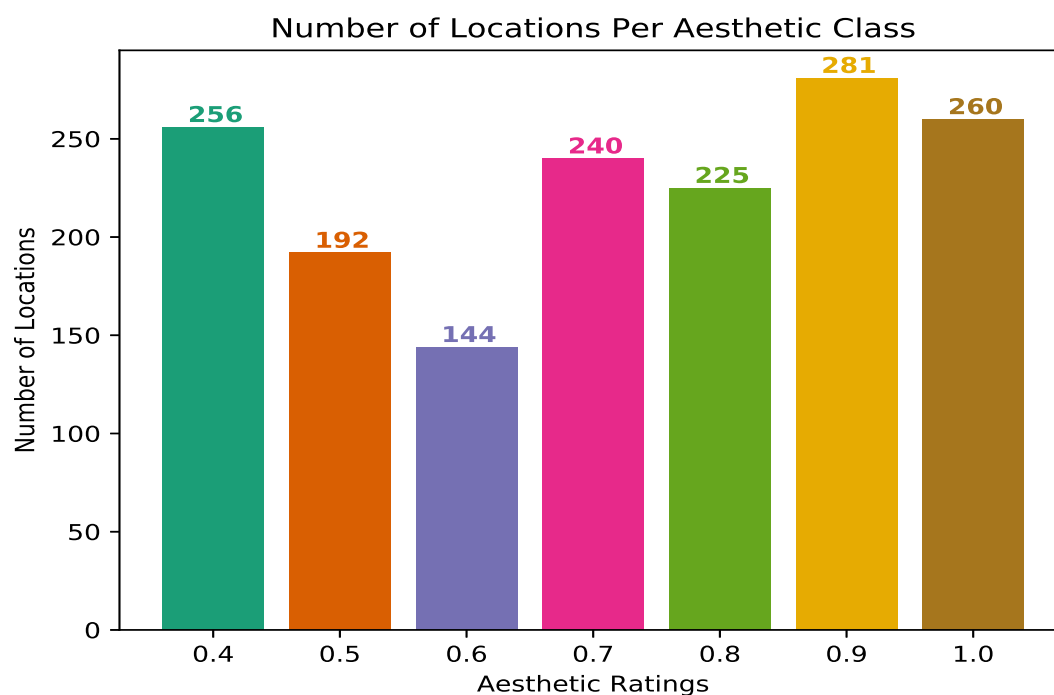


Figure 4.14: Number of locations in each class of oversampled Paris dataset

Using these balanced datasets, we have trained decision tree variants, KNN, naive bayesian classifier and several variants of ANN. Table 4.6 and Table 4.7 contain the accuracies, precisions and recalls of each classifier trained on oversampled Rome and Paris dataset. From the classifier performances obtained on Rome dataset, we can observe that k-NN and Random Tree achieves highest accuracies. On the other hand, neural network with less hidden layers performs better than deeper neural networks on Rome dataset. Similarly, Table 4.9 reports that k-NN and Random Tree performs better than other classifiers in Paris dataset. At the same time simpler architectures perform better than larger architectures in Paris dataset too.

Table 4.6: Accuracy, Precision and Recall of the classifiers trained on oversampled Rome Dataset

Classifier	Accuracy	Precision	Recall
J48	75.16	74.44	75.19
K-Nearest Neighbour	74.49	73.47	74.90
Naive Bayes	46.28	47.22	48.34
Random Tree	76.01	75.57	76.28
REPTree	71.71	70.12	71.83
Neural Network(5*5*5*5, Learning Rate: 0.1, Iterations: 1000)	20.51	NaN	16.62
Neural Network(5*5*5, Learning Rate: 0.1, Iterations: 1000)	53.39	NaN	53.47
Neural Network(5*5*5, Learning Rate: 0.1, Iterations: 5000)	49.28	50.87	49.93
Neural Network(5*5*5, Learning Rate: 0.2, Iterations: 1000)	53.13	52.76	53.85
Neural Network(5*5*5, Learning Rate: 0.3, Iterations: 1000)	54.20	52.87	54.59
Neural Network(5*5*5, Learning Rate: 0.5, Iterations: 1000)	47.72	50.14	47.38

## 4.6 Ensemble Learning

Ensemble learning is a method which uses several machine learning algorithms or multiple instances of the same algorithm to achieve a better performance than any one of them could achieve alone. In other words, ensemble method tries to train multiple hypothesis and combines them to get a better hypothesis. In this work, two state-of-the-art ensemble techniques are used. They are bagging and boosting. Bagging is an ensemble learning method that learns classifiers on various different distributions of the training set and uses all the classifiers for classification. On the other hand, in each iteration of boosting technique, it tries to learn classifiers on the samples that were wrongly classified using previous classifiers and assign a corresponding weight to each classifier.

In order to boost the accuracy of classifiers, we have applied Bagging and Adaboost technique on the decision tree classifiers, naive bayesian classifier and k-nearest neighbor classifiers. Both bagging and boosting help to reduce error rate of an individual classifier. In Table 4.8 and Table



Table 4.7: Accuracy, Precision and Recall of the classifiers trained on oversampled Paris Dataset

Classifier	Accuracy	Precision	Recall
J48	62.20	66.57	67.55
K-Nearest Neighbour	66.65	69.91	72.02
Naive Bayes	39.17	42.39	45.87
Random Tree	65.14	70.30	69.91
REPTree	58.20	61.91	64.71
Neural Network(5*5*5*5, Learning Rate: 0.1, Iterations: 1000)	17.21	NaN	17.19
Neural Network(5*5*5, Learning Rate: 0.1, Iterations: 1000)	42.87	42.47	44.98
Neural Network(5*5*5, Learning Rate: 0.1, Iterations: 5000)	45.93	48.71	50.48
Neural Network(5*5*5, Learning Rate: 0.2, Iterations: 1000)	43.18	47.28	49.96
Neural Network(5*5*5, Learning Rate: 0.3, Iterations: 1000)	44.12	43.68	49.36
Neural Network(5*5*5, Learning Rate: 0.5, Iterations: 1000)	39.17	35.93	43.72

4.9 the performance of each of the classifiers are reported. From the reported accuracies, we can notice both the ensemble learning method results in improved performance. However, in Rome dataset boosting with J48 performs better than other classifiers whereas in Paris dataset Random Tree outperforms other classifiers when it is coupled with boosting. As a result, the maximum accuracy reported on Rome dataset is 80% and the best performance of classifier reported on Paris dataset is 71%.

## 4.7 Summary

In this chapter, we have discussed about the problem of predicting aesthetic rating of a location from the social metadata of Flickr photos. In this problem, we used aesthetic rating ground

Table 4.8: Ensembled Accuracy, Precision and Recall of the classifiers trained on oversampled Rome Dataset

Classifier	Accuracy	Precision	Recall
Bagging with J48	78.97	77.82	78.87
Bagging with K-Nearest Neighbour	74.97	73.76	75.40
Bagging with Naive Bayes	46.39	47.22	48.45
Bagging with Random Tree	80.23	79.39	80.42
Bagging with REPTree	77.97	76.31	77.65
Boosting with J48	80.90	80.69	80.93
Boosting with K-Nearest Neighbour	65.95	66.12	65.95
Boosting with Naive Bayes	46.28	46.93	48.32
Boosting with Random Tree	80.12	79.58	80.39
Boosting with REPTree	78.23	77.56	78.12

truths from TripAdvisor. We gathered two datasets that contain aesthetic rating ground truth and Flickr social metadata of 850 locations in Rome and 650 locations in Paris respectively. Then, we have designed 11 empirical features and generated them from the social metadata of the photos of each location. We trained several decision tree variants, k-NN, Naive Bayesian classifiers and a number of neural networks on these datasets. Since preliminary classifiers were suffering due to dataset imbalance, we oversampled our datasets with state-of-the-art oversampling method namely, SMOTE. Finally, to further improve classifier performances we have applied bagging and boosting ensemble learning methods. Our classifiers reported maximum 80% accuracy on Rome dataset and 71% accuracy on Paris dataset.

Table 4.9: Ensembled Accuracy, Precision and Recall of the classifiers trained on oversampled Paris Dataset

Classifier	Accuracy	Precision	Recall
Bagging with J48	68.59	71.54	73.99
Bagging with K-Nearest Neighbour	66.21	69.43	71.91
Bagging with Naive Bayes	39.55	43.27	46.39
Bagging with Random Tree	71.09	73.48	75.50
Bagging with REPTree	65.08	67.96	71.37
Boosting with J48	70.65	73.70	75.17
Boosting with K-Nearest Neighbour	66.65	69.91	72.02
Boosting with Naive Bayes	39.17	42.39	45.87
Boosting with Random Tree	64.89	68.71	69.47
Boosting with REPTree	64.27	68.79	69.64

# Chapter 5

## Weather Condition Detection

Our second problem is to detect the weather condition of an image with the help of wild Flickr tags. In order to develop our solution, we have accumulated an image dataset that contains about 70k images which are tagged with one of four wild weather tags, namely, sunny, rainy, cloudy and snowy. We trained a small architecture as our initial solution. Later on, we transfer trained large scale architectures such as VGG16, InceptionV2, and Inception-ResnetV2. In this section, we first provide a short description of the available dataset related to our work in Section 5.1. Then we discuss about the process through which we have gathered our dataset in Section 5.2. In Section 5.4, we have provided short introduction about the classifiers we have used in this work and the performance of our classifiers. Finally, in Section 5.5 we have discussed about transfer learning and the architectures under consideration. Under the same section we have also presented the performance of our transfer trained classifiers.

### 5.1 Available Datasets

In order to perform weather detection, we need to train appropriate classifiers which classifies images into different weather. Therefore, we need an image dataset is annotated with weather condition ground truths. Preliminary studies on weather detection from image analysis used to consider images taken from a static viewpoint [10,11,52]. Therefore several dataset are available where images captured from one or more fixed viewpoints are annotated with weather condition

ground truths such as WILD [53] and AMOS-C. Weather and Illumination Database (WILD) is a dataset of images of urban landscapes which were collected to illustrate the variations of scene appearance due to weather, illumination and change of seasons. On other hand, AMOS is an archive of outdoor images captured by webcams of fixed viewpoints. The authors of [54] have annotated these images with the help of Weather Underground [55], Weather Central [56] and the National Climatic Data Center.

The authors of [12] have prepared a dataset of 10k images captured in the wild. Among these images 5k images are labeled as sunny and the other half of the images are labeled as cloudy. These images were collected from image datasets such as SUN dataset [42], LabelMe dataset [43] and Flickr. This dataset was annotated by the help of helpers each of whom labeled same amount of images using their common sense. These helpers also performed removal of almost similar images. In short, with the help of helpers the authors of [12] were able to prepare a dataset of 10k images with unambiguous weather condition annotation. This dataset was later used in further studies such as [15, 41]. This dataset was enhanced with auxiliary weather cue annotation by the authors of [41]. That means they employed several helpers to annotate weather cues such as shadows, clouds etc, with bounding boxes.

In contrast to binary class weather annotated image datasets, the authors of [13, 57], have accumulated a dataset named *Multi class Weather Image* (MWI) which contains 20k outdoor images with dynamic viewpoint and are classified into four weather conditions.

Finally, in [44] the authors have prepared a large-scale image dataset captured in the wild and each image is associated with rich weather annotation. The images in this dataset were collected from Flickr. With the help of Flickr API, the geo-location of each image was gathered. Using the weather history data from *Weather Underground* [55], each image was labeled as one of five weather conditions. These weather conditions are clear, cloudy, rainy, foggy and snowy. However, they focused mainly on data filtering based on criteria such as sky region ratio and so on, to develop a somewhat less ambiguous image dataset with weather condition ground truths.

In this we problem, we need to use an image dataset that contains images with dynamic viewpoints and wild weather tags as ground truth. Although recent dataset such as the one published by [12, 44] contains a collection of outdoor images from dynamic viewpoints they do

not contain wild weather tags. Therefore, we have decided to accumulate a dataset of our own to satisfy our requirement.

## 5.2 Dataset Generation

In order to generate the desired dataset, we took the help of Flickr API. More specifically, we used *flickr.photos.search* method in Flickr API. Using this method we can fetch all the photo ids that satisfy a certain criteria. In *flickr.photos.search* method we can specify uploader's user id, tags, latitude, longitude, minimum and maximum upload date and several other parameters. The *search* method can respond in JSON, XML and some other formats. In our case, we fetched all the photo ids of Flickr images who have a wild weather tag. That means we performed the *search* method once for each weather condition, namely, sunny, cloudy, rainy and snowy, by specifying the *tags* parameter of *search* method. We received responses in JSON format. The JSON responses, received from Flickr API, are paginated and we parsed all of the pages. Therefore a single query should be enough to retrieve all the photo ids who have the same wild weather tags. However, the Flickr API is designed in such a way that one single query doesn't return information of more than 4000 distinct photos. To solve this issue, we split the query into time segments by specifying maximum upload date and minimum upload date. We divided the entire period between 01 Jan 2010 to 01 Jan 2017 into weeks and retrieved all the photo ids which were taken in that time period using Flickr API.

After parsing the JSON response, we recorded photo ID, farm ID, owner ID, server ID and secret key of each photo. These are the information required to generate an URL of the image and later download it. According to [58], the format of the URL of the downloadable format of a Flickr photo is as follows.

$$https://farm\{farm-id\}.staticflickr.com/\{server-id\}/\{id\}-\{secret\}-[mstzb].jpg$$

The final segment represents the size of the photo to be fetched. There are multiple available options. In our work, we used the option *z* in order to retrieve medium sized (maximum 640 pixel on the longest side) images. Then, each image were resized into (256 \* 256) size.

Finally, we put aside a portion of the images for testing purpose. The dataset was split into approximately 80:20 ratio of training and testing images. For each image a random number between 0 to 1 was generated and if this number was greater than .80 then it was moved to the test set.

It is to be mentioned here that there is a parameter of *search* method which is named *geo-context*. According to Flickr API documentation, geo context is a numeric value which represents the photo's geotagging beyond latitude and longitude. For example, if we wish to search for photos that were taken "indoors" or "outdoors" we can set geo-context 1 and 2 respectively. Since, only outdoor images are appropriate in our problem, the most logical thing to do is to set *geo-context* as 2. However, setting *geo-context* to 2 results in empty responses for most of the queries. Therefore, we filtered out the "indoor" images using manual observation.

## 5.3 Background

Convolution neural networks(ConvNet) have become very popular due to its extraordinary performance in image recognition, image classification and computer vision. Architecture of a convolution neural network is a major concern when training a ConvNet. Therefore, extensive studies have been performed on ConvNet architectures and several state-of-the-art architectures have been published. In this section, we discuss about popular convolution neural network architectures in Section 5.3.1 and a special style of training neural networks which is known as Transfer Learning in Section 5.3.2.

### 5.3.1 Convolution Neural Network(ConvNet)

*Convolution neural network*(ConvNet) is a category of neural network that is designed and usually used in the task of image recognition. There are four major operations in convolution neural networks and each layer of nodes in a convolution neural network corresponds to one of these operations. The types of layers in a ConvNet are usually confined to convolution layer, non-linear layer, pooling layer and finally fully connected layer or classification layer. In convolution layer, several *filters* or feature detectors are slid over the pixels of an image to generate Feature

Maps. The purpose of non-linear layer is to introduce non-linearity in the ConvNet since it is seen that most real-world data have at least some degree of non-linearity. The spatial pooling layers are used to reduce dimensions while at the same time retain most popular information. Finally, the classification layers are used to predict the class labels from the trained feature maps.

### 5.3.2 Transfer Learning and Fine Tuning

Training a convolution network is a computation and resource extensive operation. Training a deep convolution neural network not only requires large datasets but also requires high performance machines(GPUs) and a lot of time. Therefore, it is often not feasible with limited resources. That's why researchers often choose to use pre-trained models and adapt those models with their problem. For this reason, a number of pre-trained networks have been published in recent years. This method of adapting a model trained for one task to perform another task is called *Transfer Learning*.

Transfer learning can be performed in the following [59] ways. First, a pre-trained model without the final fully-connected network can be used as a feature extractor. Then these feature can be used to train another fully-connected network tailor made for the task in hand. The second strategy is to not only retrain a new fully-connected model but also allow the training of the weights of the model.

#### Popular Pre-trained Architectures

- **VGG16** VGG16 [60] is a network designed by Simonyan et al where they demonstrated that increasing network depth and keeping the filters small can achieve higher accuracy and generalization than smaller networks. Their network is an improvement of AlexNet [2]. In this architecture, there are 5 convolution layers each followed by a pooling layer. Each of the convolution layers are combinations of multiple convolution operations. After the convolution layers there are 3 fully-connected layers and a softmax layer for classification. Figure 5.1 shows the VGG16 architecture. This architecture was trained on Imagenet [19] dataset.



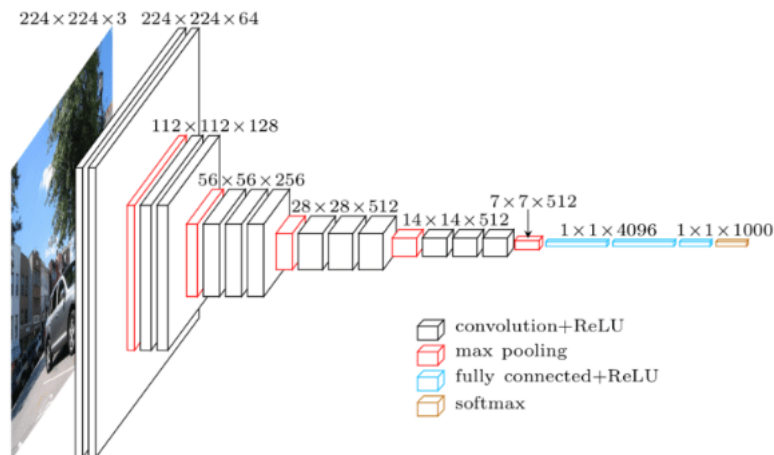


Figure 5.1: VGG-16 Architecture

- Inception** Inception model was introduced in GoogleNet [1]. Inception model is a smaller model inside a bigger architecture. Figure 5.2 shows the inception model introduced in [1]. The intuition behind such a model is that it is often unclear whether a  $3 \times 3$  convolution is better than a  $5 \times 5$  convolution layer. Rather than choosing one of them, the authors have decided to perform the convolutions in parallel and concatenate the resulting feature maps before passing them to the next layer. Recently, google have published Inception V3. Figure 5.3 shows the entire architecture of InceptionV3 where several inception module are stacked one after another along with some other layers.

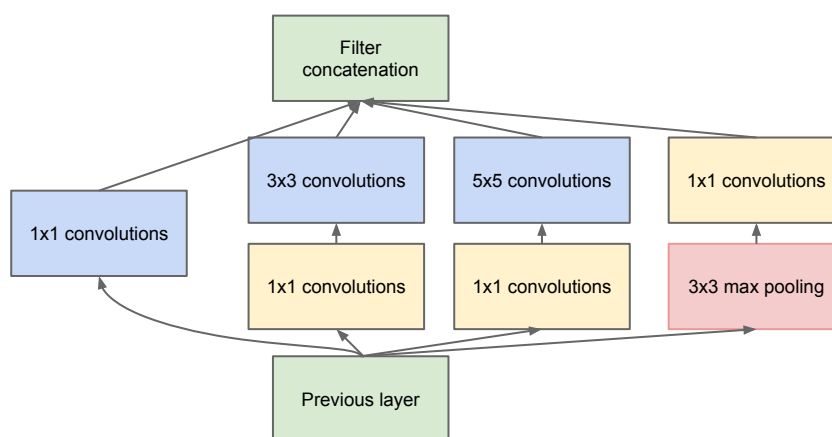


Figure 5.2: Inception module from [1]

- Inception-Resnet** Residual networks were introduced by [61] to tackle the depth problem

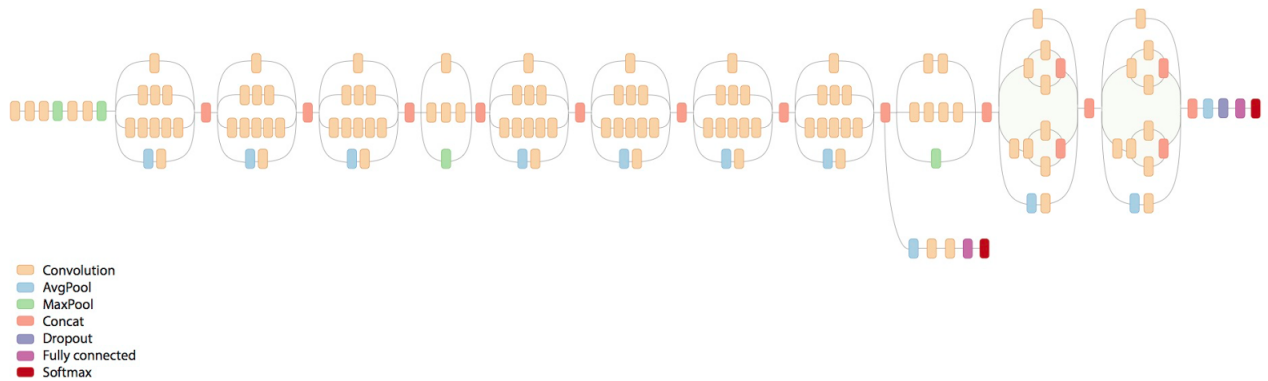


Figure 5.3: InceptionV3 architecture in Tensorflow

of neural networks. They observed that increasing layers of neural network results in higher error rate for higher number of layers. This is caused due to vanishing gradient. As layers go deep the gradients become smaller and smaller and performance drops. To tackle this problem they have introduced residual connections which means adding the output of previous layer to the output of current layer. In [62], the authors have modified the Inception model by introducing residual connections and improved the accuracy of InceptionV3. The improved model is known as Inception-ResnetV2.

## 5.4 Training Classifiers

Having collected the dataset, we can train a classifier to predict the weather conditions from images with dynamic viewpoints. If we consider applying traditional machine learning methods such as decision tree, SVM etc, use of raw images is not likely to work. In that case numeric features such as SWIFT, HOG, color histogram, GIST, brightness etc can be generated. In [12, 13, 57] the authors have also generated multiple weather dependent features. However, in this work we focus on training a classifier that can learn these representations itself from raw images. Therefore, in our solution we have trained convolution neural network (CNN) classifiers and analyzed their performance on the dataset.

Several convolution neural network architectures have demonstrated outstanding results in

recent computer vision works. Among them one of the simplest architecture is the one proposed by Krizhevsky et al. [2] which is also known as Alexnet. The original Alexnet was designed to address the Imagenet object detection challenge. It is composed of 8 layers and the output layer has 1000 nodes. The first 5 layers of Alexnet are convolution layers (CONV) with ReLu activation functions. There are two max pooling layers between the 1<sup>st</sup> and 2<sup>nd</sup> layer and the 2<sup>nd</sup> and 3<sup>rd</sup> layer. The final three layers are fully connected layers (FC). Also there are dropout layers interleaved between the fully connected layers. Figure 5.4 shows a digram of the origin architecture.

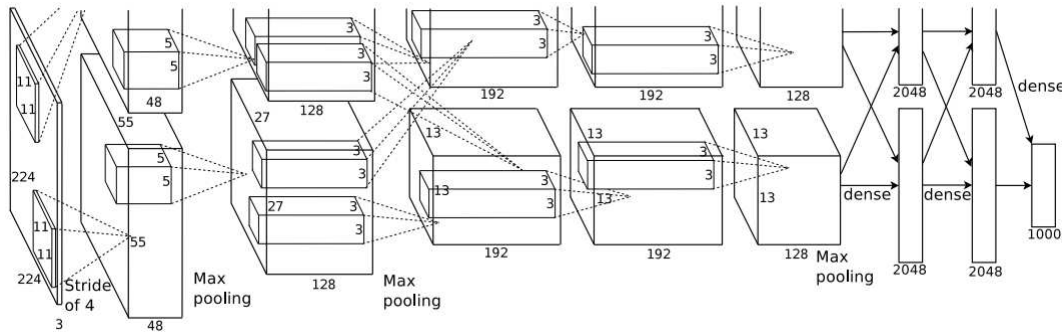


Figure 5.4: Alexnet Architecture as presented in [2]

Our first network architecture is inspired by Alexnet. However, in this problem there is only 4 weather conditions. So, we changed the final layer with a layer of 4 nodes. We have also removed the dropout layers in our network. We optimized the CNN parameters by minimizing the categorical cross-entropy loss function.

The training of CNN-models were performed using back-propagation algorithm with batch stochastic gradient descent such that the categorical cross-entropy is minimized. In back-propagation, the neural network parameters are updated by propagating the gradient of loss, multiplied by learning rate, backwards. The training process puts aside a small validation set to report validation accuracies. Each step of the training phase, computes the loss for the entire batch used during that step, updates model parameters and reports the validation set accuracy of the updated model.

### 5.4.1 Hyper-parameter Tuning

Hyper-parameters are those parameters in a machine learning process which are set before the learning process starts. The values of all the other parameters are learned during the training phase. Therefore, selection of proper hyper-parameters is an important factor of classifier performance. In order to find out an appropriate combination of the hyper-parameter values several experiments are carried out by varying hyper-parameters.

In this work we considered batch-size, learning rate and optimizers as our hyper-parameters. For each hyper-parameter, we trained several models by varying that hyper-parameter while at the same time keeping all the other hyper-parameters at their default values. Table 5.1 shows the value of hyper-parameters with their default values highlighted.

Table 5.1: AlexNet hyper-parameters and their values

hyper-parameter	Values
Batch Size	10, <b>20</b> , 50
Optimizer	<b>ADAM</b> , SGD, ADAGRAD
Learning Rate	<b>0.01</b> , 0.001, 0.0001
Number of Epochs	<b>50</b> , 100, 200

Figures 5.5, 5.6, 5.7 and 5.8 show the learning curves resulted after varying batch size, learning rate and optimizer respectively. However, it is observed that Alexnet fails to capture the representation of the images and the learning curves demonstrates erratic behavior. Even after 200 epochs the training accuracies tend to vary frequently and validation accuracy is shown to be 27.5% which resulted from classifying all the images into one class.

## 5.5 Transfer Learning

Our experiments on Alexnet gave us the insight that simpler neural network architecture may not be able to appropriately learn the weather cues from images. Therefore, to achieve better

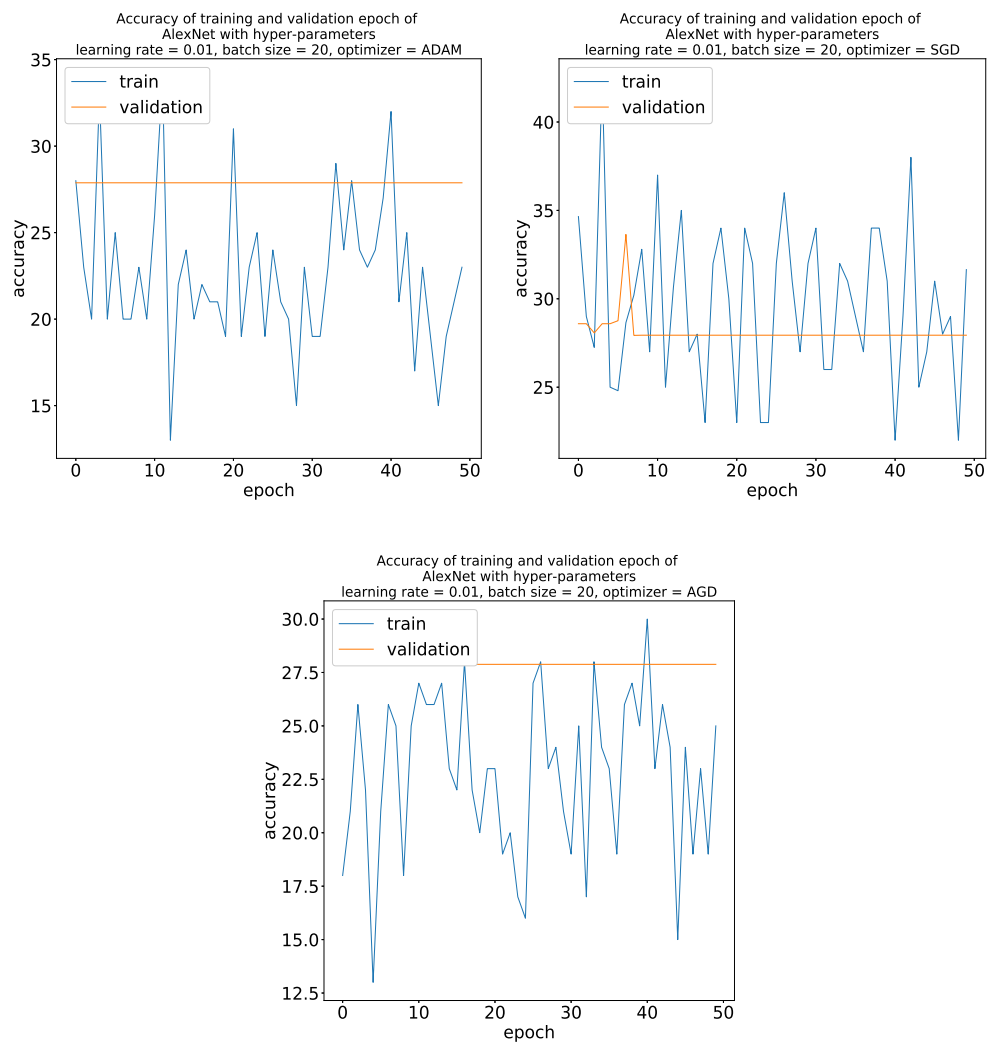


Figure 5.5: Learning curves for AlexNet, learning rate = 0.01, epochs = 50, batch size = 20 and different optimizers.

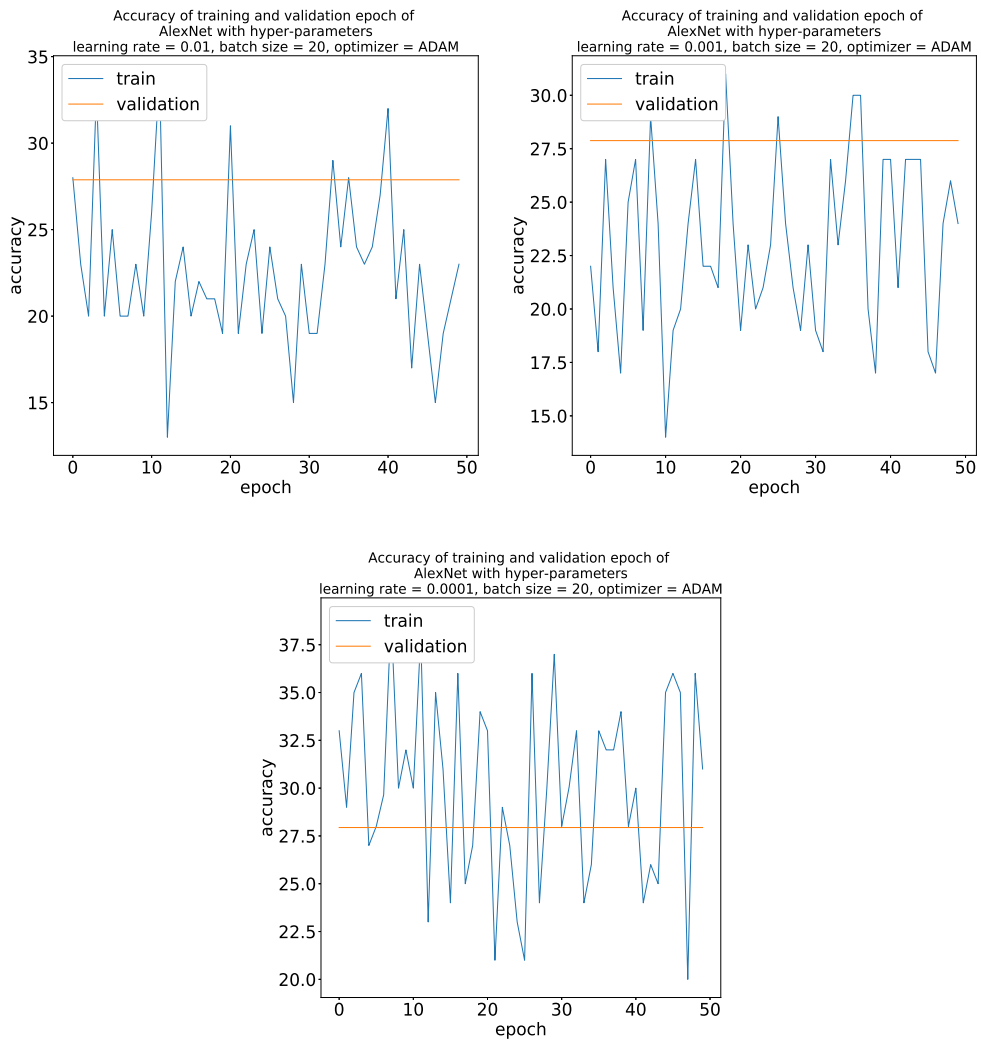


Figure 5.6: Learning curves for AlexNet with ADAM, epochs = 50, batch size = 20 and different learning rates.

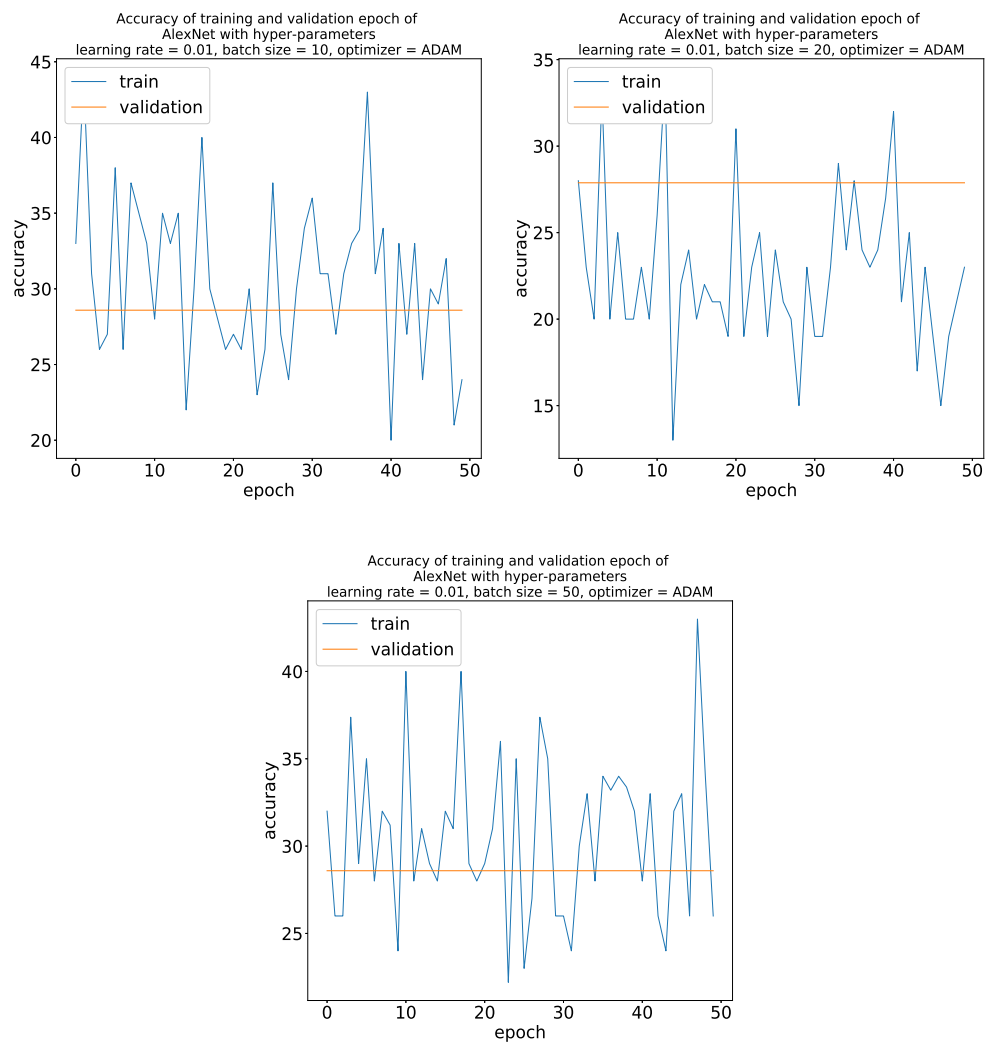


Figure 5.7: Learning curves for AlexNet with ADAM, learning rate = 0.01, epochs = 50 and different batch sizes.

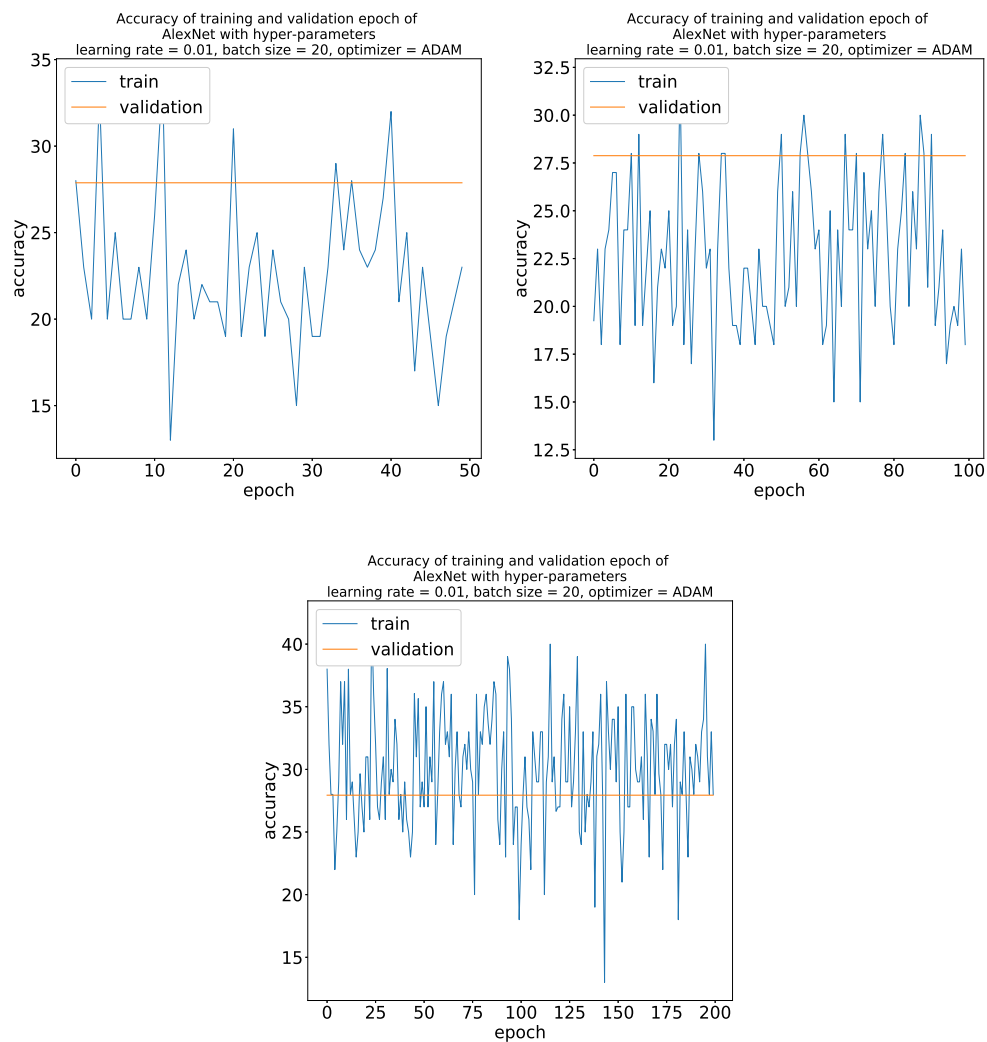


Figure 5.8: Learning curves for AlexNet with ADAM, learning rate = 0.01, batch size = 20 and different number of epochs



performance we need to train complex neural networks. However, as mentioned in Section 5.3.2 training large complex networks is a time and resource exhaustive process.

As a result, we have performed transfer learning on popular networks namely, VGG, Inception and Inception-Resnet architectures. The trained parameters of these networks are published so that others can make use of them. Here, we want to keep parameters of lower layers fixed because lower layers often correspond to basic shapes. At the same time, we need to adapt the classifier output to our problem. So we dropped the output layer and placed a fully connected layer( $FC_1$ ). Then, the outputs of  $FC_1$  were fed to a fully connected layer( $FC_{out}$ ) which is the output layer of our classifier.

Now, we can perform transfer learning by feeding each image in the network and back-propagate the gradient of loss only up to  $FC_1$  to keep the previous layers unmodified. Since these network architectures are quite large in size feeding an image to the network to generate inputs of  $FC_1$  takes significant amount of time. At the same time it can be observed that since all the layers previous to  $FC_1$  are fixed for a single image, the inputs to  $FC_1$  will always be the same for a specific image.

In order to make the training process faster, we first compute all inputs to  $FC_1$  for each image of both the training and the testing set only once. These are called bottlenecks. Then the training set bottlenecks were used to train  $FC_1$  and  $FC_{out}$ .

We have carried out experiments by varying several hyper-parameters. In our experiments we have varied four hyper-parameters. They are batch size, learning rate, optimizers and number of epochs. Table 5.2 shows the hyper-parameters and their corresponding values that were used to conduct our experiments. The default values of each hyper-parameters are highlighted.

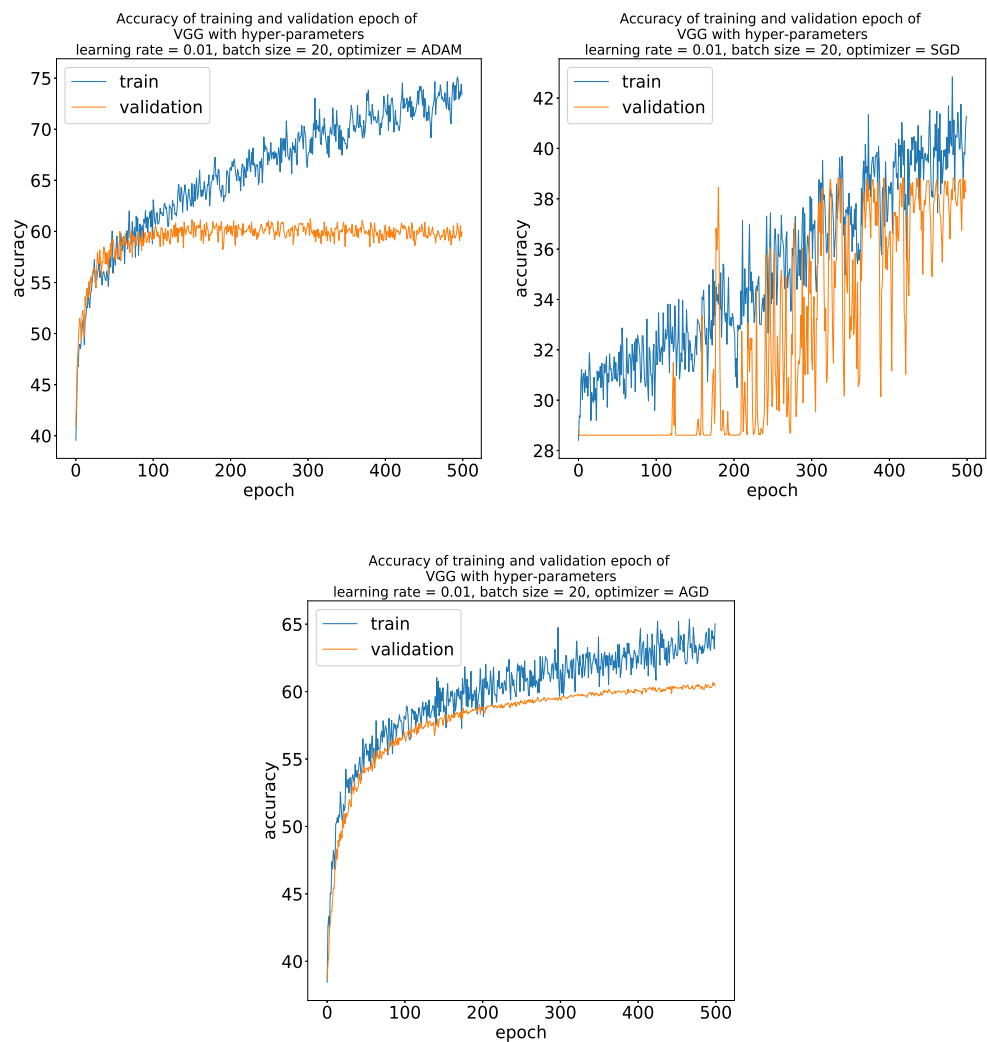


Figure 5.9: Learning curves for VGG, learning rate = 0.01, epochs = 500, batch size = 20 and different optimizers.

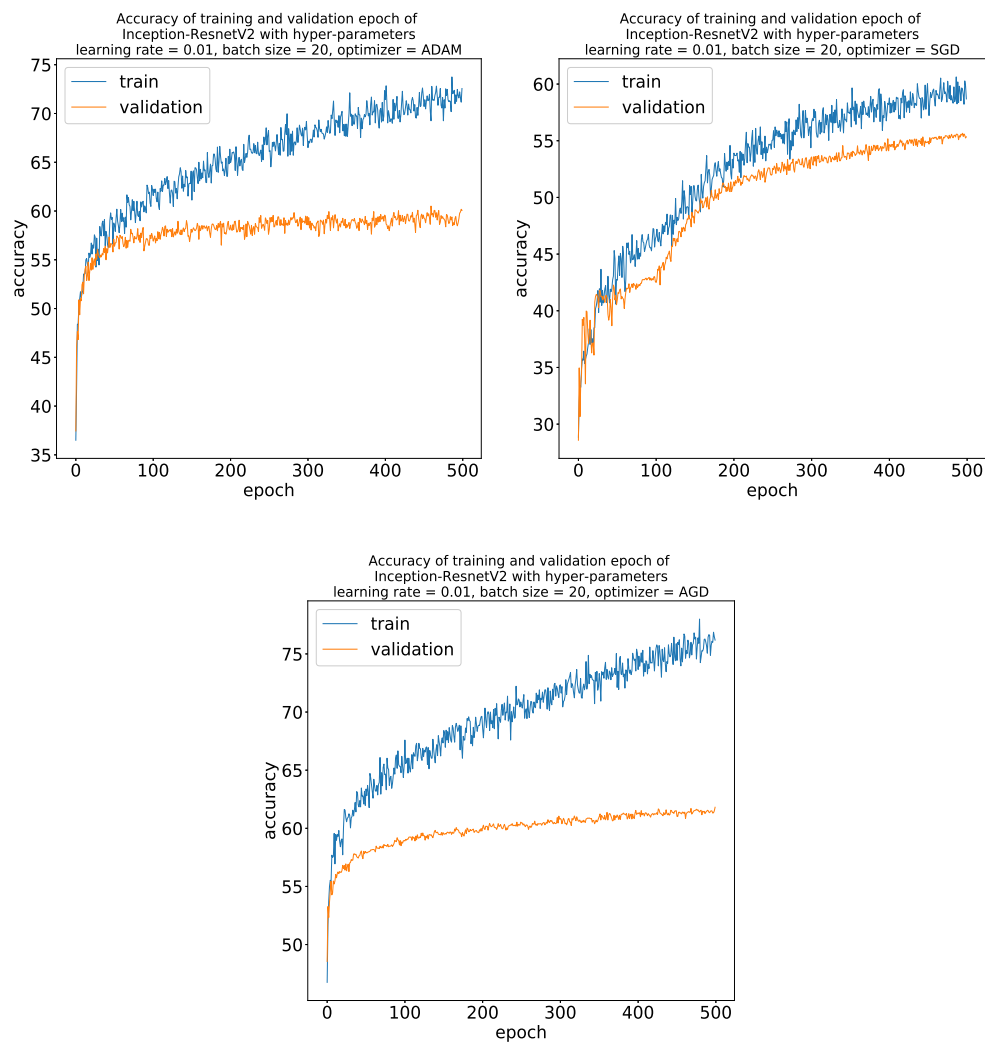


Figure 5.10: Learning curves for Inception-ResnetV2, learning rate = 0.01, epochs = 500, batch size = 20 and different optimizers.

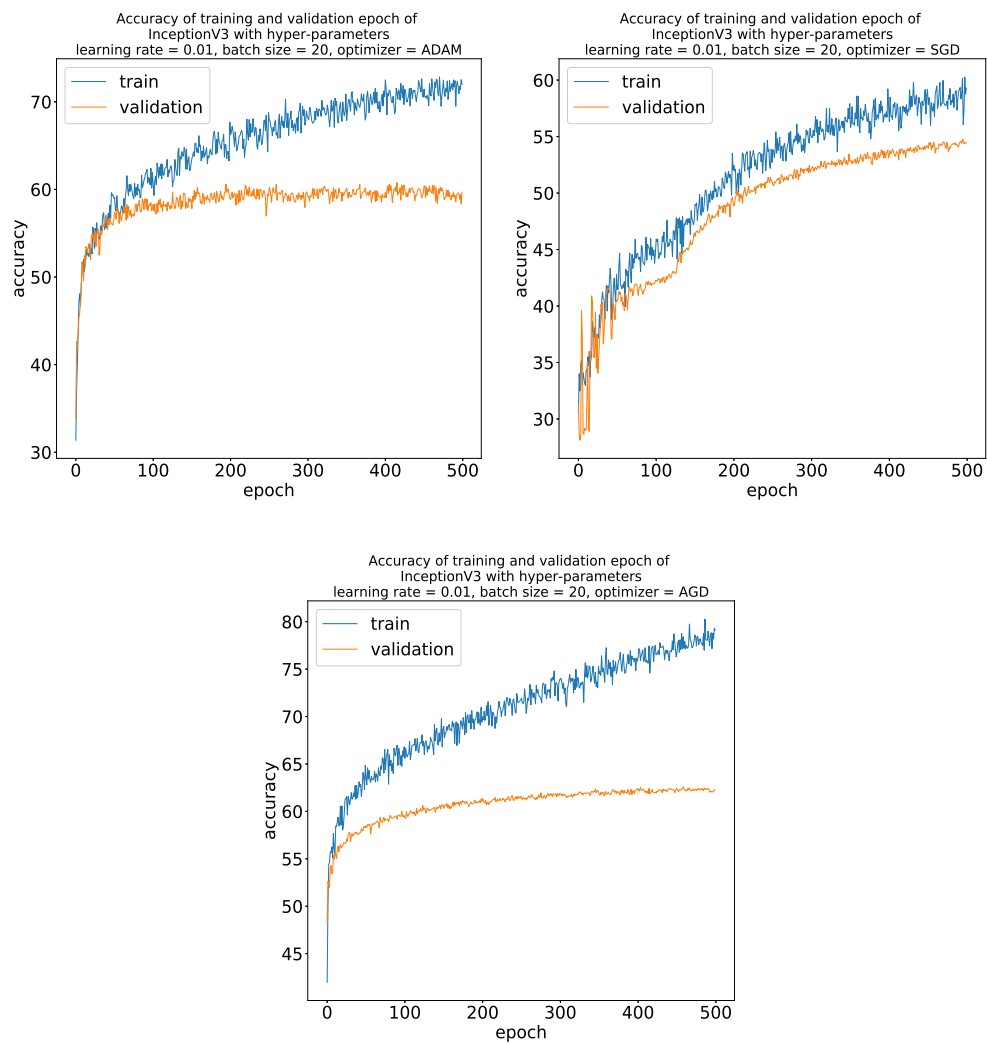


Figure 5.11: Learning curves for InceptionV3, learning rate = 0.01, epochs = 500, batch size = 20 and different optimizers.

Table 5.2: Hyper-parameters used in transfer learning and their values

hyper-parameter	Values
Batch Size	<b>20</b> , 50, 100, 200
Optimizer	<b>ADAM</b> , SGD, ADAGRAD
Learning Rate	<b>0.01</b> , 0.001, 0.0001, 0.00001
Number of Epochs	<b>500</b> , 1000

Table 5.3: Classifier Accuracy for various Optimizer of the classifier VGG

Optimizer	Training Accuracy	Validation Accuracy	Maximum Training Accuracy	Maximum Validation Accuracy
ADAM	73.38	59.90	75.12	61.24
AdaGrad	65.03	60.41	65.38	60.66
SGD	41.28	38.28	42.84	38.84

Figures 5.9, 5.10 and 5.11 show the learning curves for the classifier where we transfer train VGG, Inception-Resnet and Inception by varying optimizers respectively. It can be seen from the learning curves that use of AdaGrad optimizer resulted in smooth learning curves. On the other hand, SGD showed erratic behavior during training. However, all of them achieves about 60% accuracy on detecting weather conditions.

Figures 5.12, 5.13 and 5.14 show the learning curves for the classifier where we transfer trained VGG, Inception-Resnet and Inception upto 1000 epochs. It is observed that even after 1000 epochs of training the classifiers doesn't overfit neither their accuracies increase. Most of the classifiers reach saturation state after around 100 epochs.

Figures 5.15, 5.16 and 5.17 show the learning curves for the classifier where we transfer train VGG, Inception-Resnet and Inception with different batch sizes. However, from the learning curves we can comment that varying the batch size doesn't have much impact on the learning process since all the values of batch sizes results in similar learning curves. Finally, Figures 5.18, 5.19 and 5.20 show the learning curves for the classifier where we transfer trained VGG,

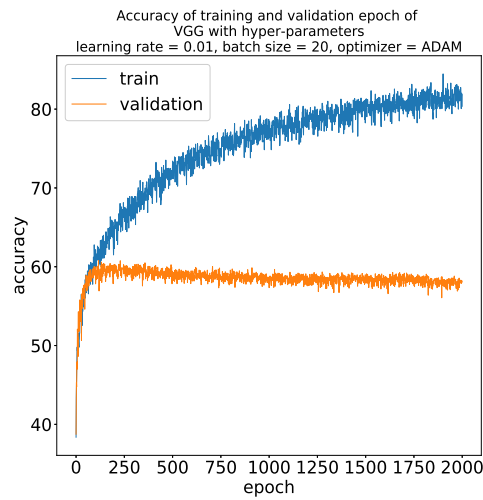


Figure 5.12: Learning curves for VGG with ADAM, learning rate = 0.01, epochs = 2000, batch size = 20

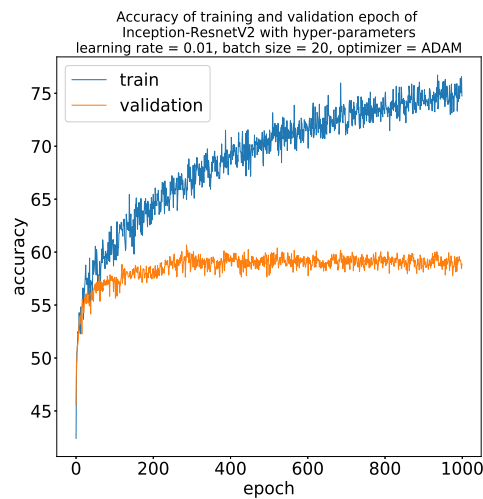


Figure 5.13: Learning curves for Inception-ResnetV2 with ADAM, learning rate = 0.01, epochs = 1000, batch size = 20

Table 5.4: Classifier Accuracy for various Optimizer of the classifier Inception-ResnetV2

Optimizer	Training Accuracy	Validation Accuracy	Maximum Training Accuracy	Maximum Validation Accuracy
ADAM	72.56	60.07	73.75	60.51
AdaGrad	76.19	61.81	78.00	61.81
SGD	58.69	55.34	60.63	55.66

Table 5.5: Classifier Accuracy for various Optimizer of the classifier InceptionV3

Optimizer	Training Accuracy	Validation Accuracy	Maximum Training Accuracy	Maximum Validation Accuracy
ADAM	72.00	59.69	72.84	60.77
AdaGrad	79.13	62.25	80.28	62.58
SGD	59.28	54.43	60.28	54.75

Inception-Resnet and Inception with different learning rates.

Tables 5.3, 5.5, 5.4, 5.6, 5.8, 5.7, 5.9, 5.11 and 5.10 provide a summary of the learning process by reporting the final training accuracy of the classifier, final validation accuracy, maximum training accuracy at any point in the training phase and maximum validation accuracy that the classifier achieved. Almost all of the classifiers achieves higher training accuracy (as much as 75%) however they perform at about 60% on the validation set. In order to improve performance, we can try to train large architecture such as VGG16 from scratch. On the other hand, an analysis of accuracy of wild Flickr tags can also be done.

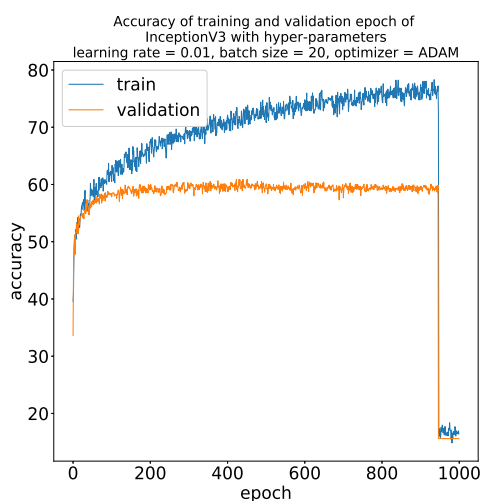


Figure 5.14: Learning curves for InceptionV3 with ADAM, learning rate = 0.01, epochs = 1000, batch size = 20

Table 5.6: Classifier Accuracy for various batch size of the classifier VGG

Batch size	Training Accuracy	Validation Accuracy	Maximum Training Accuracy	Maximum Validation Accuracy
100	72.16	59.34	74.03	60.83
200	72.47	58.75	73.78	60.78
20	73.38	59.90	75.12	61.24
50	73.06	59.86	74.56	61.18

## 5.6 Summary

In this chapter, we have discussed about the problem of detecting weather conditions of Flickr images using their wild weather tags. In our problem, we considered only four wild weather tags. We gathered about 70k images using Flickr API. Then, we trained AlexNet architecture



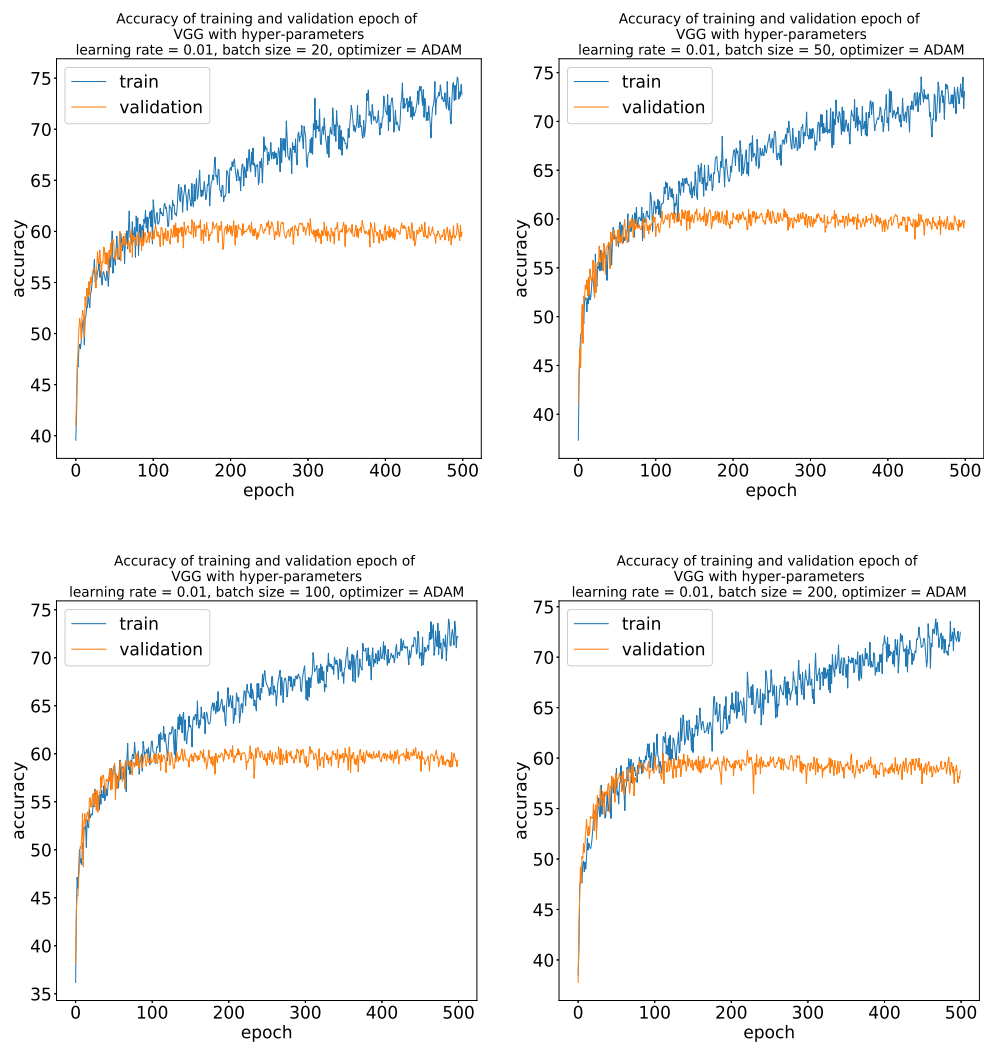


Figure 5.15: Learning curves for VGG with ADAM, learning rate = 0.01, epochs = 500 and different batch sizes.

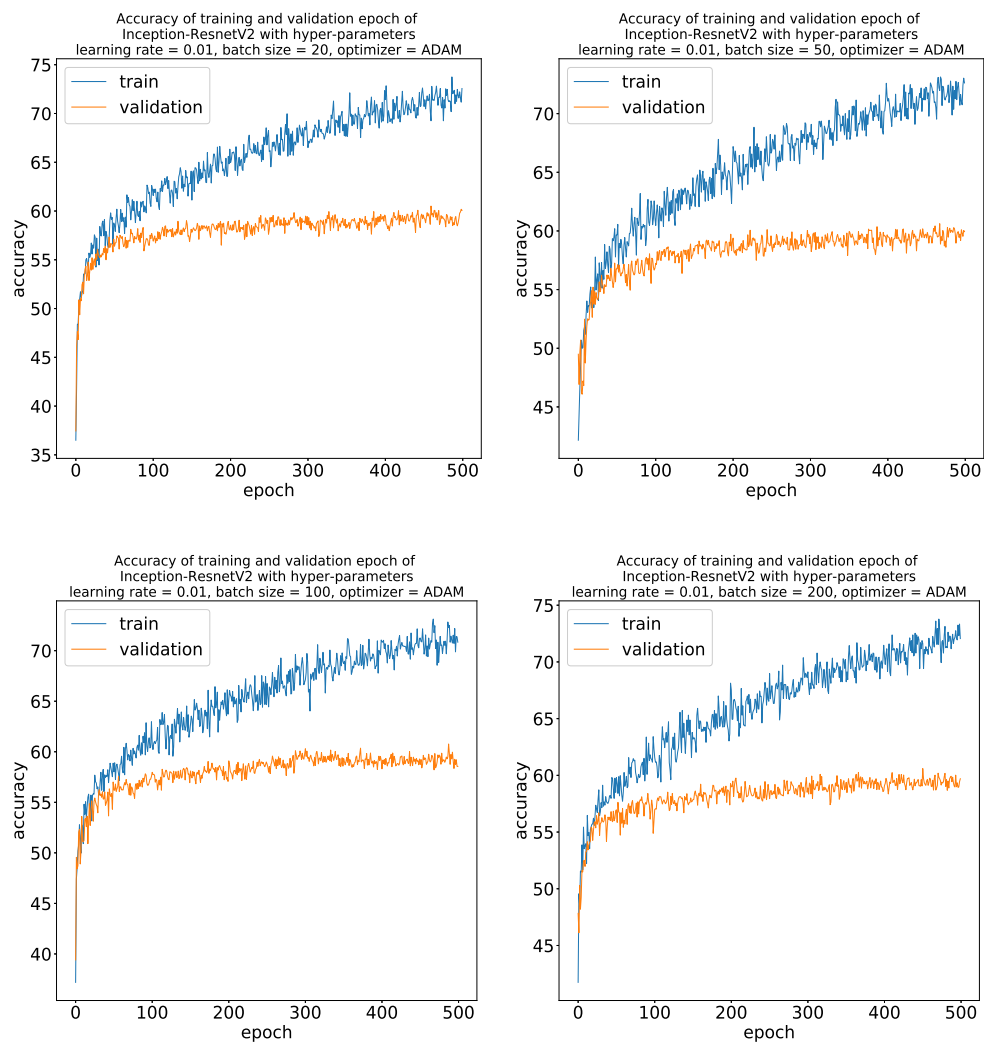


Figure 5.16: Learning curves for Inception-ResnetV2 with ADAM, learning rate = 0.01, epochs = 500 and different batch sizes.

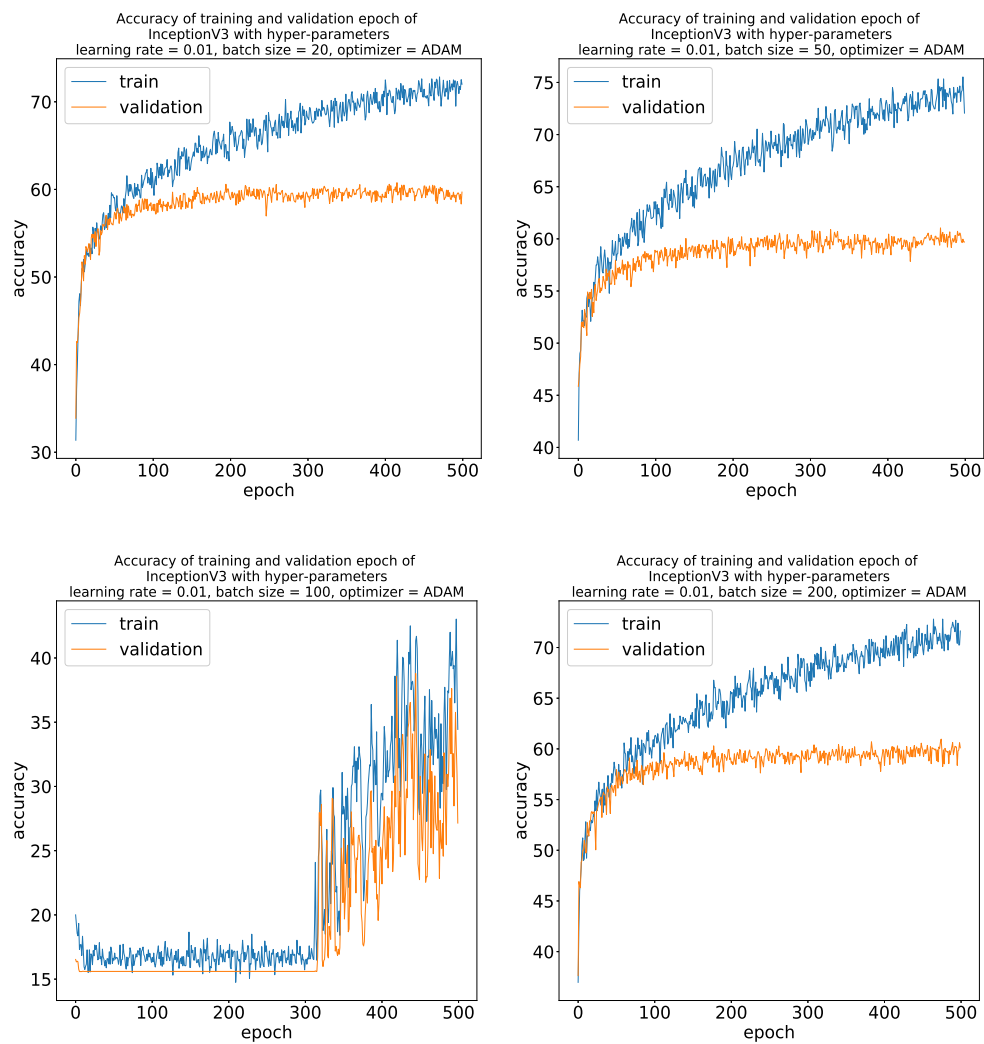


Figure 5.17: Learning curves for InceptionV3 with ADAM, learning rate = 0.01, epochs = 500 and different batch sizes.

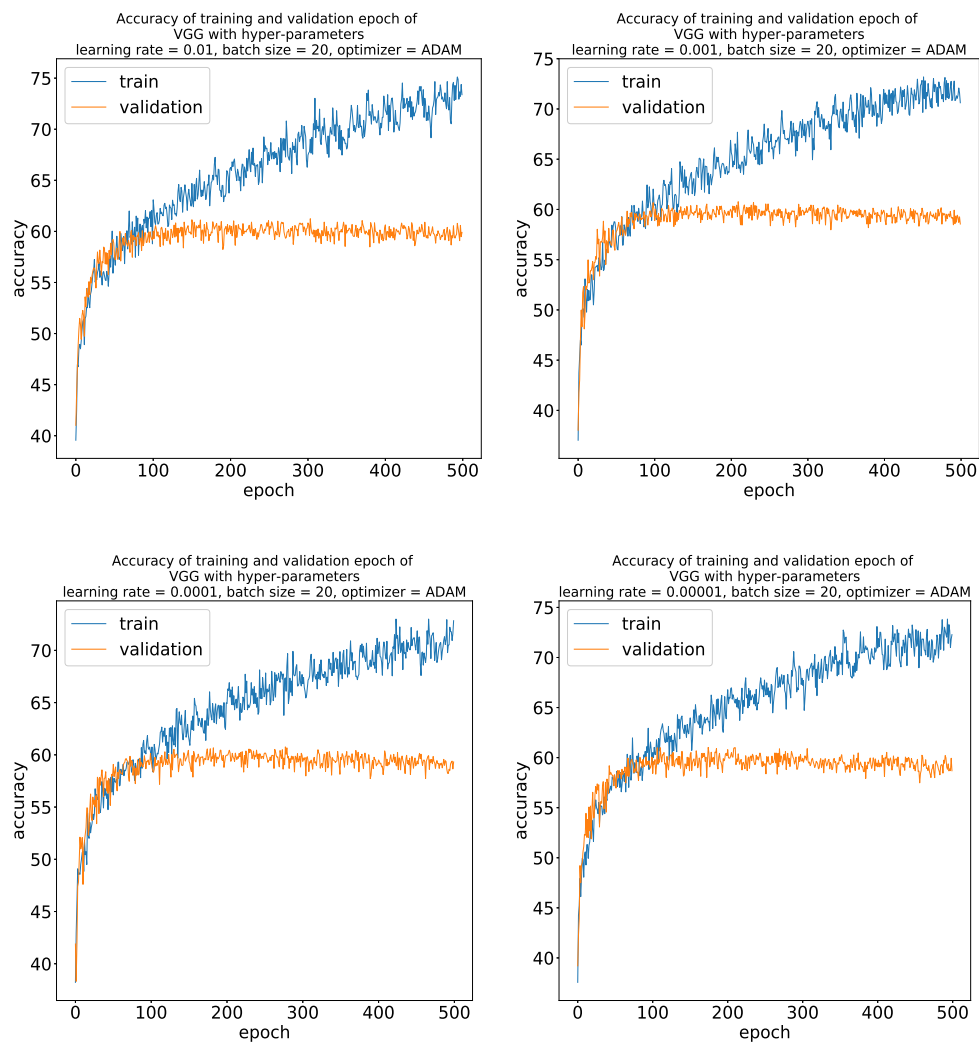


Figure 5.18: Learning curves for VGG with ADAM, epochs = 500, batch size = 20 and different learning rates.

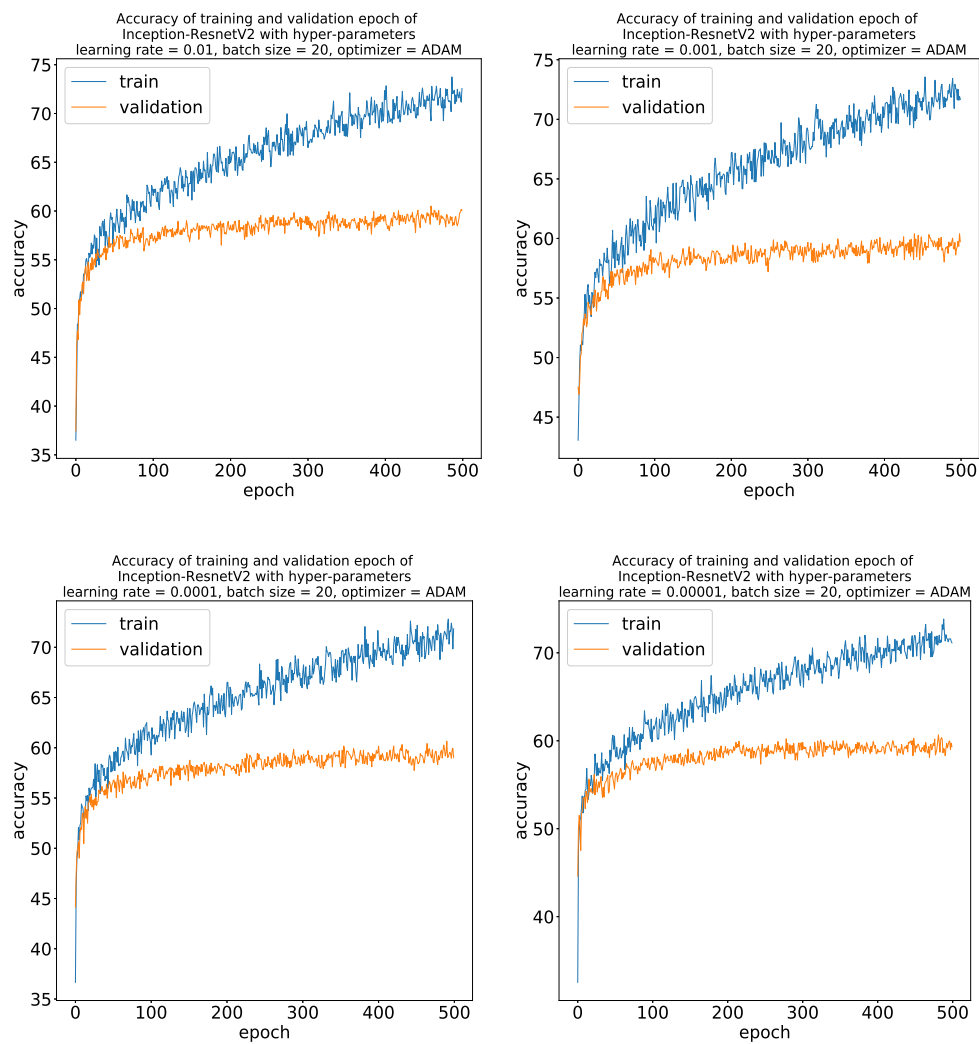


Figure 5.19: Learning curves for Inception-ResnetV2 with ADAM, epochs = 500, batch size = 20 and different learning rates.

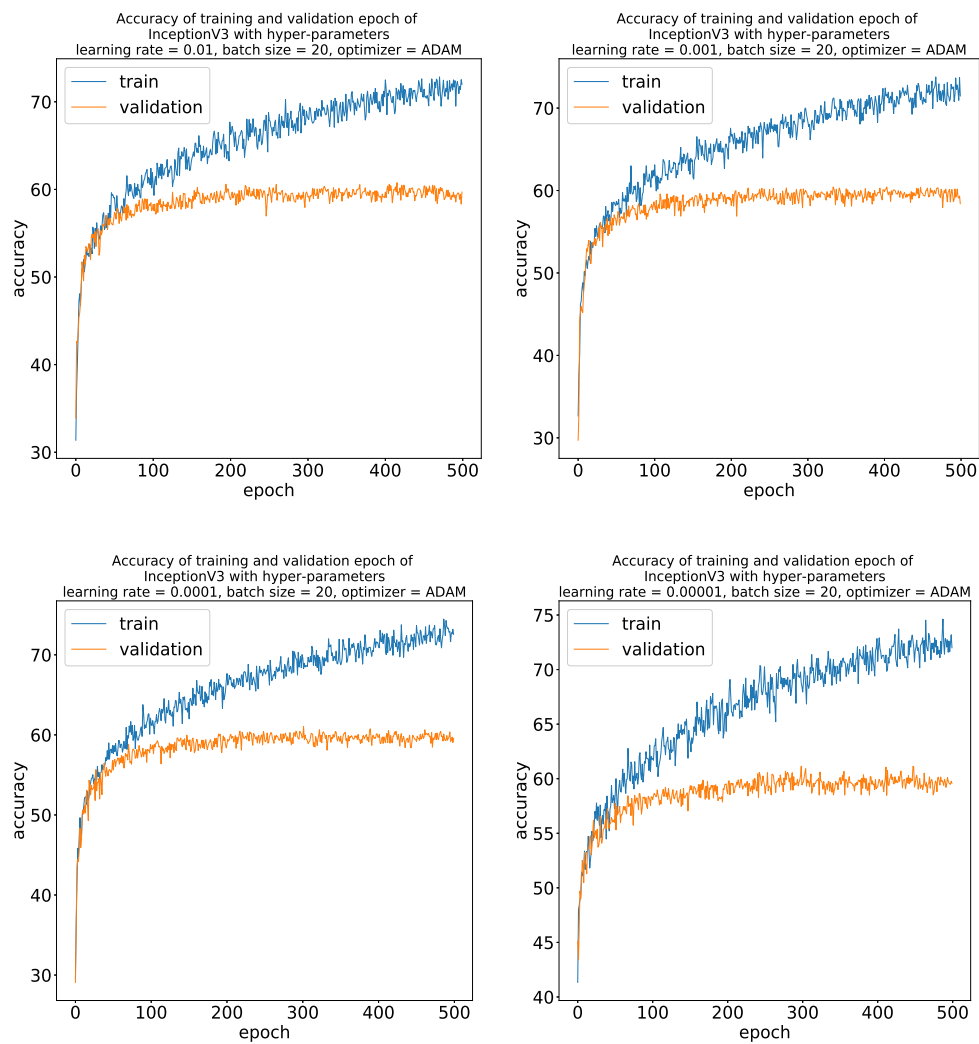


Figure 5.20: Learning curves for InceptionV3 with ADAM, epochs = 500, batch size = 20 and different learning rates.

Table 5.7: Classifier Accuracy for various batch size of the classifier Inception-ResnetV2

Batch size	Training Accuracy	Validation Accuracy	Maximum Training Accuracy	Maximum Validation Accuracy
100	70.84	58.55	73.13	60.77
200	72.06	59.70	73.78	60.60
50	72.62	59.94	73.13	60.62
20	72.56	60.07	73.75	60.51

Table 5.8: Classifier Accuracy for various batch size of the classifier InceptionV3

Batch size	Training Accuracy	Validation Accuracy	Maximum Training Accuracy	Maximum Validation Accuracy
100	34.44	27.15	43.03	38.80
200	71.63	60.11	72.81	60.97
50	72.06	59.70	75.53	61.05
20	72.00	59.69	72.84	60.77

by varying several hyper-parameters this image dataset. However, AlexNet couldn't learn the representation of weather from the images. Therefore, we performed transfer learning of three popular neural network architectures, namely, VGG16, InceptionV3, Inception-ResnetV2. We have also performed extensive experiments by varying hyper-parameters of transfer learning these architectures. Our classifiers reported about 60% accuracy on our dataset.

Table 5.9: Classifier Accuracy for various learning rate of the classifier VGG

Learning rate	Training Accuracy	Validation Accuracy	Maximum Training Accuracy	Maximum Validation Accuracy
0.00001	72.28	58.69	73.84	61.02
0.0001	72.84	59.32	73.00	60.76
0.001	70.62	58.55	73.19	60.77
0.01	73.38	59.90	75.12	61.24

Table 5.10: Classifier Accuracy for various learning rate of the classifier Inception-ResnetV2

Learning rate	Training Accuracy	Validation Accuracy	Maximum Training Accuracy	Maximum Validation Accuracy
0.00001	71.13	59.33	73.84	60.67
0.0001	71.84	59.02	72.81	60.66
0.001	71.66	59.73	73.56	60.39
0.01	72.56	60.07	73.75	60.51

Table 5.11: Classifier Accuracy for various learning rate of the classifier InceptionV3

Learning rate	Training Accuracy	Validation Accuracy	Maximum Training Accuracy	Maximum Validation Accuracy
0.00001	72.00	59.60	74.63	61.14
0.0001	72.56	59.57	74.47	61.06
0.001	71.47	58.40	73.78	60.44
0.01	72.00	59.69	72.84	60.77



# Chapter 6

## Conclusion

In this thesis, we have addressed two Flickr data mining problems. They are predicting the aesthetic rating of a location from Flickr metadata and detecting weather condition of a Flickr image using their wild tags. Although estimating scenic beauty from images have been studied for quite a while, they required setting up specialized cameras at certain locations. At the same time, quite a lot of data mining studies have been conducted on Flickr photos and their data, none of them concentrates on either aesthetic rating prediction or Flickr social metadata. At the same time, weather detection from images have been studied for quite a while. However, none of them concentrated on wild Flickr tags. From that perspective, this study provides newer insights on these two problems.

In order to address aesthetic rating prediction problem, we gathered two datasets that contain the locations of Rome and Paris and their corresponding aesthetic rating ground truths. Later on, we trained several classifiers and applied ensemble methods. We achieved as much as 81% accuracy on Rome dataset from J48 decision tree classifier when it was ensembled using Boosting technique. Similarly, the best accuracy of 71% on Paris dataset was reported when Random Tree classifier was ensembled with bagging method. On the other hand, we gathered a dataset of 70k images with their wild weather tags and trained several variations of neural networks to obtained about 60% accuracy in correctly predicting the wild weather tag of a Flickr image.

In future, textual reviews obtained from TripAdvisor and Flickr can be used to perform sentiment analysis and incorporate users' sentiment in predicting location aesthetic rating. Addition-

---

ally, Flickr raw images might be exploited to capture the aesthetic rating of a location. In case of weather detection from Flickr image and its wild tags further studies can include training large neural networks like VGG16 from scratch. Also, weather cue segmentation as performed in [41] can be applied to help the classifiers improve improve their performances.

# Bibliography

- [1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 1–9. [Online]. Available: <https://doi.org/10.1109/CVPR.2015.7298594>
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, 2012, pp. 1106–1114. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>
- [3] “20 interesting Flickr Stats(December 2018),” <https://expandedramblings.com/index.php/flickr-stats/>, 2019, [Online; accessed 02-Jan-2019].
- [4] E. Spyrou and P. Mylonas, “Analyzing flickr metadata to extract location-based information and semantically organize its photo content,” *Neurocomputing*, vol. 172, pp. 114–133, 2016. [Online]. Available: <https://doi.org/10.1016/j.neucom.2014.12.104>
- [5] N. Kalidindi, A. Le, J. Picone, L. Zheng, H. Yaqin, and V. Rudis, “Scenic beauty estimation of forestry images,” in *Southeastcon’97. Engineering new New Century, Proceedings. IEEE*. IEEE, 1997, pp. 337–339.

- [6] S. D. Bergen, C. A. Ulbricht, J. L. Fridley, and M. A. Ganter, "The validity of computer-generated graphic images of forest landscape," *Journal of Environmental Psychology*, vol. 15, no. 2, pp. 135–146, 1995.
- [7] Z. Bulut and H. Yilmaz, "Determination of landscape beauties through visual quality assessment method: a case study for kemaliye (erzincan/turkey)," *Environmental Monitoring and Assessment*, vol. 141, no. 1-3, pp. 121–129, 2008.
- [8] U. Y. Ozkan, "Assessment of visual landscape quality using ikonos imagery," *Environmental monitoring and assessment*, vol. 186, no. 7, pp. 4067–4080, 2014.
- [9] D. Quercia, R. Schifanella, and L. M. Aiello, "The shortest path to happiness: recommending beautiful, quiet, and happy routes in the city," in *25th ACM Conference on Hypertext and Social Media, HT '14, Santiago, Chile, September 1-4, 2014*, 2014, pp. 116–125. [Online]. Available: <https://doi.org/10.1145/2631775.2631799>
- [10] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu, and T. Miyahara, "Rainy weather recognition from in-vehicle camera images for driver assistance," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*. IEEE, 2005, pp. 205–210.
- [11] M. Roser and F. Moosmann, "Classification of weather situations on single color images," in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 798–803.
- [12] C. Lu, D. Lin, J. Jia, and C. Tang, "Two-class weather classification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 3718–3725. [Online]. Available: <https://doi.org/10.1109/CVPR.2014.475>
- [13] Z. Zhang and H. Ma, "Multi-class weather classification on single images," in *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015*, 2015, pp. 4396–4400. [Online]. Available: <https://doi.org/10.1109/ICIP.2015.7351637>

- [14] Q. Li, Y. Kong, and S. Xia, "A method of weather recognition based on outdoor images," in *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications, Volume 2, Lisbon, Portugal, 5-8 January, 2014*, 2014, pp. 510–516. [Online]. Available: <https://doi.org/10.5220/0004724005100516>
- [15] M. Elhoseiny, S. Huang, and A. M. Elgammal, "Weather classification with deep convolutional neural networks," in *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015*, 2015, pp. 3349–3353. [Online]. Available: <https://doi.org/10.1109/ICIP.2015.7351424>
- [16] "35 amazing tripadvisor statistics and facts (December 2018)," <https://expandedramblings.com/index.php/tripadvisor-statistics/>, 2019, [Online; accessed 02-Jan-2019].
- [17] "The world's largest travel site. Know better. Book better. Go better." <https://www.tripadvisor.com/>, 2019, [Online; accessed 02-Jan-2019].
- [18] "Expedia," <https://www.expedia.com/>, 2019, [Online; accessed 02-Jan-2019].
- [19] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 2009, pp. 248–255. [Online]. Available: <https://doi.org/10.1109/CVPRW.2009.5206848>
- [20] Y. Arase, X. Xie, T. Hara, and S. Nishio, "Mining people's trips from large scale geo-tagged photos," in *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*, 2010, pp. 133–142. [Online]. Available: <https://doi.org/10.1145/1873951.1873971>
- [21] A. Popescu and G. Grefenstette, "Deducing trip related information from flickr," in *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, 2009, pp. 1183–1184. [Online]. Available: <https://doi.org/10.1145/1526709.1526919>

- [22] G. Cai, C. Hio, L. Bermingham, K. Lee, and I. Lee, “Mining frequent trajectory patterns and regions-of-interest from flickr photos,” in *47th Hawaii International Conference on System Sciences, HICSS 2014, Waikoloa, HI, USA, January 6-9, 2014*, 2014, pp. 1454–1463. [Online]. Available: <https://doi.org/10.1109/HICSS.2014.188>
- [23] S. Shafique and M. E. Ali, “Recommending most popular travel path within a region of interest from historical trajectory data,” in *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, MobiGIS 2016, Burlingame, CA, USA, October 31, 2016*, 2016, pp. 2–11. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3004728>
- [24] A. Popescu, G. Grefenstette, and P. Moëllic, “Mining tourist information from user-supplied collections,” in *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, 2009, pp. 1713–1716. [Online]. Available: <https://doi.org/10.1145/1645953.1646211>
- [25] S. Jain, S. Seufert, and S. J. Bedathur, “Antourage: mining distance-constrained trips from flickr,” in *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, 2010, pp. 1121–1122. [Online]. Available: <https://doi.org/10.1145/1772690.1772834>
- [26] K. H. Lim, “Recommending tours and places-of-interest based on user interests from geo-tagged photos,” in *Proceedings of the 2015 ACM SIGMOD PhD Symposium, Melbourne, VIC, Australia, May 31 - June 04, 2015*, 2015, pp. 33–38. [Online]. Available: <https://doi.org/10.1145/2744680.2744693>
- [27] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, “Personalized tour recommendation based on user interests and points of interest visit durations,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2015, pp. 1778–1784. [Online]. Available: <http://ijcai.org/Abstract/15/253>

- [28] A. Majid, L. Chen, G. Chen, H. T. Mirza, I. Hussain, and J. Woodward, "A context-aware personalized travel recommendation system based on geotagged social media data mining," *International Journal of Geographical Information Science*, vol. 27, no. 4, pp. 662–684, 2013. [Online]. Available: <https://doi.org/10.1080/13658816.2012.696649>
- [29] L. Chen and A. Roy, "Event detection from flickr data through wavelet-based spatial analysis," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, 2009, pp. 523–532. [Online]. Available: <https://doi.org/10.1145/1645953.1646021>
- [30] N. Nitta, Y. Kumihashi, T. Kato, and N. Babaguchi, "Real-world event detection using flickr images," in *MultiMedia Modeling - 20th Anniversary International Conference, MMM 2014, Dublin, Ireland, January 6-10, 2014, Proceedings, Part II*, 2014, pp. 307–314. [Online]. Available: [https://doi.org/10.1007/978-3-319-04117-9\\_29](https://doi.org/10.1007/978-3-319-04117-9_29)
- [31] S. Van Canneyt, S. Schockaert, O. Van Laere, and B. Dhoedt, "Time-dependent recommendation of tourist attractions using flickr," in *23rd benelux conference on artificial intelligence (bnaic 2011)*, 2011.
- [32] L. Hollenstein and R. Purves, "Exploring place through user-generated content: Using flickr tags to describe city cores," *J. Spatial Information Science*, vol. 1, no. 1, pp. 21–48, 2010. [Online]. Available: <https://doi.org/10.5311/JOSIS.2010.1.3>
- [33] A. Khosla, A. D. Sarma, and R. Hamid, "What makes an image popular?" in *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, 2014, pp. 867–876. [Online]. Available: <https://doi.org/10.1145/2566486.2567996>
- [34] M. Rehman, M. Iqbal, M. Sharif, and M. Raza, "Content based image retrieval: survey," *World Applied Sciences Journal*, vol. 19, no. 3, pp. 404–412, 2012.
- [35] N. Singhai and S. K. Shandilya, "A survey on: content based image retrieval systems," *International Journal of Computer Applications*, vol. 4, no. 2, pp. 22–26, 2010.

- [36] J. Othman, “Assessing scenic beauty of nature-based landscapes of fraser’s hill,” *Procedia Environmental Sciences*, vol. 30, pp. 115–120, 2015.
- [37] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Studying aesthetics in photographic images using a computational approach,” in *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part III*, 2006, pp. 288–301. [Online]. Available: [https://doi.org/10.1007/11744078\\_23](https://doi.org/10.1007/11744078_23)
- [38] Y. Lu and C. Shahabi, “An arc orienteering algorithm to find the most scenic path on a large-scale road network,” in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Bellevue, WA, USA, November 3-6, 2015*, 2015, pp. 46:1–46:10. [Online]. Available: <https://doi.org/10.1145/2820783.2820835>
- [39] Y. Lu, G. Jossé, T. Emrich, U. Demiryurek, M. Renz, C. Shahabi, and M. Schubert, “Scenic routes now: Efficiently solving the time-dependent arc orienteering problem,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, 2017, pp. 487–496. [Online]. Available: <https://doi.org/10.1145/3132847.3132874>
- [40] K. H. Lim, “Recommending tours and places-of-interest based on user interests from geo-tagged photos,” in *Proceedings of the 2015 ACM SIGMOD PhD Symposium, Melbourne, VIC, Australia, May 31 - June 04, 2015*, 2015, pp. 33–38. [Online]. Available: <https://doi.org/10.1145/2744680.2744693>
- [41] X. Li, Z. Wang, and X. Lu, “A multi-task framework for weather recognition,” in *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, 2017, pp. 1318–1326. [Online]. Available: <https://doi.org/10.1145/3123266.3123382>
- [42] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “SUN database: Large-scale scene recognition from abbey to zoo,” in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, 2010, pp. 3485–3492. [Online]. Available: <https://doi.org/10.1109/CVPR.2010.5539970>



- [43] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “Labelme: A database and web-based tool for image annotation,” *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, 2008. [Online]. Available: <https://doi.org/10.1007/s11263-007-0090-8>
- [44] W. Chu, X. Zheng, and D. Ding, “Image2weather: A large-scale image dataset for weather property estimation,” in *IEEE Second International Conference on Multimedia Big Data, BigMM 2016, Taipei, Taiwan, April 20-22, 2016*, 2016, pp. 137–144. [Online]. Available: <https://doi.org/10.1109/BigMM.2016.9>
- [45] H. Izadinia, B. C. Russell, A. Farhadi, M. D. Hoffman, and A. Hertzmann, “Deep classifiers from image tags in the wild,” in *Proceedings of the 2015 Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions, MMCommons 2015, Brisbane, Australia, October 30, 2015*, 2015, pp. 13–18. [Online]. Available: <https://doi.org/10.1145/2814815.2814821>
- [46] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li, “YFCC100M: the new data in multimedia research,” *Commun. ACM*, vol. 59, no. 2, pp. 64–73, 2016. [Online]. Available: <http://doi.acm.org/10.1145/2812802>
- [47] M. J. Huiskes, B. Thomee, and M. S. Lew, “New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative,” in *Proceedings of the 11th ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2010, Philadelphia, Pennsylvania, USA, March 29-31, 2010*, 2010, pp. 527–536. [Online]. Available: <https://doi.org/10.1145/1743384.1743475>
- [48] “The App Garden,” <https://www.flickr.com/services/api/>, 2019, [Online; accessed 03-Jan-2019].
- [49] M. A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: an update,” *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009. [Online]. Available: <https://doi.org/10.1145/1656274.1656278>

- [50] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, 2009. [Online]. Available: <https://doi.org/10.1016/j.ipm.2009.03.002>
- [51] J. Finlay, R. Pears, and A. M. Connor, “Synthetic minority over-sampling technique (SMOTE) for predicting software build outcomes,” in *The 26th International Conference on Software Engineering and Knowledge Engineering, Hyatt Regency, Vancouver, BC, Canada, July 1-3, 2013.*, 2014, pp. 546–551.
- [52] Z. Chen, F. Yang, A. Lindner, G. Barrenetxea, and M. Vetterli, “How is the weather: Automatic inference from images,” in *19th IEEE International Conference on Image Processing, ICIP 2012, Lake Buena Vista, Orlando, FL, USA, September 30 - October 3, 2012*, 2012, pp. 1853–1856. [Online]. Available: <https://doi.org/10.1109/ICIP.2012.6467244>
- [53] S. G. Narasimhan, C. Wang, and S. K. Nayar, “All the images of an outdoor scene,” in *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part III*, 2002, pp. 148–162. [Online]. Available: [https://doi.org/10.1007/3-540-47977-5\\_10](https://doi.org/10.1007/3-540-47977-5_10)
- [54] N. Jacobs, W. Burgin, R. Speyer, D. Ross, and R. Pless, “Adventures in archiving and using three years of webcam images,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2009, Miami, FL, USA, 20-25 June, 2009*, 2009, pp. 39–46. [Online]. Available: <https://doi.org/10.1109/CVPRW.2009.5204185>
- [55] “Weather Underground,” <https://www.wunderground.com/>, 2018, [Online; accessed 30-Dec-2018].
- [56] “Weather Central,” <http://www.wxc.com/>, 2018, [Online; accessed 30-Dec-2018].
- [57] Z. Zhang, H. Ma, H. Fu, and C. Zhang, “Scene-free multi-class weather classification on single images,” *Neurocomputing*, vol. 207, pp. 365–373, 2016. [Online]. Available: <https://doi.org/10.1016/j.neucom.2016.05.015>

- [58] “Weather UnderGround,” <https://www.flickr.com/services/api/misc.urls.html>, 2019, [Online; accessed 02-Jan-2019].
- [59] “Transfer Learning,” <http://cs231n.github.io/transfer-learning/>, 2019, [Online; accessed 03-Jan-2019].
- [60] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [61] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.90>
- [62] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017, pp. 4278–4284. [Online]. Available: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806>