The impact of image filters on machine classification and human perception of weather conditions

by

C.M. Valsamos

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday February 28, 2019 at 14:30 PM.

Student number:	4613554	
Thesis committee:	Prof. Dr. Martha Larson	TU Delft (supervisor)
	Dr. Hayley Hung,	TU Delft
	Dr. Michael Alexander Riegler	University of Oslo

Abstract

Nowadays with the growth of social media, users upload millions of photos in different platforms online. Researchers in the field of computer vision devote their time and effort to analyze images in order to gain valuable insight. Data analysis and classification can be impeded by different factors. One of which is the image filters that are studied in this work. People greatly change the appearance of their photos by adding filters in order to make them more appealing. *Instagram* is arguably one of the most popular social media platforms online. With the platform's growth, filtering images has also become more popular. In this thesis a subset of Instagram filters has been selected in order to study their impact with a series of experiments. To our knowledge, no mention has been made of image filters' impact in prior work, in the domain of machine classification and human perception.

Image filters can create many challenges depending on the application they are used in. In this thesis, focus has been given on classification of weather conditions. Systems have been designed to receive images and accurately identify the weather conditions that exist in them solely using visual features and no prior knowledge. In weather forecasting a lot of resources are spent in order to study past and current weather conditions so as to predict the state of the weather in the future. Gathering and documenting weather related information can be aided by these aforementioned systems. However, if researchers would like to use social images to extract insight, they need to change their approach accordingly.

As it is documented in the following chapters, dealing with these photos can be problematic and can cause huge decline in performance. For this reason, the algorithmic design has been changed by using different techniques inspired from the domain of *Adversarial Machine Learning* to measure their effect. In addition to machine classification, filters can influence human perception as well. A study is conducted that measures the impact filters have on the ability of humans identifying the weather conditions in images. From the quantitative and qualitative analysis of the results several key findings are extracted regarding the effect of filters and the visual cues that are used by people. People have identified certain visual cues that have not been encoded in the classifier such as the type of clothing people are wearing. Instead much simpler features have been engineered and the performance of the classifier is still quite high.

Acknowledgements

This MSc thesis constitutes the closing chapter in a long and educating journey at TU Delft. Throughout the process of working on this project, I received the support from many people. I would like to begin by expressing my sincere gratitude towards my supervisor Martha Larson for her guidance during the course of this project. Martha was very kind, patient and she always showed great interest in the work that is being carried out. After the many discussions that we had I was able to think critically about problems and issues that appeared during this work. Furthermore, she always challenged me to think further and search deeper during our cooperation. Without her guidance this thesis would not be possible. I would also like to thank Michael Riegler, Giulia Boato and Duc-Tien Dang-Nguyen who offered help during our meetings and participated in the pilot study and provided useful feedback. Closing, I would like to thank my family and friends for their constant support and encouragement. In crucial moments I was able to move forward and continue working towards finalizing this project. Special thanks to my parents who always showed their support and gave me the freedom and means to pursue my studies in TU Delft.

C.M. Valsamos
Delft, February 2019

Contents

List of Figures

List of Tables

1

Introduction

1.1. Background

With the growth of social media and platforms such as Instagram, millions of photos are uploaded online every day. Users tend to edit their photos before they upload them online in order to make them more unique and distinguishable [2]. This phenomenon has led researchers to deal with these image enhancements or filters in various and different ways. There are a lot of domains that deal with filters from different angles. One of them is image forgery detection which has the goal of detecting photos that have been edited. A different domain is photo filter enhancement. Researchers have recognized the fact that there is a big set of filters and it is not always for a user to choose the most suitable filter for each case. In their work, Bakhshi *et al.* [2] carry out research in order to understand image enhancements in social media. With their work a lot of questions are answered including who are the mobile photographers and why/when do they use filters. In addition, with their interviews they document and reveal how casual photographers differ from more serious users.

Computer vision systems need to deal with photos that have been manipulated by users. These photos can introduce noise into the system that can reduce its performance. In [12], the authors test the effect that image enhancements have on a geo-location system. In their experiments, they reveal that up to 19% of images whose location would have been identified, it was undetectable after enhancement. This work investigates and documents the interaction between image manipulation and a machine learning system and can educate users on how to protect themselves from malicious people that can retrieve their location.

Researchers are continuously developing new systems or building upon prior research in order to improve current ones. One problem that researchers have tried to tackle is weather forecasting. In prior literature, researchers have designed classifiers in order to detect the weather class (cloudy, sunny, rainy, etc.) or approximate certain weather attributes (e.g. temperature, humidity) [11, 25, 47, 57]. These systems use pre-existing knowledge in a statistical time series manner in order to accurately estimate the weather. For instance, information from four days is encoded in order to determine the weather conditions that are going to take place on the fifth day. In contrast with the aforementioned domain, where researchers use prior knowledge there is a closely related domain that uses images and does not require prior knowledge. In this domain researchers create systems which can forecast weather by looking into and exploiting visual cues in images. The research that is reported in this thesis extends the work of [14] where they create a visual classifier that detects six types of weather, namely, cloudy, sunny, foggy, rainy, snowy and other. In this thesis, this system is replicated and tested with enhanced images. The results are reported in the following chapters of this thesis.

1.2. Research questions

In this section the research questions that are going to be addressed in the following chapters are formally stated. The interaction between image filters and weather conditions depicted in images is investigated from different angles (machine, adversarial machine learning and human) with each research question.

***RQ1**: How does image enhancement affect the performance of the weather class classifier?*

When referring to image enhancement, three different types of manipulations-filters have been included in the experiments, namely, Grayscale, Toaster and Hudson. In addition, the classifier was tested with an ensemble of manipulations. It is important to note that one image is manipulated only with one type of manipulation each time. No combination of enhancements have been used on the same image simultaneously. This research question is answered in chapter 5.

***RQ2**: Can we improve the performance of the weather class classifier by using techniques motivated by Adversarial Machine Learning?*

The second research question has been designed in order to test different approaches that can benefit the classifier from the potential drop in its detection accuracy from the first research question. As part of this thesis, different approaches have been tested all inspired from adversarial machine learning. These include: Adversarial training, feature squeezing and detection of enhancement. These aforementioned techniques are applied separately from each other. Lastly, an ensemble of adversarial machine learning techniques is applied in order to investigate how it impacts the classifier's performance. All of the experiments run are extensively discussed in chapter 6.

***RQ3**: How do filters affect users' perception in detecting the weather conditions depicted in images?*

One goal of this thesis is to document how image enhancement impacts not only the classifier but also people. To this end, a crowdsourcing study is designed in order to measure peoples' responses to image enhancement and their perception of the weather depicted in images. Similarly, with the first research question the image enhancements that are tested are: Grayscale, Hudson and Toaster respectively. In chapter 7 the study is presented and there is quantitative and qualitative investigation of the results.

1.3. Technical overview

Several authors in prior literature have developed systems that are able to identify the weather depicted in images. Visual weather prediction [38, 55] differs from prior work that uses prior knowledge in a statistical time series [11] in order to predict feature events. In this work, the *Image2weather* [13] dataset is used that combines images from *Flickr* with weather attributes such as temperature, humidity and weather conditions (sunny, cloudy, foggy, rainy etc.) extracted from the *Weather Underground API*. Due to having imbalanced and sparse data for certain conditions, images which are labelled as 'cloudy' and 'sunny' have been selected and treated as a binary classification problem. Although, weather related information such as temperature is available, this type of information has not been used here. The dataset passes through different preprocessing stages, which include the implementation of image forgery detection algorithms. For the purposes of this thesis, the dataset needs to be cleansed from already enhanced images before measuring the impact of Instagram filters. This step is needed in order to compare and contrast the performance achieved by the classifier on images that have been filtered and unmodified images.

In addition, the feature design from [13] is also replicated in order to design an algorithm and have a performance baseline that can be used to measure the impact of image filters. The feature set is a 516-D vector that consists of primarily visual features but also temporal information. The features that are used in the algorithm in [13] are: The hour and month the picture was taken, the RGB histogram, the intensity histogram, cloud features [36], the *Local Binary Pattern* histogram, haze features and contrast features [38]. Certain features use all the information of the image and others focus on the sky. In order to determine if a pixel belongs to the sky or not a sky segmentation algorithm has been tuned accordingly using the *AMOS* dataset [30] which is a publicly available source photos that have been taken by static cameras.

Photos have become one of the most popular ways of communicating in online social media. Platforms such as Instagram have grown exponentially. Subsequently, the number of images that is uploaded daily online has also risen. These images are used by researchers and they are incorporated into algorithms in order to extract useful insight. However, these images are often pre-enhanced introducing variance which systems may not be able to handle. In this thesis, we test how a visual weather dataset from the prior literature can handle a set that consists of images that have been edited with filters from Instagram. *Instagram* has attracted a lot of attention from the research community. For instance, Bakhshi *et al.* [2] analyze users' behaviour and investigate how filters impact engagement. Several interesting findings have been revealed by their work which help guide the research carried out here. Instagram has a very rich and diverse set of filters out of which a subset of Instagram filters has been selected to be included in the experiments. The filter set consists of *Toaster* and *Hudson* which are two very popular Instagram filters and *Grayscale* which is a simple filter that removes the colour from images and gives them a more vintage look. These filters have been selected, in consideration of their strength and similarity with each other. These two factors have been calculated with the help of the structural similarity (*SSIM*) index. It is important to include in the experiments filters with mild and strong effect in order to compare their impact on the classifier and people.

In machine learning, systems are often tested and evaluated with cross-validation. This ensures that the system can handle different types of data and the choice of the training set does not majorly influence the results. Overfitting is an issue in statistical models in machine learning that occurs when the algorithm can deal with very specific cases and cannot generalize and handle other types of data. In computer vision, the training data and the data that the algorithm will be ultimately be assigned to label can vary significantly. For this purpose, algorithms are often trained by receiving as input edited-slightly altered versions of the original training set. In prior literature, this practise is called *data augmentation* and it is widely used in different problems, not exclusively in the field of Computer Vision. Focus has been put on testing different adversarial machine learning techniques and their effect on the system. There are: adversarial training, feature squeezing, detection of enhancement and an ensemble of techniques.

Finally, we are interested in investigating the impact of different filters on human perception of weather conditions in images. For this purpose, a crowdsourcing survey is designed and shared online. From this study, a data collection is gathered that is used in order to conduct quantitative and qualitative analysis. Several interesting findings are made regarding the strength of filters and the ability of people to estimate the weather at the time that a photo was taken. The quantitative analysis has been conducted with the calculation of measures such as *Accuracy* and *Fleiss' Kappa*. Furthermore, in their answers, participants highlight several visual cues that helped them when the colour was absent or changed.

1.4. Motivation

There are online real estate platforms where realtors often upload online images of houses and related information in order to attract potential buyers and tenants. In a preliminary study, it was seen that these images are often edited in many different ways. A house dataset was shared with us that contained photos of houses and surroundings. This house dataset was consisted of pairs of images that were different versions of images before and after editing. It is important to mention that these images have not been edited in a way that majorly changes the layout of the house. For instance, no fireplaces or pools have been artificially edited in the photos. Instead, the photos have been edited by changing how sharp they are and by removing cords of coffee machines and other electrical tools. The most major change is the appearance of the sky in outdoor images which was converted from cloudy to sunny. This has been done potentially, because buyers are more likely to be attracted to a house because of the nice weather in these photos. Changing the sky that is depicted in images majorly affects the user's perception of the weather. The dataset allowed us to identify weather as an important aspect of the image that can be changed and also may have an effect on how much users can trust an image.

Real estate photos are an example of a case in which manipulation could prove harmful if it impacts users' interpretations. It is very important to determine what is the impact of these filters and understand how heavily the users are deceived online. The deception of users online can arise in many different scenarios some more serious than others. The image enhancements that are included in the experiments are more subtle and more likely to be found online in platforms such as Instagram. To our knowledge no prior work has explored the interaction between image enhancements and weather prediction.

During the preliminary study, this house dataset was very interesting to go through and understand how these images were changed. This dataset was of high importance because it contained images that had been edited with a specific purpose in mind which is selling a house and attracting possible customers. Unfortunately, the connection between these images and weather records was sparse. One of the advantages of the actual dataset used here in this work is that it contains images from big European cities. In these cities, the weather records not only can be found easily due to the high number of weather stations but also the weather is recorded more frequently.

Weather data collection can be quite expensive due to different factors that affect the cost to rise. Special equipment is required in order to accurately monitor weather and gather related information without big inconsistencies. In addition, human supervision of these equipment is necessary. The exponential growth of social images offers the chance to researchers to extract useful information such as weather attributes from these photos. However, these photos are often edited and changed in different ways resulting into badly trained models. The work presented here is highly motivated by the portion of photos that is edited online. The use of image enhancements is very popular among users that use various tools to edit, each one with different intents. Even though, these photos constitute the minority of the whole photo dataset, they can still affect the system's performance as it is proved in the following chapters. It is also documented that the impact edited photos have on the system is not negligible, making the necessity of adjustment techniques more important.

1.5. Contributions

In total, five main contributions have been identified and they are listed below:

1. Although, researchers have designed classifiers that detect the type of weather in images no special mention has been made on how enhanced images affect the performance of systems. It is proved in this work that certain effects can have up to 0.17 absolute reduction in accuracy which is not negligible. In the future, authors need to take proper actions to deal with this type of issue.

2. The reader can find in chapter 5 an extensive analysis of 22 Instagram filters that was run in order to select the most suitable filters to experiment with in this study. Several useful information can be found regarding the strength and similarity of filters with each other.

3. It is evident in the experiments that the impact of filters on the classifier's performance is severe. Subsequently, investigation is carried out in order to test the effect of different adversarial techniques. Adversarial training

have been proved to be the most effective way to improved the classifier's performance. All the experiments are documented in detail in order to ensure that the results are transparent.

4. The dataset that is used in this thesis is called *Image2weather* [14] and it contains images from Flickr. As part of this thesis, an additional check was conducted to gather the licenses of images in order to share these with the participants in the crowdsourcing study. As it is seen in this work not all images in that dataset can be shared or adapted. In chapter 3, there is a table listing all images and their type of license. The table was created during this thesis and the numbers may change due to users taking down images.

5. As last contribution of this thesis, several key findings are extracted related to the effect filters have on human in identifying the weather conditions in images.

1.6. Thesis outline

This thesis is organized in 8 chapters, followed by the bibliography and the appendix. This chapter has been the introduction which presents to the reader the topic of this thesis. While reading this chapter the reader can understand the scope and the history of the domain. In addition, the research questions are formally stated in order to present which are the gaps in research that this thesis tries to fill. In the motivation, the path towards the formulation of the research questions is explained. Lastly, it is important to discuss the way this thesis extend the related work and what is its contribution in research.

In the second chapter, prior to the chapters where the technical details and experiments are presented, the reader can understand the background and how prior work is connected with this thesis. The three main domains that have been identified that are related to this thesis are image enhancement, weather prediction and adversarial machine learning. In these domains, several authors have carried out research that is strongly related to this thesis.

In the third chapter, an introduction is being made about the experiments that will be carried out by first describing the dataset *Image2weather* which has been created by other authors in past work. In chapter 3, all of its attributes, source and details are analyzed. This chapter also includes a discussion about the preprocessing steps and explanatory analysis that were carried out. The preprocessing steps were vital for the purpose of this work.

In chapter 4, the pipeline is depicted in order to give to the reader a graphical overview of chapters 4,5 and 6. In the chapter with title Weather Classification, all the information regarding the classifier and features can be found, closing with the evaluation of the system.

The classifier from chapter 4 is tested with edited images in chapter 5. The images are edited with Instagram Filters and in the subsections of that chapter there is a detailed description of the choices that were made regarding the chosen enhancements. Finally, tables are presented with all the results.

Adversarial machine learning is the last chapter of the experiments that were executed with the classifier. In this chapter, the classifier and dataset are adjusted with techniques inspired from the adversarial machine learning domain. The effect of these techniques is extensively discussed in that chapter.

In chapter 7, the design of the crowdsourcing study is presented. As part of this thesis, the effect of filters is tested not only on the machine but also on people. In that chapter the methodology of the framing of the study and the collection of annotations from participants is presented.

This thesis concludes with chapter 8. It is important to reflect on the work that was carried out in this thesis and also realize the possible limitations in the methodology. The work presented in this thesis can be expanded in different directions in the future by other researchers. In the appendix, the reader can find additional information about the methodology and results.

2
Related work

In this section, work that has influenced this thesis is presented. The work that has been carried out here builds upon the work of others in prior literature. A number of works from different domains have helped guide and provided solutions to problems that were encountered in this thesis.

2.1. Image Enhancement

2.1.1. Image Forgery Detection

In prior work, researchers have tackled the challenge of image forgery detection and have defined three different types of image forgery: Copy-move forgery, image splicing and image retouching. Image retouching is when the image is uniformly changed. This can be performed by raising the contrast or brightness of the image or when adding an Instagram filter. Although, image retouching is one of the most common practices in social media, it is the domain that has attracted the least attention. This is evident in [48], where the authors measure the amount of publications for the three main categories up to 2014 and can be seen in figure 2.1. Copy-move and image splicing are often used to cover certain parts of images and hide information. As a result, these types of manipulations can be considered as more malicious compared to image retouching which is often used to make images look better. As a result, researchers have devoted more effort in the first two categories.

In this thesis, emphasis has been given on the third category, image retouching-enhancement because it is considered to be the most popular method of editing photos online among users. In contrast with the first two categories of image forgery, what constitutes image enhancement is more broad. Looking into prior work, researchers have designed systems which are able to detect several types of image enhancement: Brightness adjustment [4, 26], median filtering [7], contrast [26, 50] and sharpening enhancement [4] respectively. These image processing operations generally do not have such a huge impact on the image in terms of *Structural Similarity* (SSIM) index [61] which makes them even more challenging to detect. This is evident in [15] where the SSIM is computed for doctored, retouched and enhanced images respectively.

Two systems have been used in this work from prior literature with the goal of detecting enhancement. Firstly, Avcibas *et al.* [26] create a system where they are able to detect several types of image forgery. These include: Scale-up, Scale-down, Brightness, Contrast, Rotation, Sharpening and Equalization. The system can detect multiple and different types of manipulation, which is really useful to our problem where the type of manipulation is unknown. Regarding the feature design, two types of *Singular Value Decomposition (SVD)* based features are designed and fed into a logistic classifier. Their hypothesis is that natural images involve local linear dependencies that get affected after manipulation. In their paper, the authors present their results in three scenarios, blind, semi-blind and clairvoyant.

The image forgery detection component in this theiss has been enriched by an algorithm that differs from the image forgery detection algorithm presented in [26]. Ding *et al.* [17], create features based on the *Local Binary Pattern* (*LBP*) in order to detect images that have been sharpened. Increasing the sharpness of an image causes an increase in contrast near the edges. With *LBP* the authors perform texture analysis in order to discriminate the manipulated images from the non-manipulated ones in a blind scenario. The features are extracted from the images and fed into a *Support Vector Machine* (*SVM*) classifier.

In engineering, a lot of statistical metrics have been developed in order to understand how two (audio/visual) signals differ with each other and to measure image quality. These include the *Mean Squared Error* (*MSE*) and the *Peak Signal-To-Noise Ratio* (*PSNR*). These two metrics can be calculated when the reference image is available. MSE is the average squared difference between the two signals and can be its formally defined as follows:

$$MSE = \frac{1}{n}\sum_{i-1}^{n}(Y_i - \hat{Y}_i)^2$$

The aforementioned metrics can be easily computed and they measure the image quality on a pixel basis, whereas SSIM calculates the structural difference. In this thesis, SSIM is extensively used in order to measure perceived difference between a pair of photos.

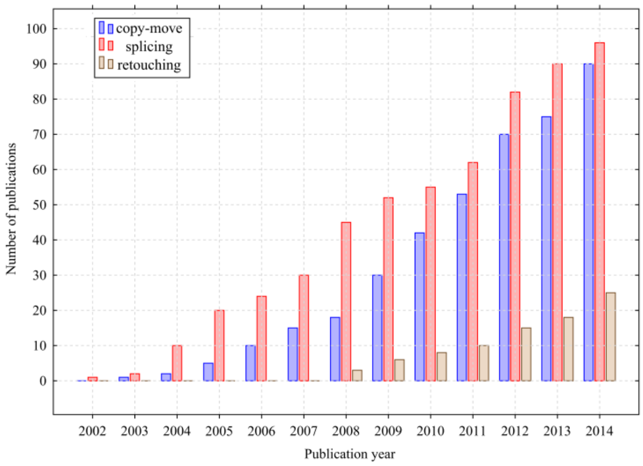


Figure 2.1: Number of publications for each category of image forgery detection in the period of 2002-2014 [48].

Users spend a lot of time online sharing their experiences via text and photos. *Instagram*'s main focus is photo-sharing after the application of photo filters. As a result, users upload millions of images online many of which have been retouched/edited beforehand. Users manipulate their images for different reasons dependent on the context. As part of their research, Bakhshi *et al.* [2] investigate why do people filter images in social images. The authors document user responses and discuss the results. As they describe, their motives can be categorized into the following five categories: Improving aesthetics, adding vintage effects, highlighting objects, manipulating colors and making photos appear more fun and unique. There are a lot of interesting findings in [2] that help guide the research carried out in this thesis. For instance, with their survey, they reveal how professional photographers think and how they differ from casual photographers.

2.1.2. Image Manipulations-Filters

People online tend to spend quite some time experimenting and testing different enhancements in order to decide what is the most suitable filter. For a user is very important to upload an image that is more appealing to their friends and inner cycle. It is becoming increasingly more challenging to distinguish what is real and what has been artificially enhanced due to the popularity of image retouching tools like Adobe Photoshop[1] that allows people with little to no expertise achieve perfect results in photo editing. People spend quite some time navigating all the possible filters and options that are provided by photo editing tools (e.g. Adobe Photoshop, GIMP[2]) and social media (e.g. Instagram [3]).

As mentioned in the previous subsection, researchers detect the use of several enhancements such as brightness and contrast enhancement. Similarly, researchers in a different setting recommend these enhancements in order to make photographs more presentable. Researchers have devoted resources in designing algorithms that recommend image enhancements for each case [6, 16, 21]. In contrast with prior works, in 2017 Sun *et al.* [52], designed a system that chooses the most appropriate filter among twenty two Instagram filters. It differs from all prior works because their algorithm does not recommend image enhancements but rather pre-defined Instagram filters.

In this thesis the selection has been made to experiment with pre-defined Instagram filters instead of image enhancements. In chapter 5, the process of using the Instagram filters in this thesis is explained. Several key insights have been extracted from [52] that are taken into consideration. One potential limitation of using Instagram filters is that some filters can be more popular than others and users choose certain filters more than others. This does not seem to be true. The researchers in [52], graphically present evidence proving that filter selection is diverse among users in a pilot study despite having similar backgrounds. As a result choosing a filter is considered to be subjective and no patterns can be found. In their work, Instagram filters were applied with the help of the free and open source image editor called *GNU Image Manipulation Program* commonly known as *GIMP*. Using GIMP as means of editing photos has many advantages including batch editing but most importantly that it is available to all via its website. This allows to reproduce not only the work of Sun *et al.* [52] but also the work that is carried out in this thesis.

2.2. Photos - user interaction

Photos in online social media have been studied in order to measure how they affect users' behaviour and perception of them. *Selfies* (self-portrait photos) have been the primary focus of much prior work that studies the behaviour of not

[1] https://www.adobe.com/products/photoshopfamily.html
[2] https://www.gimp.org/
[3] https://www.instagram.com/

only the one who is posting them but also the people who view them. The connection between self-perception and selfie-posting has been investigated in [3, 39]. McCain *et al.* [39], report in their findings that excessive selfie posting is correlated with grandiose narcissism. In contrast with [3, 39], authors have dedicated resources investigating this paradigm from its other side. In [32, 56], authors carry out research in order to reveal how do people who view selfies perceive them and how do they influence them. Several key findings can be highlighted. For instance, in [56] the authors found evidence that frequent selfie viewing led to decreased self-esteem. In [32], the authors advised people who post selfies to be aware of their possible effect on their audience. People perceive people within selfies compared to the same people taken by others as less trustworthy, less socially attractive, less open to new experiences, more narcissistic and more extroverted. In [18], the authors found that picture features such as brightness are related to personality traits and implement a system that can accurately predict these.

2.3. Weather prediction

Weather affects every part of our daily lives not only in a psychological level but also in a practical manner as well. Despite meteorologists having the expertise and the proper equipment to make future estimations, weather forecasting still remains very challenging due to different factors that contribute to this problem. Researchers try to innovate and build upon prior work with the goal of improving and making the predictions more accurate. Weather attributes hold significant value to a number of people, not exclusively meteorologists but to people from other disciplines as well. Weather can be described by a diverse and extensive set of values such as: temperature, wind speed, wind direction, humidity, precipitation, etc. Collecting these aforementioned type of attributes can be quite expensive since meteorologists make use of expensive sensors or manual inspection is needed to monitor the weather conditions.

In this century, researchers have experienced a very big and sudden increase in the volume of data. This phenomenon has created a lot of opportunities in a lot of domains. One of which is weather forecasting. The data can be encoded into machine learning systems that have been properly trained with the goal of providing reliable estimates of weather conditions. In prior work, it is evident that a big portion of researchers have focused on creating systems that predict weather attributes by using prior knowledge [11, 25, 47, 57]. Data can be encoded in a statistical time series (e.g. *ARMA*, *ARIMA*) to detect re-occuring patterns across time. In contrast with the previous models, in this thesis no prior knowledge is used in order to make the final decision regarding the weather attributes of a picture. In addition, the majority of features is extracted from the image of interested. Images can be found in abundance online, whereas weather data records can be quite expensive to gather and in certain cases access is restricted. Estimating the weather conditions that are depicted in an image can be a hard task to perform even for humans. Researchers have tackled different problems such as binary or multiclass classification of weather conditions (e.g. sunny, cloudy, rainy, foggy, etc.). Furthermore, focus has been given on the estimation of weather numerical attributes (e.g. temperature and humidity).

2.3.1. Weather condition prediction

Researchers have devoted resources in understanding what visual cues can be used in order to identify the overlaying weather class of a picture. This is a supervised learning problem where the labels are known beforehand and models are designed to distinguish the different classes. The data that are used in prior work belong to one of the following weather classes: sunny, cloudy, snowy, rainy, and foggy-haze. Researchers create systems that can identify two classes of weather conditions such as sunny and cloudy [38] or multiple classes [14, 63]. The weather conditions can be hard to identify solely using visual information and no temporal and geographical related information. Location is a huge factor in weather forecasting since there are different weather climates.

Features that have been used effectively in order to determine the weather class and weather attributes consist of cloud features, local binary pattern (LBP), contrast and haze features. In order to extract certain sky related features sky segmentation is performed. It is important to note that sky segmentation algorithms are often used in weather prediction in order to properly encode visual information. Sky has been proven to be very useful in discriminating the weather properties, since cloudiness can be quantified and used in the algorithm. The weather prediction system replicated for this work is explained in more detail in 4.

2.3.2. Weather Datasets

Researchers often combine information from different sources in order to create systems that are able to encode images into visual features in order to predict weather related information. Researchers need to retrieve images and combine them with weather metadata in order to form a single dataset. Researchers highlight the importance of both time and geographical information of the image in order to retrieve the proper weather record. However, time and geolocation information is not always needed, in cases where researchers only need the weather class of the image. In that case, researchers enlist the help of crowdworkers to annotate images with the proper labels. Although, this is not gold truth data, the labels are provided by multiple annotators and there are voting schemes to remove any possible noise. In [63], the authors gather *20K* images from different web albums such as Flickr, Picasa, etc. Each image is then labelled five times by different people and with a majority voting scheme the final label is chosen. Similarly, the authors in [38], they make use of three sources, namely the *Sun* dataset [58], the *Labelme* dataset [49] and Flickr. After, alternating stages of filtering, crowd-workers remove noise and bias and a set of 10K images is gathered with images labeled as cloudy or sunny. In both aforementioned cases, through manual annotation the researchers were able to gather essential information for their work.

A dataset that is used in prior work and has also been included here is the *Archive of Many Outdoor Scenes (AMOS)* [30] which is an archive of web-cam images. It is a dataset that contains huge amounts of data including the geolocation and timestamp of images that can help researchers in weather data collection. AMOS has functioned as a core component in prior research [10, 22, 29, 40] and it is often combined with weather related information extracted from other sources such as the *Weather Underground API* [4] or the *NOAA* website [5] formerly known as *National Climatic Data Center (NCDC)*. The *AMOS* dataset is used in this work in order to effectively implement a sky segmentation algorithm which is described in chapter 4 in more detail.

From the AMOS dataset, several others have been generated. In [40], the authors create a sky segmentation algorithm with images extracted from AMOS. As contribution, a dataset is created with 94803 entries that consists of images and weather related information. However, the data are sparse and 9.7% of the data in the weather class column (named as *"Icon"* in their data) is missing. Zhang *et al.* [63], create a dataset of 20K images called *MWI (Multi-class Weather Image)*. Their dataset [6] consists of four weather classes, namely sunny, rainy, snowy and haze. In [38], there exists a set of 10K images that has been annotated and shared with crowdworkers. Although, a portion of images has been extracted from Flickr, no information about the licensing is provided. Without the image identification number, it is very difficult to locate the image in Flickr, to get the necessary information about the type of license, restricting us of using them.

Paper - Dataset	Viewpoint	Weather	#images
Jacobs *et al.* [30]	static	No	17M
Islam et al. [29]	static	Yes	3.5K
Narasimhan *et al.* [41]	static	Yes	3K
Zhang *et al.* [64]	dynamic	Yes	20K
Lu *et al.* [38]	dynamic	Yes	10K
Glasner *et al.* [22]	static	Yes	6K
Volokitin *et al.* [55]	static	Yes	23K
Chen *et al.* [10]	static	Yes	500K
Chu *et al.* [13]	dynamic	Yes	183K

Table 2.1: Weather datasets in prior work.

In table 2.1, there is detailed information on weather related datasets that have been used in prior work. The dataset -*Image2weather*- used in this work, is chosen for several reasons such as that it "social" images with the proper licensing information. Further information is provided about the dataset in chapter 3

2.3.3. Sky Segmentation

Sky segmentation is not a trivial problem because the diverse weather conditions can introduce noise and have a very significant effect on the visual appearance of the skyline. This challenge has attracted a lot of attention from researchers with a diverse set of goals. For instance sky segmentation is essential in [59], where the authors detect obstacles to effectively create a route for autonomous air and ground vehicles for *NASA*'s Mars rovers. In [53], the researchers are interested in determining five classes of images depending on the portion of sky that can be seen in the image. These classes are: full-sky, object-in-sky, landscape, normal-sky and others. Researchers have come to the conclusion that the sky plays an important role in estimating weather related properties. In [13], certain features are extracted specifically from the sky in order to distinguish weather classes(cloudy, foggy, rainy, snowy and sunny) and estimate the temperature and the humidity. Before feature extraction, sky segmentation algorithm is run in order to separate the sky from the rest of the image.

In previous years, researchers have created innovative ways of segmenting the sky in both supervised [38] and unsupervised manner [51]. On the first case, the challenge is approached as a binary supervised problem where the pixel is labeled as "sky" or "no sky", with positive class being "sky". Running a pixel-wise feature extraction algorithm can be intensive and it uses a lot of resources, thus in [38] the binary prediction is made for patches with size 15×15. On the second approach, unsupervised learning is implemented and the pixels are primarily grouped into two clusters. Stone *et al.*[51], implement sky segmentation on UV images in an unsupervised manner with the help of k-means and Gaussian mixture model (GMM). However, in [59], the authors implement a fusion of approaches, where both supervised and unsupervised learning are incorporated into their model.

In the previous paragraph two categories of sky segmentation algorithms were presented supervised and unsupervised. In contrast with these algorithms that are utilizing information on a pixel level, the authors in [31, 37], design

[4] https://www.wunderground.com/
[5] http://www.ncdc.noaa.gov
[6] https://mwidataset.weebly.com

algorithms that detect imaginary horizontal lines that can be created in order to separate the sky from the ground. With edge detection the authors are able to detect lines and as a final step classify every pixel above them as sky and everything under it as no sky. It is important to note that the data in [31, 37] consist of images in mountainous environments where the images depict a limited number of objects-people. The algorithm can work in images that there is not that much action such as in the AMOS dataset [30]. Running a skyline detection system on the Flickr[7] data can be problematic since there will be a lot of interfering objects.

2.4. Adversarial Machine Learning

Adversarial machine learning is a domain where researchers design machine learning systems that can deal with data designed to be mis-classified. The data are often images that have been slightly changed so as to confuse the classifier. By mislabelling examples, the security of the system is breached and this can result into terrible outcomes. For instance, the autonomous driving system can be breached by an attacker and will not be able to identify stop sings. An example of adversarial machine learning is depicted in figure 2.2. In this case the attacker pertubes the data with a mask. As a result, the image is wrongly classified as “gibbon” instead of “panda”.

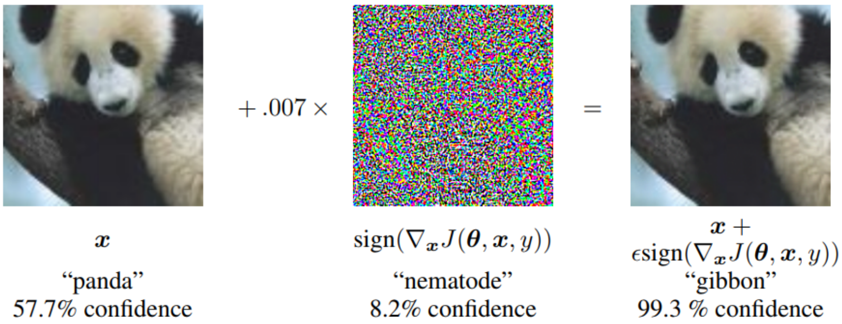

$+ .007 \times$
$=$
$\boldsymbol{x}$
“panda”
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
“nematode”
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
“gibbon”
99.3 % confidence

Figure 2.2: Adversarial machine learning example. The example has been used by Goodfellow *et al.* in [24].

A lot of similarities can be found between adversarial machine learning and the problem of dealing with enhanced pictures in weather prediction. One key difference is that in adversarial machine learning the attacker purposely edits the image to have the system make a mistake. In contrast with our case, where the user does not intend to fool the system and most likely edits images in order to improve the aesthetics as noted in [2]. In both cases the performance drops and researchers need to act accordingly. Enhanced images introduce noise and create challenges in extracting the appropriate visual features. The systems need to be designed in a way so they can be robust and cope with images that have been filtered.

With reference to adversarial machine learning, researchers have engineered ways of dealing with adversarial examples. As they observe there are advantages and disadvantages in every type of defence. Zantedeschi *et al.* [62] mentions the following strategies in order to cope with adversarial examples:

1. Adversarial training. Including in your training set, examples that have been enhanced-altered.

2. Feature squeezing. The data are represented with less features. In this case either the three colour channels are reduced into one- grayscale or the three colours are represented with smaller amount of bins.

3. Detection Systems, that aim to detect perturbations. These systems are trained to distinguish real data from data that have been enhanced by the attacker.

The techniques discussed above will be adjusted accordingly and tested in the visual weather prediction classification problem.

2.5. Data augmentation

Data augmentation is applied to different problems in machine learning. Although, it has been used for several years, recently data augmentation has been at the center of many research works. This can be explained by the growth of neural networks and especially of *Generative Adversarial Networks* [23] (*GANs* for short). Data augmentation is implemented to solve a number of issues. One of them is having imbalanced classes in supervised learning which can be solved by oversampling data instances from the minority class, undersampling data from the majority one or a combination of both. or a combination of both. In 2002, *SMOTE* (*Synthetic Minority Over-sampling Technique*) [9] was presented. Although, oversampling increases the size of the minority class, new useful information is not added

[7] https://www.flickr.com/

and as a result the classifier does not benefit. *SMOTE* differs from traditional oversampling techniques because the data that are synthesized are different from what already exists in that class.

In Computer Vision, data augmentation is frequently used to enrich and increase the size of the dataset. As a result, the model benefits as it learns from more examples. In [28], a very simple technique called *SamplePairing* is created in order to synthesize new data. For one image, a different image is randomly selected and is applied on top of the first one. Then the average of each pixel is calculated and used as value in the new synthetic image. The size of the dataset increases from N images to a size of N^2 images. The authors evaluate their approach on different datasets. Namely, on the *CIFAR-10, CIFAR-100* [33], *ILSVRC-2012* [34] and S*treet View House Numbers (SVHN)* [43]. In their findings, classification error rates on the aforementioned datasets drop after the implementation of *SamplePairing*. In computer vision, the datasets are enriched by augmenting duplicate images that have been edited in various ways. A library in *python* is offered for this exact purpose called *Augmentor* [5]. In [46], different ways of data augmentation are tested, one of which is called *Traditional* in their experiments. In their *Traditional* approach, the images are shifted, zoomed in/out, rotated, flipped, distorted or shaded with a hue. These effects positively affect the performance of the algorithm and they have relatively low computation cost in contrast with other approaches.

A more advanced technique that is tested in their methodology involves *GANs*. In machine learning GANs have been used in several applications including image colourization [8, 42], image inpainting [44, 60] and image manipulation [65]. Perez and Wang [46] test the effectiveness of the *CycleGAN* [66] in their work. This GAN is able to learn the style and certain characteristics of an image collection and transfer these to a new image collection. For instance, *CycleGAN* is able to transfer seasons (winter-summer) into images, edit images to resemble famous artist's style (Monet, Van Gogh, etc.). Perez and Wang [46] selected six styles: Cezanne, Enhance, Monet, Ukiyoe, Van Gogh and Winter. Despite the increase in performance, the *Traditional* augmentation approach still surpasses *CycleGAN*.

3 Dataset

3.1. Description

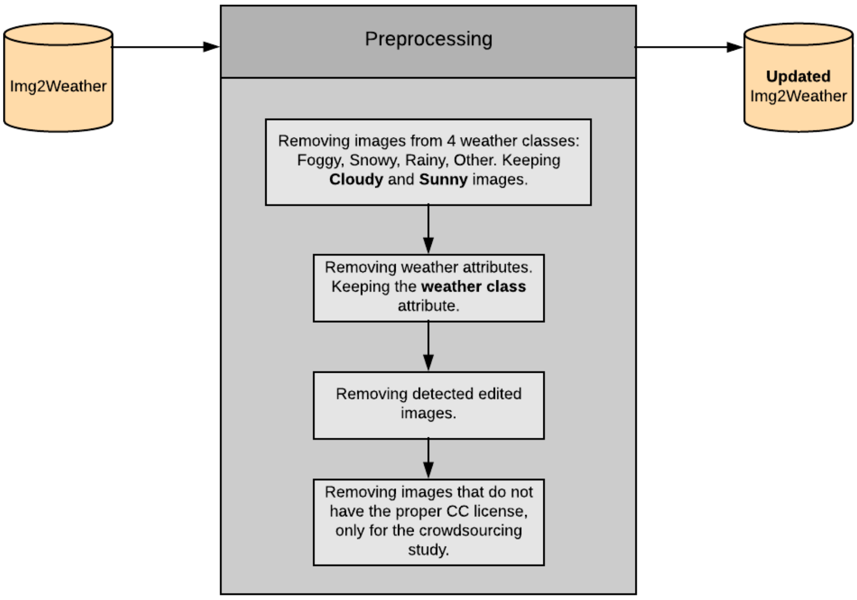

Img2Weather
Preprocessing
Updated Img2Weather
Removing images from 4 weather classes: Foggy, Snowy, Rainy, Other. Keeping **Cloudy** and **Sunny** images.
Removing weather attributes. Keeping the **weather class** attribute.
Removing detected edited images.
Removing images that do not have the proper CC license, only for the crowdsourcing study.

Figure 3.1: Pipeline of preprocessing.

In this section, the most important components of the *Image2weather* dataset are presented. The reader can find the full description of the dataset in [13], where it was first presented. Wei-Ta Chu *et al.* [13], make use of the *EC1M* (European City 1 Million) [1] which consists of *Flickr* images (witht their identintification numbers). With the help of the Flickr API, the timestamp and the location of the images are retrieved and later are used to pair the images with the closest weather records from the *Weather Underground API*.

It is important to note that both the temporal and geographical information of the image do not match exactly the information of the weather record. In order to control the amount of noise embedded into the dataset, the researchers select images that are at a maximum distance of four kilometers from the closest meteorological station. In addition, the temporal distance between the weather record and the image needs to be under two hours. As the researchers in [13] argue, one of the advantages of using the *EC1M* dataset is that it contains images from big European cities where weather records can be easily found because there are numerous weather stations. In their work, the authors explain how a preprocessing stage has been designed to check the dataset to only consist of photos that have been taken outdoors. This is performed by implementing a sky segmentation algorithm and keeping images where the sky occupies more than 10% of the image. This is performed in order to eliminate images which are taken indoors and weather estimation is not possible. As a result, a dataset of 183.798 images paired with weather related information is created. Each image belongs to one of six weather classes that are listed in table 3.1 where there is a detailed breakdown for each class.

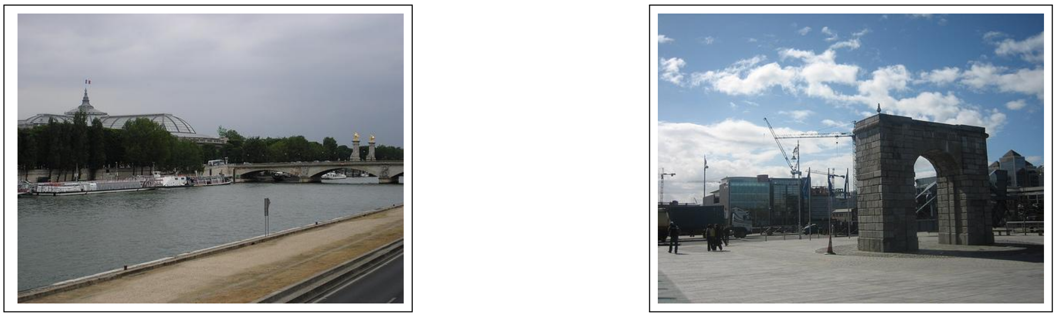
	Weather class						
	Cloudy	**Sunny**	**Foggy**	**Snowy**	**Rainy**	**Other**	**Total**
#images	45,662	70,501	357	1,252	1,369	64,657	183,798

Table 3.1: Overview of weather classes of *Image2weather* dataset.

The dataset consists of several weather attributes such as temperature, humidity, visibility in kilometers, etc. These attributes have been retrieved via the Weather Underground API. The weather class of each image has been determined with the help of manual annotation. If an image is confusing to the human eye and its weather class cannot be determined, it is classified as “other”. By examining [13], the exact characteristics of this human annotation process are not clear. It is important to note that for several images, some of the weather attributes are missing. This does not affect this work, since focus has been given on the weather class. Although, there is mention of textual metadata such as title and description, in the version of the data that is available online, no such information is available. As a result, the final decision of the classifiers is made based on the visual cues and temporal characteristics of the photo.

Focus has been given on classifying images that are either cloudy or sunny. Several reasons lead to that choice. First of all, these two classes contain a very high number of images. In addition, these two classes are very distinguishable with each other, whereas cloudy and foggy images can be hard to tell apart. Regarding the possible multi-class weather detection problem, it would be challenging due to the imbalance between the classes. As it can be seen in table 3.1, for the foggy class there are only 357 images. Class oversampling or undersampling would be necessary in order to effectively deal with the unequal number of instances for each class.

3.2. Cloudy class definition



(a) Overcast.

(b) Sunshine with a few clouds.

Figure 3.2: Images from the cloudy class.

Although, the authors in [13] keep images that are temporally and geographically close to their corresponding weather record (from the Weather Underground API) there is not a strict definition of what is cloudy. In [38], the authors create a set of 5000 cloudy and sunny images respectively. The cloudy images follow a pattern and are very distinguishable from the sunny images. It seems that authors have defined ’cloudy’ differently. In contrast with *Img2weather*, where there are two distinct classes of cloudy:

1. Overcast, little to no sunshine.

2. Plenty of sunshine, blue sky with a few clouds.

Two cloudy images from *Img2weather* with different characteristics have been selected and depicted in figure 3.2. This results into noise in the dataset which can be problematic. Clustering was tested in order to group and identify different cases of cloudy. Unsupervised learning was preferred over classification because it can run without any labels. The features used in the clustering algorithm are the same features used in weather classification. Several experiments were run with different number of clusters ranging from 2 to 6 but unfortunately no cluster was found to contain exclusively overcast images. As a result, no preprocessing has been implemented and all cloudy images have been included in the machine classification experiment. However, for the crowdsourcing experiment, cloudy images from the second class are not selected, instead overcast images have been manually selected and used.

Despite the absence of strict definition for the cloudy class this dataset has many advantages of which has been selected for this work. The images *IDs* are known and their licenses can be retrieved. The subject of licensing is going to be discussed in the next section. In addition, one other factor for which this dataset has been selected is its size and its rich set of weather metadata. In future work, researchers may extend this work to other weather metadata such as temperature or to other weather classes. Although, for some images, some of the metadata are missing, the data are not completely sparse.

3.3. Detection of edited images

One of the goals of this thesis is to understand how filters impact the classifier's performance. For this purpose, images from *Img2weather* are retrieved and edited with Instagram filters. As a result, a set of original-unedited and edited images respectively is created. However, it is not known if these images are already enhanced in any way (sharpness-contrast-brightness enhancement, object removal, cropping, etc.) by the users that originally uploaded them.

A short exploratory analysis was conducted in order to understand when were these images uploaded on Flickr. In the Appendix in figure A.1 there is a histogram that depicts the distribution of the year the images were taken. The year attribute ranges from 1996 to 2013 with 2007 being the mean and with standard deviation of one and a half years. The accuracy of the camera timestamp is an attribute that is extensively discussed in [54] where the authors warn about possible inaccuracies. If there are possible inaccuracies, these do not affect the distribution dramatically because focus has been given on the year attribute. In contrast with 2007, nowadays, with the growth of platforms such as Instagram the portion of images that exist online and has been edited has grown exponentially. However, extensive check is carried out in order to ensure that the set of images contains unfiltered-unedited images.

Manually checking 45,662 cloudy and 70,501 sunny images is cumbersome, thus an image enhancement detection system is introduced into the pipeline. In this section this image enhancement detection component is analyzed. It is challenging to create such a system without knowing what type of enhancement is applied. Two image enhancement detection algorithms have been chosen from prior literature [17, 26].

These two aforementioned algorithms were replicated for this work. It is important to note that the implementation of both algorithms is close but not identical to the original ones. Decisions have been made in order to simplify the algorithms and reduce the size of the feature space. For instance, in [26], Avcibas *et al.* set a window size where within that window they extract some statistics. Depending on the size, the neighbouring windows overlap in some ratio. In this work, none of the windows overlap. In addition, features are calculated for a fixed windows size, whereas Avcibas *et al.* include features with different windows sizes.

A dataset was shared with us that contained images from houses and their surroundings and are edited with the purpose of attracting possible buyers. These photos are not solely taken outdoors but also indoors. In total, a set of 2000 images is used in order to train and test the two aforementioned enhancement detection systems. The dataset consists of 1000 original images and their edited versions. The photo editor has also changed completely the appearance of the sky by either replacing it or making it look brighter and sunnier. The manipulations that have been applied are numerous. Mainly these include: Image contrast enhancement, rotation, blur removal and inpainting.

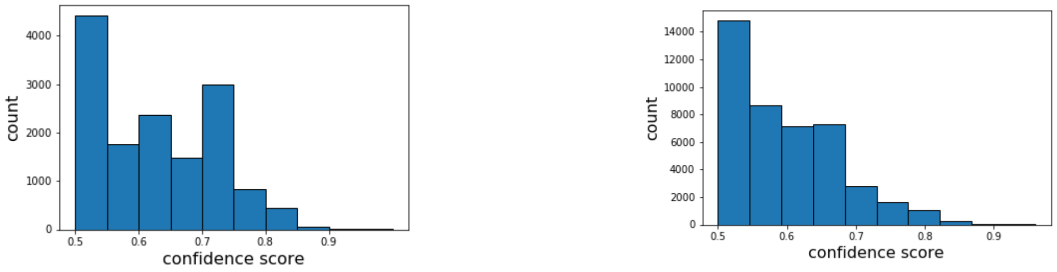
The design of the enhancement detection pipeline in this work is as follows. First the two algorithms are trained and tested independently on the house dataset. In the next step, each algorithm is used in order to annotate and detect the enhanced images from *Img2weather*. Two sets are created with the possible manipulated images. In order to ensure that these images are indeed manipulated, decisions from both systems are combined. Combining judgments is necessary since the mean accuracy on the house dataset is below 0.7 and recall does not achieve a value higher than 0.72. Having a high recall is more important than having a high accuracy. It is of high importance to remove all enhanced images even if some images get wrongly removed. In the end, the size of the dataset still remains big and the experiments can be conducted normally with the rest of the images. Lastly, the two systems differ from each other in terms of their design and what they are trying to accomplish, so combining their confidence scores minimizes the probability of missing a manipulated image.

Paper	**Algorithm**	**Mean Accuracy ± std**	**Mean Recall ± std**
Ding *et al.* [17]	Random Forest	0.69 ± 0.03	0.72 ± 0.05
Ding *et al.* [17]	AdaBoost	0.66 ± 0.02	0.69 ± 0.03
Avcibas *et al.* [26]	Random Forest	0.64 ± 0.02	0.62 ± 0.05
Avcibas *et al.* [26]	AdaBoost	0.63 ± 0.04	0.63 ± 0.05

Table 3.2: Enhancement detection on house dataset.

The results of both detection systems on the house dataset are shown in table 3.2. The best results are achieved with the *Random Forest* classifier. The training procedure for both systems is identical. The algorithms are trained ten times, with different training and test set each time. The following statement holds true for both training procedures: For one image in the training set neither the same image neither its edited/original version is included in the training or test set. This way, the algorithm does not overfit on the original-manipulated pairs. The training subset consists of 800 images and the test set of 200 images with the ratios of original and edited images being equal on each set.

In total 43291 out of 116163 (approximately 37%) images were detected as manipulated with any of the image enhancement classifiers. In figure 3.3, the reader can find two plots of the distribution of the confidence scores for the manipulated class for each of the classifiers. They are both left skewed and there are not many cases that have been detected as enhanced with confidence score greater than 0.75. The confidence scores of these 43291 images are averaged into one value. Subsequently, the top 70% of the images with the highest confidence is removed from the weather dataset and for the following experiments they are not taken into consideration. In total, 30303 images got removed,



(a) Avcibas *et al.* [26] algorithm.

(b) Ding *et al.* [17] algorithm.

Figure 3.3: Confidence scores for the detected manipulated images, for each classifier.

out of which 20600 are sunny and 9703 are cloudy images. The threshold was chosen after much consideration about the size of the resulting dataset and rick of keeping enhanced images in *Img2weather*. Both classifiers have almost equal detection accuracy on the house dataset, thus they have been weighted equally. It would be possible adding a slightly higher weight on the Ding *et al.* [17] classifier since it performed better, but this would not majorly affect the result.

Detecting enhanced images in the *Img2weather* dataset is challenging because the type of manipulation is not known. Running and combining different algorithms minimizes the chance of mislabelling images. Incorporating human annotators into the process would be best, but due to limited resources images that were identified solely by the system are removed. In the end, *Img2weather* dataset still contains a small unknown portion of enhanced images. As it was mentioned in the previous paragraph, most of the images were taken in 2007 when Instagram was not available and users did not spend as much time editing as they do nowadays. Both classifiers achieve accuracy lower than 0.70 and using exclusively either classifier would lead into mislabelling a lot of images. In figure 3.4, some examples of suspected manipulated images that got removed are depicted. To the human eye these seem to be enhanced but it is not known if these images are indeed manipulated. The classifiers are able to detect images that have been sharpened and with high contrast as depicted in figure 3.4. However, not all grayscale images have been detected by the classifier, only those that have been sharpened or additionally enhanced.

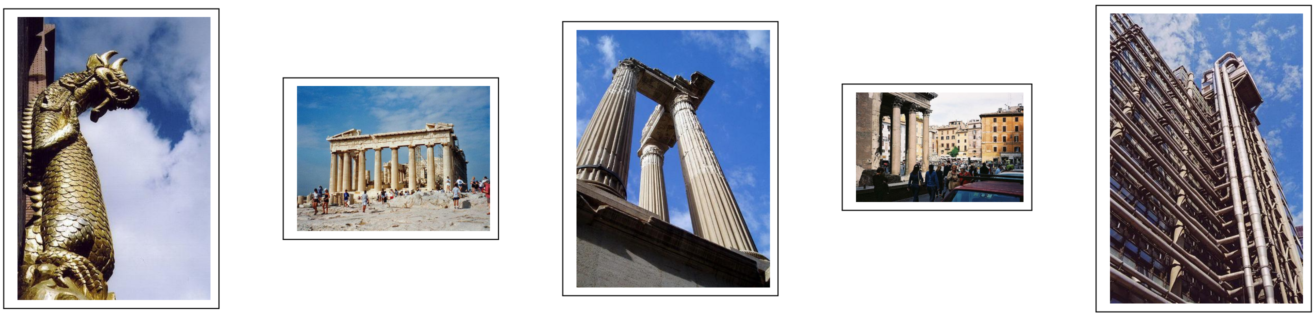


Figure 3.4: Enhanced images from *Image2weather* that got removed.

3.4. Creative Commons - Licenses

There is no mention of image licensing in [13], thus a search was needed in order to ensure that it is permitted to share the photos with the participants of the study. Users that upload their photos on Flickr often select for their photos to be used under the *Creative Commons* license. There are different types of licenses all listed in the Creative Commons site[1] in Flickr. Certain licenses are very strict and do not allow to use the photos in any way. Other licenses[2] allow to share but not adapt the photos in any way. The images that are used in the survey are licensed under one of the following categories:

- *CC BY-NC-SA 2.0*, *CC BY-NC 2.0*, *CC BY-SA 2.0* and *CC BY 2.0*.

The rest of the images cannot be used because they are more strictly licensed, they are private or they no longer exist. A script was created in order to automatically check if the photos have the proper CC license. In table 3.3, all the information about the licenses can be found. The reader can see how many images are licensed under each license and class accordingly. In this work, images from the sunny and cloudy class are shared with the participants. However, information about the other classes is also provided, in order to help researchers that may want to use these images in

[1] https://www.flickr.com/creativecommons

[2] https://creativecommons.org/licenses/by-nc-nd/2.0

Class	**Type of license**			
	CC BY-NC-SA 2.0	**CC BY-NC 2.0**	**CC BY-SA 2.0**	**CC BY 2.0**
Cloudy	4430	2042	1430	1512
Sunny	6673	3122	1705	2255
Foggy	60	20	16	7
Snowy	171	111	30	39
Rainy	158	66	28	39

Table 3.3: Creative commons licenses for images from different weather classes.

their research. The images with the proper licenses were extracted from the sunny and cloudy class in order to include them in the crowdsourcing task. As it is shown in table 3.3 for some classes, the number of properly licensed images is very limited. For instance, 103 out of 357 foggy images can be shared and adapted by researchers. The cloudy and sunny class consist of a lot of images that they are licensed and can be used in the crowdsourcing task. In chapter 7 there is a detailed discussion on the crowdsourcing task and how the selection was made for certain images to be included in the task.

4

Weather Classification

4.1. Pipeline

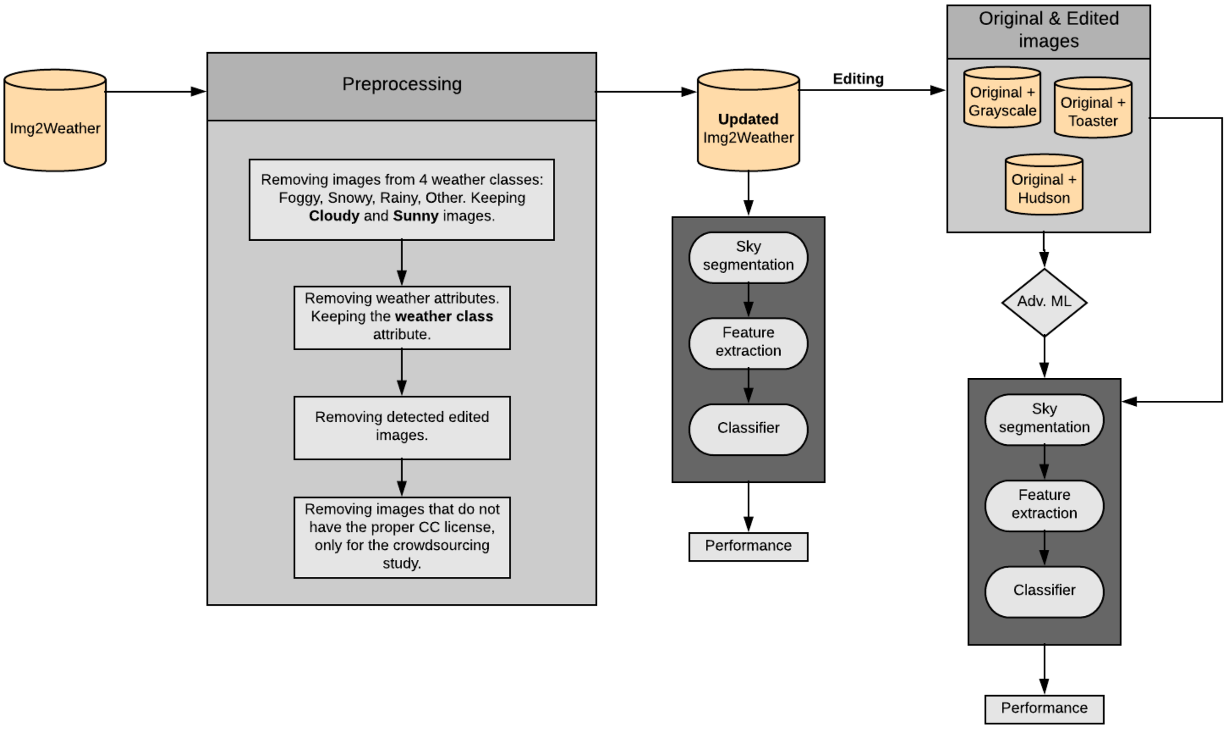

Img2Weather
Preprocessing
Removing images from 4 weather classes: Foggy, Snowy, Rainy, Other. Keeping **Cloudy** and **Sunny** images.
Removing weather attributes. Keeping the **weather class** attribute.
Removing detected edited images.
Removing images that do not have the proper CC license, only for the crowdsourcing study.
Updated Img2Weather
Sky segmentation
Feature extraction
Classifier
Performance
Editing
Original & Edited images
Original + Grayscale
Original + Toaster
Original + Hudson
Adv. ML
Sky segmentation
Feature extraction
Classifier
Performance

Figure 4.1: Pipeline and structure of the design of the algorithm implemented in this thesis.

An overview of the algorithms involved is presented in figure 4.1. On the top left corner of the figure, there is the dataset that is used in this work - *Image2weather*. After some preprocessing steps (that were explained in the previous chapter) the updated version of the dataset is extracted. The figure also includes the layout of the current and following chapters. In this chapter the classification of images is carried out without experimenting with any edited images. In chapter 5, the introduction of filters is being made and the classifier is tested with images that have been filtered. Finally, in chapter 6 several experiments are presented that involve adversarial machine learning in order to deal with these edited instances.

4.2. Weather Attributes

Weather plays an important role in our daily lives and can affect our activities. People spend time reading weather forecast news in order to be prepared for the near future. As we mentioned before, weather can be characterized by several and different types of attributes. Two main categories of attributes exist, the numerical and categorical attributes. There are the *Celsius* and *Fahrenheit* scales which are used to measure temperature. The degrees °C and

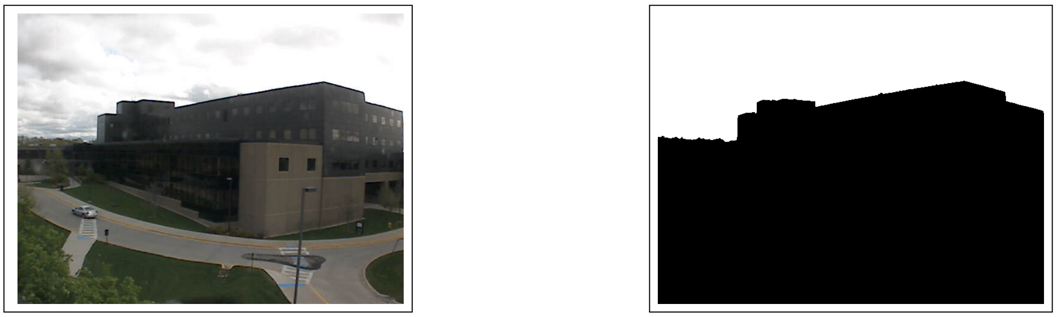
°F can be used to describe a specific temperature on the Celsius and Fahrenheit scale respectively. Another numerical attribute that is used to describe weather is wind speed which is measured with special equipment (e.g. anemometer).

Wind can also be described by its direction (West, East, North, South, etc) which is a categorical attribute. Information such as cloudy, rainy can be very useful to a lot of people that want to learn the general phenomenons that will take place. Prior research has been devoted in creating systems that are able to identify the weather class. There are different types of weather phenomena such as cloudy, sunny, rainy, foggy, snowy, windy, etc. Authors tackle the problem of separating cloudy images from sunny images but also identifying six different types of weather. On the multi-class case the authors make one assumption that two weather phenomena cannot co-exist at the same time. For instance, it cannot be rainy and sunny or rainy and foggy at the same time.

The weather class and temperature are two of the most significant attributes. The overall state of the weather is very important to people for different reasons, whereas attributes such as wind speed is interesting to a limited number of people. In this work, emphasis has been given on identifying the weather class and more specifically two classes of weather, namely cloudy and sunny. Although, in the data there exist six classes in total, all the effort has been given on two classes because they are very distinguishable with each other and contain a lot of information. Researchers are encouraged to explore if this work's findings are applicable to other weather classes or attributes.

4.2.1. Sky Segmentation

Sky segmentation-detection is a necessary step before feature extraction since several features are extracted exclusively from the sky of the image. Without including this very important component in the system, all features would be extracted globally from the whole image. As a result, the feature set of the weather prediction algorithm would be noisy and the information value would be low. In this section, the feature design and the choice behind the chosen sky segmentation algorithm are explained. Lastly, the evaluation procedure is presented.



(a) Train image example.

(b) Binary image - mask.

Figure 4.2: Webcam id: 1093.

Chu *et al.* [13] manually gather a training set by selecting 180 images randomly and creating their binary masks. These binary masks are not available online and it is time-consuming to create these masks. Therefore a different approach has been followed in this work. In [40], Mihail *et al.* create a dataset that consists of images and their masks. Their images have been taken from the publicly available dataset AMOS [30]. The AMOS dataset contains webcam images which differ from the images that are used in this work in terms of context, angle, style, resolution, content, etc. This can negatively affect the performance of the classifier. The images and their masks are available online on the dataset's web page[1].

In figure 4.2, one image and its mask is depicted. In total, the dataset that is constructed for this work contains images from four webcams. In the appendix, in figures A.2, A.3, A.4 the reader can see the rest image-mask webcam pairs. The sky segmentation algorithm is trained and tested on the AMOS dataset but it is later used in order to identify the sky in "social" images. Due to missing gold-truth data, the performance on "social" images is not measured, instead the performance on the webcam images is presented.

Although, in prior literature, researchers have extracted different types of features with high dimensionality and huge computation costs, in this work a much simpler approach has been followed. A pixel is classified as sky by looking into its hue, saturation and brightness (HSV). The HSV space is used as a 3-dimensional feature vector which despite its low dimensionality, this feature set seems to achieve great results as it is shown in table 4.1.

Different types of classifiers were tested in order to create an accurate sky detection system. Namely, *Support Vector Machines* (SVM), *Random Forest*, *K-nn* with k equal to 10 and *Naive Bayes*. The k was set to be equal with 10, after experimentation with different values randing from 2 to 50. Classifiers were evaluated based on cross-validation performance. In each fold, the classes (sky, no-sky) are balanced, thus accuracy can be used as a reliable evaluation metric. The training set consists of 900 training examples and 100 examples are used to test detection accuracy. In table 4.1 the performance of multiple selected classifiers is shown. The *Random Forest* classifier was selected since it

[1] https://mypages.valdosta.edu/rpmihail/skyfinder

Classifier	Mean accuracy ± std
SVM	0.62 ± 0.04
Random Forest	0.81 ± 0.04
K-nn	0.81 ± 0.01
Naive Bayes	0.63 ± 0.03

Table 4.1: Sky detection performance for different types of classifiers.

reached the highest accuracy. Random forest is chosen because usually it generalizes better than *K-nn* which is very important because the training set differs from the images that need to be ultimately labelled.

In prior work, sky detection algorithms try to identify a line that separates the sky from the rest of the image. This line is naturally created by the scene and surroundings. As a next step, the algorithm classifies all pixels above that line as sky pixels and everything beneath it as non-sky. The classifier presented here does not take into account the neighbouring pixels and makes a decision about each pixel independently. Although, this can affect the performance of the classifier, its detection ratio is quite high.

4.3. Experimental Setup

In the previous section the sky segmentation procedure was discussed. The weather detection algorithm follows which is the main part of this work. This section is organized as follows: Firstly, the feature design from [13] is discussed in depth. The subsection that follows has all the details of the experimentation with different classifiers. In subsection 4.3.3 different feature selection and extraction schemes are tested in order to improve performance. Lastly, this section closes with a discussion of the weather condition prediction results.

4.3.1. Feature Design

The work of [13] has been partially replicated in terms of feature design. All of the features regarding the weather classifier are described in this subsection to give to the reader an understanding of what do these features measure and why they are important. In table 4.2 all the feature characteristics that are encoded into the classifier are presented.

Feature	Type	Region	Dimensionality	Gini Importance$\ast 10^3$ ± std
Hour	metadata	global	1	1.22 ± 0
Month	metadata	global	1	0.93 ± 0
RGB_hist	visual	sky	64	1.87 ± 0.78
Insy_hist	visual	sky	64	2.92 ± 1.38
Cloud	visual	sky	64	1.83 ± 0.48
LBP_hist	visual	sky	64	1.29 ± 3.17
Haze	visual	global	84	2.03 ± 3.16
Contrast	visual	global	174	1.86 ± 1.82

Table 4.2: Feature characteristics (type, region, dimensionality and gini importance).

In total there are eight types of features, that differ with each other in terms of dimensionality and what they encode. In the end one image is converted into a 516 dimensional vector. This vector consists of visual cues and temporal information. The vector consists of the hour attribute, the month, the RGB histogram, the intensity of pixels aggregated into a histogram, cloud features, the *LBP* (Local Binary Pattern) histogram, haze features and contrast features. Some features are extracted solely from the sky by using the sky segmentation algorithm that was described in the last section. Without narrowing the region of focus for some features, the classifier will be encoded with noise since certain features are designed to exclusively measure the cloudiness of the image.

The temporal attributes are very important and contribute to the classifier's decision. The hour of the photo is used in the classifier and it is a very important attribute since it encodes if the photo was taken during the day or night. The sun is at its peak some time between 10 a.m. and 3 p.m. Therefore, the classifier can correlate this fact with sunny images. The second temporal feature is the month when the image was taken. Information about the season of one image is very important because cloudy days are very different from season to season and it can be claimed that they are more rare during the summer. One can even support that overall the weather changes drastically from one month to the next one. The classifier handles differently images that were taken in November with images taken in April. Information about the month of the image can be really helpful in a multi-class scenario where there are weather classes such as snowy and rainy. Although, climate can differ across different locations, snowy weather can be expected to take place some time during the winter. The month feature is more significant in a multiclass scenario because sunny and cloudy weather is not as season/month-dependent as snowy weather. Closing, these two temporal attributes have the advantage of low dimensionality and low computation cost during feature extraction but they are often not known to the machine learning engineer.

A pixel is represented by three values in the *RGB* colour space. The final colour of the pixel is determined by the intensity of three individual channels (red, green and blue) which range from 0 to 255. In the feature design a histogram of the RGB values is created with 64 bins and equal bin size. The height of each bin represents the frequency of pixel values within that bin. Instead of using having 255 bins, the values are aggregated into a histogram where it is easier to identify patterns and avoid overfitting of the model. A similar feature has been designed with the only difference being that it is extracted from a different colour space, the *HSV* space. An intensity histogram is created, that aggregates the values of the second channel (saturation) into 64 equally sized bins.

The cloud features [36] are consisted of two different types of features, the *blue to red ratio* and the *euclidean geometric distance*. Both features, are used in order to detect clouds in a supervised manner in all-sky images. These features are extracted solely from the sky of the image on a pixel by pixel basis. The blue to red ratio is calculated for each pixel s and it is defined as $r_o(s) = (b-r)/(b+r)$, where b and r represent the intensities of the blue and red channel respectively. It is important to note that these features are extracted from all-sky images where a bigger portion of sky is visible compared to the images used in prior work [13] and this work. Although, the type of image is different, the cloud features are very useful and help detect clouds in the sky. As the authors argue in [36], clear sky is more blue, whereas cloud particles have almost equal amount of red and blue. The euclidean geometric distance (*EGD*) is defined as:

$$ed(s) = \sqrt{r^2 + g^2 + b^2 - \frac{(r+g+b)^2}{3}}$$

The intuition behind this feature is that sky pixels will often have high *ed* and cloud pixels low *ed*.

A very common feature that is used extensively in numerous works in Computer Vision is the *Local Binary Pattern (LBP)*. Local binary pattern enables researchers to encode texture and understand how neighbourhoods of pixels behave. It compares the intensity of the center pixel with its eight neighbour pixels resulting into a binary code which is later transformed into a decimal. This decimal value is the new value of the center pixel. This process is repeated for all pixels of the image and later all the values are aggregated into a histogram with 64 bins. In the end, a 64-D feature vector is extracted and used in the model.

An important component in the feature vector are the haze features which can reveal the presence of cloudy weather as discussed in [38]. The *dark channel prior* is firstly presented in [27] and it is effectively used to remove haze from images. The dark channel is extracted from the RGB colour space as:

$$J^k(x) = \min_{r,g,b} \min_{y \in \Omega(x)} J^c(y)$$

Where J^c is a colour channel, $\Omega(x)$ the local patch with dimensions 8×8. The image is separated into 2^2, 4^2, and 8^2 non-overlapping regions to obtain in total 84 regions. For each region the median value of the dark channel is extracted forming the 84-D feature vector.

Lastly, the contrast features is the last component that contribute to this classifier's decision. Contrast is an important measure and it is encoded into the classifier in order to discriminate cloudy from sunny images. Its effectiveness is proved in prior literature [38]. In this work, the saturation value of the *HSV* colour space is used. The feature vector is constructed collecting all saturation percentile ratios of the saturation values of the image. The percentile ratios are denoted as $r = p_i/p_j \ \forall i > j$. The i^{th} percentile is denoted as p_i where i and j are multiples of 5, resulting into 171 percentile ratios.

In table 4.2 the fifth column from the left lists the importance of each feature which is computed with the help of the *Gini* coefficient. This measure is often used in economics to measure wealth distribution in populations, and more specifically its inequality. It ranges from zero (perfect balance, zero inequality) to one (maximum inequality). In machine learning it is used in a *Random Forest* classifier in order to measure the quality of a split. With the Gini coefficient, the features can be ranked according to their importance and give the engineer an understanding of the algorithm design.

It is often referenced as Gini importance or Mean Decrease Impurity [35]. Total decrease in node impurity is summed and averaged across all trees in a Random Forest. The mean gini importance has been computed for 10 runs of cross validation, then these values have been averaged a second time for each type of feature. The final values are graphically depicted in figure 4.3b and listed in table 4.2. The month attribute is the feature with the least predictive power, whereas the intensity histogram features are the most important ones in this model.

4.3.2. Classifier

The classifier is chosen after the execution of multiple cross-validation tests with different type of classifiers. Cross validation tests were run with the following classifiers:*AdaBoost*, *Quadratic Discriminant analysis (QDA)*, *Random Forest*, *K-nn* and *Naive Bayes*. In table 4.3 the performance for each classifier is listed. Regarding the *K-nn* classifier, different values of k were tested ranging within [10,50] and with the *Minkowski* distance. *K-nn* achieved the highest performance with k equal to 30 . As it can be seen in table 4.3, the AdaBoost and Random Forest classifier achieve the highest mean accuracy of 0.82. Both algorithms were selected for further experimentation but solely the results achieved by AdaBoost are presented because it achieved higher performance with the different feature reduction algorithms.

In addition, there is experimentation excluding the temporal featuresand solely including visual features into the model. The classifier manages to detect cloudy and sunny instances with 0.82 being the highest accuracy achieved by the AdaBoost algorithm.

Classifier	Mean accuracy ± std	
	Visual & Temporal features	Visual features
AdaBoost	0.82 ± 0.02	0.82 ± 0.02
QDA	0.75 ± 0.03	0.74 ± 0.02
Random Forest	0.82 ± 0.02	0.80 ± 0.02
K-nn	0.77 ± 0.01	0.75 ± 0.03
Naive Bayes	0.70 ± 0.03	0.70 ± 0.02

Table 4.3: Weather classification with different types of algorithms.

The AdaBoost classifier was first introduced by Freund and Schapire in [20]. Boosting algorithms is a common technique in machine learning that is used by engineers that want to further improve the system's performance. It differs from other popular algorithms because it actually learns "weak" classifiers and at the final stage it combines them in order to make the final decision. These classifiers are "weak" meaning that a large portion of the data is classified within the wrong class. At the beginning of the algorithm all samples have the same weight and after each round the wrongly classified instances get re-weighted with a higher value. As a result, the next classifier is able to correct the decision made by its prior. It is important to have "weak" learners instead of very good learners, in order to avoid overfitting the model. The Adaboost used in this work consists of 50 weak classifiers - decision stamps and it is implemented with the help of the *Scikit-learn* library [45] in python.

4.3.3. Feature Reduction

In the previous section, the feature design is explained in detail. One goal of this work is to design of a classifier that achieves high performance in terms of detection accuracy. In addition, the classifier needs to be efficient and the feature extraction computation cost needs to be minimized. For that reason, there is experimentation with different feature reduction schemes in order to lower the dimensionality of the data. In its design the Random Forest algorithm randomly selects different features. Yet feature reduction schemes are still included in the pipeline as a pre-processing filtering component.

Gini importance is calculated to give to the reader an understanding about the significance of each feature. Two main categories of feature reduction have been chosen in order to reduce the dimensionality of data:

1. Feature selection (Chi- squared Test, Variance threshold) and

2. Feature extraction (PCA).

In table 4.5, the mean accuracy can be found for different feature reduction schemes and with different number of features. The mean accuracy is extracted after 10 runs of cross validation with 1000 train and 100 test examples for each class. The classes are balanced, thus accuracy accurately depicts the quality of the output. Different number of features were tested ranging from 100 to 500 for all feature reduction strategies.

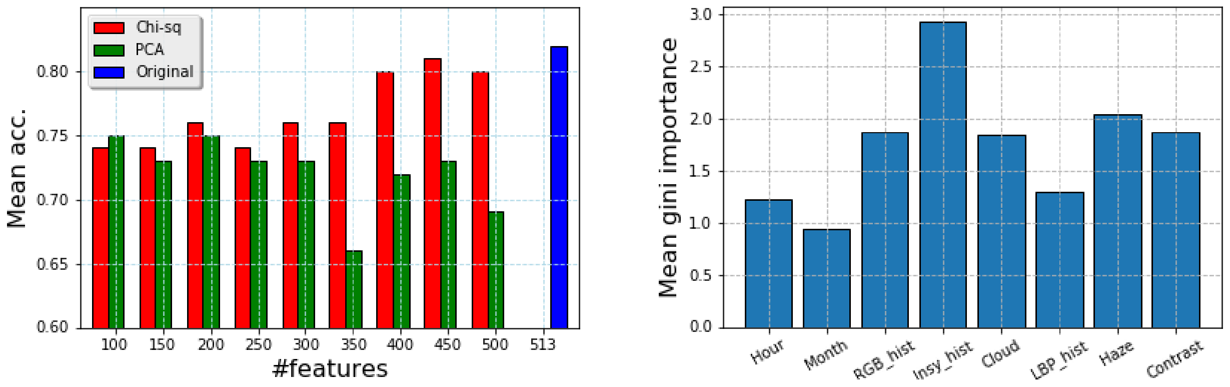
Feature selection can be performed in numerous ways one of which is very simple and it merely requires the calculation of each feature's variance. An advantage of this feature selection scheme is that it is performed in an unsupervised manner meaning that the label of the image is not needed. In order to select features, a threshold is set and features with variance lower than that threshold are removed from the model. The intuition is that features with low variance do not contribute as much as features with high variance. In table 4.4, there is detailed information about the experiments that were run with thresholds ranging from 0.4 to 0.8. None of the experiments exceed the performance of the classifier with the original set of features.

Var. threshold	Mean accuracy ± std
0.4	0.77 ± 0.02
0.5	0.75 ± 0.02
0.6	0.74 ± 0.02
0.7	0.75 ± 0.01
0.8	0.76 ± 0.01
original set of features	0.82 ± 0.02

Table 4.4: Feature reduction with variance thresholding.

Another way of performing feature selection is with statistical hypothesis testing. Chi-squared stats are calculated for each feature and class and the most important features are kept into the model. In table 4.5 the results of this scheme are presented. With the 450 most important features AdaBoost classifier reaches mean accuracy of 0.81, which is marginally lower than the performance achieved with the original set of features. It is also noteworthy to point out that with feature selection the hour and month attribute were not included in the feature set.

Lastly, a different feature reduction scheme was tested, feature extraction. Unlike feature selection, the feature space changes and the features cannot be interpreted by the engineer. As a result, the engineer does not have insight into the system and it can be problematic when the result of the classifier needs to be explained to others. A common way of creating new features based on the current ones is with the help of *Principal Component Analysis* commonly known as *PCA*. It is important to normalize features before running PCA, therefore the features are standardized by removing the mean and scaling to unit variance. In table 4.5, it is seen that PCA does not benefit the classifier since it reaches mean accuracy of 0.75 with 100 features. In a second experiment, a different normalization approach was tested. Features were scaled into a range of [0,1] but the performances were similar to the standardization experiment. In conclusion, feature selection benefit the algorith more than feature extraction but still the best performance is achieved with the original set of features. Performances reached by hypothesis testing and PCA are also graphically depicted in figure 4.3a.



(a) Mean accuracy for different feature reduction schemes.

(b) Mean gini importance*10^3 for each type of feature.

Figure 4.3: Mean accuracy for different feature reduction schemes and Mean gini importance*10^3 for each type of feature.

#features	**Feature scheme**	
	Chi-squared test	**PCA**
100	0.74 ± 0.02	0.75 ± 0.03
150	0.74 ± 0.03	0.73 ± 0.04
200	0.76 ± 0.02	0.75 ± 0.03
250	0.74 ± 0.03	0.73 ± 0.04
300	0.76 ± 0.02	0.73 ± 0.05
350	0.76 ± 0.03	0.66 ± 0.08
400	0.80 ± 0.02	0.72 ± 0.05
450	0.81 ± 0.02	0.73 ± 0.05
500	0.80 ± 0.02	0.69 ± 0.05

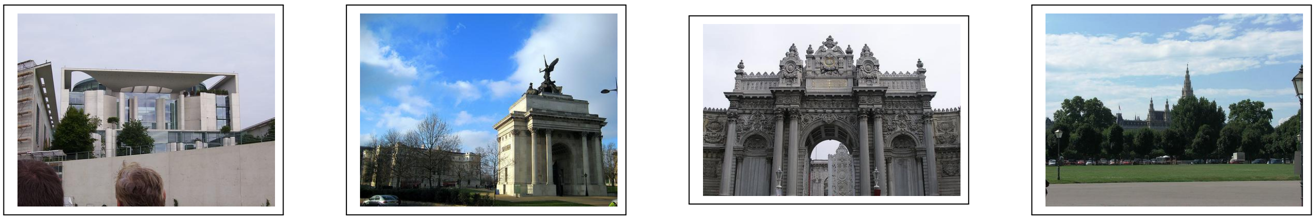
Table 4.5: Mean accuracy scores ± std for different number of features and different feature extraction schemes.

4.4. Discussion

The classifier identifies cloudy and sunny images in a diverse set of images. As it is mentioned in chapter 3 the images have their weather conditions manually annotated by people in [13] and there is not a strict definition of what constitutes as cloudy weather. As a result, some images have not been labelled accurately and there is some noise in the results. In figure 4.4 some images have been selected in order to highlight that fact. These images are under the *Creative Commons* license and can be shared and viewed by others. For this example, the cloudy weather class is considered to be the positive class and sunny the negative one. The results of the classifier fall into the following categories:

- True positives (true label: cloudy, predicted label: cloudy)
- True negatives (true label: sunny, predicted label: sunny)

- False Positives (true label: sunny, predicted label: cloudy) and
- False Negatives (true label: cloudy, predicted label: sunny).



(a) TP example.

(b) TN example.

(c) FP example.

(d) FN example.

Figure 4.4: Example images that were classified correctly (TP, TN) and wrongly (FP, FN).

There are several observations that can be made about figure 4.4. In the true negative case, the classifier correctly classifies a sunny image. Images where the sky is blue filled with puffs of clouds and there is sunshine can be found in both classes of the dataset. In contrast with the dataset in [38], where the images within the cloudy class do not vary as much. As a result, the images in figure 4.4c and 4.4d are wrongly identified as false positive and false negative respectively. One image has been identified by the classifier as cloudy but its real label seems to be sunny. These two cases have been selected from the set of miss-classifications and they constitute a small portion of images.

5

Weather Classification of Filtered Images

5.1. Filters

Instagram is one of the most popular social media platforms online. In June 2018 it hit a very important milestone. According to the story presented here[1] Instagram has grown from 800 million users in September 2017 up to 1 billion of users in June 2018. The decision has been made to experiment with Instagram filters instead of using general image enhancements. Several libraries in many programming languages offer the tools and means to automatically enhance images in any way the programmer desires. The problem with using a type of enhancement is that it can be used in combination with other enhancements and the strength of the manipulation can vary dramatically for different images. With Instagram filters it is made possible to replicate what is happening online and create images that can indeed be found online. *Gimp*[2], has plugins to edit images in batches and offers 22 Instagram filters that can be applied. Instagram filters by Gimp have been used in [52], where a system is able to recommend the proper filter for an image. In Instagram, users can find in total 40 different filters to edit their images with. In figure 5.1, the reader can see the original version, its grayscale and 22 other versions that have been produced with the Instagram filters provided by Gimp.

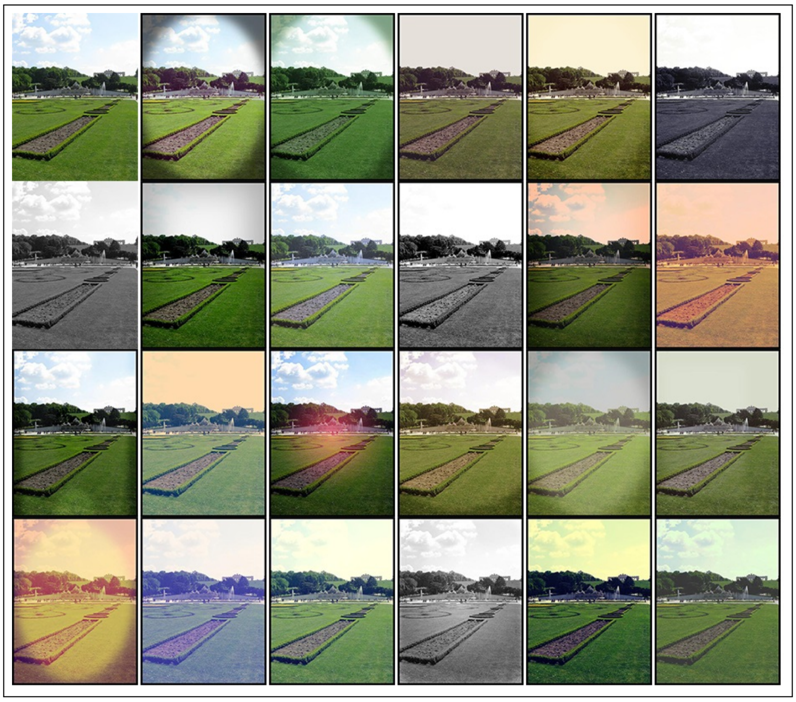


Figure 5.1: From left to right: Original, Amaro, Apollo, Brannan, Earlybird, Gotham, Grayscale, Hefe, Hudson, Inkwell, Lofi, LordKelvin, Mayfair, Nashville, Poprocket, Rise, Sierra, Sutro, Toaster, Valencia, Walden, Willow, Xpro2, 1977.

Adobe Photoshop has a rich set of tools that possibly allow expert users to replicate Instagram filters. There are several online guides[3] that describe the process of making an Instagram filter with several steps included. Using Adobe

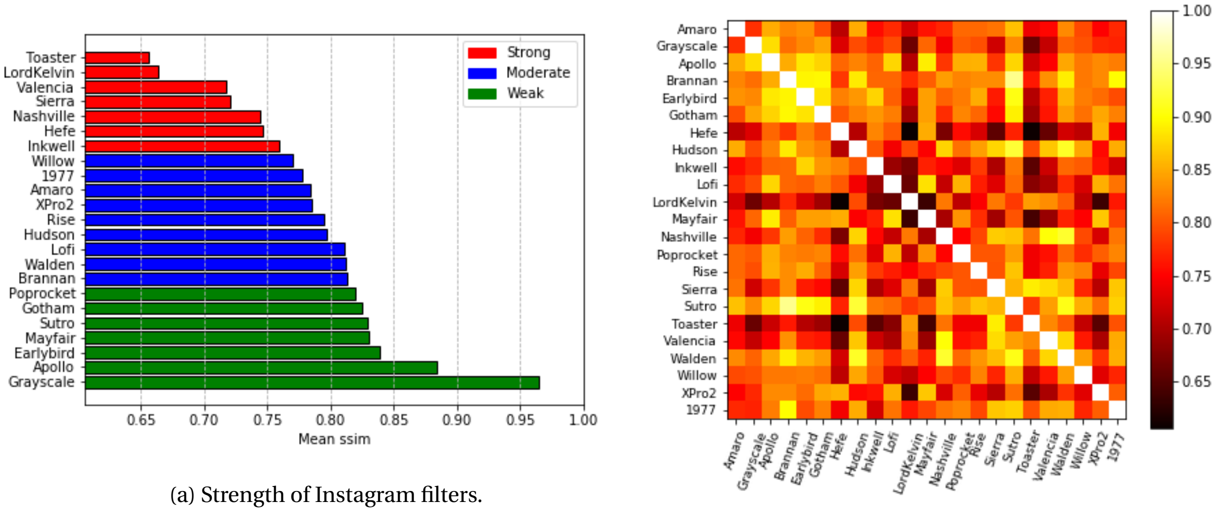
[1] https://techcrunch.com/2018/06/20/instagram-1-billion-users/

[2] https://www.gimp.org/

[3] https://mashable.com/2013/10/20/photoshop-instagram-filters/?europe=true

Photoshop requires skill and the filters reproduced online do not seem to match the original ones. In addition, there are online platforms such as Insta-editor[4] that allow users to edit and add different effects. Unfortunately, (at the time writing this thesis) this platform does not have the option of batch-editing which is essential for the amount of images that need to be edited in this work.

Understanding how and why users edit their images is a complicated issue which researchers have tried to investigate. Several filter/image enhancement recommendation algorithms have been designed in prior work [16, 21, 52] to offer guidance to users in order to edit their images accordingly. In [2], one of their findings is that users edit their images in order to make their images more beautiful. What makes an image beautiful and choosing a filter can be subjective, thus making very difficult to recommend the proper filter. In addition, social media influencers tend to copy image looks that are not always in a good taste. Consequently, finding patterns in filter use online does not have a straightforward solution. Bakhshi *et al.* [2] carry out interviews and the difference between professional and casual photographers is highlighted. In particular, professional photographers edit photos to correct errors, they are more selective with filters and they express their dislike in Instagram-like filters which completely alter the looks of an image. On the other hand, casual users are interested in making their photos more special and fun as it was highlighted in their interviews.



(a) Strength of Instagram filters.

(b) Similarity of filters.

Figure 5.2: Strength and similarity of Instagram filters.

In this work, the filter selection has been made by taking into consideration two factors, strength and diversity. These two factors have been measured with the help of the SSIM index which has been selected due to its widespread use in computer vision. The filters need to be selected in such a way so they have different level of impact on images. In figure 5.2a, the filters have been categorized into three categories according to their level of impact. These filters have been put on a scale after measuring their impact on a set of 60 images. The set of 60 images has been manually selected in such a way to include diverse colours, scenes and weather. In order to ensure that the images are diverse, a check is conducted with the help of the SSIM. For each image in the set, its SSIM is calculated with all the rest 59 images. This process is repeated for all of the images and then the average SSIM is extracted which ranges from 0.03 to 0.55, has average of 0.25 and standard deviation of 0.07. As it can be seen the SSIM is low and not close to one indicating that the images are far from identical.

After, the set of 60 images has been selected and checked in terms of similarity the strength and diversity of filters are measured as follows. The mean SSIM is extracted between the 60 edited images and their original versions, for each filter. As a final step, the mean SSIM values are put on a scale for each filter.

The stength of filters in terms of mean SSIM ranges from 0.65 to 0.96. The filters depicted in red have low mean SSIM indicating the big impact they have on images, whereas the filters depicted in green, do not drastically change the images, thus the high SSIM is explained. Three filters have been selected, one from each strength category. It is a goal of this thesis provide insight on the effect different filters have on human perception and on the classifier. It is important to use three distinct filters instead of using various filters within a category because the three categories are very close with each other and all filters have generally high SSIM. Consider the following case, *Poprocket* has mean SSIM of 0.82 and *Brannan* achieved mean SSIM of 0.81. These two filters have been put in different strength categories despite their low SSIM difference.

The reader can find in the appendix a similar graph in figure A.5 that ranks filters based on their strength by using

[4] http://insta-editor.com/

the *Mean squared error (MSE)*. The two ranking lists are very similar with each other and the decision on which metric to use does not majorly influence the strength ranking.

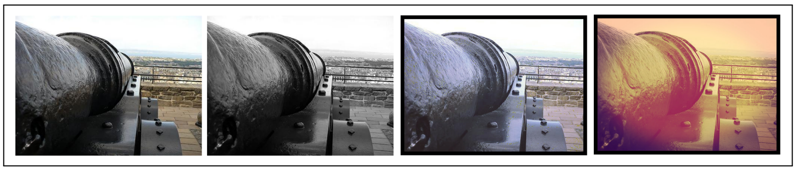


Figure 5.3: From left to right: Original, Grayscale, Hudson, Toaster.

In figure 5.2b, a heatmap is displayed that consists of the mean SSIM values between the edited sets of 60 images that were produced with the Instagram filters. In the main diagonal of the heatmap, we have high similarity between Instagram filters which is indicated by white colour. The grayscale effect differs the most with *Toaster*, so selecting these two is the best option, since they are also filters with the lowest and highest impact respectively. Lastly, a filter is selected with moderate effect called *Hudson*. These filters, are very popular among users online nad they are depicted side by side together with the original version of the image in figure 5.3. Grayscale is the most well known and extensively used filter that can be found in numerous devices. Hudson makes photos look cold and icy and Toaster adds a dodged center to the photos with a strong red tint and a burnt edge.

5.2. Classifier Performance

In this section, the first research question is answered: *How does image enhancement affect the performance of the weather class classifier?* This question can be split into four sub-questions for each of the three filters and the ensemble that are included in the experiments.

The algorithm, feature design and the evaluation procedure are similar to the ones presented in chapter 4. The classifier is evaluated by measuring the mean accuracy achieved in 10 folds. The classes are balanced in both training and test set. The training set consists of 1000 training examples and the algorithm is evaluated on 100 test images in each fold. It is important to note that if one image is selected in the training set, none of its edited versions will be selected in the test set. This check is performed in order to avoid possible overfitting. As it was mentioned in previous chapter, the detected enhanced images have been removed from *Img2weather*.

The test set needs to reflect the reality and what exists in social media. This has been ensured by using images from Flickr and testing with popular image filters from Instagram. It is challenging to determine and decide on a fixed ratio for the edited class. Therefore, the effect of filters was evaluated in different ratios ranging from 10% up to 100% in the test set. By running experiments in different ratios, it is possible to see what happens in low ratios but also in extreme ones. In the appendix, the experiments for all different ratios are organized in tables. In addition, the reader can find the performances achieved on the edited & original dataset and on the edited part of the dataset.

Table 5.1, summarizes all the results and provides an overview of the performances that were achieved without changing anything from the classifier presented in chapter 4. In the appendix there are four tables with all the details for each type of experiment in different edited ratios (tables A.1, A.2, A.3, A.4). Filters clearly degrade the performance of the classifier answering the first research question of this work. The grayscale images have the most severe effect on the algorithm, whereas Hudson has the least impact. In the ensemble of filters, the dataset consists of Grayscale, Hudson and Toaster images.

Dataset - Filter	Mean accuracy ± std
Original	0.82 ± 0.02
Grayscale + Original	0.65 ± 0.02
Hudson + Original	0.77 ± 0.02
Toaster + Original	0.66 ± 0.02
Ensemble of filters (Edited) + Original	0.70 ± 0.02

Table 5.1: Results with Edited images.

5.3. Discussion

Despite having the lowest impact in terms of SSIM, grayscale images cause the performance to drop the most. This can be explained by the feature design where certain features calculate the difference between the RGB colour channels. With grayscale images the difference is equal to zero, since all the colour channels have the same values. The Toaster filter is a very distinct filter that it was ranked first in the analysis of strength of filters. The filter has also a very big impact on the classifier and on its decision to determine if the weather depicted in an image is sunny or cloudy. In contrast, the Hudson filter has the least impact on the classifier causing the performance to drop by an absolute value of 5%. The filters can be ranked in descending order based on how strong their impact was on the classifier as follows: Grayscale, Toaster, Ensemble, Hudson and Original. In the next chapter the design is adjusted in order to cope with the filtered images.

6

Adversarial Machine Learning

6.1. Overview

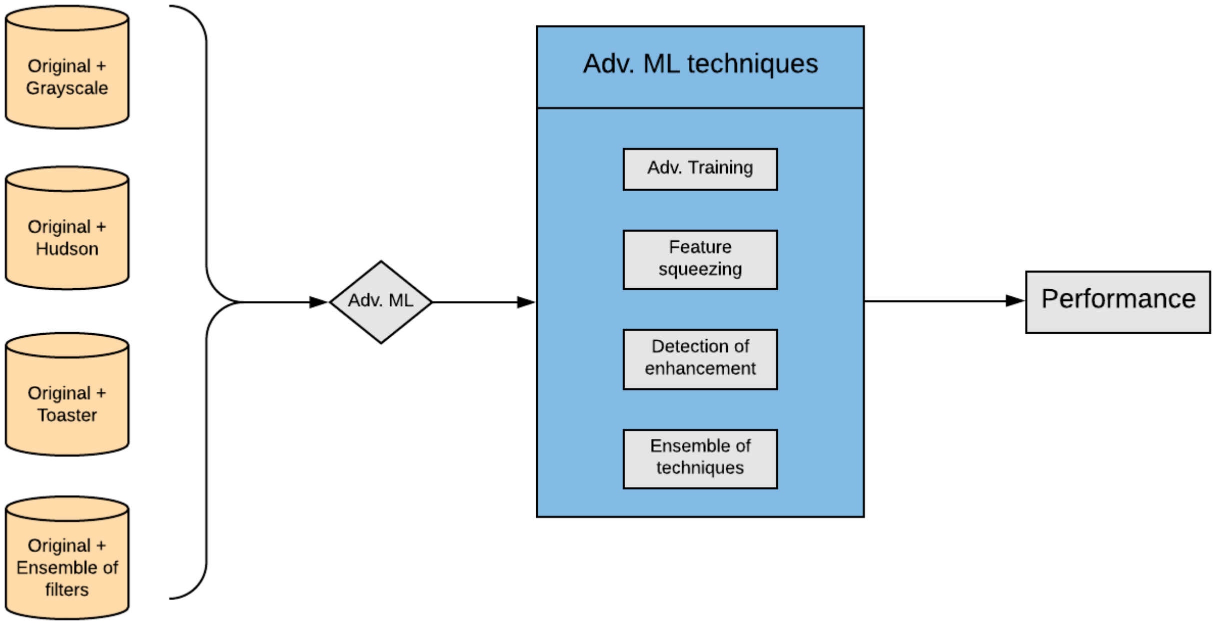


Figure 6.1: Overview of datasets and adversarial machine learning techniques.

The purpose of this chapter is to investigate the impact of adversarial machine learning in our problem. By the end of this chapter the second research question will be answered which is formally stated as: *Can we improve the performance of the weather class classifier by using techniques motivated by Adversarial Machine Learning?* In figure 6.1, an overview of this chapter is presented.

6.2. Adversarial training

The first method that is designed to protect the classifier from filtered images is called adversarial training. For the purposes of this thesis, filtered images are considered as adversarial examples that introduce noise and variance into the dataset. Adversarial training can be considered as data augmentation because it is the process of incorporating new training examples into the corpus. With this approach, the classifier learns not only from unedited images but also from edited ones. The results after experimentation with adversarial training are listed in table 6.1. More details about each specific filter can be found in the appendix's tables A.5, A.6, A.7, A.8. Similarly with the experiments explained in chapter 5, the filtered/no filtered image ratio in the test set is not fixed and it ranges from 0.1 to 1. Overall, two different experiments were run. In the first experiment, 50% of the images in the training set were edited and in the second the ratio was set to 30%. The third column from the left indicates the original scenario where none of the images included in the training set have been edited. For each type of filter, the test set consists of images that have been edited with the same filter. The training ratios were selected to be no greater than 50% since in [46] the portion of edited images is also set to 50%. One key difference from [46], is that in this work in the training set there are not two versions of the same image because the algorithm can generalize more effectively.

	Ratio of Filtered images in the training set		
Dataset - Filter	**ratio = 0.5**	**ratio = 0.3**	**ratio = 0**
Grayscale + Original	0.77 ± 0.02	0.77 ± 0.02	0.65 ± 0.02
Hudson + Original	0.79 ± 0.02	0.79 ± 0.01	0.77 ± 0.02
Toaster + Original	0.77 ± 0.02	0.77 ± 0.02	0.66 ± 0.02
Ensemble of filters (Edited) + Original	0.77 ± 0.02	0.76 ± 0.02	0.70 ± 0.02

Table 6.1: Adversarial training results on different filters.

The effect of adversarial training is evident even at a ratio of 30%. By introducing a small portion of grayscale images into the training set, the mean accuracy on the grayscale test set rises drastically. The model possibly learns from the grayscale images and adjusts accordingly. Of course the type of manipulation is known and the training set is adjusted each time depending on the dataset-filter. This can be considered impossible in a real life scenario where the type of manipulation is unknown to the machine learning engineer. One additional experiment was run where the classifier was tested on images from all three datasets. Adversarial training had a positive effect on the classifier by increasing its performance by an absolute value of *0.07*.

6.3. Feature squeezing

In adversarial machine learning, *feature squeezing* is often used to reduce the complexity of the representation of the data. In our problem, this has been achieved by encoding the colour channels with 171 and 84 bins respectively. Testing all 256 values for the number of bins would be infeasible, thus two values were selected in such a way to represent the whole range. In the original scenario the data representation is consisted of 256 bins. Within a compressed data representation the values need to be properly dispersed across the range of 0-255. For instance, if the images are encoded with 84 values these values need to be scattered and not very close with each other. The quantized bins are randomly and uniformly selected from 0 to 255 for all images. Although, running a clustering algorithm for each image would probably benefit the classifier, this requires too many resources that increase the computation cost. The representation of the enhanced images is reduced down to 171 and 84 respectively, whereas the original images are represented with 256 bins. The overview of the experiments with feature squeezing can be found in table 6.2. The first column from the left indicates the performance achieved with the original classifier and by keeping the original representation. More details about all the experiments with feature squeezing can be found in the appendix in tables A.5, A.6, A.7, A.8.

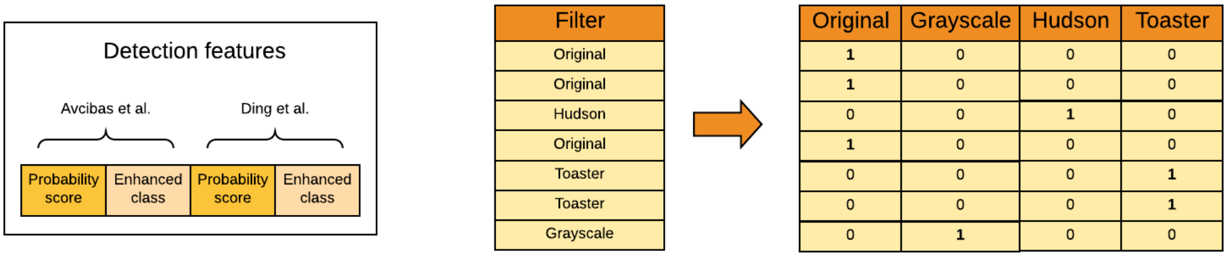
	Feature squeezing		
Dataset - Filter	**bins = 256**	**bins = 171**	**bins = 84**
Grayscale + Original	0.65 ± 0.02	0.66 ± 0.02	0.66 ± 0.03
Hudson + Original	0.77 ± 0.02	0.77 ± 0.02	0.77 ± 0.02
Toaster + Original	0.66 ± 0.02	0.67 ± 0.02	0.66 ± 0.02
Ensemble of filters (Edited) + Original	0.70 ± 0.02	0.70 ± 0.02	0.70 ± 0.02

Table 6.2: Feature squeezing results on different filters.

In contrast with adversarial training, reducing the complexity of data does not always have the desirable effect. With some filters the average performance remains stable across different data representations. With grayscale images the classifier performance increases by a negligible absolute value of 1%. One possible reason why feature squeezing has a mild impact is that the data representation needs to be further reduced with a value lower than 84. Despite feature squeezing not having a big effect, reducing the complexity of data has other benefits such as computational benefits. Feature extraction is very demanding in terms of resources, especially if it is run on a dataset that consists of thousands of images.

6.4. Detection of enhancement

Two different types of features are designed and included in the classifier, *Detection features* and the *Filter features* which have been named as such for the purpose of this work. In chapter 3 two image enhancement classifiers were designed, inspired from prior work [17], [26] in order to cleanse the dataset from images that have already been enhanced. These two classifiers are used in adversarial machine learning in order to provide additional information to the classifier. On the left part of figure 6.2, the reader can find the composition of the detection features. From each



Filter
Original
Original
Hudson
Original
Toaster
Toaster
Grayscale

Original	Grayscale	Hudson	Toaster
1	0	0	0
1	0	0	0
0	0	1	0
1	0	0	0
0	0	0	1
0	0	0	1
0	1	0	0

Figure 6.2: Detection features (left) and example of one-hot encoding filters (right).

classifier, the probability score of the positive class is extracted together with a label indicating if the image is categorized as enhanced. The detection features were not tested for the grayscale images because no (contrast, sharpness, etc.) enhancement has been applied.

	Feature set used in training		
Dataset - Filter	**Detection features + Original features**	**Original features**	**Filter features + Original features**
Grayscale + Original	—	0.65 ± 0.02	0.65 ± 0.02
Hudson + Original	0.77 ± 0.03	0.77 ± 0.02	0.77 ± 0.02
Toaster + Original	0.67 ± 0.03	0.66 ± 0.02	0.66 ± 0.03
Hudson + Toaster + Original	0.71 ± 0.03	0.72 ± 0.02	0.72 ± 0.02
Ensemble of filters (Edited) + Original	—	0.70 ± 0.02	0.70 ± 0.02

Table 6.3: Detection and Filter features results on different filters.

In table 6.3, the results are shown for both types of features. The detection features have no real impact on the classifier's performance. Subsequently, the filter features are introduced into the algorithm. These features indicate the type of manipulation that has been applied to the image. This categorical attribute is one hot encoded into binary flags that indicate the type of manipulation. In figure 6.2, an example of one hot encoding can be found. If the dataset contains one enhancement (e.g. grayscale), two columns are encoded into the algorithm. If the dataset contains two enhancements (e.g. grayscale and hudson), then three columns are added and so forth. The performance of the filter features resembles the one the detection features achieved. Simply the indication of the enhancement of one image is not sufficient to increase the classifier's performance. Both of these type of features are very similar with each other with one key difference between them. In the detection features, it is assumed that the type of manipulation is not known to the machine learning engineer, whereas in the filter features this information is known.

6.5. Ensemble of adversarial techniques

This chapter concludes with experiments testing the effect of all aforementioned techniques. Certain techniques have been selected and used in combination in order to improve the detection rate. It is proved that the most effective way of increasing performance when having edited images in the dataset is adversarial training. Therefore, 50% of the images in the training set would be edited versions of the original images. In addition, the data representation has been dropped to 84 quantized bins. Detection and Filter features have not been selected to be included in this set because they did not perform as well as other techniques. The results that are shown in table 6.4 are the mean performances achieved for each type of enhancement. The full tables with all the detailed results are listed in tables A.17, A.18, A.19, A.20 in the appendix.

	Ensemble of adv. techniques	
Dataset - Filter	**train ratio = 0.5 & bins = 84**	**train ratio = 0 & bins = 256**
Grayscale + Original	0.84 ± 0.02	0.65 ± 0.02
Hudson + Original	0.80 ± 0.02	0.77 ± 0.02
Toaster + Original	0.78 ± 0.03	0.66 ± 0.02
Ensemble of filters (Edited) + Original	0.78 ± 0.02	0.70 ± 0.02

Table 6.4: Ensemble of adv. techniques results on different filters.

Grayscale images had the most severe effect on the classifier, and as it is seen from the experiments it achieves not only the highest increase in performance but also the highest performance in general. Combining different adversarial techniques clearly benefits the classifier different parts of the model are improved. Feature squeezing reduces the complexity of the data representation and adversarial training introduces new examples in the training process.

6.6. Discussion

In summary, four main techniques inspired from adversarial machine learning were tested in this chapter to evaluate their impact in visual weather prediction in filtered images. These are: Adversarial training, feature squeezing, detection of enhancement and an ensemble of adversarial techniques. Each experiment consists of several smaller experiments. In these experiments the ratio of edited images in the test set ranges from zero up to 100%. The complete tables with all the details can be found in the appendix.

This chapter is written in order to answer a very important question in our work: *Can we improve the performance of the weather class classifier by using techniques motivated by Adversarial Machine Learning?* Overall, adversarial machine learning techniques had a positive effect on the classifier's performance. Adversarial training had the biggest impact, whereas detection of enhancement did not majorly influence the classifier. Encoding the type of feature enhancement into the classifier does not seem to contribute to the final decision and further investigation needs to be carried out.

7
Human perception study

7.1. Introduction

The goal of this chapter is to answer the third research question: *How do filters affect users' perception in detecting the weather conditions depicted in images?* The impact of filters is reflected by measuring how much do people seem to agree or disagree about the weather conditions depicted in the images. In addition, to the quantitative analysis that is being carried out, qualitative analysis is executed as well. In the survey, the participants are asked to explain their logic behind their choices with 1-3 sentences. These sentences are carefully reviewed in order to separate the low quality submissions from the higher ones but also to detect patterns in user behaviour. Several key findings are extracted from the free-text responses. The filters that are included in this chapter are the same with the previous chapter: Grayscale, Toaster and Hudson.

7.1.1. Crowdsourcing platform

The study is run on *Prolific*[1] which is a relatively new crowdsourcing platform that was founded in 2014. In the platform there are over 25.000 users around the world. A large portion of the userbase (approximately 77%) speaks English as their first language as seen in their very detailed demographics' page. Several other statistics can be found, such as the distribution of age, type of employment and highest education level. Similarly with other crowdsourcing platforms, there are two sides, on one side there are the scientists that are interested in conducting research and on the other side, there are people that can create accounts to complete tasks and earn monetary rewards.

Several key characteristics lead us into selecting Prolific over other crowdsourcing platforms to run our study. Data quality is very important and it can majorly affect this study. The website is advertised as a platform that is primarily used for research and that the participants are highly reliable. It is important to have participants that care to a degree and are legitimately interested in participating in research. Highly motivated participants can raise the quality of the data by providing detailed explanations. As it is mentioned in the platform's website it is important no exploitation is happening by the researchers and that the participants are treated are fairly. Furthermore, Prolific ensures that the rewards are fair (in comparison with other platforms) with a minimum hourly wage of 6.50 USD. Subsequently, the participants feel safe and protected, thus their productivity increases. On the other hand, *Amazon Mechanical Turk* offers flexibility on the design of the study template that can ease large-scale data annotation. In Prolific there are guides on how to integrate studies from different external platforms such as *Qualtrics* or *Google Forms*, of which the latter is used for the survey presented in this chapter.

7.1.2. Image collection

Similarly with the experiments in the previous chapters, a preselection of cloudy and sunny images was made. Only images that have been licensed under the proper categories are shared with the participants. Therefore, a large portion of images cannot be considered as it is not allowed to share or/and adapt them in any way. It is important to adhere to these licenses in order to ensure that the people who are sharing these photos are protected. The different types of *Creative Commons (CC)* licenses were discussed in chapter 3. Furthermore, a cloudy subclass is defined and selected for the human perception study. Although, the weather conditions in the images of *Image2weather* were manually identified, the cloudy class appears to have images where sunshine is visible. Images where the weather appears to be overcast and there is no sunshine are manually selected in order to avoid any confusion. In addition, sunny images are included in the study. No other weather conditions (foggy, snowy, etc.) were included. Cloudy and sunny images have distinct differences with each other, whereas foggy and overcast images can be easily confused with each other.

After the final stage of preprocessing the final set of images is selected based on the content and scenes that are displayed in them. It is important to have a diverse set in order to investigate the impact of different visual cues that

[1] https://prolific.ac/

can lead the participants towards a decision. In the final set, there are images that show people whose appearance can be very important in order to make a judgment. In these images, people wear glasses (indicating sunny weather), coats or t-shirts. However, if participants only use this type of information they can be deceived, since there are images where the semantics can point to a different weather condition than the actual one. In the diverse set of images there exists a small portion of images that does not show any people and they are taken from an upward angle with limited background. Lastly, in the selected set, there are photos depicting famous monuments/landmarks in cities around Europe.

7.1.3. Pilot study

Before running the main study on Prolific, it is important to understand if the survey is designed properly, if certain questions need to be re-phrased or if something in the description is unclear. In addition, a pilot study makes it possible to accurately estimate how much time is needed for a participant to complete the survey. The pilot study was designed in order to further to improve the main study and none of the answers are included in the final analysis of the results. A possible approach of conducting the pilot study, is to run it separately from the main study on Prolific solely to gather feedback. By running the pilot study on Prolific it is possible to get familiar with the platform and the process of reviewing submissions. However, a drawback of this approach is that it requires resources and it relies on participants to give reliable feedback. For this reason, the pilot study was shared with people that have closely related research interests. The majority of people who participated into the pilot study are professionals within academia. This resulted in a set of very useful feedback since the people who participated were mostly skilled and experienced in designing surveys. After the pilot study, only some minor changes were made to the design according to the feedback. Even though, a pilot study was run, the study needed to be adjusted once again after the comments provide by the support team of Prolific.

In total, eight different versions of the same survey have been designed and run on Prolific. These eight different versions are very similar with each other in terms of structure, but their goals and images are different. One of these eight surveys was chosen and shared in the pilot study. It is important to understand that potential flaws of a different version of the study may remain hidden until its execution on Prolific. After successfully running all eight versions on Prolific, no specific version of the studies stood out as being the most problematic in terms of design.

7.2. Demographics

Country of birth	Percentage
UK	40.8
US	11.2
Poland	7.92
Portugal	6.25
Other	33.83

Employment status	Percentage
Full-Time	43.8
Part-Time	16.2
Unemployed (and job seeking)	12.9
Not in paid work	5.42
Other	21.68

Age group	Percentage
18-25	37.1
26-35	39.2
36-45	14.2
46-55	6.25
56-65	1.67
Unknown	1.67

Sex	Percentage
Female	47.5
Male	52.5

Table 7.1: Participant demographic information (country of birth, employment status, age group, sex).

In table 7.1, there is detailed information regarding the country of birth, employment status, age group and sex of the people who participated in the studies and their submissions are included in the analysis. Prolific is founded and has offices in the *UK*. Therefore, a lot of people are originated from the *UK*. As it is noticed, approximately 76.3% are below 36 years of age. There is balance between male and female participants. This study does not require any prior knowledge or technical skills. It is not essential for the users to be familiar with Instagram or Instagram filters in particular, since photo editing and sharing are practices that do not solely exist online. However, it would be ideal to have people who are interested in the research topic or photography in general. The study tries to replicate what happens online and for that reason a sample size of 480 unique image-participant pairs was determined. In total, 240 people participated on Prolific (30 in each study). It is ensured to have different participants in each study with the help of the pre-screening options provided by Prolific. It is important to note that the eight surveys were carried out sequentially and not all at the same time in order to ensure to have new participants in each study. There are approximately 60.000 active participants in the platform from which 14.664 have participated in at least 50 studies. By increasing the amount of completed studies the amount of participants drops as follows. All of the following statistics were extracted at the time writing this thesis and represent the amount of active participants in the platform.

- 8.556 people have participated in at least 100 studies.
- 541 people have participated in at least 500 studies.

- 43 people have participated in at least 1000 studies.

After a couple of studies it was very quickly noticed that the quality of results was low and it was extremely to hard to get any data that followed the guidelines. Participants did not follow instructions, were skipping questions or were giving invalid answers. Despite the low quantity of needed responses, the amount of approved answers was still low. After communicating with the team of Prolific, it was made known that several of the participants that were rejected were banned-removed due to fraudulent activity. Therefore, additional pre-screening options were deemed necessary. Only the participants who have approval rate higher than 60%, have completed at least 15 studies, have not participated in any of our studies before and are fluent in English were allowed to work on the surveys. The available amount of participants dropped from approximately 60.000 down to approximately 20.000. Due to the pre-screening option of fluency in English, a lot of people are originated from the *UK* and the *US*. Information regarding the number of rejections and approvals is available to the researcher, thus a scatter plot is created and shown in the appendix in figure A.6. It is noticeable from the scatter plot that there are only seven participants with six rejections or higher.

Although, the slots were not filled that quickly as before without the pre-screening filters, the quality of results rose dramatically. Despite having pre-screening filters, a small portion of people's work still was rejected because something went wrong with their submission. In the instructions it is written that they need to provide three answers for each image set and leave the rest blank. In addition, in the last stage of the survey, the users are asked to explain their choices with 1-3 sentences and follow the guidelines very closely because their work may be rejected. Unfortunately, some of the sentences were very short, general, partially or completely unclear, thus these submissions were rejected. While reviewing the submissions, and looking more closely at the explanations provided something became very clear. A small portion of the sentences had bad grammar, misspelling mistakes and typos. People perhaps did not pay attention to these mistakes, were in a hurry to complete as fast as possible the survey or do not seem to possess the required level of English that is indicated in the pre-screening options. It is not known how Prolific ensures that the users provide truthful information during registration but users seem to make several mistakes.

7.3. Experimental design

The study in *Prolific* is titled *What's the weather in the picture?* and in figure 7.1 an overview is displayed. In total, there are eight different surveys, two for each filter/representation for the two weather conditions. In a cloudy survey, there are four lists that consist of different photos. In each list there are four cloudy and two sunny images. This is not made known to the participant whose task is to select three images that seem to be cloudy and rank them based on how likely it is to depict cloudy weather (no mention about sunny weather is being made). The image which seems to the participant to be the cloudiest, is ranked in the first place. In the quantitative analysis, the results are analyzed in two ways. In scenario *'No S'*, if the participant has selected the two cloudy images, these are labelled as cloudy, otherwise they are labelled as sunny. It is important to have a fair process of evaluating the participant's ability in detecting weather conditions in images. The participant is allowed (to a degree) to make false predictions and rank a sunny image higher than a cloudy one. In scenario *'S'* an image is labelled as cloudy if it is ranked in any of the first two positions. In a similar manner, the procedure is repeated for the sunny side of the experiment, but the participants are tasked to rank the images based on how sunny they appear to be. This process results into gathering human annotated labels that are either true negatives or false negatives. Designing a human study where participants efficiently use their time is important. The design of the study encourages participants to make comparisons between images that co-exist in a list (and also across lists) and it is easier than making absolute judgements. By asking from the participants to rank images the participants are really engaged in the experiment and they think about their choices.

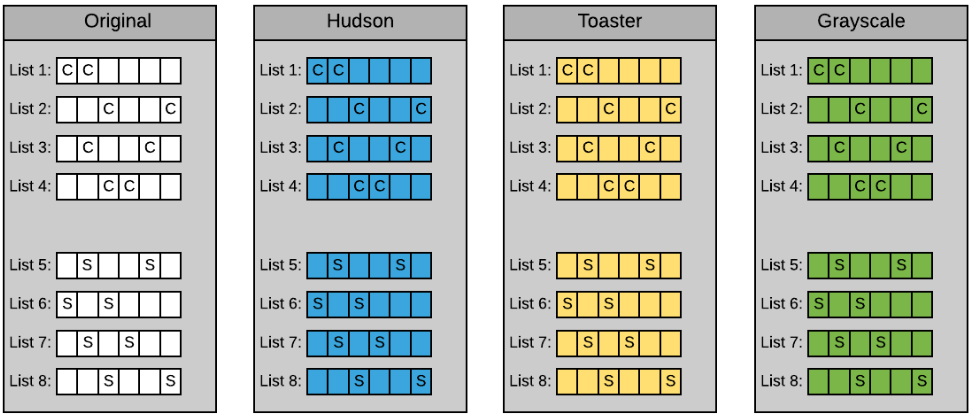


Figure 7.1: Structure of the main body of the human perception study.

In total, there are 48 unique images in different filters/representations. The order of images and which images co-exist in a list are chosen randomly. In a survey, the images belong to the same representation/filter. If there were different filters, the participant would pay more attention to these rather than in the weather conditions. A different condition that needs to be fulfilled is to have different people work on experiments that are run on different days. This is ensured by making use one of the pre-screening options offered by Prolific as it was previously mentioned. This is important because participants need to work without having seen a different version of the image that they are supposed to work on. Without this condition, the participant would make decisions based on the original representation, thus the filter would not play an important role in the final decision. After the execution of the last survey, all participant ids are unique and appeared only once in any of the eight studies. In addition, Prolific mentions in its website that they have different mechanisms in place in order to prevent duplicate accounts.

7.4. Structure of study

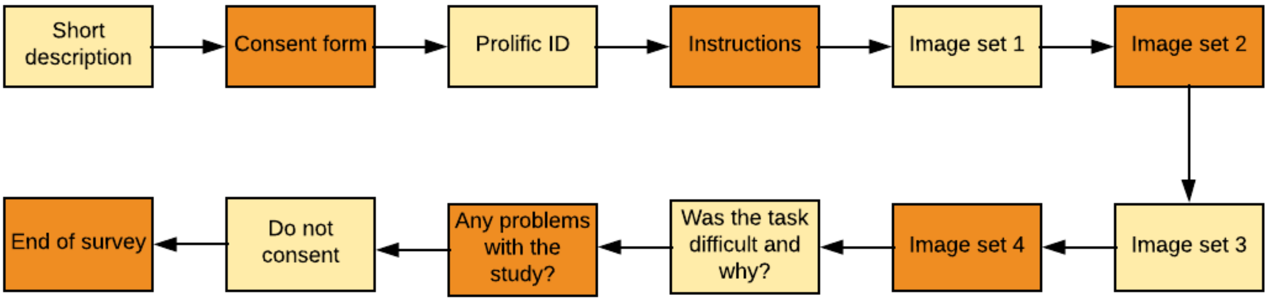


Figure 7.2: Structure of crowdsourcing survey.

In the previous section, the main body of the study was explained. There are other sections that precede the main body and other sections that follow. The study consists of the following sections: *Short description, consent form, Prolific ID, instructions, image set 1, image set 2, image set 3, image set 4, was the task difficult and why, any problems with the study, do not consent* and *end of study*. Before collecting any data, it is important to briefly describe the survey and also get the participants' consent. The consent section informs the participants about the research purpose of the study and also who is running it. There is a short explanation of what should be expected in terms of answer quality, study duration and reward. Lastly, the participants are warned that if they do not follow the instructions it is possible to have their submission rejected. The participant chooses between two multiple choice boxes: *"I consent"* which initiates the study and if for any reason the participant does not want to contribute there is the option *"I do not consent"* which redirects to the end of the survey. The section that follows simply asks for the participant's Prolific ID.

In the "Instructions", the experiment is clearly described by explaining the design and type of answers that need to be provided. The participants are reassured that there is no correct or wrong answers and that we are interested in their opinions. However, they are warned that invalid answers may lead into rejection. The main body of the study consists of four image sets with six images each. Under each image there are three multiple choices:

- 1 - First most likely to have been taken in sunny weather.
- 2 - Second most likely to have been taken in sunny weather.
- 3 - Third most likely to have been taken in sunny weather.

The participant has to only complete the multiple choice for three images. If there are less or more than three multiple choices completed for each image set or if the same answer has been used for two different pictures then the participant's work is rejected. In Prolific, there are clear guidelines on when and why to reject low quality submissions in order to judge one person's work fairly. In combination with the multiple choice answers, the participants are asked to choose the most challenging image set and also explain their choices with 1-3 sentences. In their sentences, they are asked to make reference to specific images and avoid general remarks. In a qualitative manner, their answers are checked for patterns and possible reoccurring visual cues that helped them detect the weather conditions. In conclusion, it is important to understand their logic behind their choices and it is important for us to be aware about any possible problems that they may encountered during the study or if they have other comments or suggestions. Eight different *Google Forms* are designed and shared through *Prolific*.

7.5. Quantitative analysis

Fleiss' Kappa [19] was estimated in order to determine the level of agreement between the participants. In total, 60 participants make a judgment on 16 different photos. If = 1 then there is absolute agreement between the participants. A confidence level of $\alpha = 0.05$ is defined. In addition to *Fleiss' Kappa*, the detection rate was estimated in order to see if the participants were able to accurately detect the weather conditions that exist in the images. The results are shown in table 7.2, in two different scenarios *'S'* and *'No S'*. Scenario *'S'* is stricter than scenario *'No S'*. The participants are tasked to select three out of six images and rank them based on how sunny and cloudy they are. In the sunny experiments, the image sets consist of 4 cloudy and 2 sunny images. For scenario *'A'* if the participants do not rank the sunny images in the first two positions the images are labelled as cloudy. In scenario 'B' an image is labelled as cloudy if it was selected for any of the first three positions. The images are labelled accordingly for the cloudy experiments as well.

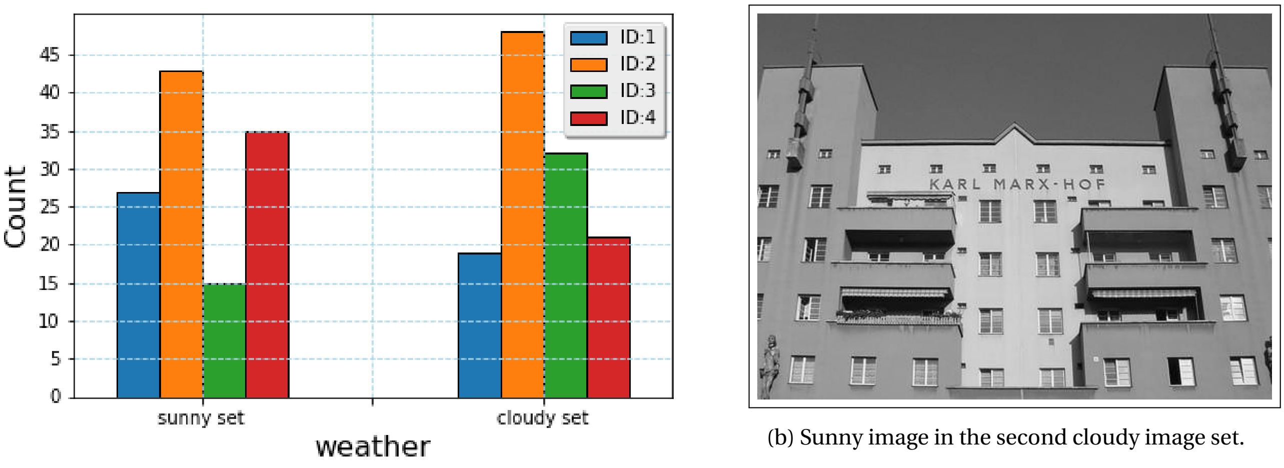
- There was an almost perfect agreement between the 60 participants' judgments on the Original images, $\kappa_A = 0.95$, $\kappa_B = 0.89$, $p - value < 0.05$.
- There was a almost perfect agreement between the 60 participants' judgments on the Hudson images, $\kappa_A = 0.91$, $\kappa_B = 0.79$, $p - value < 0.05$.
- There was a moderate agreement between the 60 participants' judgments on the Toaster images, $\kappa_A = 0.67$, $\kappa_B = 0.42$, $p - value < 0.05$.
- There was a substantial agreement between the 60 participants' judgments on the Grayscale images, $\kappa_A = 0.91$, $\kappa_B = 0.65$, $p - value < 0.05$.

Filter	**Detection rate**		**Fleiss' Kappa**		**p-value**	
	No S	S	No S	S	No S	S
Original	98%	97%	0.95	0.89	0	0
Hudson	97%	94%	0.91	0.79	0	0
Toaster	90%	81%	0.67	0.42	0	0
Grayscale	97%	89%	0.91	0.65	0	0

Table 7.2: Detection rate and Fleiss' Kappa results.

By design the Fleiss' kappa is expected to be high, since there are only two categories (cloudy-sunny). In addition, by increasing the number of annotators the evidence of agreement or disagreement increases. The filters can be ordered based on their Kappa values as follows: Toaster, Grayscale, Hudson and Original. The participants seem to struggle with the Toaster images which a Kappa value of 0.42 was extracted. Next, the absence of colour in the Grayscale images seems to cause confusion between the participants ($\kappa = 0.65$). The purpose of this chapter is to answer the third research question: *How do filters affect users' perception in detecting the weather conditions depicted in images?* It is clear after the quantitative analysis of the data that the distinct look of the Toaster images and the absence of colour in the Grayscale images majorly influences users' perception.

7.6. Qualitative analysis

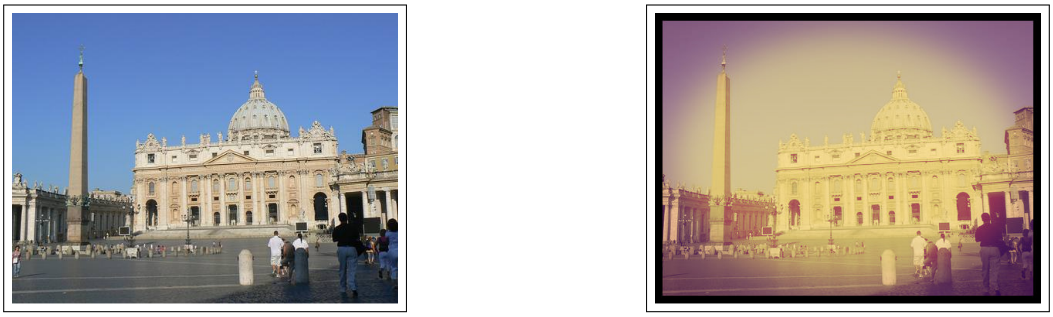


(a) Most challenging set for each type of weather.

(b) Sunny image in the second cloudy image set.

Figure 7.3

In the last stage of the survey, the participants are asked to select the most challenging set and explain their thinking. In figure 7.3a, the total amount of times an image set was selected is displayed for the eight image sets for all different representations/filters. As it can be seen from figure 7.3a, the second image sets have been selected for the sunny and cloudy images respectively. These two sets have been labelled as most challenging the most times with the grayscale versions of the images. In their responses, the participants highlight the fact that in these image sets, there are no distinct cloud shapes and that there are no shadows. In addition, in the cloudy set people make reference to a specific image which is depicted in 7.3b and is a photo of *Karl-Marx-Hof* which is located in Vienna. People make reference to this image as being the most challenging to understand the weather conditions since there are no clues such as people or shadows and only a limited portion of the sky is shown.



(a) Original representation.

(b) Toaster representation.

Figure 7.4: St. Peter's Basilic in the Vatican.

In the free-text responses, participants point out that the lack of visual cues makes it harder to determine is either cloudy or sunny. Visual cues consist of shadows, clothes, sunglasses, etc. Some of the participants, made their decision based on the location where the image was taken. One of the participants mentions that *"I´ve chosen image 4a and 4d as most likely taken at cloudy weather, because they're 100% from London and cloudy weather is typical for Great Britain."*. However, image 4a is actually an image of St. Peter's Basilica which is located in the Vatican and it is sunny as it is depicted in figure 7.4a. A different participant claims that *"For example, several of the images were taken in Italy and I recognize that there is a much greater chance of sunshine in those images than those of the UK."*. Location plays an important role in the participants' decision and can deceive them.

This task seems to be easier for someone who takes photos and has a trained eye in identifying the colors of an image. As a participant points out *"First, it was not that difficult for me because I am a photographer."*. Although, the set which this participant worked on is rather easy, it helps if you take photos, because it shows interest in photography which can motivate the participant to work.

A corpus is generated by the combination of the free-text responses. This corpus is depicted as a word-cloud in the appendix in figure A.7. Several words seem to appear quite often in the explanations of the participants. Words such as *wearing, sunglasses, sky, people, similar, bright, blue, cloud, sun* and others.

7.7. Discussion

In the survey, the participants pointed out several visual cues that were used in order to make a decision about the weather conditions that exist in an image. These cues constitute the context which surrounds the image and it is often neglected in classifiers. In the weather detection algorithm that was inspired from prior work several features have been implemented in order to measure cloud cover. However, no feature has been designed in order to encode and use the context in the classifier. Objects as sunglasses, jackets and winter clothes in general can be detected and encoded as additional information in the classifier. However, object detection may have high cost in training and properly fine-tuning. In addition, unfortunately context is not always available in photos which do not contain any people in them. Furthermore, the participants pointed out that location played an important role in their decision. The participants were able to identify the location because the image collection was consisted of photos with popular monuments and landmarks. Location can be estimated by the detection of landmarks and popular buildings in photos, but it is more challenging in photos without any easily recognisable buildings. The aforementioned features that encode the context of a photo needs to be used in combination with more traditional features in systems because this type of information is more rare.

7.7.1. Experiences with Prolific

In total eight different surveys needed to be executed. In the beginning, there were some difficulties gathering the number of responses needed for analysis. Although, in the platform's website is advertised that the participants are reliable, this was not the case. A lot of accounts were not interested in completing the survey successfully but rather fast. In addition, a lot of accounts were banned since they were deemed as fraudulent by the team of Prolific. A limit of maximum amount of rejected submissions is set at approximately 20% each time a researcher publishes a survey. The limit is set at 20% for all surveys no matter what the pre-screening options are. If one study requires from the participants to have at least 90% approval rate, the number of rejected submission is expected to be lower than a survey with no pre-screening restrictions. The threshold of 20% proved to be quite low and it was quickly reached for the first couple of surveys.

It was needed to reject more than the limit set by Prolific, thus a request was submitted from the Prolific's platform in order to increase the rejection limit. While reviewing submissions, it is important to justify to the participant why his/her submission got rejected. However, it was also needed to justify why the users got rejected to the support team of Prolific. The support team of Prolific made suggestions in order to improve the design of the survey. Certain adjustments were made to the design of the survey in order to avoid any confusion. Some of the submissions, were rejected due to their free-text response. In these cases, the support team of Prolific were convinced that the participants followed the instructions and proposed to show some leniency. It is important to have data of high quality in order to be able to extract meaningful information. Accepting submissions that failed to follow instructions can have a negative impact on the research carried out here, thus a lot of submissions were rejected.

Furthermore, it was suggested from the support team of Prolific to first contact the participants to either repeat or "return" their submission. Returning the submission means that the user does not get compensated for the work, his/her Prolific score does not change and an slot opens for a different participant. Only a participant or an admin can "return" a submission. If too many participants return their submissions in surveys, then their Prolific score is not a realistic representation of their quality as a participant. Solely the participants for whom no data seemed to be retrieved were asked to repeat the survey. Due to the low limit of rejections the messaging system of Prolific was extensively used but a lot of participants did not reply back.

Lastly, the support team of Prolific was very cooperative and helped to increase the limit of rejections whenever it was needed. As it is advertised in the platform's website, Prolific ensures that the participants are treated and rewarded fairly. However, this comes to the cost of researchers who have very limited power on whom to reject.

8
Conclusion

In this thesis, three research questions were asked regarding Instagram filters and their impact on weather detection classifiers and human perception. In chapter 4, a classifier that is able to detect sunny and cloudy images solely based on visual cues and other related metadata is designed. This classifier was tested in chapter 5 with three filters, Toaster, Hudson and Grayscale. It was evident with the experiments that were executed that the performance of the classifier drops with any type of filter. Therefore, several techniques inspired from adversarial machine learning were used in chapter 2.2. Some of the techniques, such as adversarial training had a positive effect to the classifier. Lastly, a study was designed in order to test the effect of the filters on human perception. In chapter 7, a crowdsourcing experiment is explained in detail. From the analysis of the data is proved that the distinct look of filters causes confusion between annotators.

8.1. Limitations

Throughout this thesis, certain problems and challenges appeared that needed to be overcome. The methodology presented throughout this thesis has certain limitations that will be discussed in this section. It is noteworthy to analyze and understand these not only for the purposes of this work but also for the future work.

Although, the *Image2weather* dataset [14] is rich with its different weather attributes and also its high numbers per weather class, the cloudy class is not strictly defined. In [38] the cloudy class is more strictly defined and it contains images that can be described as overcast where there is little to no sunshine. This does not hold true for the dataset used in this work as presented in chapter 3 where cloudy seems to have multiple definitions. This can be explained by the different methodology of data labelling. In [38], there is an elaborate labelling process that consists of different annotators in different steps. By reading the work presented in [14] it is unclear how the annotation of a dataset this size was performed. It was the intention of this work to focus on specific weather classes. Therefore, there is uncertainty how this work applies to other weather classes.

In the pipeline of this thesis, several classifiers were involved in order to detect different phenomena. In chapter 3 it is described how the edited/enhanced images were removed from the dataset. For this purpose, two different classifiers [17], [26] were implemented and their performance can certainly be further improved since they achieve 0.69 and 0.64 accuracy respectively. Due to the relatively low performance and unknown type of manipulations that exist in our photos, the decisions of these two classifiers were combined. Not knowing which images have been pre-enhanced and in which way could be a really difficult issue to solve. One possible solution, would be to have these images labelled by people. This approach not only can have high monetary cost but also the dataset that would be produced could still be far from an ideal gold-truth dataset.

Visual weather prediction involves detecting the sky and having certain features extracted from the sky exclusively. Due to having different multiple components in the pipeline, the feature design of the sky segmentation algorithm is simple but has been included in the pipeline due to its importance. However, it has detection rate of 0.81 tested on the AMOS dataset [30]. The algorithm was tested in a different dataset because *Image2weather* does not contain images with the sky annotated. Manually annotating the sky for these images can be cumbersome, thus the testing procedure was performed on images from a different source.

The next limitation is regarding the feature design of the visual weather classifier that was replicated and the original algorithm was presented in [14]. In their feature design one type of feature is the Gabor wavelet texture. The algorithm achieved quite high detection rate, therefore no other feature has been designed in its place. In addition, in supervised learning there are numerous classifiers, a small subset of them was tested without including *Artificial neural networks*. Fine-tuning these networks can be more challenging and they may require more resources than a different supervised learning classifier.

Instagram filters were applied with the help of *Gimp* and a similar methodology has been followed in prior work [52]. Although, images from Image2weather have not been enhanced with the actual filters from Instagram, the edited versions are not very far apart. In addition, every photo has been edited with the same three selected filters. No sophisticated methodology has been created in order to determine the most suitable filter for each photo. According to prior work [2], it is very challenging to find behavioural patterns in filter use online and in their findings users have their own preference and some of them prefer filters that stand out instead of more subtle enhancements. Modelling this behaviour is very challenging and it is not straightforward what the aim needs to be in order to ensure that the dataset resembles social images online. In addition, the dataset consists of *Flickr* images that were taken in the past and a big portion of them before *2010* which was the year of *Instagram's* release. Although, the filters are applied on photos from a different time period, this does not majorly impact this work. Since this period is not temporally far from this one. In figure A.1 in the appendix there is a histogram with the creation timestamp of these images.

8.2. Future work

The work presented here has laid the foundations for future work that can expand the investigation into other new directions. More work can be done on data augmentation. In this thesis, the effect of adversarial training was highlighted in previous chapters. The disadvantage of adversarial training is that the type of the manipulation was known and the dataset was adjusted accordingly each time. The effectiveness of data augmentation has been highlighted in [46]. Several duplicate images can be created from the original image after rotating, flipping or adding a hue. In addition, the data is augmented with new samples generated from *CycleGAN* [66]. It would be interesting to compare the effect of the techniques presented here with other data augmentation techniques that were not included in this study.

In chapter 6, detection of enhancement was tested but it proved to be insignificant. Simply encoding the type of enhancement into the classifier is insufficient. After detecting the type of enhancement a *GAN* can be implemented in order to revert the effects of the manipulation. In [42], the power of *GANs* was showcased when grayscale images were colourized. It was noted in the experiments that grayscale images had a negative effect which can be explained by looking more closely at the feature design. In the cloud features, there exists one features that calculates the difference between the colour channels. This value in the grayscale images is zero resulting into having a feature with low discriminative power. It would be interesting to investigate what needs to be achieved in order to successfully design a new type of feature that is more robust to filters and enhancements. Lastly, in future work the results of this work can be tested to see if they can be applied to other related problems. This includes other weather classes, weather attributes or other filters/enhancements.

Bibliography

[1] Yannis Avrithis, Yannis Kalantidis, Giorgos Tolias, and Evaggelos Spyrou. Retrieving landmark and non-landmark images from community photo collections. *Proceedings of the international conference on Multimedia - MM '10*, page 153, 2010. doi: 10.1145/1873951.1873973. URL `http://dl.acm.org/citation.cfm?doid=1873951.1873973`.

[2] Saeideh Bakhshi, David A Shamma, Lyndon Kennedy, and Eric Gilbert. Why We Filter Photos and How it Impacts Engagement. *International AAAI Conference on Web and Social Media (ICWSM)*, pages 12–21, 2015.

[3] Christopher T Barry, Shari R Reiter, Alexandra C Anderson, Mackenzie L Schoessler, and Chloe L Sidoti. "let me take another selfie": Further examination of the relation between narcissism, self-perception, and instagram posts. 2017.

[4] Sevinç Bayram. Image manipulation detection. *Journal of Electronic Imaging*, 15(4):041102, 2006. ISSN 1017-9909. doi: 10.1117/1.2401138. URL `http://electronicimaging.spiedigitallibrary.org/article.aspx?doi=10.1117/1.2401138`.

[5] Marcus D Bloice, Christof Stocker, and Andreas Holzinger. Augmentor: an image augmentation library for machine learning. *arXiv preprint arXiv:1708.04680*, 2017.

[6] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 97–104, 2011. ISSN 10636919. doi: 10.1109/CVPR.2011.5995413.

[7] Gang Cao, Yao Zhao, Rongrong Ni, Lifang Yu, and Huawei Tian. Forensic detection of median filtering in digital images. In *2010 IEEE International Conference on Multimedia and Expo, ICME 2010*, pages 89–94, 2010. ISBN 9781424474912. doi: 10.1109/ICME.2010.5583869.

[8] Yun Cao, Zhiming Zhou, Weinan Zhang, and Yong Yu. Unsupervised Diverse Colorization via Generative Adversarial Networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10534 LNAI, pages 151–166, 2017. ISBN 9783319712482. doi: 10.1007/978-3-319-71249-9_10.

[9] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[10] Bor Chun Chen, Pallabi Ghosh, Vlad I. Morariu, and Larry S. Davis. Detection of Metadata Tampering Through Discrepancy between Image Content and Metadata Using Multi-task Deep Learning. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July:1872–1880, 2017. ISSN 21607516. doi: 10.1109/CVPRW.2017.234.

[11] Ling Chen and Xu Lai. Comparison between ARIMA and ANN models used in short-term wind speed forecasting. *Asia-Pacific Power and Energy Engineering Conference, APPEEC*, (c), 2011. ISSN 21574839. doi: 10.1109/APPEEC.2011.5748446.

[12] Jaeyoung Choi, Martha Larson, Xinchao Li, Kevin Li, Gerald Friedland, and Alan Hanjalic. The Geo-Privacy Bonus of Popular Photo Enhancements. *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval - ICMR '17*, pages 84–92, 2017. doi: 10.1145/3078971.3080543. URL `http://dl.acm.org/citation.cfm?doid=3078971.3080543`.

[13] Wei-ta Chu, Xiang-you Zheng, and Ding-shiuan Ding. Camera as Weather Sensor : Estimating Weather Information from Single Images.

[14] Wei Ta Chu, Xiang You Zheng, and Ding Shiuan Ding. Image2weather: A large-scale image dataset for weather property estimation. *Proceedings - 2016 IEEE 2nd International Conference on Multimedia Big Data, BigMM 2016*, pages 137–144, 2016. doi: 10.1109/BigMM.2016.9.

[15] Valentina Conotter, Duc-Tien Dang-Nguyen, Giulia Boato, María Menéndez, and Martha Larson. Assessing the impact of image manipulation on users' perceptions of deception. 9014:90140Y, 2014. ISSN 1996756X. doi: 10.1117/12.2039418. URL `http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2039418`.

[16] Yubin Deng, Chen Change Loy, and Xiaoou Tang. Aesthetic-Driven Image Enhancement by Adversarial Learning. 2017. URL `http://arxiv.org/abs/1707.05251`.

[17] Feng Ding, Guopu Zhu, and Yun Qing Shi. A novel method for detecting image sharpening based on local binary pattern. In *International Workshop on Digital Watermarking*, pages 180–191. Springer, 2013.

[18] Bruce Ferwerda, Markus Schedl, and Marko Tkalcic. Using instagram picture features to predict users' personality. In *International Conference on Multimedia Modeling*, pages 850–861. Springer, 2016.

[19] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[20] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[21] Michaël Gharbi, Jiawen Chen, Jonathan T. Barron, Samuel W. Hasinoff, and Frédo Durand. Deep Bilateral Learning for Real-Time Image Enhancement. *ACM Transactions on Graphics*, 36(4), 2017. ISSN 07300301. doi: 10.1145/3072959.3073592. URL `http://arxiv.org/abs/1707.02880{%}0Ahttp://dx.doi.org/10.1145/3072959.3073592`.

[22] Daniel Glasner, Pascal Fua, Todd Zickler, and Lihi Zelnik-Manor. Hot or not: Exploring correlations between appearance and temperature. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter: 3997–4005, 2015. ISSN 15505499. doi: 10.1109/ICCV.2015.455.

[23] Ian J Goodfellow, Jean Pouget-abadie, Mehdi Mirza, Bing Xu, and David Warde-farley. Generative Adversarial Nets. pages 1–9, 2014.

[24] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. pages 1–11, 2014. ISSN 0012-7183. doi: 10.1109/CVPR.2015.7298594. URL `http://arxiv.org/abs/1412.6572`.

[25] Aditya Grover, Ashish Kapoor, and Eric Horvitz. A Deep Hybrid Model for Weather Forecasting. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, pages 379–386, 2015. doi: 10.1145/2783258.2783275. URL `http://dl.acm.org/citation.cfm?doid=2783258.2783275`.

[26] Gokhan Gul, Ismail Avcibas, and Fatih Kurugollu. SVD based image manipulation detection. In *Proceedings - International Conference on Image Processing, ICIP*, pages 1765–1768, 2010. ISBN 9781424479948. doi: 10.1109/ICIP.2010.5652854.

[27] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2011.

[28] Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

[29] Mohammad T Islam, Nathan Jacobs, Hui Wu, and Richard Souvenir. Images + Weather : Collection , Validation , and Refinement. 2013.

[30] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (July 2015), 2007. ISSN 10636919. doi: 10.1109/CVPR.2007.383258.

[31] Byung-Ju Kim Kim, Jong-Jin Shin, Hwa-Jin Nam, and Jin-Soo. Skyline Extraction using a Multistage Edge Filtering. *International Science Index*, 5(7):67 – 71, 2011. ISSN 1307-6892. URL `http://www.waset.org/publications/10437`.

[32] Nicole C Krämer, Markus Feurstein, Jan P Kluck, Yannic Meier, Marius Rother, and Stephan Winter. Beware of selfies: The impact of photo type on impression formation based on social networking profiles. *Frontiers in psychology*, 8:188, 2017.

[33] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[35] Breiman Leo, Jerome H Friedman, Richard A Olshen, and Charles J Stone. Classification and regression trees. *Wadsworth International Group*, 1984.

[36] Qingyong Li, Weitao Lu, Jun Yang, and James Z. Wang. Thin cloud detection of all-sky images using Markov random fields. *IEEE Geoscience and Remote Sensing Letters*, 9(3):417–421, 2012. ISSN 1545598X. doi: 10.1109/LGRS.2011.2170953.

[37] Wen Nung Lie, Tom C I Lin, Ting Chih Lin, and Keng Shen Hung. A robust dynamic programming algorithm to extract skyline in images for navigation. *Pattern Recognition Letters*, 26(2):221–230, 2005. ISSN 01678655. doi: 10.1016/j.patrec.2004.08.021.

[38] Cewu Lu, Di Lin, Jiaya Jia, and Chi Keung Tang. Two-Class Weather Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2510–2524, 2014. ISSN 01628828. doi: 10.1109/TPAMI.2016.2640295.

[39] Jessica L McCain, Zachary G Borg, Ariel H Rothenberg, Kristina M Churillo, Paul Weiler, and W Keith Campbell. Personality and selfies: Narcissism and the dark triad. *Computers in Human Behavior*, 64:126–133, 2016.

[40] Radu P. Mihail, Scott Workman, Zach Bessinger, and Nathan Jacobs. Sky segmentation in the wild: An empirical study. *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, 2016. doi: 10.1109/WACV.2016.7477637.

[41] Srinivasa G. Narasimhan, Chi Wang, and Shree K. Nayar. All the Images of an Outdoor Scene. *Computer Vision - ECCV 2002*, 2352:148–162, 2006. ISSN 16113349. doi: 10.1007/3-540-47977-5. URL `http://www.springerlink.com/index/10.1007/3-540-47977-5`.

[42] Kamyar Nazeri and Eric Ng. Image Colorization with Generative Adversarial Networks. pages 1–11, 2018. URL `http://arxiv.org/abs/1803.05400`.

[43] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[44] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[46] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[47] S. A. Pourmousavi Kani and M. M. Ardehali. Very short-term wind speed prediction: A new artificial neural network-Markov chain model. *Energy Conversion and Management*, 52(1):738–745, 2011. ISSN 01968904. doi: 10.1016/j.enconman.2010.07.053. URL `http://dx.doi.org/10.1016/j.enconman.2010.07.053`.

[48] Muhammad Ali Qureshi and Mohamed Deriche. A bibliography of pixel-based blind image forgery detection techniques. *Signal Processing: Image Communication*, 39:46–74, 2015. ISSN 09235965. doi: 10.1016/j.image.2015.08.008.

[49] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008. ISSN 09205691. doi: 10.1007/s11263-007-0090-8.

[50] Matthew Stamm and K. J. Ray. Blind forensics of contrast enhancement in digital images. In *Proceedings - International Conference on Image Processing, ICIP*, pages 3112–3115, 2008. ISBN 1424417643. doi: 10.1109/ICIP.2008.4712454.

[51] Thomas Stone, Michael Mangan, Paul Ardin, and Barbara Webb. Sky segmentation with ultraviolet images can be used for navigation. *Proceedings Robotics: Science and Systems*, pages 1–10, 2014. doi: 10.15607/RSS.2014.X.047. URL `http://www.roboticsproceedings.org/rss10/p47.pdf`.

[52] Wei Tse Sun, Ting Hsuan Chao, Yin Hsi Kuo, and Winston H. Hsu. Photo Filter Recommendation by Category-Aware Aesthetic Learning. *IEEE Transactions on Multimedia*, 19(8):1870–1880, 2017. ISSN 15209210. doi: 10.1109/TMM.2017.2688929.

[53] Litian Tao, Lu Yuan, and Jian Sun. SkyFinder. *ACM Transactions on Graphics*, 28(3):1, 2009. ISSN 07300301. doi: 10.1145/1531326.1531374. URL `http://portal.acm.org/citation.cfm?doid=1531326.1531374`.

[54] Bart Thomee, Jose G Moreno, and David A Shamma. Who's time is it anyway?: Investigating the accuracy of camera timestamps. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 909–912. ACM, 2014.

[55] Anna Volokitin, Radu Timofte, and Luc Van Gool. Deep Features or Not: Temperature and Time Prediction in Outdoor Scenes. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1136–1144, 2016. ISSN 21607516. doi: 10.1109/CVPRW.2016.145.

[56] Ruoxu Wang, Fan Yang, and Michel M Haigh. Let me take a selfie: exploring the psychological effects of posting and viewing selfies and groupies on social media. *Telematics and Informatics*, 34(4):274–283, 2017.

[57] C. L. Wu, K. W. Chau, and C. Fan. Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques. *Journal of Hydrology*, 389(1-2):146–167, 2010. ISSN 00221694. doi: 10.1016/j.jhydrol.2010.05.040.

[58] Jianxiong Xiao, James Hays, Krista A Ehinger, and Antonio Torralba. SUN database : Large-scale scene recognition from abbey to zoo The MIT Faculty has made this article openly available . Please share Citation Institute of Electrical and Electronics Engineers Publisher Version Accessed Citable Link Terms of Use Detailed T. 2013.

[59] Ali Pour Yazdanpanah, Emma E. Regentova, Ajay Kumar Mandava, Touqeer Ahmad, and George Bebis. Sky segmentation by fusing clustering with neural networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013. ISBN 9783642419386. doi: 10.1007/978-3-642-41939-3_65.

[60] Raymond A Yeh, Chen Chen, Teck-Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, volume 2, page 4, 2017.

[61] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. Wavelets for Image Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. ISSN 10577149. doi: 10.1109/TIP.2003.819861.

[62] Valentina Zantedeschi, Maria-irina Nicolae, and Ambrish Rawat. Efficient Defenses Against Adversarial Attacks. pages 1–16, 2017. doi: 10.1145/3128572.3140449. URL `http://arxiv.org/abs/1707.06728`.

[63] Zheng Zhang, Huadong Ma, Huiyuan Fu, and Cheng Zhang. Scene-free multi-class weather classification on single images. *Neurocomputing*, 207:365–373, 2016. ISSN 18728286. doi: 10.1016/j.neucom.2016.05.015. URL `http://dx.doi.org/10.1016/j.neucom.2016.05.015`.

[64] Huadong Ma Zheng Zhang. MULTI-CLASS WEATHER CLASSIFICATION ON SINGLE IMAGES. In *International Conference on Image Processing (ICIP)*, pages 4396–4400, 2015. ISBN 9781479983391.

[65] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[66] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.

A

Appendix

A.1. Dataset

A.1.1. Detection of edited images

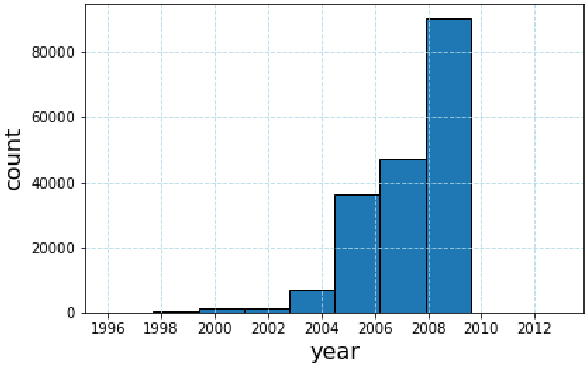


Figure A.1: Distribution of the years the images were taken.

A.2. Weather classification

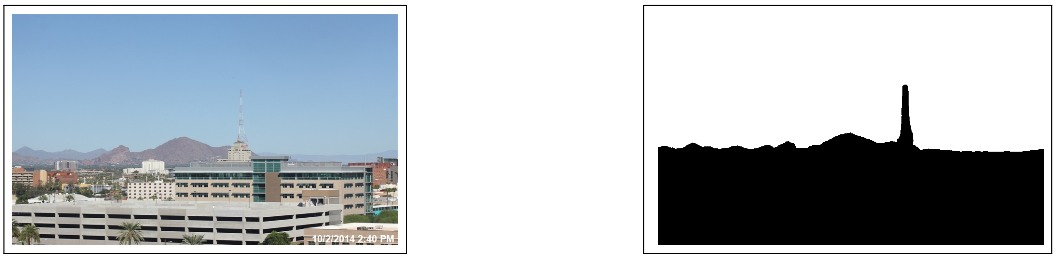
A.2.1. Sky segmentation

Lake of Biel/Bienne (Switzerland)
2011-01-14 CET 08:45:00
meteoBase :: Air: 8.7°C > Wind: 1.6km/h (0.9kn) WSW > Windpeak: 3.2km/h (1.7kn) WSW
(c)2011 Roman Christ (silentworx.ch)

(a) Train image example.

(b) Binary image - mask.

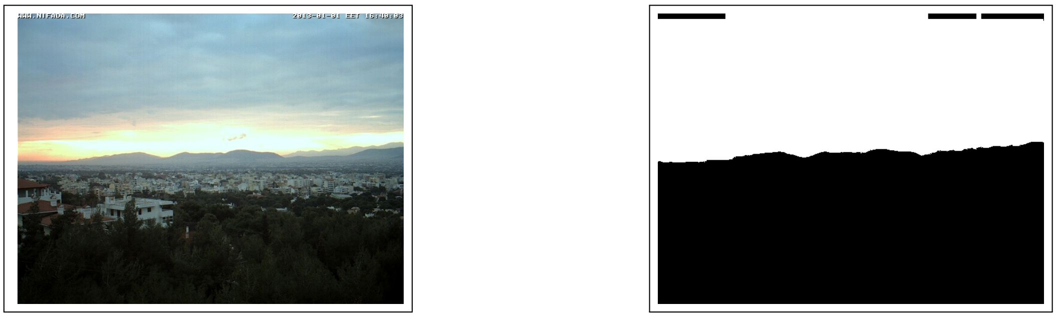
Figure A.2: Sky segmentation example, webcam id 3395.



(a) Train image example.

(b) Binary image - mask.

Figure A.3: Sky segmentation example, webcam id 4232.



(a) Train image example.

(b) Binary image - mask.

Figure A.4: Sky segmentation example, webcam id 4584.

A.3. Weather classification of Edited images

A.3.1. Instagram Filters

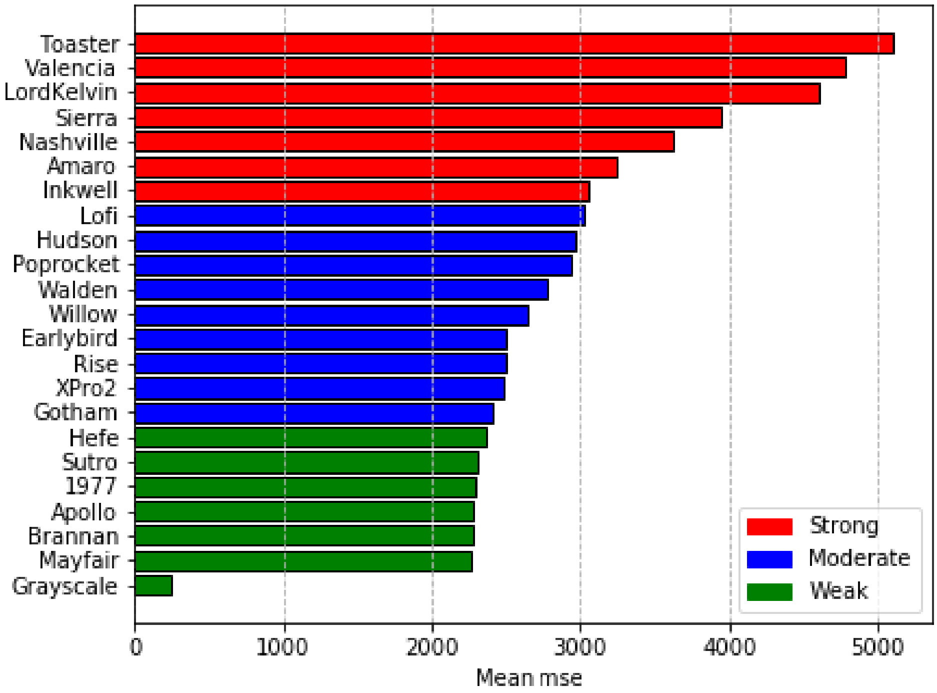


Figure A.5: Strength of Instagram filters by measuring the Mean squared error.

A.3.2. Classifier performance

Ratio of Grayscale images in the test set	**Mean accuracy ± std**	
	Grayscale+Original test set	**Grayscale test set**
0	0.82 ± 0.02	—
0.1	0.78 ± 0.02	0.52 ± 0.04
0.2	0.76 ± 0.03	0.52 ± 0.03
0.3	0.72 ± 0.03	0.55 ± 0.04
0.4	0.69 ± 0.01	0.52 ± 0.04
0.5	0.67 ± 0.01	0.50 ± 0.01
0.6	0.64 ± 0.02	0.52 ± 0.02
0.7	0.62 ± 0.03	0.55 ± 0.03
0.8	0.58 ± 0.02	0.53 ± 0.05
0.9	0.54 ± 0.03	0.52 ± 0.03
1	0.53 ± 0.05	0.53 ± 0.05
Mean accuracy	**0.65 ± 0.02**	**0.52 ± 0.03**

Table A.1: Results with Grayscale images.

Ratio of Toaster images in the test set	**Mean accuracy ± std**	
	Toaster+Original test set	**Toaster test set**
0	0.82 ± 0.02	—
0.1	0.78 ± 0.02	0.58 ± 0.11
0.2	0.77 ± 0.02	0.58 ± 0.05
0.3	0.74 ± 0.04	0.59 ± 0.07
0.4	0.71 ± 0.02	0.55 ± 0.04
0.5	0.67 ± 0.03	0.52 ± 0.05
0.6	0.65 ± 0.02	0.53 ± 0.03
0.7	0.63 ± 0.04	0.55 ± 0.06
0.8	0.59 ± 0.03	0.54 ± 0.04
0.9	0.59 ± 0.05	0.57 ± 0.05
1	0.53 ± 0.02	0.53 ± 0.02
Mean accuracy	**0.66 ± 0.02**	**0.55 ± 0.04**

Table A.2: Results with Toaster images.

Ratio of Hudson images in the test set	**Mean accuracy ± std**	
	Hudson+Original test set	**Hudson test set**
0	0.82 ± 0.02	—
0.1	0.81 ± 0.02	0.74 ± 0.08
0.2	0.79 ± 0.04	0.78 ± 0.03
0.3	0.80 ± 0.01	0.74 ± 0.08
0.4	0.77 ± 0.03	0.72 ± 0.05
0.5	0.80 ± 0.02	0.76 ± 0.03
0.6	0.76 ± 0.03	0.73 ± 0.04
0.7	0.77 ± 0.03	0.74 ± 0.04
0.8	0.75 ± 0.02	0.74 ± 0.02
0.9	0.75 ± 0.02	0.75 ± 0.02
1	0.75 ± 0.02	0.75 ± 0.02
Mean accuracy	**0.77 ± 0.02**	**0.74 ± 0.04**

Table A.3: Results with Hudson images.

Ratio of Edited images in the test set	**Mean accuracy ± std**	
	Edited+Original test set	**Edited test set**
0	0.82 ± 0.02	—
0.1	0.80 ± 0.02	0.58 ± 0.08
0.2	0.77 ± 0.03	0.60 ± 0.05
0.3	0.76 ± 0.03	0.62 ± 0.08
0.4	0.73 ± 0.03	0.62 ± 0.04
0.5	0.71 ± 0.02	0.60 ± 0.02
0.6	0.69 ± 0.02	0.61 ± 0.03
0.7	0.68 ± 0.02	0.62 ± 0.03
0.8	0.63 ± 0.02	0.60 ± 0.02
0.9	0.63 ± 0.02	0.61 ± 0.03
1	0.60 ± 0.03	0.60 ± 0.03
Mean accuracy	**0.70 ± 0.02**	**0.61 ± 0.04**

Table A.4: Results with an ensemble of edited images.

A.4. Adversarial Machine Learning

A.4.1. Adversarial training

	Ratio of Grayscale images in the training set					
	Accuracy ± std on Grayscale+Original test set			**Accuracy ± std on Grayscale test set**		
Ratio of Grayscale images in the test set	**ratio = 0.5**	**ratio = 0.3**	**ratio = 0**	**ratio = 0.5**	**ratio = 0.3**	**ratio = 0**
0	—	—	0.82 ± 0.02	—	—	—
0.1	0.80 ± 0.02	0.80 ± 0.03	0.78 ± 0.02	0.76 ± 0.07	0.74 ± 0.01	0.52 ± 0.04
0.2	0.79 ± 0.01	0.81 ± 0.01	0.76 ± 0.03	0.75 ± 0.06	0.77 ± 0.05	0.52 ± 0.03
0.3	0.77 ± 0.02	0.79 ± 0.01	0.72 ± 0.03	0.75 ± 0.04	0.77 ± 0.06	0.55 ± 0.04
0.4	0.80 ± 0.02	0.78 ± 0.03	0.69 ± 0.01	0.77 ± 0.04	0.75 ± 0.04	0.52 ± 0.04
0.5	0.75 ± 0.03	0.78 ± 0.03	0.67 ± 0.01	0.72 ± 0.05	0.75 ± 0.03	0.50 ± 0.01
0.6	0.77 ± 0.02	0.78 ± 0.03	0.64 ± 0.02	0.73 ± 0.02	0.75 ± 0.02	0.52 ± 0.02
0.7	0.76 ± 0.02	0.76 ± 0.03	0.62 ± 0.03	0.75 ± 0.02	0.74 ± 0.05	0.55 ± 0.03
0.8	0.75 ± 0.03	0.76 ± 0.02	0.58 ± 0.02	0.73 ± 0.03	0.76 ± 0.02	0.52 ± 0.02
0.9	0.75 ± 0.03	0.74 ± 0.02	0.54 ± 0.03	0.75 ± 0.03	0.73 ± 0.02	0.52 ± 0.03
1	0.77 ± 0.02	0.74 ± 0.02	0.53 ± 0.05	0.77 ± 0.02	0.74 ± 0.02	0.53 ± 0.05
Mean accuracy	**0.77 ± 0.02**	**0.77 ± 0.02**	**0.65 ± 0.02**	**0.74 ± 0.03**	**0.75 ± 0.03**	**0.52 ± 0.03**

Table A.5: Adversarial training results on the Grayscale+Original dataset.

	Ratio of Toaster images in the training set					
	Accuracy ± std on Toaster+Original test set			**Accuracy ± std on Toaster test set**		
Ratio of Toaster images in the test set	**ratio = 0.5**	**ratio = 0.3**	**ratio = 0**	**ratio = 0.5**	**ratio = 0.3**	**ratio = 0**
0	—	—	0.82 ± 0.02	—	—	—
0.1	0.79 ± 0.02	0.80 ± 0.01	0.78 ± 0.02	0.77 ± 0.06	0.75 ± 0.07	0.58 ± 0.11
0.2	0.78 ± 0.02	0.80 ± 0.02	0.77 ± 0.02	0.75 ± 0.04	0.75 ± 0.05	0.58 ± 0.05
0.3	0.79 ± 0.01	0.80 ± 0.02	0.74 ± 0.04	0.77 ± 0.04	0.74 ± 0.04	0.59 ± 0.07
0.4	0.76 ± 0.03	0.78 ± 0.01	0.71 ± 0.02	0.74 ± 0.05	0.77 ± 0.04	0.55 ± 0.04
0.5	0.76 ± 0.03	0.78 ± 0.03	0.67 ± 0.03	0.75 ± 0.05	0.76 ± 0.04	0.52 ± 0.05
0.6	0.78 ± 0.03	0.78 ± 0.03	0.65 ± 0.02	0.77 ± 0.03	0.75 ± 0.04	0.53 ± 0.03
0.7	0.78 ± 0.03	0.77 ± 0.02	0.63 ± 0.04	0.77 ± 0.03	0.75 ± 0.02	0.55 ± 0.06
0.8	0.77 ± 0.03	0.76 ± 0.02	0.59 ± 0.03	0.77 ± 0.03	0.76 ± 0.03	0.54 ± 0.04
0.9	0.79 ± 0.02	0.76 ± 0.03	0.59 ± 0.05	0.78 ± 0.02	0.76 ± 0.02	0.57 ± 0.05
1	0.77 ± 0.02	0.77 ± 0.04	0.53 ± 0.02	0.77 ± 0.02	0.77 ± 0.04	0.53 ± 0.02
Mean accuracy	**0.77 ± 0.02**	**0.77 ± 0.02**	**0.66 ± 0.02**	**0.76 ± 0.03**	**0.75 ± 0.03**	**0.55 ± 0.05**

Table A.6: Adversarial training results on the Toaster+Original dataset.

	Ratio of Hudson images in the training set					
	Accuracy ± std on Hudson+Original test set			**Accuracy ± std on Hudson test set**		
Ratio of Hudson images in the test set	**ratio = 0.5**	**ratio = 0.3**	**ratio = 0**	**ratio = 0.5**	**ratio = 0.3**	**ratio = 0**
0	—	—	0.82 ± 0.02	—	—	—
0.1	0.80 ± 0.01	0.81 ± 0.02	0.81 ± 0.02	0.77 ± 0.07	0.84 ± 0.04	0.74 ± 0.08
0.2	0.80 ± 0.03	0.81 ± 0.02	0.79 ± 0.04	0.79 ± 0.09	0.78 ± 0.07	0.78 ± 0.03
0.3	0.80 ± 0.03	0.80 ± 0.02	0.80 ± 0.01	0.76 ± 0.06	0.78 ± 0.04	0.74 ± 0.08
0.4	0.80 ± 0.03	0.79 ± 0.02	0.77 ± 0.03	0.81 ± 0.03	0.77 ± 0.03	0.72 ± 0.05
0.5	0.80 ± 0.02	0.81 ± 0.01	0.80 ± 0.02	0.79 ± 0.03	0.79 ± 0.03	0.76 ± 0.03
0.6	0.80 ± 0.02	0.80 ± 0.01	0.76 ± 0.03	0.78 ± 0.04	0.79 ± 0.03	0.73 ± 0.04
0.7	0.80 ± 0.03	0.78 ± 0.02	0.77 ± 0.03	0.79 ± 0.03	0.78 ± 0.01	0.74 ± 0.04
0.8	0.78 ± 0.02	0.79 ± 0.02	0.75 ± 0.02	0.78 ± 0.02	0.78 ± 0.03	0.74 ± 0.02
0.9	0.80 ± 0.02	0.78 ± 0.02	0.75 ± 0.02	0.80 ± 0.03	0.78 ± 0.02	0.75 ± 0.02
1	0.76 ± 0.02	0.76 ± 0.01	0.75 ± 0.02	0.76 ± 0.02	0.76 ± 0.01	0.75 ± 0.02
Mean accuracy	**0.79 ± 0.02**	**0.79 ± 0.01**	**0.77 ± 0.02**	**0.78 ± 0.04**	**0.78 ± 0.03**	**0.74 ± 0.04**

Table A.7: Adversarial training results on the Hudson+Original dataset.

	Ratio of Edited images in the training set					
	Accuracy ± std on Edited+Original test set			**Accuracy ± std on Edited test set**		
Ratio of Edited images in the test set	**ratio = 0.5**	**ratio = 0.3**	**ratio = 0**	**ratio = 0.5**	**ratio = 0.3**	**ratio = 0**
0	—	—	0.82 ± 0.02	—	—	—
0.1	0.79 ± 0.02	0.80± 0.02	0.80 ± 0.02	0.78 ± 0.09	0.71 ± 0.09	0.58 ± 0.08
0.2	0.79 ± 0.02	0.78 ± 0.03	0.77 ± 0.03	0.76 ± 0.06	0.72 ± 0.07	0.60 ± 0.05
0.3	0.79 ± 0.02	0.79 ± 0.03	0.76 ± 0.03	0.73 ± 0.07	0.72 ± 0.05	0.62 ± 0.08
0.4	0.78 ± 0.03	0.77 ± 0.03	0.73 ± 0.03	0.75 ± 0.07	0.70 ± 0.04	0.62 ± 0.04
0.5	0.77 ± 0.03	0.76 ± 0.02	0.71 ± 0.02	0.74 ± 0.04	0.70 ± 0.04	0.60 ± 0.02
0.6	0.77 ± 0.01	0.76 ± 0.01	0.69 ± 0.02	0.74 ± 0.02	0.72 ± 0.02	0.61 ± 0.03
0.7	0.75 ± 0.02	0.75 ± 0.04	0.68 ± 0.02	0.74 ± 0.03	0.72 ± 0.05	0.62 ± 0.03
0.8	0.75 ± 0.02	0.75 ± 0.01	0.63 ± 0.02	0.73± 0.02	0.74 ± 0.02	0.60 ± 0.02
0.9	0.72 ± 0.02	0.71 ± 0.03	0.63 ± 0.02	0.72 ± 0.02	0.72 ± 0.03	0.61 ± 0.03
1	0.73 ± 0.04	0.72 ± 0.02	0.60 ± 0.03	0.73 ± 0.04	0.72 ± 0.02	0.60 ± 0.03
Mean accuracy	**0.77 ± 0.02**	**0.76 ± 0.02**	**0.70 ± 0.02**	**0.74 ± 0.04**	**0.72 ± 0.04**	**0.61 ± 0.04**

Table A.8: Adversarial training results on the Edited+Original dataset.

A.4.2. Feature squeezing

	Number of quantized bins					
	Accuracy ± std on Grayscale+Original test set			**Accuracy ± std on Grayscale test set**		
Ratio of Grayscale images in the test set	**bins = 256**	**bins = 171**	**bins = 84**	**bins = 256**	**bins = 171**	**bins = 84**
0	0.82 ± 0.02	—	—	—	—	—
0.1	0.78 ± 0.02	0.79 ± 0.02	0.77 ± 0.04	0.52 ± 0.04	0.51 ± 0.02	0.55 ± 0.07
0.2	0.76 ± 0.03	0.75 ± 0.02	0.75 ± 0.02	0.52 ± 0.03	0.53 ± 0.05	0.53 ± 0.06
0.3	0.72 ± 0.03	0.73 ± 0.02	0.73 ± 0.03	0.55 ± 0.04	0.52 ± 0.03	0.54 ± 0.04
0.4	0.69 ± 0.01	0.69 ± 0.01	0.72 ± 0.02	0.52 ± 0.04	0.52 ± 0.03	0.59 ± 0.07
0.5	0.67 ± 0.01	0.67 ± 0.02	0.67 ± 0.02	0.50 ± 0.01	0.53 ± 0.04	0.54 ± 0.06
0.6	0.64 ± 0.02	0.66 ± 0.03	0.65 ± 0.02	0.52 ± 0.02	0.55 ± 0.04	0.52 ± 0.02
0.7	0.62 ± 0.03	0.61 ± 0.03	0.61 ± 0.02	0.55 ± 0.03	0.53 ± 0.03	0.52 ± 0.01
0.8	0.58 ± 0.02	0.59 ± 0.02	0.59 ± 0.04	0.52 ± 0.02	0.53 ± 0.03	0.53 ± 0.05
0.9	0.54 ± 0.03	0.62 ± 0.06	0.57 ± 0.04	0.52 ± 0.03	0.60 ± 0.06	0.55 ± 0.05
1	0.53 ± 0.05	0.52 ± 0.03	0.56 ± 0.05	0.53 ± 0.05	0.52 ± 0.03	0.56 ± 0.05
Mean accuracy	**0.65 ± 0.02**	**0.66 ± 0.02**	**0.66 ± 0.03**	**0.52 ± 0.03**	**0.53 ± 0.03**	**0.54 ± 0.04**

Table A.9: Feature squeezing results on the Grayscale+Original dataset.

	Number of quantized bins					
	Accuracy ± std on Toaster+Original test set			**Accuracy ± std on Toaster test set**		
Ratio of Toaster images in the test set	**bins = 256**	**bins = 171**	**bins = 84**	**bins = 256**	**bins = 171**	**bins = 84**
0	0.82 ± 0.02	—	—	—	—	—
0.1	0.78 ± 0.02	0.78 ± 0.02	0.79 ± 0.04	0.58 ± 0.11	0.54 ± 0.08	0.49 ± 0.09
0.2	0.77 ± 0.02	0.78 ± 0.02	0.74 ± 0.04	0.58 ± 0.05	0.56 ± 0.05	0.54 ± 0.12
0.3	0.74 ± 0.04	0.75 ± 0.02	0.74 ± 0.03	0.59 ± 0.07	0.58 ± 0.06	0.56 ± 0.05
0.4	0.71 ± 0.02	0.71 ± 0.03	0.70 ± 0.03	0.55 ± 0.04	0.56 ± 0.05	0.52 ± 0.05
0.5	0.67 ± 0.03	0.70 ± 0.02	0.66 ± 0.01	0.52 ± 0.05	0.57 ± 0.04	0.52 ± 0.02
0.6	0.65 ± 0.02	0.65 ± 0.03	0.67 ± 0.02	0.53 ± 0.03	0.55 ± 0.05	0.57 ± 0.04
0.7	0.63 ± 0.04	0.62 ± 0.04	0.62 ± 0.04	0.55 ± 0.06	0.54 ± 0.05	0.55 ± 0.05
0.8	0.59 ± 0.03	0.62 ± 0.05	0.60 ± 0.03	0.54 ± 0.04	0.57 ± 0.06	0.54 ± 0.05
0.9	0.59 ± 0.05	0.56 ± 0.04	0.57 ± 0.04	0.57 ± 0.05	0.53 ± 0.05	0.55 ± 0.04
1	0.53 ± 0.02	0.56 ± 0.05	0.55 ± 0.04	0.53 ± 0.02	0.56 ± 0.05	0.55 ± 0.04
Mean accuracy	**0.66 ± 0.02**	**0.67 ± 0.03**	**0.66 ± 0.03**	**0.55 ± 0.05**	**0.56 ± 0.05**	**0.54 ± 0.05**

Table A.10: Feature squeezing results on the Toaster+Original dataset.

	Number of quantized bins					
	Accuracy ± std on Hudson+Original test set			**Accuracy ± std on Hudson test set**		
Ratio of Hudson images in the test set	**bins = 256**	**bins = 171**	**bins = 84**	**bins = 256**	**bins = 171**	**bins = 84**
0	0.82 ± 0.02	—	—	—	—	—
0.1	0.81 ± 0.02	0.79 ± 0.02	0.80 ± 0.02	0.74 ± 0.08	0.73 ± 0.07	0.77 ± 0.07
0.2	0.79 ± 0.04	0.80 ± 0.03	0.79 ± 0.02	0.78 ± 0.03	0.74 ± 0.06	0.74 ± 0.06
0.3	0.80 ± 0.01	0.78 ± 0.01	0.81 ± 0.02	0.74 ± 0.08	0.72 ± 0.02	0.76 ± 0.04
0.4	0.77 ± 0.03	0.80 ± 0.02	0.78 ± 0.02	0.72 ± 0.05	0.76 ± 0.05	0.73 ± 0.05
0.5	0.80 ± 0.02	0.78 ± 0.02	0.77 ± 0.02	0.76 ± 0.03	0.75 ± 0.03	0.74 ± 0.04
0.6	0.76 ± 0.03	0.76 ± 0.02	0.77 ± 0.02	0.73 ± 0.04	0.71 ± 0.03	0.74 ± 0.03
0.7	0.77 ± 0.03	0.77 ± 0.03	0.76 ± 0.02	0.74 ± 0.04	0.75 ± 0.02	0.74 ± 0.02
0.8	0.75 ± 0.02	0.76 ± 0.02	0.74 ± 0.03	0.74 ± 0.02	0.74 ± 0.02	0.73 ± 0.03
0.9	0.75 ± 0.02	0.75 ± 0.02	0.74 ± 0.02	0.75 ± 0.02	0.74 ± 0.02	0.73 ± 0.02
1	0.75 ± 0.02	0.76 ± 0.03	0.74 ± 0.03	0.75 ± 0.02	0.76 ± 0.03	0.74 ± 0.03
Mean accuracy	**0.77 ± 0.02**	**0.77 ± 0.02**	**0.77 ± 0.02**	**0.74 ± 0.04**	**0.74 ± 0.04**	**0.74 ± 0.04**

Table A.11: Feature squeezing results on the Hudson+Original dataset.

	Number of quantized bins					
	Accuracy ± std on Edited+Original test set			**Accuracy ± std on Edited test set**		
Ratio of Edited images in the test set	**bins = 256**	**bins = 171**	**bins = 84**	**bins = 256**	**bins = 171**	**bins = 84**
0	0.82 ± 0.02	—	—	—	—	—
0.1	0.80 ± 0.02	0.80 ± 0.02	0.80 ± 0.03	0.58 ± 0.08	0.66 ± 0.08	0.66 ± 0.10
0.2	0.77 ± 0.03	0.77 ± 0.03	0.77 ± 0.02	0.60 ± 0.05	0.61 ± 0.08	0.63 ± 0.08
0.3	0.76 ± 0.03	0.74 ± 0.02	0.75 ± 0.02	0.62 ± 0.08	0.61 ± 0.04	0.58 ± 0.05
0.4	0.73 ± 0.03	0.72 ± 0.02	0.71 ± 0.03	0.62 ± 0.04	0.59 ± 0.03	0.58 ± 0.05
0.5	0.71 ± 0.02	0.71 ± 0.02	0.71 ± 0.03	0.60 ± 0.02	0.61 ± 0.03	0.58 ± 0.05
0.6	0.69 ± 0.02	0.67 ± 0.01	0.68 ± 0.03	0.61 ± 0.03	0.58 ± 0.02	0.60 ± 0.03
0.7	0.68 ± 0.02	0.66 ± 0.03	0.66 ± 0.02	0.62 ± 0.03	0.59 ± 0.04	0.60 ± 0.01
0.8	0.63 ± 0.02	0.65 ± 0.02	0.66 ± 0.01	0.60 ± 0.02	0.62 ± 0.03	0.61 ± 0.02
0.9	0.63 ± 0.02	0.65 ± 0.02	0.62 ± 0.03	0.61 ± 0.03	0.63 ± 0.03	0.60 ± 0.03
1	0.60 ± 0.02	0.60 ± 0.03	0.62 ± 0.02	0.60 ± 0.03	0.60 ± 0.03	0.62 ± 0.02
Mean accuracy	**0.70 ± 0.02**	**0.70 ± 0.02**	**0.70 ± 0.02**	**0.61 ± 0.04**	**0.61 ± 0.04**	**0.61 ± 0.04**

Table A.12: Feature squeezing results on the Edited+Original dataset.

A.4.3. Detection of enhancement

	Feature set used in training			
	Accuracy ± std on Toaster+Original test set		Accuracy ± std on Toaster test set	
Ratio of Toaster images in the test set	Detection features + Original features	Original features	Detection features + Original features	Original features
0	—	0.82 ± 0.02	—	—
0.1	0.79 ± 0.02	0.78 ± 0.02	0.58 ± 0.09	0.58 ± 0.11
0.2	0.78 ± 0.02	0.77 ± 0.02	0.57 ± 0.07	0.58 ± 0.05
0.3	0.74 ± 0.02	0.74 ± 0.04	0.55 ± 0.04	0.59 ± 0.07
0.4	0.71 ± 0.04	0.71 ± 0.02	0.55 ± 0.08	0.55 ± 0.04
0.5	0.69 ± 0.03	0.67 ± 0.03	0.57 ± 0.05	0.52 ± 0.05
0.6	0.66 ± 0.02	0.65 ± 0.02	0.55 ± 0.04	0.53 ± 0.03
0.7	0.63 ± 0.04	0.63 ± 0.04	0.56 ± 0.04	0.55 ± 0.06
0.8	0.61 ± 0.03	0.59 ± 0.03	0.56 ± 0.03	0.54 ± 0.04
0.9	0.57 ± 0.04	0.59 ± 0.05	0.55 ± 0.05	0.57 ± 0.05
1	0.56 ± 0.04	0.53 ± 0.02	0.56 ± 0.04	0.53 ± 0.02
Mean accuracy	**0.67 ± 0.03**	**0.66 ± 0.02**	**0.56 ± 0.05**	**0.55 ± 0.05**

Table A.13: Detection features results on the Toaster+Original dataset.

	Feature set used in training			
	Accuracy ± std on Hudson+Original test set		Accuracy ± std on Hudson test set	
Ratio of Hudson images in the test set	Detection features + Original features	Original features	Detection features + Original features	Original features
0	—	0.82 ± 0.02	—	—
0.1	0.79 ± 0.03	0.81 ± 0.02	0.70 ± 0.04	0.74 ± 0.08
0.2	0.79 ± 0.02	0.79 ± 0.04	0.72 ± 0.06	0.78 ± 0.03
0.3	0.80 ± 0.03	0.80 ± 0.01	0.75 ± 0.06	0.74 ± 0.08
0.4	0.80 ± 0.03	0.77 ± 0.03	0.76 ± 0.05	0.72 ± 0.05
0.5	0.77 ± 0.03	0.80 ± 0.02	0.75 ± 0.06	0.76 ± 0.03
0.6	0.79 ± 0.02	0.76 ± 0.03	0.79 ± 0.02	0.73 ± 0.04
0.7	0.75 ± 0.02	0.77 ± 0.03	0.72 ± 0.02	0.74 ± 0.04
0.8	0.76 ± 0.01	0.75 ± 0.02	0.75 ± 0.01	0.74 ± 0.02
0.9	0.74 ± 0.04	0.75 ± 0.02	0.73 ± 0.04	0.75 ± 0.02
1	0.75 ± 0.02	0.75 ± 0.02	0.75 ± 0.02	0.75 ± 0.02
Mean accuracy	**0.77 ± 0.03**	**0.77 ± 0.02**	**0.74 ± 0.03**	**0.74 ± 0.04**

Table A.14: Detection features results on the Hudson+Original dataset.

	Feature set used in training			
	Accuracy ± std on Toaster+Hudson+Original test set		Accuracy ± std on Toaster+Hudson test set	
Ratio of Toaster+Hudson images in the test set	**Detection features + Original features**	**Original features**	**Detection features + Original features**	**Original features**
0	—	0.82 ± 0.02	—	—
0.1	0.80 ± 0.02	0.78 ± 0.03	0.58 ± 0.10	0.60 ± 0.09
0.2	0.78 ± 0.04	0.78 ± 0.01	0.67 ± 0.06	0.64 ± 0.06
0.3	0.76 ± 0.02	0.76 ± 0.02	0.65 ± 0.02	0.60 ± 0.06
0.4	0.73 ± 0.02	0.75 ± 0.02	0.63 ± 0.04	0.66 ± 0.05
0.5	0.74 ± 0.02	0.72 ± 0.02	0.66 ± 0.02	0.63 ± 0.04
0.6	0.71 ± 0.02	0.72 ± 0.02	0.65 ± 0.04	0.65 ± 0.03
0.7	0.68 ± 0.03	0.69 ± 0.02	0.64 ± 0.02	0.64 ± 0.02
0.8	0.67 ± 0.04	0.68 ± 0.02	0.64 ± 0.04	0.64 ± 0.03
0.9	0.66 ± 0.02	0.66 ± 0.02	0.65 ± 0.02	0.65 ± 0.03
1	0.62 ± 0.04	0.65 ± 0.02	0.62 ± 0.04	0.65 ± 0.02
Mean accuracy	**0.71 ± 0.03**	**0.72 ± 0.02**	**0.63 ± 0.03**	**0.63 ± 0.04**

Table A.15: Detection features results on the Hudson+Toaster+Original dataset.

	Feature set used in training			
	Accuracy ± std on Edited+Original test set		Accuracy ± std on Edited test set	
Ratio of Edited images in the test set	**Filter features + Original features**	**Original features**	**Filter features + Original features**	**Original features**
0	—	0.82 ± 0.02	—	—
0.1	0.80 ± 0.03	0.80 ± 0.02	0.66 ± 0.07	0.58 ± 0.08
0.2	0.76 ± 0.02	0.77 ± 0.03	0.63 ± 0.07	0.60 ± 0.05
0.3	0.75 ± 0.02	0.76 ± 0.03	0.58 ± 0.06	0.62 ± 0.08
0.4	0.72 ± 0.02	0.73 ± 0.03	0.61 ± 0.03	0.62 ± 0.04
0.5	0.71 ± 0.02	0.71 ± 0.02	0.60 ± 0.05	0.60 ± 0.02
0.6	0.69 ± 0.03	0.69 ± 0.02	0.61 ± 0.04	0.61 ± 0.03
0.7	0.67 ± 0.02	0.68 ± 0.02	0.61 ± 0.03	0.62 ± 0.03
0.8	0.75 ± 0.02	0.63 ± 0.02	0.60 ± 0.03	0.60 ± 0.02
0.9	0.72 ± 0.02	0.63 ± 0.02	0.61 ± 0.02	0.61 ± 0.03
1	0.73 ± 0.04	0.60 ± 0.03	0.61 ± 0.02	0.60 ± 0.03
Mean accuracy	**0.70 ± 0.02**	**0.70 ± 0.02**	**0.61 ± 0.04**	**0.61 ± 0.04**

Table A.16: Filter feature results on the Edited+Original dataset.

A.4.4. Ensemble of adversarial techniques

	Ensemble of adv. techniques			
	Accuracy ± std on Grayscale+Original test set		**Accuracy ± std on Grayscale test set**	
Ratio of Grayscale images in the test set	**train ratio = 0.5 & bins = 84**	**train ratio = 0 & bins = 256**	**train ratio = 0.5 & bins = 84**	**train ratio = 0 & bins = 256**
0	—	0.82 ± 0.02	—	—
0.1	0.80 ± 0.02	0.78 ± 0.02	0.88 ± 0.06	0.52 ± 0.04
0.2	0.81 ± 0.02	0.76 ± 0.03	0.85 ± 0.05	0.52 ± 0.03
0.3	0.81 ± 0.02	0.72 ± 0.03	0.87 ± 0.03	0.55 ± 0.04
0.4	0.84 ± 0.03	0.69 ± 0.01	0.90 ± 0.03	0.52 ± 0.04
0.5	0.83 ± 0.01	0.67 ± 0.01	0.89 ± 0.02	0.50 ± 0.01
0.6	0.84 ± 0.03	0.64 ± 0.02	0.88 ± 0.03	0.52 ± 0.02
0.7	0.87 ± 0.02	0.62 ± 0.03	0.89 ± 0.01	0.55 ± 0.03
0.8	0.88 ± 0.02	0.58 ± 0.02	0.90 ± 0.02	0.52 ± 0.02
0.9	0.87 ± 0.01	0.54 ± 0.03	0.88 ± 0.02	0.52 ± 0.03
1	0.87 ± 0.02	0.53 ± 0.05	0.87 ± 0.02	0.53 ± 0.03
Mean accuracy	**0.84 ± 0.02**	**0.65 ± 0.02**	**0.88 ± 0.03**	**0.52 ± 0.03**

Table A.17: Ensemble of adv. techniques results on the Grayscale+Original dataset.

	Ensemble of adv. techniques			
	Accuracy ± std on Toaster+Original test set		**Accuracy ± std on Toaster test set**	
Ratio of Toaster images in the test set	**train ratio = 0.5 & bins = 84**	**train ratio = 0 & bins = 256**	**train ratio = 0.5 & bins = 84**	**train ratio = 0 & bins = 256**
0	—	0.82 ± 0.02	—	—
0.1	0.79 ± 0.02	0.78 ± 0.02	0.78 ± 0.07	0.58 ± 0.11
0.2	0.79 ± 0.04	0.77 ± 0.02	0.75 ± 0.08	0.58 ± 0.05
0.3	0.79 ± 0.02	0.74 ± 0.04	0.76 ± 0.05	0.59 ± 0.07
0.4	0.79 ± 0.03	0.71 ± 0.02	0.78 ± 0.04	0.55 ± 0.04
0.5	0.78 ± 0.04	0.67 ± 0.03	0.76 ± 0.05	0.52 ± 0.05
0.6	0.79 ± 0.02	0.65 ± 0.02	0.79 ± 0.01	0.53 ± 0.03
0.7	0.79 ± 0.03	0.63 ± 0.04	0.79 ± 0.03	0.55 ± 0.06
0.8	0.77 ± 0.03	0.59 ± 0.03	0.75 ± 0.04	0.54 ± 0.04
0.9	0.78 ± 0.02	0.59 ± 0.05	0.78 ± 0.02	0.57 ± 0.05
1	0.78 ± 0.02	0.53 ± 0.02	0.78 ± 0.02	0.53 ± 0.02
Mean accuracy	**0.78 ± 0.03**	**0.66 ± 0.02**	**0.77 ± 0.04**	**0.55 ± 0.05**

Table A.18: Ensemble of adv. techniques results on the Toaster+Original dataset.

	Ensemble of adv. techniques			
	Accuracy ± std on Hudson+Original test set		**Accuracy ± std on Hudson test set**	
Ratio of Hudson images in the test set	**train ratio = 0.5 & bins = 84**	**train ratio = 0 & bins = 256**	**train ratio = 0.5 & bins = 84**	**train ratio = 0 & bins = 256**
0	—	0.82 ± 0.02	—	—
0.1	0.81 ± 0.03	0.81 ± 0.02	0.82 ± 0.06	0.74 ± 0.08
0.2	0.80 ± 0.02	0.79 ± 0.04	0.79 ± 0.06	0.78 ± 0.03
0.3	0.81 ± 0.01	0.80 ± 0.01	0.80 ± 0.03	0.74 ± 0.08
0.4	0.79 ± 0.02	0.77 ± 0.03	0.77 ± 0.03	0.72 ± 0.05
0.5	0.79 ± 0.02	0.80 ± 0.02	0.78 ± 0.03	0.76 ± 0.03
0.6	0.79 ± 0.04	0.76 ± 0.03	0.77 ± 0.04	0.73 ± 0.04
0.7	0.78 ± 0.02	0.77 ± 0.03	0.77 ± 0.02	0.74 ± 0.04
0.8	0.80 ± 0.02	0.75 ± 0.02	0.80 ± 0.03	0.74 ± 0.02
0.9	0.80 ± 0.02	0.75 ± 0.02	0.80 ± 0.02	0.75 ± 0.02
1	0.80 ± 0.02	0.75 ± 0.02	0.80 ± 0.02	0.75 ± 0.02
Mean accuracy	**0.80 ± 0.02**	**0.77 ± 0.02**	**0.79 ± 0.03**	**0.74 ± 0.04**

Table A.19: Ensemble of adv. techniques results on the Hudson+Original dataset.

	Ensemble of adv. techniques			
	Accuracy ± std on Edited+Original test set		**Accuracy ± std on Edited test set**	
Ratio of Edited images in the test set	**train ratio = 0.5 & bins = 84**	**train ratio = 0 & bins = 256**	**train ratio = 0.5 & bins = 84**	**train ratio = 0 & bins = 256**
0	—	0.82 ± 0.02	—	—
0.1	0.80 ± 0.01	0.80 ± 0.02	0.74 ± 0.06	0.58 ± 0.08
0.2	0.81 ± 0.03	0.77 ± 0.03	0.77 ± 0.06	0.60 ± 0.05
0.3	0.79 ± 0.02	0.76 ± 0.03	0.76 ± 0.05	0.62 ± 0.08
0.4	0.77 ± 0.01	0.73 ± 0.03	0.74 ± 0.03	0.62 ± 0.04
0.5	0.78 ± 0.05	0.71 ± 0.02	0.76 ± 0.07	0.60 ± 0.02
0.6	0.76 ± 0.02	0.69 ± 0.02	0.75 ± 0.04	0.61 ± 0.03
0.7	0.77 ± 0.02	0.68 ± 0.02	0.75 ± 0.03	0.62 ± 0.03
0.8	0.77 ± 0.02	0.63 ± 0.02	0.77 ± 0.02	0.60 ± 0.02
0.9	0.77 ± 0.02	0.63 ± 0.02	0.77 ± 0.02	0.61 ± 0.03
1	0.77 ± 0.02	0.60 ± 0.03	0.77 ± 0.02	0.61 ± 0.03
Mean accuracy	**0.78 ± 0.02**	**0.70 ± 0.02**	**0.76 ± 0.04**	**0.61 ± 0.04**

Table A.20: Ensemble of adv. techniques results on the Edited+Original dataset.

A.5. Human perception study

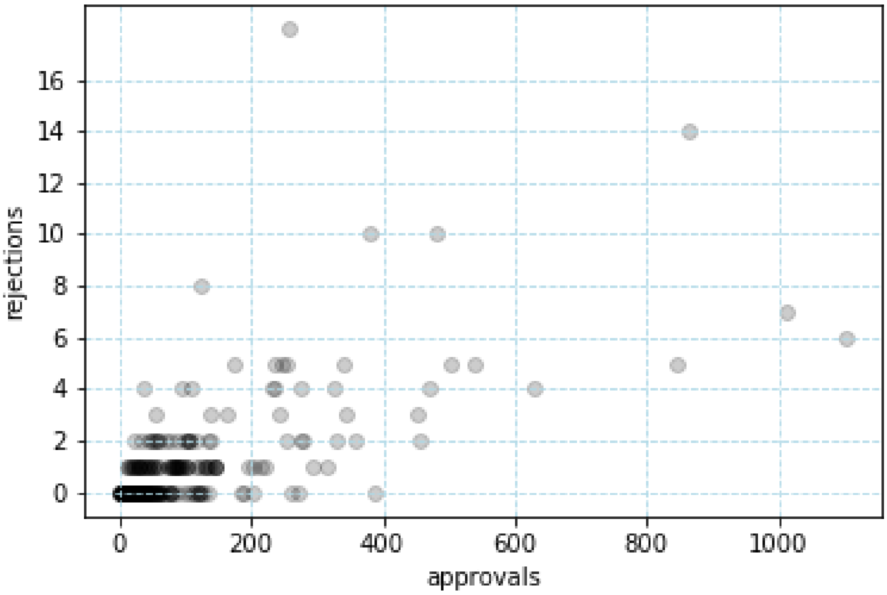


Figure A.6: Participant amount of rejections and approvals in Prolific.

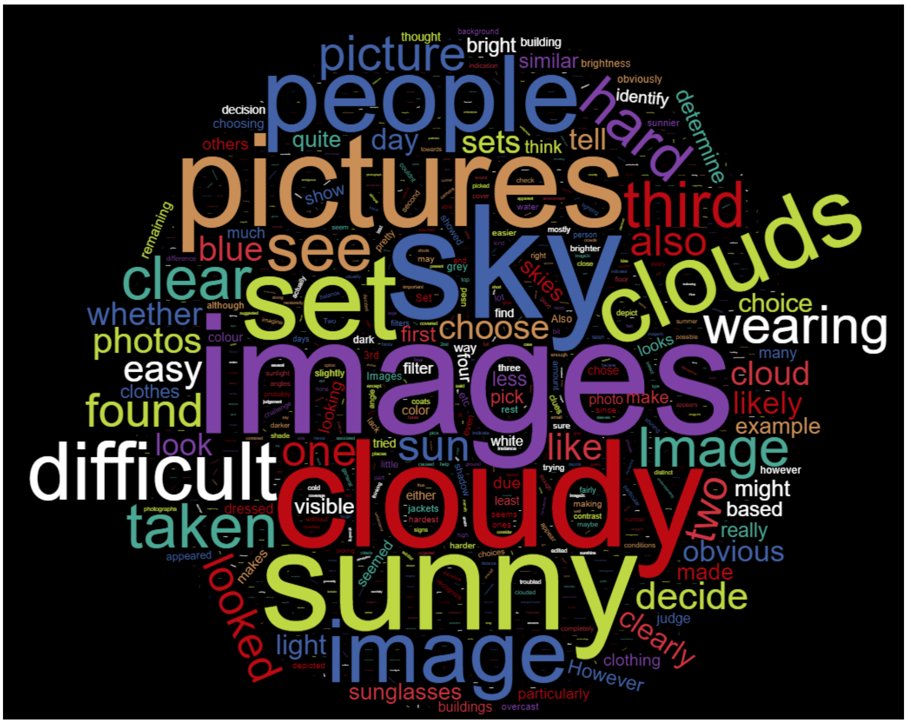


Figure A.7: Wordcloud of the free-text responses corpus.