Predicting Human Behavior using Visual Media

by

Aditya Khosla

B.S., California Institute of Technology (2009)
M.S., Stanford University (2011)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2017

Signature redacted

Author .....
Department of Electrical Engineering and Computer Science
September 19, 2016

Signature redacted

Certified by .....
Antonio Torralba
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Signature redacted

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Theses

Predicting Human Behavior using Visual Media

by

Aditya Khosla

Submitted to the Department of Electrical Engineering and Computer Science
on September 19, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The ability to predict human behavior has applications in many domains ranging from advertising to education to medicine. In this thesis, I focus on the use of visual media such as images and videos to predict human behavior. Can we predict what images people remember or forget? Can we predict the type of images people will like? Can we use a photograph of someone to determine their state of mind? These are some of the questions I tackle in this thesis. Through my work, I demonstrate: (1) It is possible to predict with near human-level correlation, the probability with which people will remember images, (2) it is possible to predictably modify the extent to which a face photograph is remembered, (3) it is possible to predict, with a high correlation, the number of views an image will receive even before it is uploaded, (4) it is possible to accurately identify the gaze of people in images, both from the perspective of a device, and third-person. Further, I develop techniques to visualize and understand machine learning algorithms that could help humans better understand themselves through the analysis of algorithms capable of predicting behavior. Overall, I demonstrate that visual media is a rich resource for the prediction of human behavior.

Thesis Supervisor: Antonio Torralba
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

Being at MIT was a fantastic experience. I could not have asked for more. I would like to express my gratitude to everyone who supported me in one way or another through the wonderful journey of obtaining my PhD.

First and foremost, I would like to thank my advisor, Antonio Torralba. Antonio has been the best advisor one could ask for, largely by acting more like a collaborator than an advisor. He has never made me feel that I am working to please him[1], instead, he is there as a sounding board when I need him. Meetings with him have been amazing – I have always come out feeling in awe of his brilliance, and smarter at the same time having learned something from him. His boundless passion, ingenious creativity and relentless optimism have pushed me to achieve more than I ever thought possible.

I would also like to thank my *co-advisor*[2], Aude Oliva. She has always been a great source of inspiration and motivation for me. I have always come out of meetings with her, feeling like I could conquer the world. Her amazing foresight to *predict the future* has led to ideas with brilliant outcomes. Despite her ridiculously busy schedule, she has always taken time out to learn about me, and my life outside of research, which I have always appreciated deeply.

I do not have the words to express the gratitude I feel towards Antonio and Aude. I just hope that some day I can be even half as good a mentor to someone as they have been to me.

I also want to thank my thesis committee members, Wojciech Matusik and Bill Freeman for their excellent feedback and thought-provoking questions. Further, I sincerely thank my academic advisor, Joel Emer, who I have always looked forward to meeting at the beginning of each semester. I also wanted to express my gratitude towards my mentors at various internships, Piotr Dollar, Raffay Hamid and Larry Zitnick, who allowed to explore novel directions of research. Additionally, I want to thank Andrew Beck and Michal Kosinski for being excellent collaborators and helping

[1] He might feel differently about this.

[2] Unofficial, but only on paper.

me navigate the faculty application process.

I took my baby steps into the world of computer vision, and scientific research, while pursuing my Masters at Stanford. I want to take this opportunity to thank everyone who contributed to those formative years. First and foremost, I wanted to thank my advisor, Fei-Fei Li. Among other things, she demonstrated how to keep in sight the big picture while tackling the low level details. She taught me the importance of giving a great presentation and has been a constant source of inspiration. I would also like to thank Navneet Dalal for teaching me HOG in great detail and being a mentor to me as I navigated my PhD, Honglak Lee for teaching me to always strive for perfection, and Bangpeng Yao for helping me achieve my dream of pursuing a PhD at MIT. I was also fortunate to work with Daphne Koller and Andrew Ng who continue to remain some of the smartest people I have ever interacted with.

No man is an island. My research has been fueled by working with the following amazing collaborators: Byoungkwon An, Wilma Bainbridge, Greg Beams, Michael Bernstein, Suchendra Bhandarkar, Amartya Biswas, Yu Cao, Hsu-Kuang Chiu, Radoslaw Cichy, Gary Cottrell, Atish Das Sarma, Akshat Dave, Jia Deng, George Djorgovski, Ciro Donalek, Rachit Dubey, Alyosha Efros, Bernard Ghanem, Will Grathwohl, Leonidas Guibas, Tracey Ho, Junling Hu, Zhiheng Huang, Phillip Isola, Nityananda Jayadevaprakash, Xiaoye Jiang, Harini Kannan, Andrej Karpathy, Petr Kellnhofer, Mingyu Kim, Kyle Krafka, Jonathan Krause, Tejas Kulkarni, Agata Lapedriza, Eric Lau, Sebastian Leon, Anying Li, Joseph Lim, Andy Lin, Chih-Jen Lin, Cliff Lin, Dion Low, Sean Ma, Ashish Mahabal, Tomasz Maliesiewicz, Mikael Mengistu, Juhan Nam, Jiquan Ngiam, Dimitrios Pantazis, Joshua Peterson, Hamed Pirsiavash, Xavier Puig, Akhil Raju, Adria Recasens, Olga Russakovsky, Tossaporn Sangja, Sanjeev Satheesh, Shuran Song, Hao Su, Neel Sundaresan, Xiaoou Tang, Carl Vondrick, Svetlana Vyetrenko, Dayong Wang, Jiajun Wu, Zhirong Wu, Jianxiong Xiao, Ming-Hsuan Yang, Fisher Yu, Linguang Zhang, Bolei Zhou and Tinghui Zhou. They have been a constant source of motivation and their intellectual contributions have made my achievement possible. Apart from research, I have also had the pleasure of working with various people on organizing workshops and challenges

at a number of top conferences: Serge Belongie, Alex Berg, John Canny, Polo Chau, Jia Deng, Ryan Farrell, James Hays, Derek Hoiem, Biye Jiang, Jonathan Krause, Agata Lapedriza, Li-Jia Li, Fei-Fei Li, Wei Liu, Subhransu Maji, Aude Oliva, Olga Russakovsky, Silvio Savarese, Antonio Torralba, Jianxiong Xiao and Bolei Zhou. Together, we have been able to impact the research community at large.

I also want to thank a number of my close friends from MIT who have spent countless hours with me discussing everything under the sun: George Chen, Phillip Isola, Tejas Kulkarni, Joseph Lim, Hossein Mobahi, Andrew Owens, Adria Recasens, Carl Vondrick and Bolei Zhou. I would also like to thank other friends in the Boston area and those around the MIT vision lab who have contributed significantly to my graduate experience: Yusuf Aytar, Katie Bouman, Abe Davis, Zoya Gavrilov, David Hayden, Seyed-Mahdi Khaligh-Razavi, Gunhee Kim, Michael Rubinstein, Palvi Raikar, Maria Rodriguez, Subramanium Sundaram, Santani Teng, Donglai Wei, Tianfan Xue and Jenny Yuen. It has been amazing to hang out with such extraordinary people.

I also want to thank the various CSAIL/MIT resources that were critical to my work: TIG for maintaining the machines we routinely abuse, Janet Fischer for putting up with me as I struggled with paperwork, Adam Conner-Simons for bringing millions of eye balls to my work, and Fern Keniston and Bryt Bradley for helping with logistics and reimbursements. I also want to thank Facebook for their generous fellowship in support of my research.

Last but in no way least, I want to thank my family: the unwavering belief of my parents and brother in my abilities drove me to achieve more than I ever thought possible. They have taught me never to falter and to keep pushing ahead no matter the challenges. Finally, I want to thank my best friend and life partner, my wife, Radhika Marathe. Without her, I might never have got in, or come to MIT[3] Without her, I would have never got through the PhD – from celebrating the acceptance of papers to bringing me supplies when I decided to *live* in the office, she has always been there for me. I dedicate this thesis to her.

[3]She came to MIT two years before me to pursue her PhD. This motivated me to do the same.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Visual media makes up the majority of internet traffic. People are constantly taking and uploading photos and videos of their daily lives. From this barrage of visual media, can we learn anything about human behavior? Can we predict what images people remember or forget? Can we predict the type of images people will like? Can we use a photograph of someone to determine their state of mind? These are some of the questions I tackle in this thesis. The ability to predict human behavior has applications in many domains ranging from advertising to education to medicine.

In this thesis, I focus on three specific topics with the potential to impact a variety of application domains: (1) predicting visual memory (Section 1.1), (2) predicting image popularity(Section 1.2), (3) predicting state of mind (Section 1.3) and (4) visualizing and understanding convolutional neural networks (Section 1.4). I describe each of these in detail below, and summarize the main contributions of my work.

1.1 Predicting Visual Memory

One hallmark of human cognition is its massive capacity for remembering lots of different images, many in great detail, and after only a single view. Interestingly, people tend to collectively remember and forget the same pictures and faces, a property known as *memorability* [78]. This suggests that despite different personal experiences, people naturally encode and discard the same types of information.

During my PhD, I developed the first models for understanding, predicting and modifying the images people remember. My research has furthered our understanding of the relationship of memorability to a variety of image attributes such as image emotions, saliency and popularity. Notably, the model achieves near-human level consistency at predicting image memory. It further identifies exactly which regions of an image are memorable, and which are forgettable allowing for immediate deployment in applications such as advertising, video summarization [77] and generating mnemonic aids. Combining these insights, I formulated and implemented an algorithm to modify faces to predictably change the extent to which they are remembered. I summarize the key findings below, and provide the details in Chapter 2.

Understanding memorability: By introducing a novel experimental method [78] for efficiently collecting human memory scores (at about one-tenth the cost of prior work), I collected LaMem, the first large-scale benchmark dataset containing 60,000 images with memorability scores from human observers (about 27$\times$ larger than the previous dataset). Among other observations, I discovered that the most memorable images tended to be more popular on social networks, and that images portraying negative emotions tend to be more memorable.

Predicting memorability: Using LaMem, I formulated and trained MemNet [78], a deep network that achieves near human level performance at predicting memorability. While deep networks have led to significant improvements in a variety of tasks, the secret to their performance remains hidden in their complex high-dimensional representation. As such, I have worked on a variety of approaches for visualizing [160, 173] and semantically understanding the internal representation of deep networks. By visualizing the learned representation of the layers of MemNet, we discovered the emergent representations, or diagnostic objects, that explain what makes an image memorable or forgettable. I then applied MemNet to overlapping image regions to produce memorability maps [80]. Using a simple technique based on non-photorealistic rendering we demonstrated that our deep memorability network correctly isolates the important components of visual memorability.

Modifying memorability: Imagine if it were possible to modify the extent to

which an image is remembered – this could have far-reaching applications in various domains ranging from advertising and gaming to education and social networking. For example, we could modify educational diagrams to make them easier to remember while preserving their critical content. In [75], I propose and implement the first algorithm of its kind to automatically modify the memorability of faces without affecting their identity. Despite the small modifications, my approach produced a significant impact on the memorability of the faces. This raises a natural question: if we were to make all visual content more memorable, would we just shift the baseline? My early research suggests that this is not the case – participants presented with all highly memorable stimuli tended to remember all of them. This implies that by making visual content more memorable, we might be able to increase human memory capacity.

1.2 Predicting Image Popularity

Hundreds of thousands of photographs are uploaded to the internet every minute through various social networking and photo sharing platforms. While some images get millions of views, others are completely ignored. Even from the same users, different photographs receive different number of views. This begs the question: What photographs do people like? Can we predict the number of views a photograph will receive even before it is uploaded? In [76], I develop an algorithm that can reliably predict the normalized view count of images with a rank correlation of 0.81 using both image content and social cues. My work is the first to show that image content alone can be used to predict popularity.

By investigating various image cues such as color, gradients, deep learning features and the set of objects present, I show key insights from our method that identify crucial aspects of the image that influence its popularity. On average, we observe that the greenish and bluish colors tend to have lower importance as compared to more reddish colors. Further, we find that objects such as *mini skirts* and *revolvers* tended to increase the popularity of images while *laptops* and *spatulas* tended to

decrease it. The detailed results are provided in Chapter 3.

1.3 Predicting State of Mind

Ultimately, I plan to build systems that automatically predict both short-term (e.g., what someone is thinking, feeling) and long-term (e.g., personality, political inclination, satisfaction with life) state of mind of each individual. Building such systems would allow us to better understand the underlying human cognitive processes through inspection of our automatic systems, and also allow artificially intelligent agents such as robots to have more meaningful interactions with people. My recent and ongoing research focuses on developing critical components for realizing this feat. For predicting short-term state of mind, I believe people's gaze reveal significant information about their state of mind. I have developed approaches to tackle this problem from two different perspectives: first from perspective of the device (known as eye tracking) and another from the perspective of another individual (known as gaze following). I summarize the key findings below, and provide the details in Chapter 4.

Eye tracking: From human-computer interaction techniques to medical diagnosis to psychological studies to computer vision, eye tracking has applications in many areas. Gaze is the externally-observable indicator of human visual attention, and initial attempts to record it date back to the late eighteenth century. Today, a variety of solutions exist (many of them commercial) but all suffer from one or more of the following: high cost, custom or invasive hardware or inaccuracy under real-world conditions. These factors prevent eye tracking from becoming a pervasive technology that should be available to anyone with a reasonable camera (e.g., a smartphone or a webcam). My research combines the generalization power of deep learning with big data gathered via crowdsourcing ($> 20\times$ larger than any existing dataset) to achieve unprecedented results in calibration-free eye tracking [86]. Using only the front-facing smartphone camera, my approach achieves an accuracy of under 2cm on average and can run in real-time (10 - 15fps) on mobile devices, significantly outperforming all existing approaches. Overall, the technology I have developed is a significant step

towards putting the power of eye tracking in everyone's palm.

Gaze following: Humans have the remarkable ability to follow the gaze of other people to identify what they are looking at. Following eye gaze, or gaze-following, is an important ability that allows us to understand what other people are thinking, the actions they are performing, and even predict what they might do next. Despite the importance of this topic, this problem has only been studied in limited scenarios within the computer vision community. In [129], I propose a deep neural network-based approach for gaze-following and a large-scale benchmark dataset for thorough evaluation. Given an image and the location of a head, my approach follows the gaze of the person and identifies the object being looked at. The quantitative evaluation shows that my approach produces reliable results matching human prediction, even when viewing only the back of the head.

1.4 Visualizing and Understanding Convolutional Neural Networks

The ability to predict human behavior using state-of-the-art machine learning algorithms is useful, but it begs the question of, can we learn more about ourselves by understanding the algorithms making the inferences? This motivates my work in visualizing and understanding convolutional neural networks, the current state-of-the-art approach in machine learning.

With the success of new computational architectures for visual processing, such as convolutional neural networks (CNN) and access to image databases with millions of labeled examples (e.g., ImageNet, Places), the state of the art in computer vision is advancing rapidly. One important factor for continued progress is to understand the representations that are learned by the inner layers of these deep architectures. Here we show that object detectors emerge from training CNNs to perform scene classification. As scenes are composed of objects, the CNN for scene classification automatically discovers meaningful objects detectors, representative of the learned

scene categories. With object detectors emerging as a result of learning to recognize scenes, our work demonstrates that the same network can perform both scene recognition and object localization in a single forward-pass, without ever having been explicitly taught the notion of objects.

Chapter 2

Predicting Visual Memory

One hallmark of human cognition is our massive capacity for remembering lots of different images [15, 84], many in great detail, and after only a single view. Interestingly, we also tend to remember and forget the same pictures and faces as each other [4, 63]. This suggests that despite different personal experiences, people naturally encode and discard the same types of information. For example, pictures with people, salient actions and events, or central objects are more memorable to all of us than natural landscapes. Images that are consistently forgotten seem to lack distinctiveness and a fine-grained representation in human memory [15, 84]. These results suggest that memorable and forgettable images have different intrinsic visual features, making some information easier to remember than others. Indeed, computer vision works [62, 80, 75, 33] have been able to reliably estimate the memorability ranks of novel pictures, or faces, accounting for half of the variance in human consistency. However, to date, experiments and models for predicting visual memorability have been limited to very small datasets and specific image domains.

Intuitively, the question of an artificial system successfully predicting human visual memory seems out of reach. Unlike visual classification, images that are memorable, or forgettable, do not even look alike: an elephant, a kitchen, an abstract painting, a face and a billboard can all share the same level of memorability, but no visual recognition algorithms would cluster these images together. What are the common visual features of memorable, or forgettable, images? How far we can we go in predicting

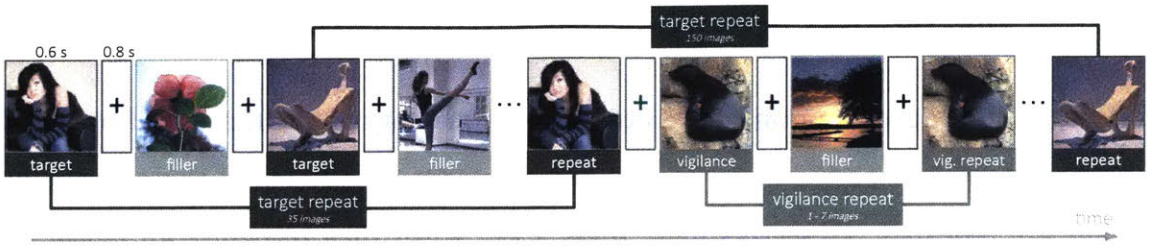
with high accuracy which images people will remember, or not?

In this work, we demonstrate that a deep network trained to represent the diversity of human visual experience can reach astonishing performance in predicting visual memorability, at a near-human level, and for a large variety of images. Combining the versatility of many benchmarks and a novel experimental method for efficiently collecting human memory scores (about one-tenth the cost of [63]), we introduce the *La Mem* dataset, containing 60,000 images with memorability scores from human observers (about 27 times larger than the previous dataset [63]).

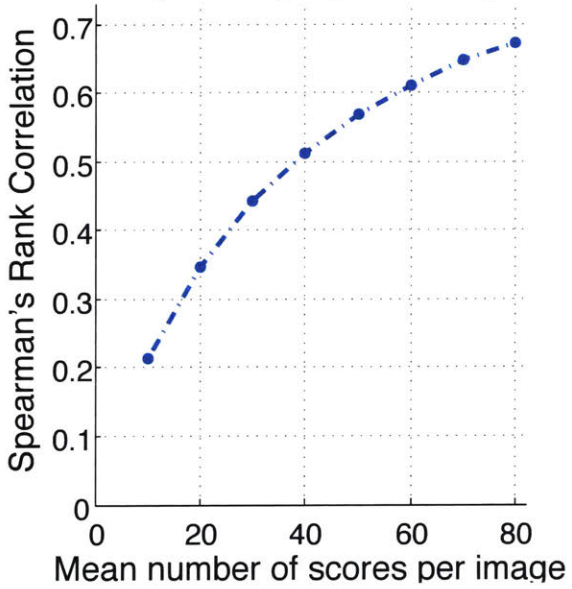
Figure 2-1: Sample images from *La Mem* arranged by their memorability score (decreasing from left to right). *La Mem* contains a very large variety of images ranging from object-centric to scene-centric images, and objects from unconventional viewpoints.

By fine-tuning Hybrid-CNN [174], a convolutional neural network (CNN) [92, 87] trained to classify more than a thousand categories of objects and scenes, we show that our model, MemNet, achieves a rank correlation of 0.64 on novel images, reaching near human consistency rank correlation (0.68) for memorability. By visualizing the learned representation of the layers of MemNet, we discover the emergent representations, or diagnostic objects, that explain what makes an image memorable or forgettable. We then apply MemNet to overlapping image regions to produce a memorability map. We propose a simple technique based on non-photorealistic rendering to evaluate these memorability maps. We find a causal effect of this manipulation on human memory performance, demonstrating that our deep memorability network has been able to isolate the correct components of visual memorability.

Altogether, this work stands as the first near-human performance benchmark of



(a) The efficient visual memory game. Each image is shown for 600ms, separated by a blank fixation of 800ms. The worker can press a key anytime during this 1.4s.



(b) Human consistency.

Figure 2-2: Illustration of the efficient visual memory game (left), and the resulting human consistency averaged over 25 random splits (right) obtained using the proposed method on the *La Mem* dataset.

human visual memory, offering an understanding and a concrete algorithm for predicting the memorability of an image and its regions. We envision that many applications can be developed out of deep memorability features, akin to the recent astonishing impact that deep networks have had on object and scene classification tasks. Our work shows that predicting human cognitive abilities is within reach for the field of computer vision.

2.1 *La Mem*: Large-scale Memorability Dataset

Here, we introduce an optimized protocol of the memory game introduced by [63] to collect human memory scores. In this game, images are presented successively, and some are repeated. Observers must press a key when they recognize an image seen before. This allows us to collect ground truth scores on how memorable images are. The basic idea of our novel procedure is to allow the second occurrence of an image to occur at variable time intervals. This procedure is based on the finding that the memorability ranks of images are time-independent [63]. We propose an algorithm to account for this varied time interval allowing us to obtain high consistency with the existing benchmark [63]. Furthermore, using this new experimental setting, we build a novel massive memorability dataset, with scores on $60,000$ images (~ 27 times the previous largest benchmark), while keeping a low cost. Our dataset contains significantly more variety in the types of images (see Figure 2-1), while still maintaining a high human consistency on memorability.

First, in Section 2.1.1, we briefly describe the sources of images used for building the dataset to demonstrate its variety as compared to existing datasets. Then, in Section 2.1.2, we describe the efficient visual memory game for obtaining large-scale memorability annotations. Last, in Section 2.1.3, we provide experimental validation of the proposed method.

2.1.1 Collecting images

To create a varied dataset, we sampled images from a number of existing datasets such as MIR Flickr [60], AVA dataset [109], affective images dataset [102] (consisting of Art and Abstract datasets), image saliency datasets (MIT1003 [68] and NUSEF [126]), SUN [164], image popularity dataset [76], Abnormal Objects dataset [135] and aPascal dataset [39]. Thus, our dataset contains scene-centric images, object-centric images and other types such as images of art, images evoking certain emotions, and other user-generated images such as 'selfies'. We explore the correlation between a variety of these attributes and memorability in Section 2.2.2.

2.1.2 Efficient Visual Memory Game

Our experimental procedure consists of showing target repeats (the second occurrence of an image) at variable time intervals. For example, some targets may be repeated after just 30 images, while others are repeated after 100. As shown in [63], memorability scores change predictably as a function of the time interval between repeats, while memorability ranks are largely conserved i.e., if the time between the showing of a target and its repeat is increased, the memorability scores of all images decrease by a similar amount, thereby preserving the rank ordering. In our method, we use this information to propose a method based on coordinate descent that explicitly accounts for the difference in interval lengths. This allowed us to collect ground truth memorability scores for a large number of images (here $60,000$), in a short amount of time, and at a very reasonable cost.

Model: We first describe one possible interpretation of the memorability score computation proposed by [63], and extend that to our setting. Let us define $m^{(i)}$ as the memorability of image i. For image i, we have some $n^{(i)}$ observations given by $x_j^{(i)} \in \{0,1\}$ and $t_j^{(i)}$ where $x_j = 1$ implies that the image repeat was correctly detected when it was shown after time t_j. The memorability score proposed by [63] is the average hit rate per image, which can also be seen as the value that minimizes the ℓ_2 error $\sum_j ||x_j^{(i)} - m^{(i)}||_2^2 \implies m^{(i)} = \frac{1}{n^i}\sum_j x_j^{(i)}$. In this case, the different times of repeat presentation, t_j, are not taken into account explicitly as all repeats are shown at about the same delay to all participants. Next, we modify the above model to suit our new scenario with variable delays.

Memorability follows a log-linear relationship with time delay between images [63]. Let us assume that the memorability of image i is $m_T^{(i)}$ when the time interval between repeated displays is T. Thus, we can write the memorability of image i as $m_T^{(i)} = \alpha \log(T) + c^{(i)}$, where $c^{(i)}$ is the *base memorability* for the given image and α is the decay factor of memorability over time. Similarly, for some other time t, we can write the memorability of the same image as $m_t^{(i)} = \alpha \log(t) + c^{(i)}$. Thus, we obtain the

relationship:

$$m_t^{(i)} - m_T^{(i)} = \alpha \log(t) - \alpha \log(T) \tag{2.1}$$

$$\implies m_t^{(i)} = m_T^{(i)} + \alpha \log\left(\frac{t^{(i)}}{T}\right) \tag{2.2}$$

As before, we have some n observations for image i given by $x_j^{(i)} \in \{0, 1\}$ and $t_j^{(i)}$ where $x_j = 1$ implies that the image repeat was correctly detected when it was shown after time t_j. For N images, we can now write the overall ℓ_2 error, E, as:

$$E(\alpha, m_T^{(i)}) = \sum_{i=1}^{N} \sum_{j=1}^{n^{(i)}} ||x_j^{(i)} - m_{t_j}^{(i)}||_2^2 \tag{2.3}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{n^{(i)}} \left|\left| x_j^{(i)} - \left[m_T^{(i)} + \alpha \log\left(\frac{t_j^{(i)}}{T}\right)\right]\right|\right|_2^2 \tag{2.4}$$

Note that we write the combined error (as compared to individual errors per image) as the decay factor α is shared across all images. Our goal is to find $m_T^{(i)}$ and α that minimize E. By adjusting the value of T, we can adjust the time delay at which we want to find the memorability score. Also, by finding all scores at a fixed delay T, the scores for all images become comparable, as is the case in the model proposed by [63].

Optimization: We observe that we can find the global minima of E with respect to $m_T^{(i)}$ if we fix the value of α, and similarly, we can find α if we fix the value of $m_T^{(i)}$. Thus, we can minimize E by iteratively updating α, followed by $m_T^{(i)}$ and so on. By differentiating E with respect to each of the variables, and setting it to 0, we can find the update equations:

$$\alpha \leftarrow \frac{\sum_{i=1}^{N} \frac{1}{n^{(i)}} \sum_{j=1}^{n^{(i)}} \log(t_j^{(i)}/T) \left[x_j^{(i)} - m_T^{(i)}\right]}{\sum_{i=1}^{N} \frac{1}{n^{(i)}} \sum_{j=1}^{n^{(i)}} \left[\log(t_j^{(i)}/T)\right]^2} \tag{2.5}$$

and

$$m_T^{(i)} \leftarrow \frac{1}{n^{(i)}} \sum_{j=1}^{n^{(i)}} \left[x_j^{(i)} - \alpha \log\left(\frac{t_j^{(i)}}{T}\right)\right] \tag{2.6}$$

As the update equations find the global optima of E when keeping the other fixed, we ensure that the error is always decreasing, guaranteeing convergence. In practice, we initialize $m_T^{(i)}$ to the mean hit rate ignoring time delay, and find that approximately 10 iterations are enough for convergence. Note that our model has no hyperparameters.

2.1.3 Dataset experiments

In this section, we describe in detail the experimental setup of our efficient visual memory game, and conduct several experiments comparing the results of the proposed methodology with [63]. Further, we demonstrate that the proposed model can increase human consistency by accounting for variable time delays between repeats and it results in a consistent decay factor, α, across splits.

Experimental setup: The efficient visual memory game is summarized in Figure 2-2a. We conducted memorability experiments using Amazon's Mechanical Turk (AMT) on the 60,000 target images obtained by sampling the various datasets mentioned in Section 2.1.1. Each task lasted about 4.5 minutes consisting of a total of 186 images divided into 66 targets, 30 fillers, and 12 vigilance repeats. Targets were repeated after at least 35 images, and at most 150 images. Vigilance repeats were shown within 7 images from the first showing. The vigilance repeats ensured that workers were paying attention leading to a higher quality of results. Workers who failed more than 25% of the vigilance repeats were blocked, and all their results discarded. Further, we used a qualification test to ensure the workers understood the task well. We obtained 80 scores per image on average, resulting in a total of about 5 million data points. Similar to [63], we use rank correlation to measure consistency.

Comparison with [63]: Before describing the results on our new dataset, we first compare the performance of our method to the one proposed by Isola et al [63] on the SUN memorability dataset to ensure that our modifications are valid. We randomly selected 500 images from their dataset, and collected 80 scores per image. After applying our algorithm to *correct* the memorability scores, we obtained a within-dataset human rank correlation of 0.77 (averaged over 25 random splits), as compared

to 0.75 using the data provided by [63]. Further, we obtain a rank correlation of 0.76 when comparing the independently obtained scores from the two methods. This shows that our method is well suited for collecting memorability scores.

Results on *La Mem*: Figure 2-2b shows the human consistency as the number of number of human annotations per image increases. At 80 scores per image, we obtain a human rank correlation of 0.67 (averaged over 25 random splits) if we simply take the average of the correct responses ignoring the difference in time delays (i.e., same formula as [63]) which increases to 0.68 after applying our method. Note that the impact of using our method is small in this case as the range of average delays of each image is relatively small, ranging only from 62 to 101 intervening images. While our method can rectify the errors caused by variable delays, the error here is rather insignificant.

To further verify our algorithm, we created *adversarial* splits of the data where the responses for each image are divided based on the delays i.e., all the responses when delays are low go into one split, and all the responses when delays are high go into the other split. We randomly assign the low and high delay split of each image to different overall splits. Using the method of [63] (i.e., simple averaging) in this case significantly reduces the human rank correlation to 0.61, which can be restored to 0.67 using our method. This demonstrates the importance of applying our method when the interval distribution is more diverse.

Interestingly, we find that the decay factor, α, found by our method is largely consistent across various splits of data, having a standard deviation of less than 1% from the mean. This further verifies the finding made by [63] that memorability decays consistently over time, and our method provides a robust way to estimate this decay factor.

Overall, the high human consistency obtained on *La Mem* despite the large variety of images strengthens the importance of the concept of memorability and shows that it is a universal and intrinsic property of each image.

2.2 Understanding Memorability

As described in Section 2.1.1, *La Mem* is composed of a variety of other datasets that contain additional annotation such as aesthetics, popularity, image emotions, objects, and so on. In this section, we explore the relationship of some of these image attributes to memorability.

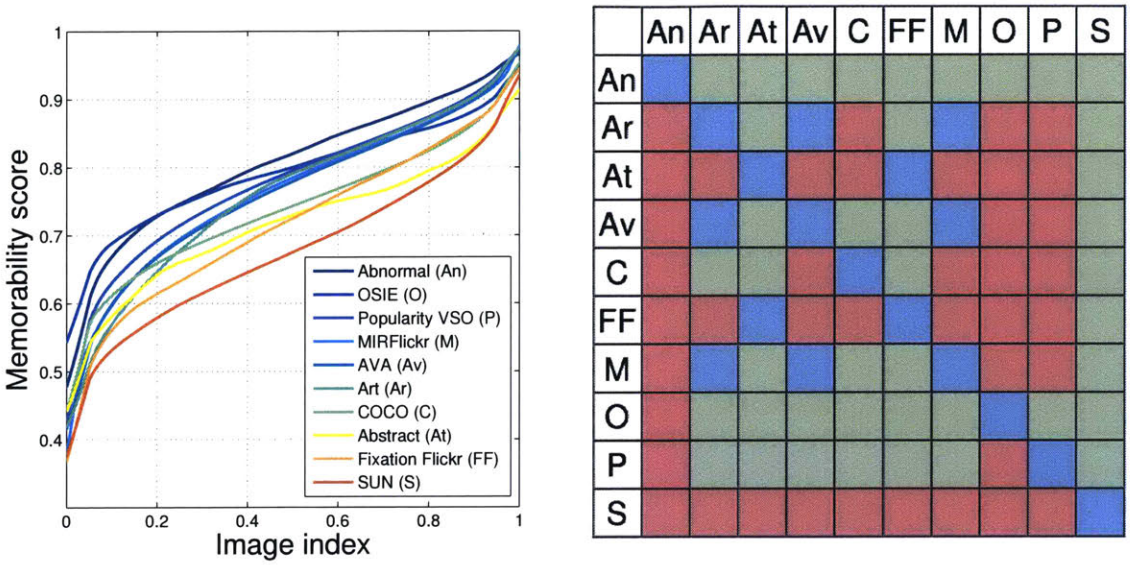
2.2.1 Differences across datasets

In Figure 2-3a, we plot the memorability scores of some of the datasets contained in *La Mem*. We find that the distribution of memorability scores for the different datasets tends to look rather different. While images from the Abnormal Objects dataset [135] and image popularity dataset [76] tend to be extremely memorable, those from the SUN dataset [164] tend to be rather forgettable. In Figure 2-3b we evaluate whether these perceived differences are statistically significant using a one-sided t-test. We find that most of the differences are significant at the 5% level.

2.2.2 Image attributes

In this section, we explore how some of the image attributes, such as popularity, saliency, emotions and aesthetics, affect the memorability of images and vice-versa. We would like to highlight that the significant diversity of *La Mem* allows for this exploration at a large-scale.

Popularity: In [76], popularity is defined as the log-normalized view-count of an image on Flickr. Using the 5000 images from this dataset contained in *La Mem*, in Figure 2-4a, we plot the popularity scores of the images divided into quartiles based on their ground-truth memorability scores. We find that the popularity scores of the most memorable images (1st quartile) are statistically higher than those of the other quartiles. The same holds if we plot the memorability scores of the 25% most and least popular images as shown in Figure 2-5b. On the other hand, when the memorability scores are low-medium, there is little difference in the popularity scores. This could be an insightful finding for people attempting to design images



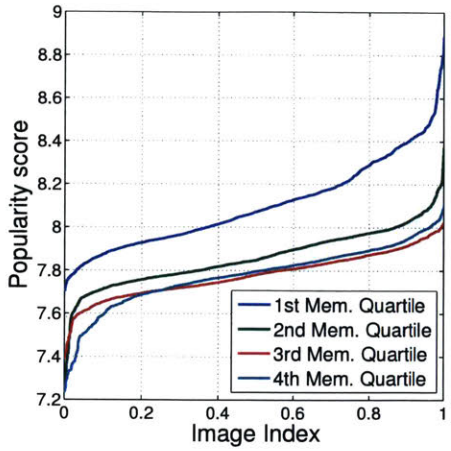
(a) Memorability scores

(b) Statistical differences in mean

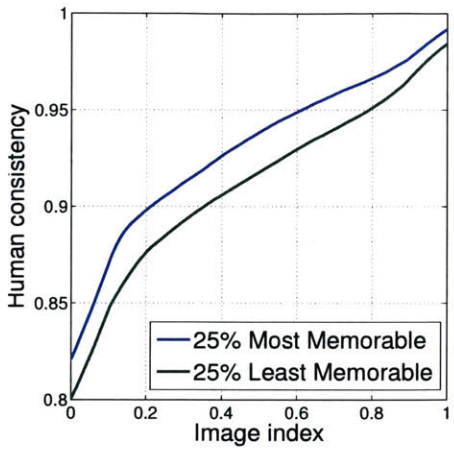
Figure 2-3: (a) Memorability scores of images from different datasets. For each dataset, the memorability scores of the images are independently sorted from low to high i.e., image index 0 to 1. Note that the image index ranges from 0 to 1 (instead of 1 to N) as each dataset has a different number of images. (b) Matrix indicating if the differences in the mean memorability scores of different datasets are statistically significant at the 5% level of significance. Blue indicates no difference, red indicates *col*>*row*, while green indicates *row*>*col*.

that become popular. Note that even though these images are popular on Flickr, we do not expect the AMT workers to have seen them before in general as the most popular image had fewer than 100k views. Furthermore, if they had seen the image before, they would have generated a false alarm on the first presentation of the image resulting in a lower memorability score for the image.

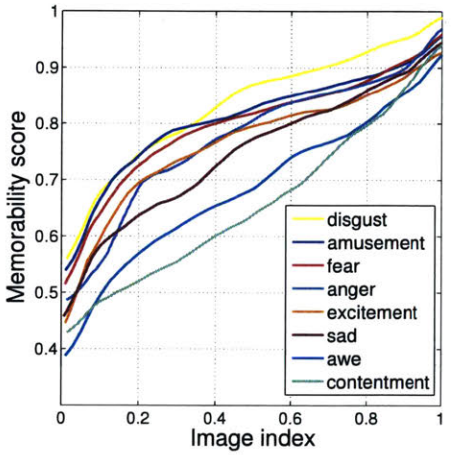
Saliency: Using images from the Fixation Flickr [68] dataset, we explore the relationship between human fixations and memorability. As shown in Figure 2-4b, we find that images that are more memorable tend to have more consistent human fixations. In fact, we find that human fixation consistency and memorability have a reasonable rank correlation of 0.24. A high human consistency on saliency often occurs when humans have one or a few specific objects to fixate on, which would tend to imply that the image contains more close-ups or larger objects. Essentially, this suggests that when humans have a specific point of focus in an image, they are



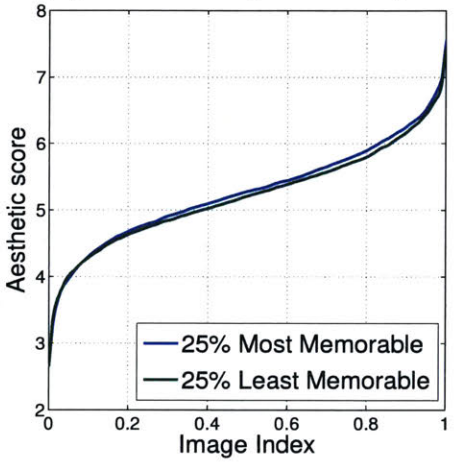
(a) Memorability vs popularity



(b) Memorability vs saliency



(c) Memorability vs emotions



(d) Memorability vs aesthetics

Figure 2-4: Plots showing the relationship of memorability and various image attributes. For each curve, the images are sorted independently using ground-truth memorability scores. As each curve may contain a different number of images, the image index above has been normalized to be from 0 to 1.

better able to remember it and vice versa. These findings are similar to those of [104] and [17].

Emotions: In Figure 2-4c, we plot the memorability scores of images portraying various emotions from the affective images dataset [102]. We find that images that evoke *disgust* are statistically more memorable than images showing most other emotions, except for *amusement*. Further, images portraying emotions like *awe* and *contentment* tend to be the least memorable. This is similar to the findings in [62] where they show attributes like 'peaceful' are strongly negatively correlated with

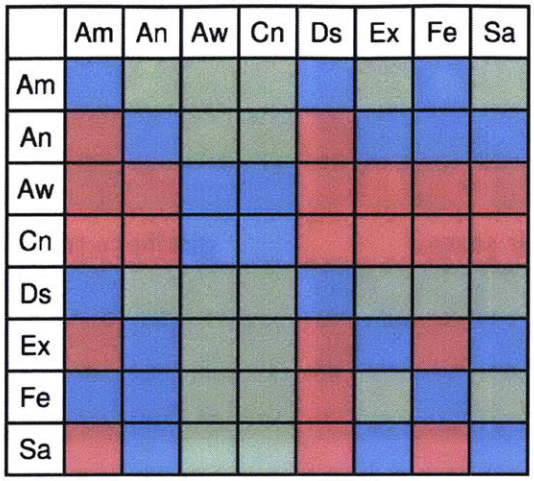
Test set:		Train set: SUN Memorability				
		fc6	fc7	fc8	fine-tune	HOG2x2
SUN Mem	no FA	0.57	0.60	0.58	0.51	0.45
	with FA	0.61	**0.63**	0.62	0.53	0.48
La Mem	no FA	0.46	0.48	0.46	0.43	0.35
	with FA	0.52	0.54	**0.55**	0.47	0.43
Test set:		Train set: ***La Mem***				
		fc6	fc7	fc8	MemNet	HOG2x2
SUN Mem	no FA	0.56	0.59	0.57	0.59	0.47
	with FA	0.57	0.59	0.58	**0.61**	0.48
La Mem	no FA	0.54	0.55	0.53	0.57	0.40
	with FA	0.61	0.61	0.60	**0.64**	0.47

Table 2.1: Rank correlation of training and testing on both *La Mem* and SUN Memorability datasets. The reported performance is averaged over various train/test splits of the data. For cross-dataset evaluation, we use the full dataset for training and evaluate on the same test splits to ensure results are comparable. `fc6`, `fc7` and `fc8` refer to the different layers of the Hybrid-CNN [174], and 'FA' refers to false alarms. Please refer to Section 2.3.1 for additional details.

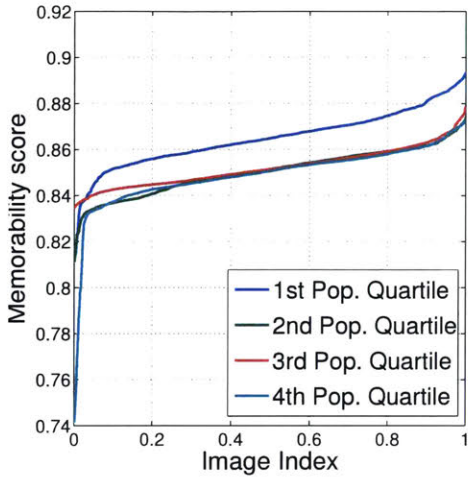
memorability. Overall, we find that images that evoke negative emotions such as *anger* and *fear* tend to be more memorable than those portraying positive ones. The analysis on the statistical differences between the memorability of emotions is shown in Figure 2-5a.

Aesthetics: Figure 2-4d shows the aesthetic scores of the 25% most and least memorable images from the AVA dataset [109]. As in [4, 63], we find that the aesthetic score of an image and its memorability have little to no correlation.

Objects: We use the ground-truth annotation of objects available in Microsoft COCO [96] to study the impact of objects on memorability (*La Mem* contains 10,000 images from COCO). We use a similar procedure as [63] here using a SVR with histogram intersection kernel, and iteratively removing objects to quantify their importance. Figure 2-6 shows some examples of our results. Overall, we find that objects that humans interact with and are more 'hand-held' tend to be more memorable than those that are large. For example, bananas tend to be a lot more memorable than giraffes.



(a) Memorability vs emotions - statistical differences in mean



(b) Popularity vs memorability

Figure 2-5: (a) Matrix indicating if the differences in the mean memorability scores of different emotions are statistically significant at the 5% level of significance. Blue indicates no difference, red indicates *col>row*, while green indicates *row>col*. (b) Plot showing the relationship of popularity and memorability. We split the images into 4 quartiles based on their popularity scores (*1st* being the one with the highest popularity scores), and plot a graph showing the sorted memorability scores. Overall, we find that images that are more popular are more memorable and vice versa.

2.3 Predicting the Memorability of Images

In this section, we focus on predicting image memorability using deep networks. In Section 2.3.1, we describe the experimental setup and our approach, MemNet, for predicting memorability. Then, in Section 2.3.2 and 2.3.3, we apply the proposed algorithms to the SUN memorability dataset and our new *La Mem* dataset respectively. Last, in Section 2.3.4 we provide additional analysis such as visualizing the internal representation learned by MemNet.

2.3.1 MemNet: CNN for Memorability

Given the recent success of convolutional neural networks (CNN) in various visual recognition tasks [47, 87, 128, 170, 149, 174], we use them here for memorability prediction. As memorability depends on both scenes and objects, we initialize the training using the pre-trained Hybrid-CNN from [174], trained on both ILSVRC 2012 [134]

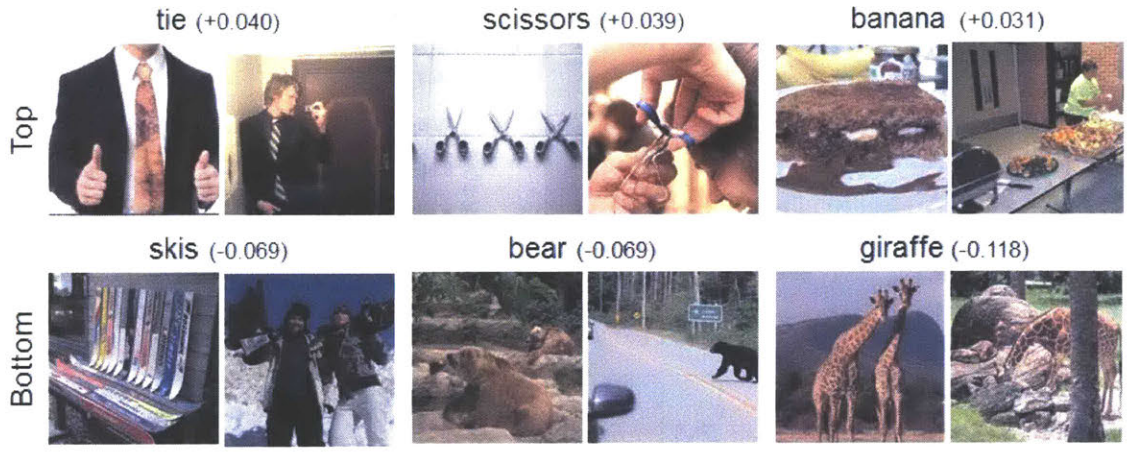


Figure 2-6: Correlation of memorability with objects using ground-truth annotation from Microsoft COCO [96]. The numbers in the parentheses show the average contribution of the corresponding objects to the memorability of the images.

and Places dataset [174]. Memorability is a single real-valued output, so we use a Euclidean loss layer to fine-tune the Hybrid-CNN. We call our final network *MemNet*.

Setup and baseline: We followed the same experimental procedure as [63] where we distribute the data into random train and test splits: the train split is scored by one half of the workers, and the test split by the other half. For the SUN Memorability dataset, we repeat the experiment for 25 splits, but for *La Mem*, we use 5 splits due to the computationally expensive fine-tuning step. As the baseline, we report performance when using HOG2x2 features that are extracted in a similar manner to [80] i.e., we densely sample HOG [24] in a regular grid and use locality-constrained linear coding [161] to assign descriptors to a dictionary of size 256. Then, we combine features in a spatial pyramid [90] resulting in a feature of dimension 5376. This is the best performing feature for predicting memorability as reported by various previous works [75, 80, 63]. For both HOG2x2 and features from CNNs, we train a linear Support Vector Regression machine [37, 32] to predict memorability. We used validation data to find the best B and C hyperparameters[1].

As proposed in [75], we evaluate two notions of memorability - one that does not account for false alarms (no FA), and one that does (with FA). It can be important

[1]Note that, since Liblinear [37] regularizes the bias term, B, we found that it was important to vary it to maximize performance.

to account for false alarms to reduce the *noise* in the signal as people may remember some images simply because they are *familiar*, but not *memorable*. Indeed, we find that this greatly improves the prediction rank correlation despite using the same features. In our experiments, we evaluate performance using both metrics. Note that the models for 'no FA' and 'with FA' as mentioned in Table 2.1 are trained independently.

2.3.2 SUN Memorability dataset

Table 2.1 (left) shows the results of training on the SUN Memorability dataset and testing on both datasets. We observe that deep features significantly outperform the existing state-of-the-art by about 0.15 (0.63 vs 0.48 with FA, and 0.60 vs 0.45 no FA). This demonstrates the strength of the deep features as shown by a variety of other works. Similar to [75], we observe that the performance increases significantly when accounting for false alarms. Apart from high performance on the SUN Memorability dataset, the features learned by CNNs generalize well to the larger *La Mem* dataset. Despite having significantly less variety in the type of images, the representational power of the features allow the model to perform well.

Fine-tuning has been shown to be important for improving performance [128], but we find that it reduces performance when using the SUN Memorability dataset. This is due to the limited size of the data, and the large number of network parameters, leading to severe overfitting of the training data. While the rank correlation of the training examples increases over backpropagation iterations, the validation performance remains constant or decreases slightly. This shows the importance of having a large-scale dataset for training a robust model of memorability.

Note that Table 2.1 only compares against having the single best feature (HOG2x2), but even with multiple features the best reported performance [80] is 0.50 (no FA), which we outperform significantly. Interestingly, our method also outperforms [63] (0.54, no FA) and [81] (0.58, no FA) which use various ground truth annotations such as objects, scenes and attributes.

2.3.3 *La Mem* dataset

Table 2.1 (right) shows the results of training on the *La Mem* dataset, and testing on both datasets. In this case, we split the data to 45k examples for training, 4k examples for validation and 10k examples for testing. We randomly split the data 5 times and average the results. Overall, we obtain the best rank correlation of 0.64 using MemNet. This is remarkably high given the human rank correlation of 0.68 for *La Mem*. Importantly, with a large-scale dataset, we are able to successfully fine-tune deep networks without overfitting severely to the training data, and preserving generalization ability in the process.

Additionally, we find that the learned models generalize well to the SUN Memorability dataset achieving a comparable performance to training on the original dataset (0.61 vs 0.63, with FA). Further, similar to the SUN Memorability dataset, we find that higher performances can be attained when accounting for the observed false alarms.

2.3.4 Analysis

In this section, we investigate the internal representation learned by MemNet. Figure 2-7 shows the average of images that maximally activate the neurons in two layers near the output of MemNet, ordered by their correlation to memorability. We see that many units near the top of `conv5` look like close-ups of humans, faces and objects while units near the bottom (so associated with more forgettable objects) look more like open and natural scenes, landscapes and textured surfaces. A similar trend has been observed in previous studies [63]. Additionally, to better understand the internal representations of the units, in Figure 2-8, we apply the methodology from [173] to visualize the segmentation produced by five neurons from `conv5` that are strongly correlated with memorability (both positively and negatively). We observe that the neurons with the highest positive correlation correspond to body parts and faces, while those with a strong negative correlation correspond to snapshots of natural scenes. Interestingly, these units emerge automatically in MemNet without

any explicit training to identify these particular categories.

Impact of objects: Here, we use a prediction model to identify the importance of different objects. As the ground truth information of objects in all images is not available, and can be rather expensive to label, we need an automatic method to predict the objects in an image. To do this, we use an approach similar to [76]. We train a linear SVR on the probability outputs of the network trained on ImageNet data [87]. This network outputs the probability of an image containing any of 1000 objects ranging from dogs to castles. The learned weights can then be used to attribute importance to the different objects, and find their correlation with memorability. We visualize some of these in Figure 2-9.

From the figure, we observe that while the most correlated objects are neck braces and band aids, they do not actually tend to exist in the images. We believe that it not these objects that increase the memorability of images, but the fact that these objects tend to co-exist with people, and as found in [63], people tend to be highly memorable. However, there are other objects that do exist in the images and are highly correlated with memorability, such as brassiere and bikinis. These objects are also closely correlated with those that make an image popular [76], and exploring this relationship could be a promising avenue for future work.

For the positively correlated objects, we observe that the images tend to have consistently high memorability scores. However for the images scoring high for objects that have little to no correlation with memorability, we find that the scores are highly variable. Essentially these objects tend to be uninformative about memorability. This might be because they don't tend to exist sufficiently in our dataset, or they truly do not affect the memorability of images. Expanding the memorability dataset further to contain a larger, and more curated variety of objects could be a promising direction for further research.

Last, we find that open scenes with few objects, and buildings without people in the surroundings tend to have low memorability scores. Indeed the object categories that are negatively correlated with memorability tend to suggest this phenomenon. It is important to note that the current models are limited in being able to identify

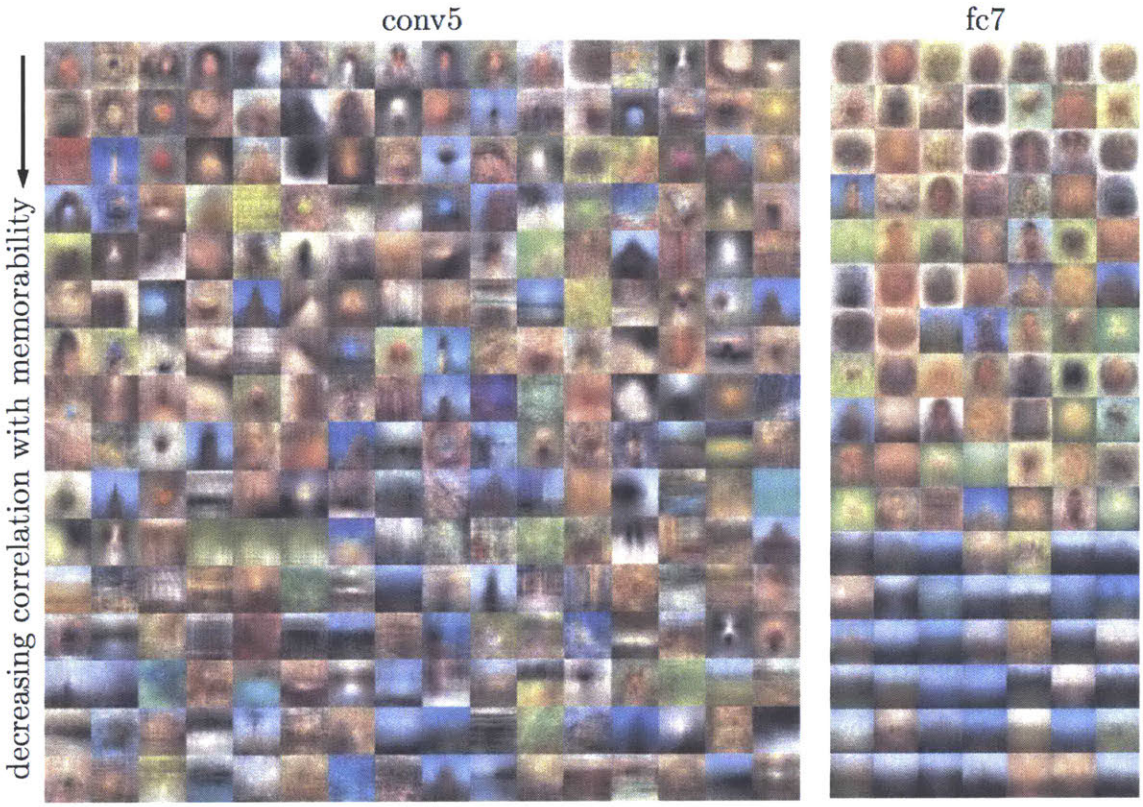


Figure 2-7: Visualizing the CNN features after fine-tuning, arranged in the order of their correlation to memorability from highest (top) to lowest (bottom). The visualization is obtained by computing a weighted average of the top 30 scoring image regions (for `conv5`, this corresponds to its theoretical receptive field size of $163 * 163$, while for `fc7` it corresponds to the full image) for each neuron in the two layers. From top to bottom, we find the neurons could be specializing for the following: people, busy images (lots of gradients), specific objects, buildings, and finally open scenes. This matches our intuition of what objects might make an image memorable. Note that `fc7` consists of 4096 units, and we only visualize a random subset of those here.

the set of objects in images, and we can better understand the impact of objects as we develop better visual recognition systems over time.

2.4 Predicting the Memorability of Image Regions

In this section, we investigate whether our model can be applied to understanding the contribution of image regions to memorability [80]. Predicting the memorability of image regions could allow us to build tools for automatically modifying the memorability of images [79], which could have far-reaching applications in various domains

Figure 2-8: The segmentations produced by neurons in `conv5` that are strongly correlated, either positively or negatively, with memorability. Each row corresponds to a different neuron. The segmentations are obtained using the data-driven receptive field method proposed in [173].

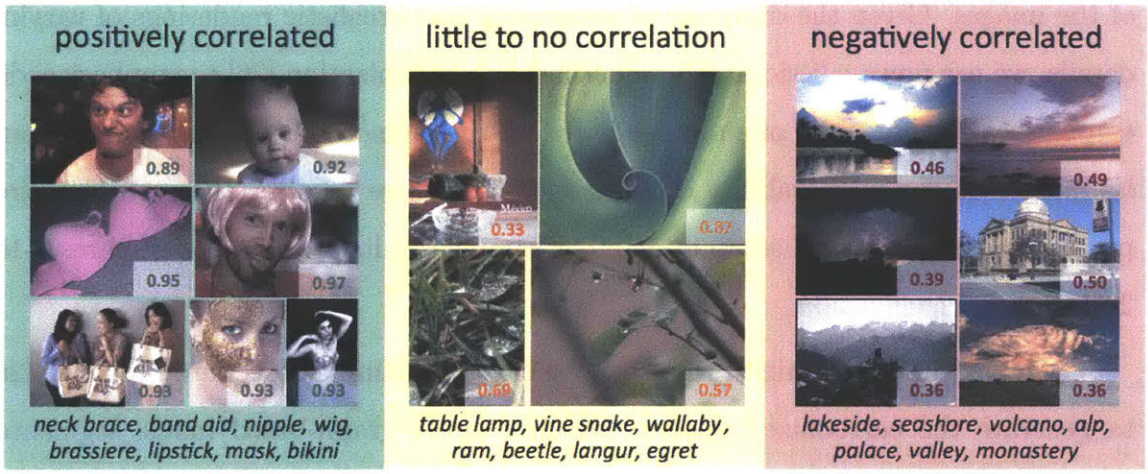


Figure 2-9: Correlation of memorability with different objects, decreasing from left to right. The number on each image indicates its memorability score as measured in our experiment. The images shown are sampled randomly from those that had a high score for at least one of the objects in the three correlation ranges.

ranging from advertising and gaming to education and social networking. First, we describe the method of obtaining memorability maps, and then propose a method to evaluate them using human experiments. Overall, using MemNet, we can accurately predict the memorability of image regions.

To generate memorability maps, we simply scale up the image and apply MemNet to overlapping regions of the image. We do this for multiple scales of the image and average the resulting memorability maps. To make this process computationally efficient, we use an approach similar to [97]: we convert the fully-connected layers, `fc6` and `fc7` to convolutional layers of size $1 * 1$, making the network fully-convolutional. This fully-convolutional network can now be applied to images of arbitrary sizes to generate different sized memorability maps e.g., an image of size 451×451 would generate an output of size 8×8. We do this for several different image sizes and average the outputs to generate the final memorability map (takes ~1s on a typical GPU). The second column of Figure 2-10 shows some of the resulting memorability maps. As expected, the memorability maps tend to capture cognitively salient regions that contain meaningful objects such as people, animals or text.

While the maps appear semantically meaningful, we still need to evaluate whether the highlighted regions are truly the ones leading to the high/low memorability scores of the images. Given the difficulty of generating photorealistic renderings of varying details, we use non-realistic photo-renderings or cartoonization [26] to emphasize/de-emphasize different parts of an image based on the memorability maps, and evaluate its impact on the memorability of an image. Below, we describe the cartoonization process and experimental setup in detail. Example results are shown in Figures 2-10, 2-11 and 2-12.

Generating cartoonized images: We develop a simple procedure similar to [26] to generate the cartoonized images: we first apply graph-based image segmentation [43] with two sets of parameters to generate a high detail and low detail images, containing a large and small number of segments respectively. The cartoonized version of an image is simply a combination of segments from each of these two pairs of images, where the segments are represented by the average color of the pixels within

it. The difference arises in the way we pick what segments come from the high detail or low detail images. To ensure a uniform level of detail across all cartoonized versions of an image, we ensure that approximately 50% of the pixels of each of the cartoonized versions of an image (high, medium and low) are from the high detail image, and the remaining 50% are from the low detail image.

For the *high* image as shown in our examples, we use the highest scoring regions of the memorability map to select the segments that come from the high detail image. Specifically, we assign each segment a memorability score as the max memorability score within that segment from the memorability map. Then we iteratively select segments from the high detail image until the selected segments occupy approximately 50% of the image area. Then for the remaining empty space, we use segments from the low detail image. For the *low* case, we do exactly the opposite, picking segments from the high detail image where the memorability score is minimized. For the *medium* case, we select segments randomly such that the image is composed of 50% high detail segments and 50% low detail segments. While this strategy of normalization may not be perfect, it seems to work reasonably in practice. This is an area that we hope to explore and improve further in future work.

Additionally, similar to [26], we apply Canny edge detection to the original image and add tapered lines of varying thickness to the cartoonized images. The thickness is varied based on the length of the line, and the first third and last third of the lines are tapered to give a more cartoonized appearance. Further, we add and remove lines in regions based on the amount of detail we want to preserve in that region.

Experimental setup: Given an image and a heatmap, we investigate the difference in human memory for the following scenarios: (1) *high* – emphasizing regions of high memorability and de-emphasizing regions of low memorability (Figure 2-10 col 3), (2) *medium* – having an *average* emphasis across the entire image (Figure 2-10 col 4), and (3) *low* – emphasizing regions of low memorability and de-emphasizing regions of high memorability (Figure 2-10 col 5). If our algorithm is identifying the *correct* memorability of image regions, we would expect the memorability of the images from the above three scenarios to rank as *high* > *medium* > *low*.

Following the above procedure, we generate three cartoonized versions of 250 randomly sampled images based on the memorability maps generated by our algorithm. We use our efficient visual memory game (Section 2.1) to collect memorability scores of the cartoonized images on AMT. We ensure that a specific worker can see exactly one modification of each image. Further, we also cartoonize the filler and vigilance images to ensure that our target images do not stand out. We collect 80 scores per image, on average.

Results: The results of this experiment are summarized in Figure 2-13. Interestingly, we find that our algorithm is able to reliably identify the memorability of image regions. All pairwise relationships, *low*<*medium*, *low*<*high* and *medium*<*high* are statistically significant (5% level). This shows that the memorability maps produced with our method are reliable estimates of what makes an image memorable or forgettable, serving as a building block for future applications. We also observe that the memorability of all cartoonized versions of an image tends to be lower than the original image, even though the *high* version emphasizes the more memorable regions. We expect that this is because even the *high* version of the image loses significant details of objects as compared to the original photograph. This might make it harder for people to distinguish between images and/or identify the objects.

Further, observe that the graphs in Figure 2-13 are independently sorted. Here we provide the results when we consider examples as tuples of either pairwise images such as *low* and *high*, or triplets of images (*low*, *medium* and *high*). This is a more strict test to check if our method produces desirable results. First, we find that 45% of the images obey the *low* < *medium* < *high* relationship. Note that this is significantly above chance as chance accuracy here is 16.7%. Our result is also statistically above chance showing that our method is capable of accurately predicting, to a reasonable extent, which regions of an image are memorable and vice versa. Now we consider pairwise relationships where the chance accuracy is 50%. For the following pairwise relationships, *low*<*high*, *medium*<*high*, *low*<*meidum*, we obtain the following accuracies: 75.6%, 70.0%, 60.0%. We find that all of these are statistically significantly above chance.

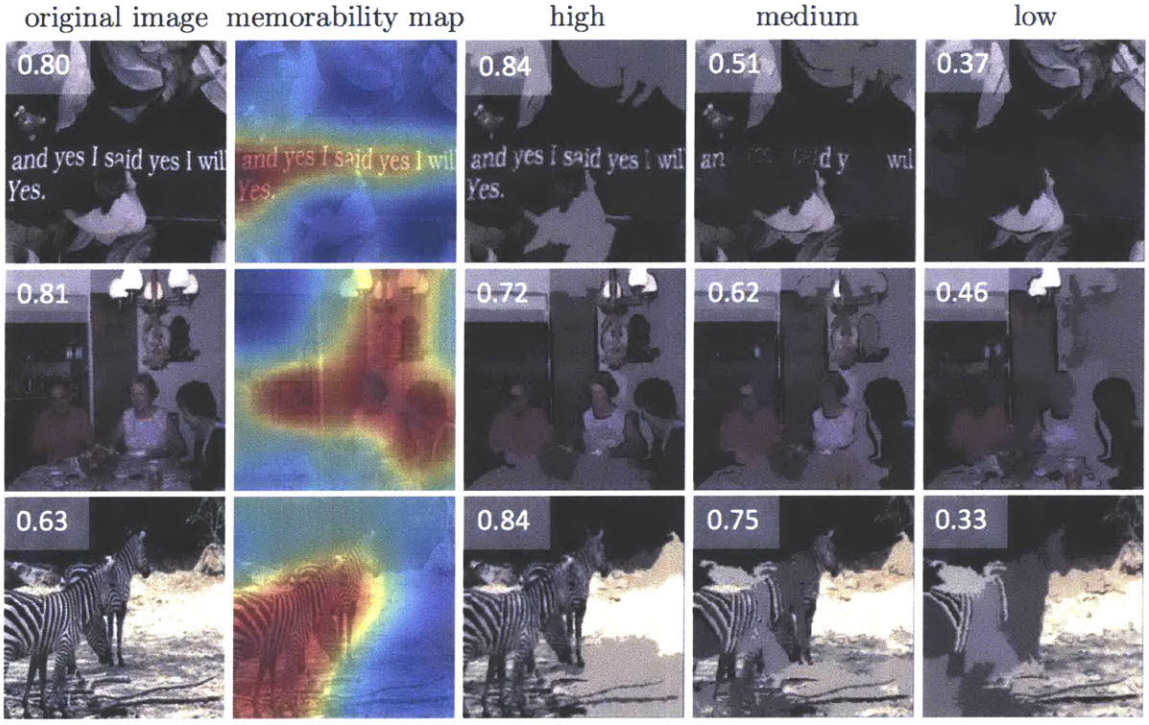


Figure 2-10: The memorability maps for several images. The memorability maps are shown in the jet color scheme where the color ranges from blue to red (lowest to highest). Note that the memorability maps are independently normalized to lie from 0 to 1. The last three columns show the same image modified using [26] based on the predicted memorability map: *high* image – regions of high memorability are emphasized while those of low memorability are de-emphasized e.g., in the first image text is visible but leaves are indistinguishable, *medium* image – half the image is emphasized at random while the other half is de-emphasized e.g., some text and some leaves are visible for the first image, and *low* image – regions of low memorability are emphasized while those of high memorability are de-emphasized e.g., text is not visible in first image but leaves have high detail. The numbers in white are the resulting memorability scores of the corresponding images.

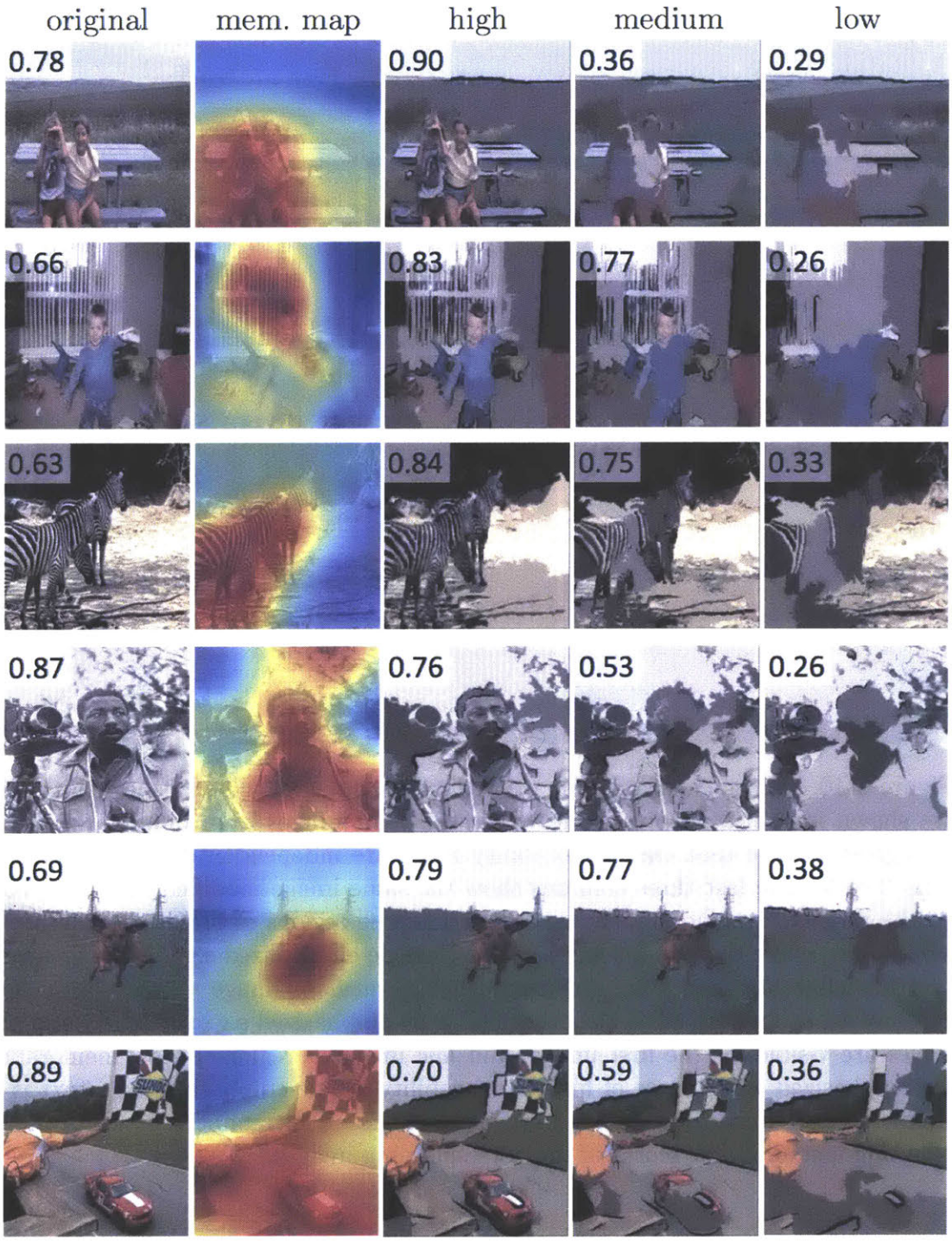


Figure 2-11: Refer to the caption of Figure 2-10 for a detailed explanation.

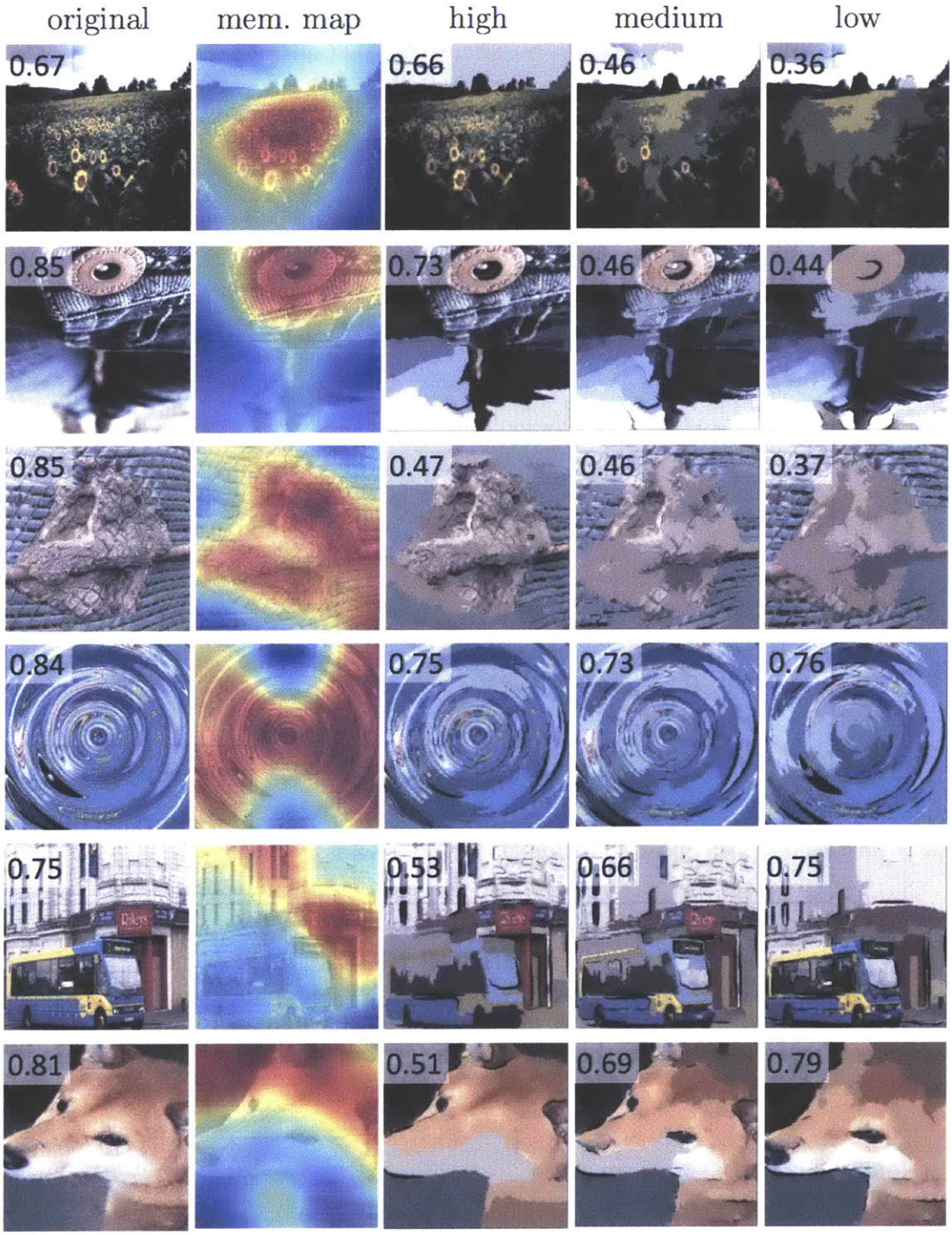


Figure 2-12: Refer to the caption of Figure 2-10 for a detailed explanation. The last three rows show failure cases where the memorability from the human experiment does not match what we expect. We find that the CNN fails at predicting the salient objects in some cases (last two rows), and in other cases (third row from the bottom) the heatmap does not have a significant impact on the image.

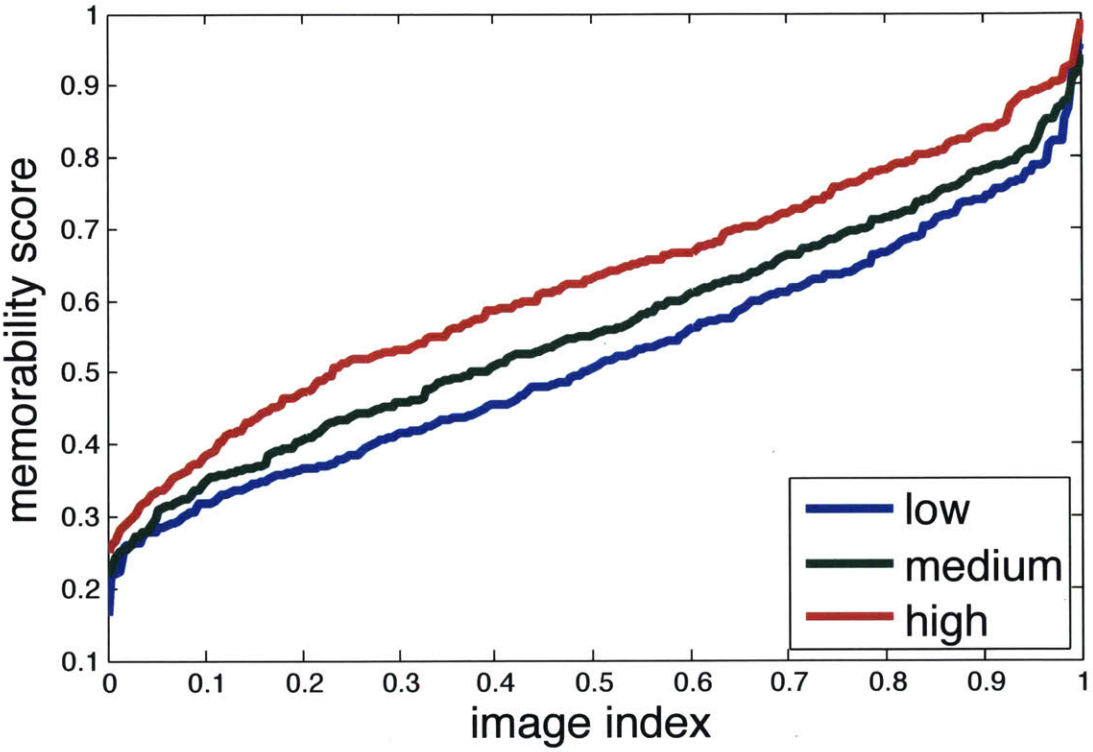


Figure 2-13: Memorability scores of the cartoonized images for the three settings shown in Figure 2-10. Note that the scores for *low*, *medium* and *high* are independently sorted.

2.5 Modifying the Memorability of Faces

April 2016, London. "Will this make me more memorable?" she wonders, playing with the options on her new smart phone app. Her face on the screen has gotten an instant lift, the expression is somehow more interesting. "This is for the job application, I want them to remember me."

One ubiquitous fact about people is that we cannot avoid evaluating the faces we see in daily life. In fact, we automatically tag faces with personality, social, and emotional traits within a single glance: according to [163], an emotionally neutral face is judged in the instance it is seen, on traits such as level of attractiveness, likeability, and aggressiveness. However, in this flash judgment of a face, an underlying decision is happening in the brain – should I remember this face or not? Even after seeing a picture for only half a second we can often remember it [124]. Face memorability is in fact a critical factor that dictates many of our social interactions; even if a face seems friendly or attractive, if it is easily forgotten, then it is meaningless to us. Work in computer graphics has shown how to select a candid portrait from videos [46], or how faces can be slightly adjusted to look more attractive [94]. With the rapid expansion of social networks and virtual worlds, we are becoming increasingly concerned with selecting and creating our best selves – our most remembered selves.

Back to 2013: Can we make a portrait more memorable? In this work, we show that it is indeed possible to change the memorability of a face photograph. Figure 1 shows some photographs manipulated using our method (one individual per row) such that each face is more or less memorable than the original photograph, while maintaining the identity, gender, emotions and other traits of the person. The changes are subtle, difficult to point out precisely, or even to describe in words. However, despite these subtle changes, when testing people's visual memory of faces, the modification is successful: after glancing at hundreds of faces, observers remember better seeing the faces warped towards memorability, than the ones warped away from it.

It is not immediately intuitive what qualities cause a face to be remembered. Several memory researchers [157, 159] have found that measures of distinctiveness

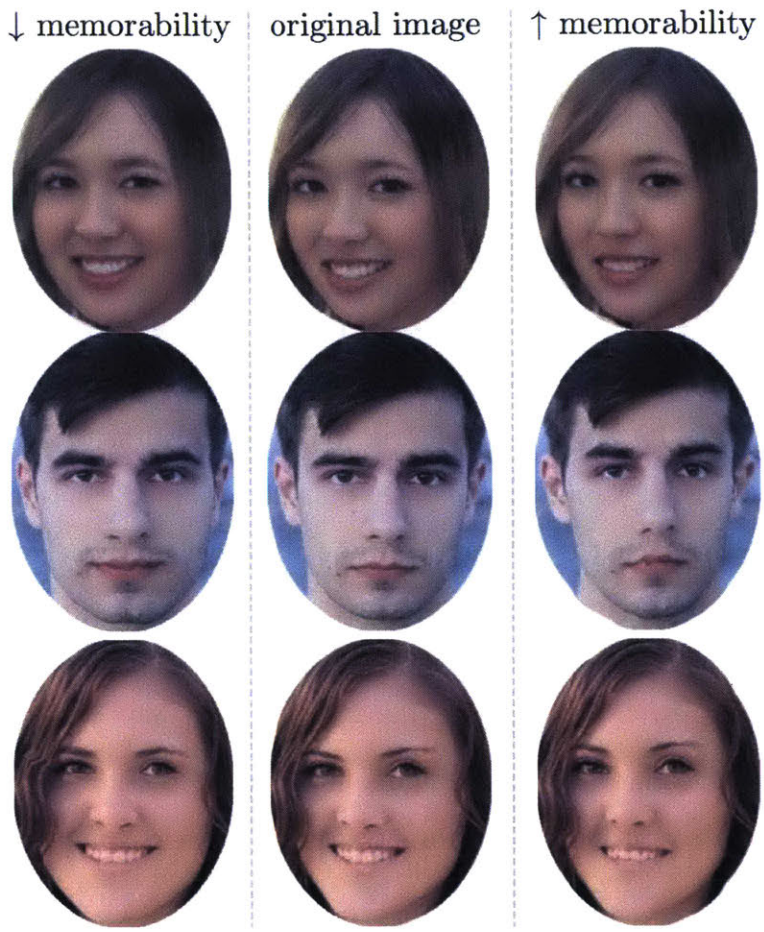


Figure 2-14: Examples of modifying the memorability of faces while keeping identity and other attributes fixed. Despite subtle changes, there is a significant impact on the memorability of the modified images.

are correlated with memorability, while familiarity is correlated with increased false memories (believing you saw something you have not seen). Another line of work has discussed the use of geometric face space models of face distinctiveness to test memorability [16]. Importantly, recent research has found that memorability is a trait intrinsic to images, regardless of the components that make up memorability [4, 63, 79]. Thus, a fresh approach looking at memorability itself rather than its individual components (such as gender or distinctiveness) allows us here to create a method to change the memorability of a face while keeping identity and other important facial characteristics (like age, attractiveness, and emotional magnitude) intact.

To overcome the complex combination of factors that determine the memorability of a face, we propose a data-driven approach to modify face memorability. In our method, we combine the representational power of features based on Active Appearance Models (AAMs) with the predictive power of global features such as Histograms of Oriented Gradients (HOG) [24], to achieve desired effects on face memorability. Our experiments show that our method can accurately modify the memorability of faces with an accuracy of 74%.

Related work (Face modification): The major contribution of this work is modifying faces to make them more or less memorable. There has been significant work in modifying faces along other axes or attributes, such as gender [83], age [89, 146], facial expressions [2] and attractiveness [94]. However, while these works focus on intuitive and describable attributes, our work focuses on a property that is yet not well understood. Further, our method differs from existing works as it combines powerful global image features such as SIFT [99] and HOG [24] to make modifications instead of using only shape and appearance information as done in AAMs [22].

Related work (Face caricatures): Work in computer vision and psychology has also looked at face caricatures [7, 116], where the distinctive (i.e., deviant from the physical average) features of a face are accentuated to create an exaggerated face. The distinctiveness of a face is known to affect its later recognition in humans [16], so increasing the memorability of a face may caricaturize it to some degree. However, unlike face caricature work, the current study aims to maintain the realism of the faces, by preserving face identity. Recent memorability work finds that distinctiveness is not the sole predictor of face memorability [4], so the algorithm presented in this section is likely to change the faces in more subtle ways than simply enlarging distinctive physical traits.

2.5.1 Predicting Face Memorability

As we make changes to a face, we need a model to reliably predict its memorability; if we cannot predict memorability, then we cannot hope to modify it in a predictable way. Thus, in this section, we explore various features for predicting face memorability

and propose a robust memorability metric to significantly improve face memorability prediction. We also note that the task of automatically predicting the memorability of faces using computer vision features has not been explored in prior works.

In Section 2.5.1 we describe the dataset used in our experiments and the method used to measure memorability scores. Then, in Section 2.5.1, we describe our robust memorability metric that accounts for false alarms leading to significantly improved prediction performance (Section 2.5.1).

Measuring Memorability

Memorability scores of images are obtained using a visual memory game, as described in [4, 63]: Amazon Mechanical Turk (AMT) workers are presented with a sequence of faces, each shown for 1.4 seconds with a 1 second blank interval, and their task is to press a key when they encounter a face they believe they have seen before. The task is structured in levels of 120 images each and includes a combination of target images (i.e. images that repeat) and filler images (i.e. images shown only once). Target image repeats are spaced $91-109$ images apart. A given target image and its repeat are shown to N unique individuals. Of the N individuals, we define H to be the number of people that correctly detected the repeat, and F as the number of people that false alarmed on the first showing. Based on this, Bainbridge et al [4] investigated two *memorability scores* for an image, (1) the proportion of correct responses , $\frac{H}{N}$, also called the hit rate and (2) the proportion of errors, $\frac{F}{N}$, also called false alarm rate. However, they do not combine the two scores into a metric that allows for a trade-off between correct responses and false alarms. We address this in the following section.

Memorability Score with False Alarms

As noted in [4], faces tend to have a significantly higher false alarm rate than scenes [63], i.e. there are certain faces that a large proportion of people believe they have seen before which they actually have not. Rather than being memorable (with high correct detections), these faces are in fact "familiar" [159] - people are more likely to

Metric	Face database [4]		Scene database [63]	
	Human	Pred	Human	Pred
$\frac{H}{N}$ [63]	0.68	0.33	0.75	0.43
$\frac{(H-F)}{N}$ (Ours)	0.69	**0.51**	0.73	**0.45**

Table 2.2: **Memorability score with false alarms:** Spearman's rank correlation (ρ) using the existing [63] and proposed (Section 2.5.1) memorability score metrics. 'Human' refers to human consistency evaluated using 25 train/test splits of data (similar to [63]), while 'prediction' refers to using support vector regression (SVR) trained on dense HOG [24] (details in Section 2.5.1).

report having seen them, leading to both correct detections and false alarms. To account for this effect, we propose a slight modification to the method of computing the memorability score.

To account for the *wrong* correct detections, we simply subtract the false alarms from the hit count to get an estimate of the *true* hit count of an image. Thus, the new memorability score can be computed as $\frac{H-F}{N}$, unlike $\frac{H}{N}$ as done in [63] and [4]. Note that $H, F \in [0, N]$, so $\frac{H-F}{N} \in [-1, 1]$ while $\frac{H}{N} \in [0, 1]$. The negative memorability scores can be easily adjusted to lie in the range $[0, 1]$.

The result of applying the above metric is summarized in Table 2.2. To show that our metric is robust, we apply it to both the face [4] and scene memorability [63] datasets. We observe that human consistency remains largely the same in both cases. This is expected as the false alarms tend to be consistent across participants in the study. Importantly, we observe that there is a significant increase in the prediction performance from rank correlation of 0.33 to 0.51 for face memorability. By using our new metric, we have effectively decreased noise in the prediction labels (memorability scores) caused by inflated memorability scores of *familiar* images. This allows the learning algorithm to better model the statistics of the data that best describe memorability. We note that the performance improvement is not as large in scenes because the human consistency of false alarms and the rate of false alarms is significantly lower, and effects of familiarity may function differently.

We use our proposed memorability score that takes false alarms into consideration for the remaining experiments in this section.

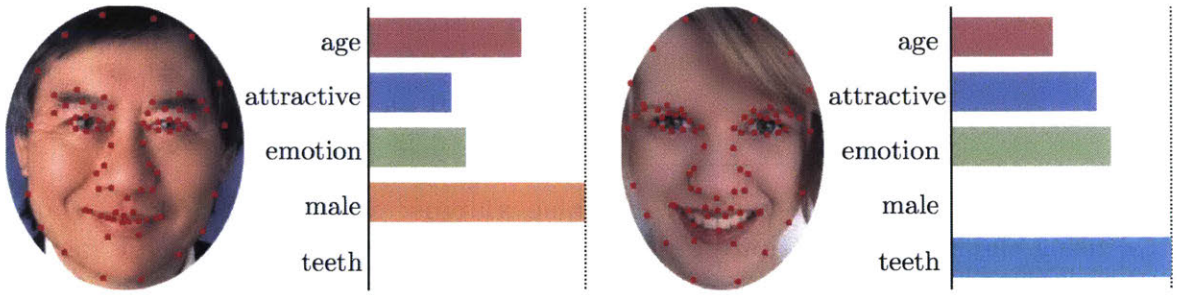


Figure 2-15: **Additional annotation:** We annotated 77 facial landmarks of key geometric points on the face and collected 19 demographic and facial attributes for each image in the 10k US Adult Faces Database[2].

Prediction experiments

In this section, we describe the additional annotations we collected to enable better face modification. Then we describe the setup of our experiments such as the evaluation metric and features used for the prediction of memorability and other attributes.

Additional annotation

In order to modify faces while preserving facial attributes such as gender and emotion, we need two sets of additional annotations: (1) facial landmark locations to allow for the easy modification of faces by moving these keypoints and generating new images through warping, and (2) facial attribute (e.g., attractiveness, emotion) ratings so we can keep them fixed. Figure 2-15 shows some examples of the additional annotation we collected on the 10k US Adult Faces Database [4]. The full set of questions asked about each image are shown in Table 2.3. Note that since we aim to modify faces instead of detect keypoints, we assume that landmark annotation is available at both train and test times. Locating these landmarks is a well-studied problem and we could obtain them using various automatic methods [22, 175] depending on the application. In our work, the landmarks were annotated by experts to ensure high consistency across the dataset.

To collect the facial attributes, we conducted a separate AMT survey similar to [88], where each of the 2222 face photographs was annotated by twelve different workers on 19 demographic and facial attributes of relevance for face memorability and face modification. We collected a variety of attributes including demographics

Physical Characteristics	Possible Answers	Consolidation
Gender?	Male or Female	Mode
Race?	White, Black, East Asian, South Asian, Hispanic, Middle Eastern or Other	Mode
Age?	< 20, $20-30$, $30-45$, $45-60$, $60+$	Mean
Is this person famous?	Yes, No or Maybe	Mode
Emotion?	Disgust, Happy, Sad, Angry, Fear, Surprise or Neutral	Mode
High-level Attributes	**Possible Answers**	**Consolidation**
Emotional magnitude?	Little – A lot $(1-5)$	Mean
How friendly?	Very unfriendly – Very friendly $(1-5)$	Mean
How attractive?	Unattractive – Attractive $(1-5)$	Mean
How memorable?	Forgettable – Memorable $(1-5)$	Mean
How common?	Uncommon – Common $(1-5)$	Mean
Computational Modeling	**Possible Answers**	**Consolidation**
Face direction?	At you, up, down, left or right	Mode
Eyes direction?	At you, up, down, left or right	Mode
Facial hair?	None – A lot $(1-3)$	Mean
How much teeth showing?	None – A lot $(1-3)$	Mean
Makeup?	None – A lot $(1-3)$	Mean
Image quality?	Poor – Very good $(1-5)$	Mean
Vision & Graphics	**Possible Answers**	**Consolidation**
Speed of expression?	Slowly – Quickly $(1-5)$	Mean
Good profile picture?	Yes, No or Maybe	Mean
Star of a movie?	Yes, No or Maybe	Mean

Table 2.3: Questions asked in the Mechanical Turk attributes study

such as gender, race and age, physical attributes such as attractiveness, facial hair and make up, and social attributes such as emotional magnitude and friendliness. These attributes are required when modifying a face so we can attempt to keep them constant or modify them jointly with memorability, as required by the user.

Experimental Setup

Dataset: In our experiments, we use the 10k US Adult Faces Database [4] that consists of 2222 face photographs annotated with memorability scores.

Evaluation: The prediction performance is evaluated either using Spearman's

rank correlation (ρ) for real-valued attributes, and accuracy for discrete-valued attributes. We evaluate the performance on 25 random train/test splits of the data (as done in [63]) with an equal number of images for training and testing. For memorability, the train splits are scored by one half of the participants and the test splits are scored by the other half with a human consistency of $\rho = 0.69$. This can be thought of as an upper bound on the performance we can hope to achieve.

Features: We use similar features as [80] for our experiments, namely the color naming feature [158], local binary pattern (LBP) [112], dense HOG2x2 [24] and dense SIFT [99]. For the bag-of-words features (i.e., color, HOG and SIFT), we sample descriptors at multiple scales and learn a dictionary of codewords using K-means clustering. Then we use Locality-constrained Linear Coding (LLC) [161] to assign descriptors to codewords, and apply max-pooling with 2 spatial pyramid levels to obtain the final feature vector. Similarly, we use a 2-level spatial pyramid for the non-uniform LBP descriptor. In addition, we also use the coordinates of the ground-truth landmark annotations (shown in Figure 2-15) normalized by image size as 'shape' features.

Model: For discrete-valued attributes, we apply a one-vs-all linear support vector machine (SVM) [37], while for real-valued attributes, we apply support vector regression (SVR) [32]. The hyperparameters, C (for SVM/SVR) and ϵ (for SVR), are found using cross-validation.

Memorability and attribute prediction

Table 2.4 summarizes the prediction performance of face memorability and other attributes when using various features. For predicting memorability, dense global features such as HOG and SIFT significantly outperform landmark-based features such as 'shape' by about 0.15 rank correlation. This implies that it is essential to use these features in our face modification algorithm to robustly predict memorability after making modifications to a face. While powerful for prediction, the dense global features tend to be computationally expensive to extract, as compared to shape. As described in Section 2.5.2, shape is used in our algorithm to parametrize faces so it essentially has zero cost of extraction for modified faces.

Similar to memorability prediction, we find that dense global features tend to outperform shape features for most attributes. However, as compared to memorability, the gap in performance between using shape features and dense features is not as large for other attributes. This might suggest why, unlike our method, existing methods [89, 94] typically use landmark-based features instead of dense global features for the modification of facial attributes.

2.5.2 Algorithm for Modifying Face Memorability

In order to modify a face photograph, we must first define an expressive yet low-dimensional representation of a face. We need to parametrize a face such that we can synthesize new, realistic-looking faces. Since faces have a largely rigid structure, simple methods such as Principal Component Analysis (PCA) e.g. AAMs [22] tend to work fairly well. In Section 2.5.2, we describe a method based on AAMs where we represent faces using two distinct features, namely shape and appearance.

While the above parametrization is extremely powerful and allows us to modify a given face along various dimensions, we require a method to evaluate the modifications in order to make predictable changes to a face. Our objective is to modify the memorability score of a face, while preserving the identity and other attributes such as age, gender, emotions, etc of the individual. We encode these requirements in a cost function as described in Section 2.5.2. Specifically, our cost function consists of three terms: (1) the cost of modifying the identity of the person, (2) the cost of not achieving the desired memorability score, and (3) the cost of modifying other attributes. By minimizing this cost function, we can achieve the desired effect on the memorability of a face photograph.

As highlighted in Section 2.5.1, it is crucial to use dense global features when predicting face memorability. These features tend to be highly non-convex in our AAM parameter space, making it difficult to optimize the cost function exactly. Thus, we propose a sampling-based optimization procedure in Section 2.5.2 that leverages the representational power of AAMs with the predictive power of dense global features in a computationally efficient manner.

Representation

Using facial landmark-based annotations (described in Section 2.5.1) in the form of AAMs is a common method for representing faces for modification because it provides an expressive, and low-dimensional feature space that is reversible, i.e., we can recover an image easily after moving in this feature space. AAMs typically have two components to represent a face, namely shape (x_s) and appearance (x_a).

To obtain x_s, we first compute the principal components of the normalized landmark locations[3] across the datasets. Then, for a particular image, x_s is given as the coefficients of these principal components. Similarly, to obtain x_a, we first warp all the faces to the mean shape and apply PCA to the concatenated RGB values of the resulting image[4] (resized to a common size). Then, x_a for a given image is given by the coefficients of these principal components. Thus, the full parametrization of a face, x, can be written as $x = [x_s\ x_a]$, i.e., the concatenation of shape and appearance features. We can now modify x and use the learned principal components to synthesize a new face.

When applying the above parametrization to faces, we observed that there were significant distortions when warping a face, and a high reconstruction error even without any modification (despite keeping 99.5% of the principal components). To improve the reconstruction, we cluster the normalized facial landmarks and apply PCA independently to each cluster. Further, as shown in Figure 2-15, images in the 10k US Adult Faces Database contain an ellipsoid around them to minimize the background effects on memorability, so we added uniformly spaced landmarks along the boundary to constrain the warping in order to obtain more realistic results. We evaluate the effects of these modifications to AAM in Section 2.5.3.

[3]The 77 normalized landmark locations are concatenated to form a 154 dimensional shape vector.

[4]As there could be components of appearance outside the face region such as hair that we would like to be able to modify, we use the entire image instead of just the face pixels (as is typically done).

Cost function

In order to modify a face to a desired memorability score while preserving identity and other facial attributes, we define a cost function with the following three terms:

- C_{id}: Cost of modifying the identity
- C_{mem}: Cost of *not* attaining the desired memorability
- C_{attr}: Cost of modifying facial attributes such as age, attractiveness, emotional magnitude, etc

We parametrize the above terms using the shape and appearance based representation, x, described in Section 2.5.2. Then, our optimization objective is:

$$\min_{x} \quad C_{id}(x) + \lambda C_{mem}(x) + \gamma C_{attr}(x) \tag{2.7}$$

where λ and γ are hyperparameters to control the relative importance of the three terms in the cost function.

Before defining the above terms explicitly, we define some common terminology. Let F define the set of image features (e.g. HOG [24] or SIFT [99]) and A, the set of facial attributes (e.g., is male or are teeth visible). Then we define $m_i(x)$ as a function to predict the memorability score of an image represented by PCA coefficients x computed using feature $i \in F$. Similarly, we define $f_{i,j}(x)$ as a function to predict the value of attribute $j \in A$ of an image defined by PCA coefficients x, computed using feature $i \in F$. Note that the landmark-based features can be directly obtained from x, while there is an implicit conversion from the PCA coefficients x to an image before extracting dense global features. For brevity, we do not write this transformation explictly. In our experiments, m_i and $f_{i,j}$ are obtained using linear SVR/SVM as described in Section 2.5.1.

Now, given an image $\hat{I}$ that we want to modify, our goal is to synthesize a new image I that has a memorability score of M (specified by the user) and preserves the identity and other facial attributes of the original image $\hat{I}$. Representing the PCA

coefficients of $\hat{I}$ by $\hat{x}$, our objective is to find the PCA coefficients x that represent I. Since landmark estimation is outside the scope of this work, we assume that the landmark annotations required to obtain $\hat{x}$ are available at both train and test time.

Based on this problem setup, we define the terms from Eqn. 2.7 as the following three equations (Eqn. 2.8, 2.9, 2.10):

$$C_{id}(x) = [w \cdot (x - \hat{x})]^2 \tag{2.8}$$

where w is the weight vector for preserving identity, learned using a Support Vector Machine (SVM) trained on the original image $\hat{I}$ as positive, and the remaining images in the dataset as negatives.

$$C_{mem}(x) = \sum_{i \in F} \frac{c_i}{|F|}(m_i(x) - M)^2 \tag{2.9}$$

where c_i represents the confidence of predictor m_i, and can be estimated using cross-validation. Since the performance of different features on memorability prediction varies significantly (Section 2.5.1), we weight the difference between M and m_i by the confidence score c_i to increase the importance of better performing features. Overall, this function penalizes the memorability score of the new image x if it does not match the desired memorability score, M.

$$C_{attr}(x) = \sum_{i \in F} \sum_{j \in A} \frac{c_{i,j}}{|F| \cdot |A|}(f_{i,j}(x) - f_{i,j}(\hat{x}))^2 \tag{2.10}$$

where $c_{i,j}$ is the confidence of predictor $f_{i,j}$, and can be estimated using cross-validation. The above function computes the distance between the estimated attribute value of the original image $f_{i,j}(\hat{x})$ and the new image $f_{i,j}(x)$. Additionally, a user could easily modify the relative importance of different attributes in the above cost function.

Overall, C_{id} and C_{attr} encourage the face to remain the same as $\hat{I}$ while C_{mem}

encourages the face to be modified to have the desired memorability score of M. By adjusting the hyperparameters appropriately, we can achieve the desired result on memorability.

Optimization

While the terms in the objective function defined in Eqn. 2.7 look deceptively simple, we note that the function is actually highly complex and non-linear because of the dense global features such as HOG or SIFT involving histograms, bag-of-words and max-pooling. Thus, typical gradient-based approaches hold little promise in this case, and finding a local minimum is the best we can hope for.

The basic idea of our algorithm is fairly straightforward: we want to find x given $\hat{x}$; since x is expected to look like $\hat{x}$, we initialize $x = \hat{x}$. Then, we randomly sample[5] some points at some fixed distance d from x. From the set of random samples, we find the sample that best minimizes the objective function and use that as our new x. We now repeat the procedure by finding samples at distance $\frac{d}{2}$ from x. We include the initial value of x as one of the samples in each iteration to ensure that the objective function always decreases. By iteratively reducing d, we can find a local minimum of our objective close to $\hat{x}$. This approach is described in greater detail in Algorithm 1.

We observe that the computational cost of feature extraction differs significantly between dense global features and landmark-based features; dense global features require the synthesis of a new image based on x, and the subsequent extraction of features while landmark-based features such as shape and appearance can be trivially obtained from x. Note that the dense global features play a significant role in accurate memorability prediction (Section 2.5.1), so we must include them in our algorithm. Since it is computationally expensive to compute dense global features for all samples, this severely restricts the total number of samples considered per iteration, making it difficult to find a good solution, or even one better than $\hat{x}$.

To overcome the computational bottleneck, we propose a two-step procedure: (1)

[5]In practice, we fit a multivariate Gaussian to the PCA coefficients of the training images (similar to [94]), and obtain random samples from this distribution. Given a random sample p, we find a point at distance d from x that lies in the direction $p - x$.

Algorithm 1: Algorithm for optimizing the cost function to modify the memorability of a face photograph (described in Section 2.5.2). The notation used is defined as follows: α is the step size, β is the update factor when an iteration fails, $maxIter$ is the maximum number of iterations of the algorithm we want to run and, k and n are the number of samples we want to evaluate with and without global features, respectively. In our experiments, we set $\alpha = 0.05, \beta = 0.5, k = 10, maxIter = 10, n = 10000$.

input: image I, landmarks L

Assign (I, L) to cluster in AAM to obtain $\hat{x}$
Define $x_{curr} = \hat{x}$
Define $minCost = cost(\hat{x})$

for $i = 1 \ldots maxIter$ **do**
  $X = n$ samples from cluster-specific Gaussian
  $\bar{X}_i = x_{curr} + \alpha \cdot (X_i - x_{curr}) \ \forall i$
  $C =$ Evaluate (2.7) for all $\bar{X}_i$ ignoring global features
  $C_G =$ Evaluate (2.7) for k lowest cost samples in C
  if $min(C_G) < minCost$ **then**
    $x_{curr} =$ lowest cost sample from C_G
    $minCost = min(C_G)$
  else
    $\alpha = \alpha * \beta$
  end if
end for
output: x_{curr}

for a large number of samples, evaluate the cost function ignoring the terms involving dense global features, and (2) obtain a small subset of the best scoring samples from step (1) and rescore them using the full cost function. In this way, we can obtain a better solution by pruning the space using a computationally efficient estimation of the cost function, and later rescoring a small set of samples using the full cost function to find the best sample for our final result.

2.5.3 Experiments

In this section, we describe the experimental evaluation of our memorability modification algorithm. Specifically, we describe the experimental setup in Section 2.5.3, the results obtained in Section 2.5.3 and additional analysis of our algorithm in Section 2.5.3. Overall, we find that our algorithm modifies 74% of the images accurately, and our result is statistically significant with a p-value of $< 10^{-4}$.

Setup

Our goal is to evaluate whether our algorithm is able to modify the memorability of faces in a predictable way. In order to do this, we use the face memory game [4]: we designed two balanced experiments where, for the first experiment, we increase the memorability of half the target images and decrease the memorability of the other half, and vice versa for the other (modified versions of the same target images are used in both experiments). Then we compare the memorability scores of the modified images; if the mean memorability of the set of images whose memorability was increased is higher than the decreased set, we can conclude that our algorithm is accurately modifying memorability.

Specifically, we randomly sample 505 of the 2222 target images from the 10k US Adult Faces Database and use them as targets, and the remaining as fillers in our experiments. We ensure that the set of participants in the two studies is disjoint, i.e., a participant does not see both high/low modifications of a single target. On average, we obtained 63 scores per image in each of the experiments.

Algorithmic details: We set the hyperparameters $\lambda = 10$, and $\gamma = 1$ and use the PCA features together with dense HOG2x2 [24] and SIFT [99] features as described in Section 2.5.1. The target images were modified to have a memorability score that differs by 0.2 from the original in the appropriate direction. To account for warping effects, the filler images are also modified by random scores in the range $[-0.1, 0.1]$ and are identical for both sets of experiments.

Memorability Modification Results

Figure 2-16 summarizes the quantitative results from the memorability games described in Section 2.5.3. In Figure 2-16(a), we show the overall memorability scores of all target images after the two types of modifications (i.e., memorability increase and decrease) sorted independently. We observe that the mean memorability score 'memorability increase' images is significantly higher than that of the 'memorability decrease' images. We perform a one-tailed paired sample t-test and find that the null hypothesis that the means are equal is rejected with a p value of $< 10^{-4}$ for both sets of experiments, indicating that our results are statistically significant.

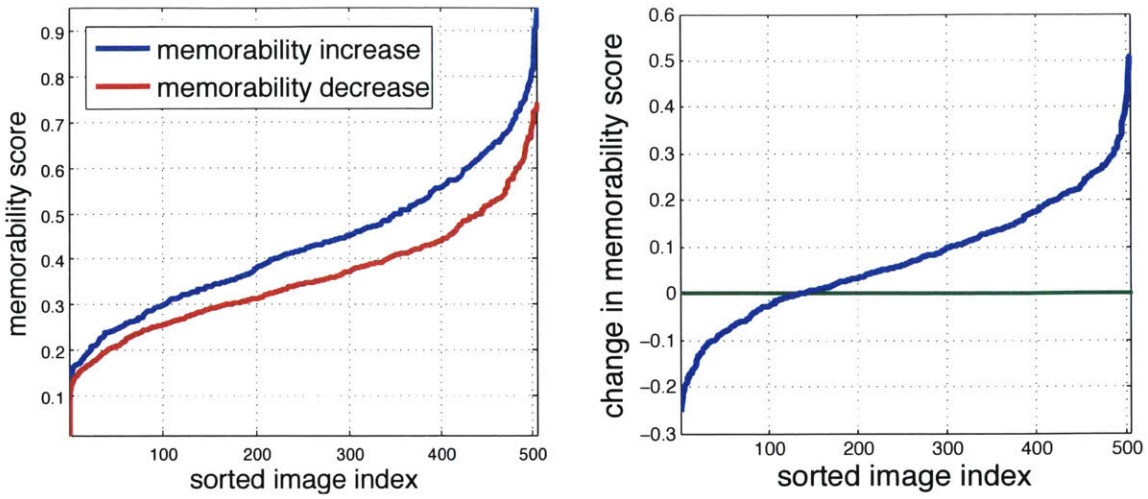
Figure 2-16(b) shows the difference in memorability scores of individual images; for a given image we subtract the observed memorability of the version modified to have lower memorability image from that of the version modified to have higher memorability. We find that the expected change in memorability (> 0) occurs in about 74% of the images (chance is 50%). This is a fairly high value given our limited understanding of face memorability and the factors affecting it. We also observe that the increase in memorability scores is much larger in magnitude than the decrease.

Figure 2-18 shows qualitative results of modifying images to have higher and lower memorability, together with the memorability scores obtained from our experiments. While we observe that the more memorable faces tend to be more 'interesting', there is no single modification axis such as distinctiveness, age, etc, that leads to more or less memorable faces. Essentially, our data-driven approach is effectively able to identify the subtle elements of a face that affect its memorability and apply those effects to novel faces.

Analysis

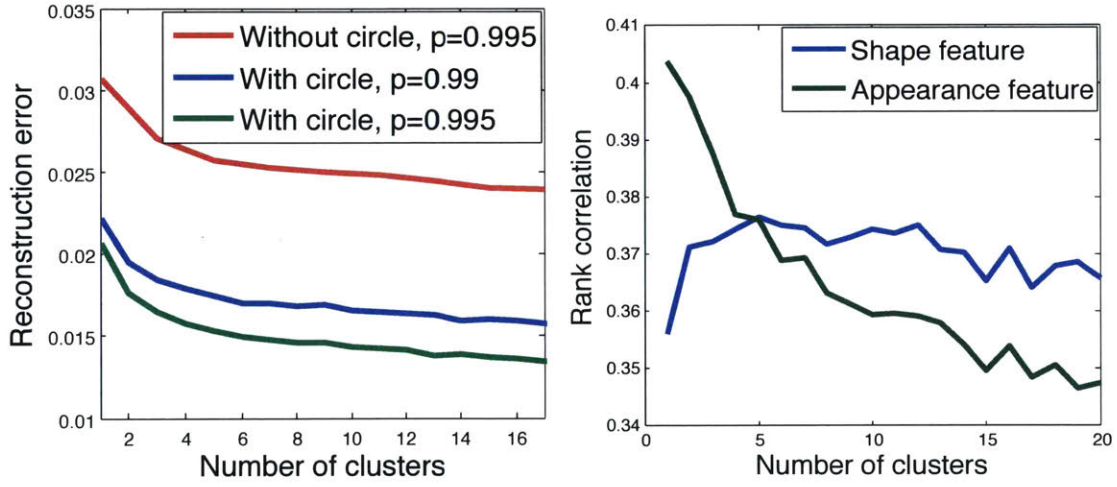
To investigate the contribution of shape and appearance features to face memorability, we conduct a second AMT study similar to the one described in Section 2.5.3, except that we only modify shape features in this case. We found that the accuracy of obtaining the expected change in memorability, as described in Section 2.5.3, dropped to 60%. In addition, the changes in memorability scores were not as significant in this case as compared to the original setting. This shows that a combination of shape and appearance features are important for modifying memorability; however, it is interesting to note that despite the limited degree of freedoms, our algorithm achieved a reasonable modification accuracy.

Figure 2-17 shows the effect of having clusters in the AAM as described in Section 2.5.2. We find that having more clusters allows us to have better reconstructions without significant sacrifice in memorability prediction performance. Thus, we choose to use 8 clusters in our experiments. Lastly, since changes in memorability lead to unintuitive modifications to faces, in Figures 2-19, 2-20, 2-21 and 2-22 we apply our algorithm to modify other attributes whose effects are better understood. Indeed, we observe that the algorithm is behaving as per our expectations.



(a) Overall memorability scores (b) Individual images

Figure 2-16: **Quantitative results:** (a) Memorability scores of all images in the increase/decrease experimental settings, and (b) change in memorability scores of individual images.



(a) Reconstruction error (b) Memorability prediction

Figure 2-17: **Analysis:** Figure showing (a) reconstruction error, and (b) memorability prediction performance as we change the number of clusters in AAM. With and without circle refers to having control points on the image boundary when doing warping.

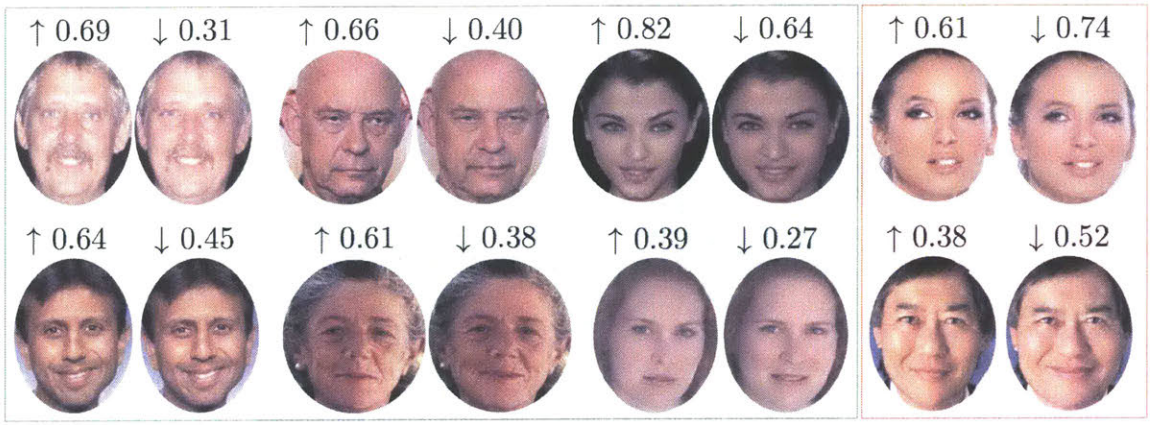


Figure 2-18: **Visualizing modification results:** Figure showing success (green background) and failure (red background) cases of the modification together with memorability scores from human experiments. Arrow direction indicates which face is expected to have higher or lower memorability of the two while numbers indicate the actual memorability scores.

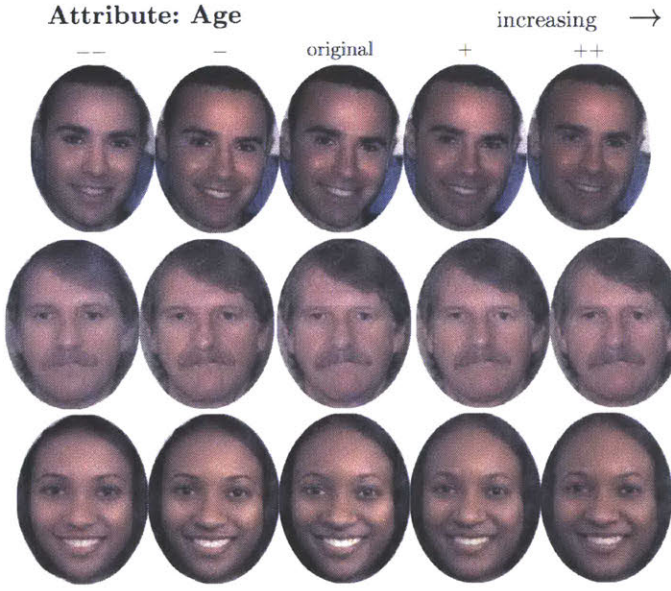


Figure 2-19: Figure showing the modification of age attribute to different extents, increasing to the right. The age is decreased for images to the left of the original image, and increased to the right. We observe that our results tend to reflect our intuition such as removal of wrinkles for people becoming younger and gray hair for people getting older. Further, the facial features move to reflect this effect.

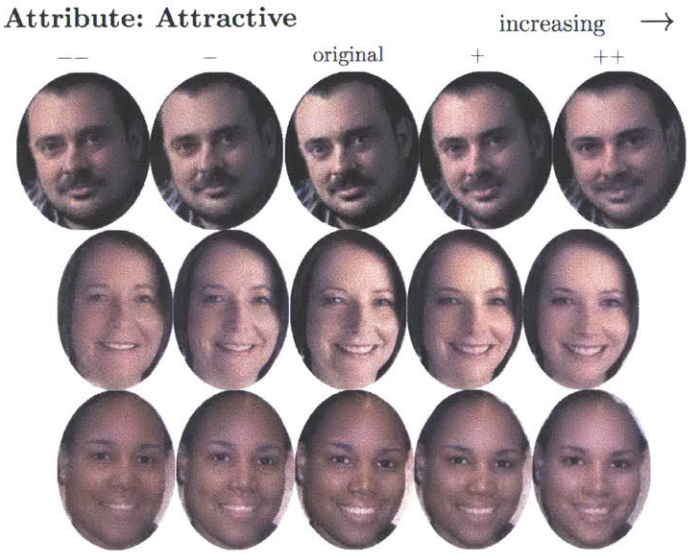


Figure 2-20: Figure showing the modification of attractiveness attribute to different extents, increasing to the right. The attractiveness is decreased for images to the left of the original image, and increased to the right. We observe that our results tend to reflect our intuition.

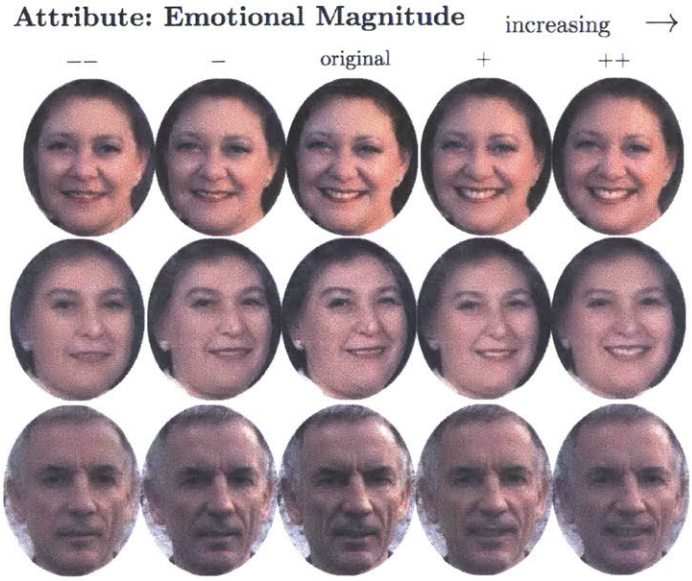


Figure 2-21: Figure showing the modification of emotional magnitude attribute to different extents, increasing to the right. The emotional magnitude is decreased for images to the left of the original image, and increased to the right. We observe that our results tend to reflect our intuition.

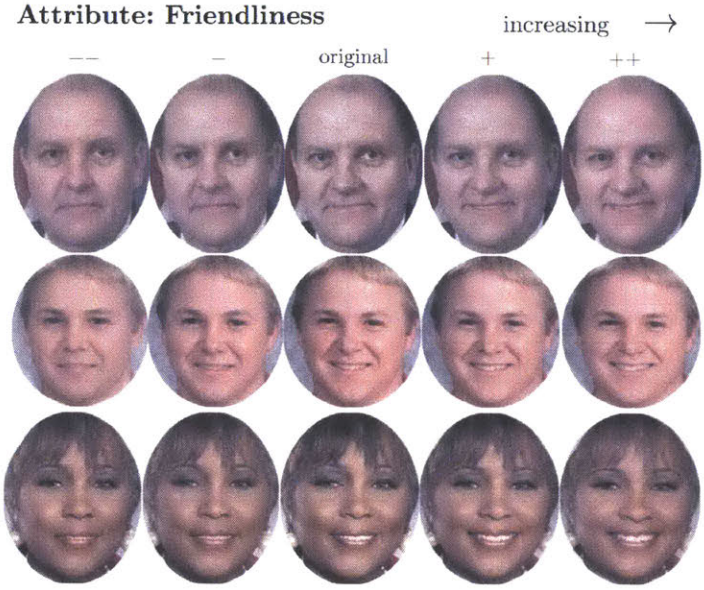


Figure 2-22: Figure showing the modification of friendliness attribute to different extents, increasing to the right. The friendliness is decreased for images to the left of the original image, and increased to the right. We observe that our results tend to reflect our intuition.

	Chance	Color 200	HOG 1024	LBP 1239	SIFT 1024	SSIM 1024	Shape 154
age	0	0.210	0.722	0.640	**0.773**	0.685	0.680
attractive	0	0.290	0.593	0.505	**0.616**	0.565	0.543
celeb	0.928/3	0.928	0.928	0.928	0.928	0.928	0.928
common	0	0.113	**0.265**	0.215	0.261	0.262	0.216
emoteMag	0	0.134	0.803	0.642	0.831	0.646	**0.881**
emotion	0.680/7	0.670	0.843	0.779	0.850	0.735	**0.869**
eyes	0.912/5	0.912	0.913	0.912	0.912	0.912	**0.916**
face	0.818/5	0.818	**0.911**	0.861	0.906	0.829	0.910
facialhair	0.780/3	0.778	**0.836**	0.831	0.824	0.783	0.787
friendly	0	0.154	0.772	0.636	0.804	0.634	**0.851**
makeup	0.636/3	0.701	0.836	0.806	**0.837**	0.804	0.804
male	0.569/2	0.721	0.927	0.889	**0.935**	0.907	0.906
movie	0	0.244	0.476	0.411	**0.484**	0.436	0.406
profilepic	0.626/3	0.618	0.717	0.677	**0.720**	0.671	0.714
quality	0	0.273	0.420	0.472	0.445	**0.490**	0.241
race	0.825/7	0.828	0.866	0.848	0.871	0.828	**0.883**
remember	0	0.212	**0.428**	0.395	0.425	0.400	0.348
speed	0	0.062	0.690	0.547	0.722	0.556	**0.777**
teeth	0.360/3	0.408	0.707	0.579	0.720	0.581	**0.770**
memorability	0	0.12	**0.51**	0.23	0.49	0.47	0.37

Table 2.4: **Prediction performance of memorability and other attributes:** The number below the feature name denotes the feature dimension (for LBP [112] and Shape) or dictionary size (for Color [158], HOG [24], SIFT [99] and SSIM [138]). For real-valued attributes and memorability, we report Spearman's rank correlation (ρ), while for discrete valued attributes such as 'male', we report classification accuracy. For chance performance, there are two cases: (1) it is 0 when rank correlation is used, or (2) it is non-zero when accuracy is used, where the first number denotes the chance performance obtained by picking the class with the largest number of examples in the training set, and the number after the slash denotes the number of classes for the particular task. We use a linear SVM or SVR [37] for training, and the above results are reported on 25 random train/test trials where the train and test sets are of equal size. The features are assigned to a dictionary using Locality-Constrained Linear Coding [161] and max-pooled at 2 pyramid levels [90]. The reported performance is averaged on 25 random train/test splits of the data.

Chapter 3

Predicting Image Popularity

Over the last decade, online social networks have exploded in terms of number of users, volume of activities, and forms of interaction. In recent years, significant effort has therefore been expended in understanding and predicting online behavior, surfacing important content, and identifying viral items. In this chapter, we focus on the problem of predicting popularity of images.

Hundreds of thousands of photographs are uploaded to the internet every minute through various social networking and photo sharing platforms. While some images get millions of views, others are completely ignored. Even from the same users, different photographs receive different number of views. This begs the question: What makes a photograph popular? Can we predict the number of views a photograph will receive even before it is uploaded?

We investigate two crucial attributes that may affect an image's popularity, namely the image content and social context. In the social context, there has been significant work in viral marketing strategies and influence propagation studies for online social networks [30, 131, 110, 29, 72]. However, most of these works adopt a view where the spread of a piece of content is primarily due to a user viewing the item and potentially sharing it. Such techniques adopt an algorithmic view and are geared towards strategies for maximizing influence. On the contrary, in this work, we use social signals such as number of friends of the photo's uploader, and focus on the *prediction* problem of overall popularity.

Previous works have focused primarily on predicting popularity of text [122, 57] or video [123, 137, 111] based items. Such research has also explored the social context as well as the content of the text itself. However, image content is significantly harder to extract, and correlate with social popularity. Text based techniques can build on the wealth of methods developed for categorizing, NLP, clustering, and sentiment analysis. Comparatively, understanding such cues from image or video content poses new challenges. While there has been some work in video popularity prediction [45, 44], these tend to naturally focus on the social cues, comment information, and associated tags. From this standpoint, our work is the first to suitably combine contextual information from the uploader's social cues, and the content-based features, for images.

In order to obtain the image content features, we apply various techniques from computer vision and machine learning. While there has been a significant push in the computer vision community towards detecting objects [42, 24], identifying contextual relationships [125, 153], or classifying scenes [114, 90], little work has been expended towards associating key image components with 'global spread' or popularity in an online social platform. This is perhaps the first work that leverages image cues such as color histograms, gradient histograms, texture and objects in an image for ascertaining their predictive power towards popularity. We demonstrate through extensive exploration the independent benefits of such image cues, as well as social cues, and highlight the insights that can be drawn from either. We further show these cues combine effectively towards an improved popularity prediction algorithm. Our experiments illustrate several benefits of these two types of features, depending on the data-type distributions.

Using a dataset of millions of images from Flickr, we demonstrate that we can reliably predict the normalized view count of images with a rank correlation of up to 0.81 using both image content and social cues. We consider tens of thousands of users and perform extensive evaluation based on prediction algorithms applied to three different settings: *one-per-user*, *user-mix*, *user-specific*. In each of these cases, we vary the number of users, and the number of images per user. In each of these

cases, the relative importance of different attributes are presented and compared against several baselines. We identify insights from our method that open-up several directions for further exploration.

We briefly summarize the **main contributions** of this chapter in the following:

- We initiate a study of popularity prediction for images uploaded on social networks on a massive dataset from Flickr. Our work is one of the first to investigate high-level and low-level image features and combine them with the social context towards predicting popularity of photographs.

- Combing various features, we present an approach that obtains more than 0.8 rank correlation on predicting normalized popularity. We contrast our prediction technique that leverages social cues and image content features with simpler methods that leverage color spaces, intensity, and simple contextual metrics. Our techniques highlight the importance of low-level computer vision features and demonstrate the power of certain semantic features extracted using deep learning.

- We investigate the relative importance of individual features, and specifically contrast the power of social context with image content across three different dataset types - one where each user has only one image, another where each user has several thousand images, and a third where we attempt to get specific predictors for users separately. This segmentation highlights benefits derived from the different signals and draws insights into the contributions of popularity prediction in comparison to simpler baseline techniques.

- As an important contribution, this work opens the doors for several interesting directions to pursue image popularity prediction in general, and pose broad social questions around online behavior, popularity, and causality. For example, while our work attempts to disentangle social and content based features, and derives new insights into their predictive power, it also begs the question on their impacts influencing each other through self-selection. Our work sheds

some light on such interesting relations between features and popularity, but also poses several questions.

3.1 Related Work

Popularity prediction in social media has recently received a lot of attention from the research community. While most of the work has focused on predicting popularity of text content, such as messages or tweets on Twitter [122, 57], and some recent works on video popularity [123, 137, 111], significantly less effort has been expended in prediction of image popularity. The challenge and opportunity for images comes from the fact that one may leverage both social cues (such as the user's context, influence etc. in the social media platform), as well as image-specific cues (such as the color spectrum, the aesthetics of the image, the quality of the contrast etc.). Text based popularity prediction has of course leveraged the social context, as well as the content of the text itself. However, image content can be significantly harder to extract, and correlate with popularity.

Recently, there has been an increasing interest in analyzing various semantic attributes of images. One such attribute is image memorability which has been shown to be an intrinsic image property [63] with different image regions contributing differently to an image's memorability [80, 79]. Similarly, image quality and aesthetics are other attributes that have been recently explored in substantial detail [25, 28, 14]. Recent work has also analyzed the more general attribute of image intrestingness [155], particularly focusing on its correlation with image memorability [50]. There have also been a variety of other works dealing with facial [88], scene [120] and object [39] attributes.

In social context, there has been significant interest in understanding behavioral aspects of users online and in social networks. There is a large body of work studying the correlation of activity among friends in online communities; see examples in [48, 132, 133]. Most are forms of diffusion research, built on the premise that user engagement is contagious. As such, a user is more likely to adopt new products or

Figure 3-1: Sample images from our image popularity dataset. The popularity of the images is sorted from more popular (left) to less popular (right).

behaviors if their friends do so [3, 93]; and large cascades of behavior can be triggered by the actions of a few individuals [48, 132]. A number of theoretical models have been developed to model influence cascades [110, 29]. The seminal work of Kempe et al. [72] also consider two models of influence; there has also been a long line of work on viral marketing starting from [30, 131]. All these works focus on influence but do not necessarily predict popularity beforehand - they are generic techniques that do not focus on a specific domain such as images in our context.

It would be interesting to explore emotions elicited from images, and their causal influence on image popularity on social networks - some interesting studies in other domains include [70, 82].

In the context of social networks, there has been a great deal of focus on understanding the effects of friends on behavior. In a very interesting piece of work, Crandall et al. [23] consider the two documented phenomenon, social influence and selection, in conjunction. They [23] suggest that users have an increased tendency to conform to their friends' interests. Another related work [85] considers homophily in social networks: they conclude that users' preference to similar people is amplified over time, due to biased selection. An interesting open question in our context is whether users, over time, have an increasing tendency to share images that cater to their friends' preferences - and if yes, how this affects overall popularity. As such, our work does not delve into these social aspects. We focus primarily on the task of popularity prediction. It would be interesting to further explore the contributions of homophily to virality of images.

3.2 What is image popularity?

There are various ways to define the popularity of an image such as the number of 'likes' on Facebook, the number of 'pins' on Pinterest or the number of 'diggs' on Digg. It is difficult to precisely pick any single one as the true notion of popularity - different factors are likely to impact these different measures of popularity in different contexts. In this work, we focus on the *number of views* on Flickr as our medium for exploring image popularity. Given the availability of a comprehensive API that provides a host of information about each image and user, together with a well established social network with significant public content, we are able to conduct a relatively large-scale study with millions of images and hundreds of thousands of users.

Figure 3-2(a) shows the histogram of the number of views received by the 2.3 million(M) images used in our study. Our dataset not only contains images that have received millions of views but also plenty of images that receive zero views. To deal with the large variation in the number of views of different images, we apply the log function as shown in Figure 3-2(b). Furthermore, as shown in [147], we know that unlike Digg, visual media tends to receive views over some period of time. To normalize for this effect, we divide the number of views by the duration since the upload date of the given image (obtained using Flickr API). The results are shown in Figure 3-2(c). We find that this resembles a Gaussian distribution of the view counts as one would expect. Throughout the rest of this chapter, image popularity refers to this log-normalized view count of images.

In the following, we provide details regarding the datasets used (Section 3.2.1), and the evaluation metric (Section 3.2.2) for predicting image popularity.

3.2.1 Datasets

Figure 3-1 shows a sample of the images in our dataset. In order to explore different data distributions that occur naturally in various applications and social networks,

[1]Note that the maximum view count of a single image in our dataset is 2.5M, but we truncate the graph on the left to amplify the remaining signal. Despite this, it is difficult to see any clear signal as most images have very few views. This graph is best seen on the screen with zoom.

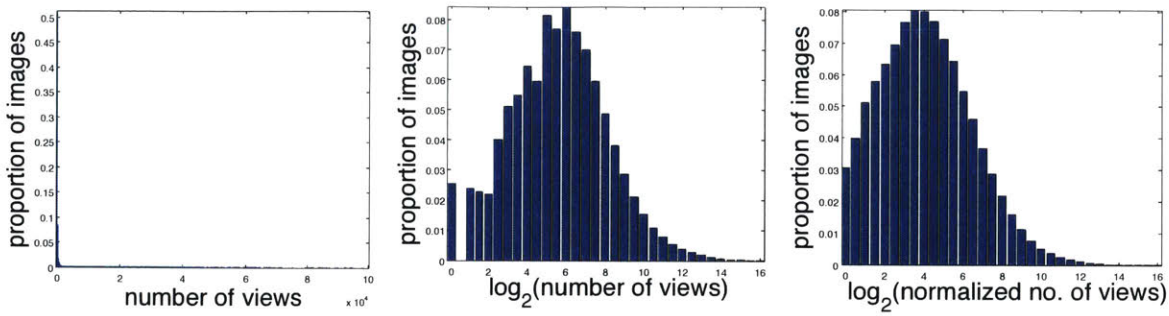


Figure 3-2: Histogram of view counts of images. The different graphs show different transformations of the data: (left) absolute view counts[1], (middle) $\log_2$ of view counts +1 and (right) $\log_2$ view counts +1 normalized by upload date.

we evaluate our algorithms in 3 different settings namely *one-per-user*, *user-mix*, and *user-specific*. The main components that we vary across these settings is the number of images per user in the dataset, and whether we perform user-specific predictions. These settings are described below. In the later sections, we will show the importance of splitting image popularity into these different settings by illustrating how the relative contribution of both content-based and social features changes across these tasks.

One-per-user: For this setting, we use the Visual Sentiment Ontology dataset [14] consisting of approximately 930k images from about 400k users, resulting in a little over two images from each user on average. This dataset was collected by searching Flickr for 3244 adjective-noun-pairs such as 'happy guy', 'broken fence', 'scary cat', etc corresponding to various image emotions. This dataset represents the setting where different images belong to different users. This is often the case in search results.

User-mix: For this setting, we randomly selected about 100 users from the one-per-user dataset that had between 10k and 20k public photos shared on Flickr, resulting in a dataset of approximately 1.4M images. In this setting, we put all these images from various users together and perform popularity prediction on the full set. This setting often occurs on newsfeeds where people see multiple images from their own contacts or friends.

User-specific: For this setting, we split the dataset from the user-mix setting into

100 different users and perform training and evaluation independently for each user and average the results. Thus, we build user-specific models to predict the popularity of different images in their own collections. This setting occurs when users are taking pictures or selecting pictures to highlight - they want to pick the images that are most likely to receive a high number of views.

3.2.2 Evaluation

For each of the settings described above, we split the data randomly into two halves, one for training and the other testing. We average the performance over 10 random splits to ensure the consistency of our results; overall, we find that our results are highly consistent with low standard deviations across splits. We report performance in terms of Spearman's rank correlation (ρ) between the predicted popularity and the actual popularity. Note that we use log-normalized view count of images as described in the beginning of this section for both training and testing. Additionally, we found that rank correlation and standard correlation give very similar results.

3.3 Predicting popularity using image content

In this section, we investigate the use of various features based on image content that could be used to explain the popularity of images. First, in Section 3.3.1 we investigate some simple human-interpretable features such as color and intensity variance. Then, in Section 3.3.2, we explore some low-level computer vision features inspired by how humans perceive images such as gradient, texture or color patches. Last, in Section 3.3.3, we explore some high-level image features such as the presence of various objects. Experimental results show that low-level computer vision features and high-level semantic features tend to be significantly more predictive of image popularity than simple image features.

3.3.1 Color and simple image features

Are simple image features enough to determine whether or not an image will be popular? To address this, we evaluate the rank correlation between popularity and basic pixel statistics such as the mean value of different color channels in HSV space, and intensity mean, variance, and skewness. The results are shown in Figure 3-3. We find that most simple features have very little correlation with popularity, with mean saturation having the largest absolute value of 0.05. Here, we use all 2.3M images independent of the settings described in Section 3.2.1. Since significant correlation does not exist between simple image features and popularity, we omit results from the individual settings for brevity.

Additionally, we look at another simple feature: the color histogram of images. As the space of all colors is very large ($256 * 256 * 256 \approx 16.8$m), and since small variations in lighting and shadows can drastically change pixel values, we group the color space into 50 distinct colors as described in [74] to be more robust to these variations. Then, we assign each pixel of the image to one of these 50 colors and form a ℓ_1-normalized histogram of colors. Using support vector regression (SVR) [32] with a linear kernel (implemented using LIBLINEAR [37]), we learn the importance of these colors in predicting image popularity[2]. The results are shown in Table 3.1 (column: *color histogram*), and visualized in Figure 3-4. Despite its simplicity, we obtain a rank correlation of 0.12 to 0.23 when using this feature on the three different data settings. We observe that on average, the greenish and bluish colors tend to have lower importance as compared to more reddish colors. This might occur because images containing more striking colors tend to catch the eye of the observer leading to a higher number of views.

While the simple features presented above are informative, more descriptive features are likely necessary to better represent the image in order to make better predictions. We explore these in the remaining part of this section.

[2]We find the hyperparameter $C \in \{0.01, 0.1, 1, 10, 100\}$ using five-fold cross-validation on the training set.

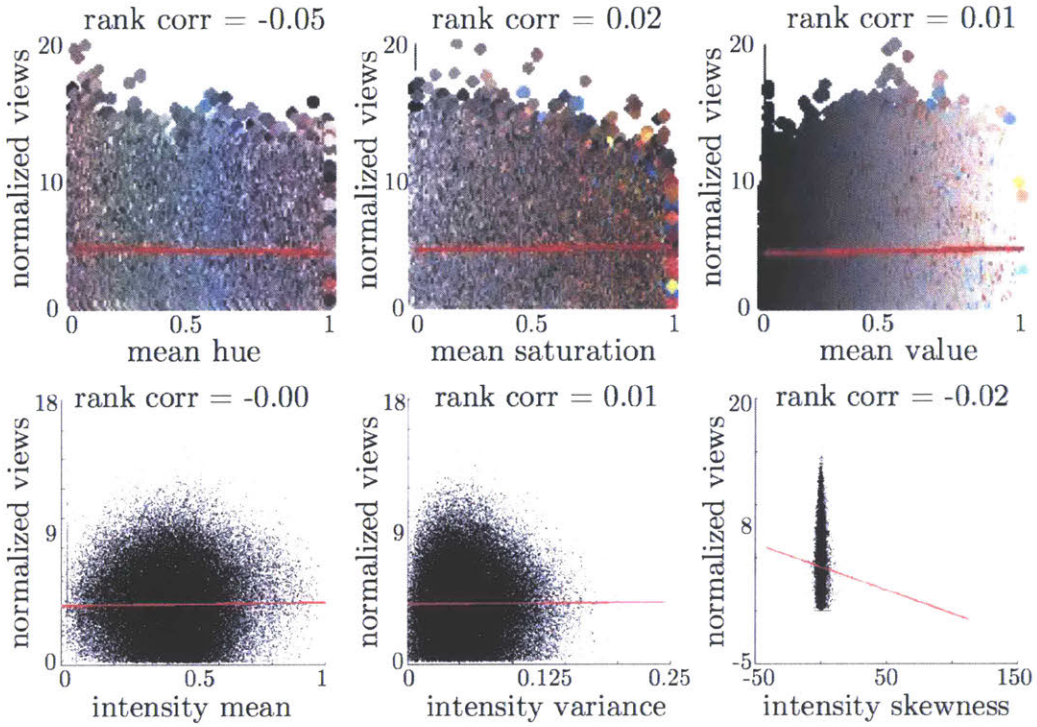


Figure 3-3: Correlation of popularity with different components of the HSV color space (top), and intensity statistics (bottom).

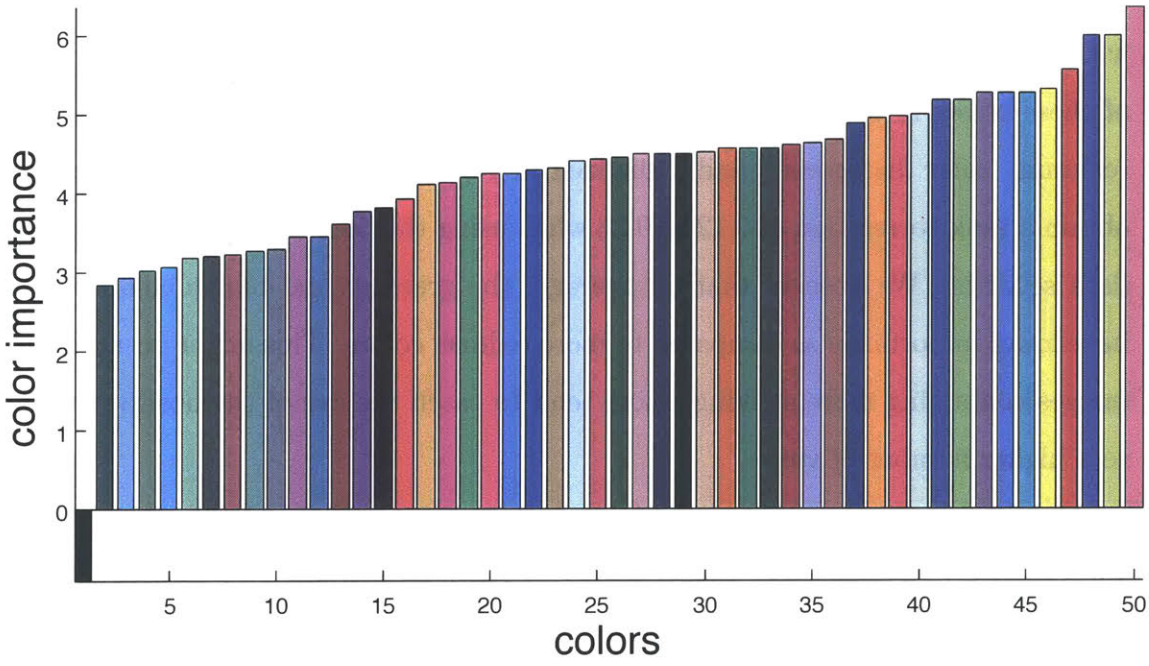


Figure 3-4: Importance of different colors to predict image popularity. The length of each bar shows importance of the color shown on the bar.

Dataset	Gist	Color histogram	Texture	Color patches
One-per-user	0.07	0.12	0.20	0.23
User-mix	0.13	0.15	0.22	0.29
User-specific	0.16	0.23	0.32	0.36

Dataset	Gradient	Deep learning	Objects	Combined
One-per-user	0.26	0.28	0.23	**0.31**
User-mix	0.32	0.33	0.30	**0.36**
User-specific	0.34	0.26	0.33	**0.40**

Table 3.1: Prediction results using image content only as described in Section 3.3.

3.3.2 Low-level computer vision features

Motivated by Khosla et al. [80, 75], we use various low-level computer vision features that are likely used by humans for visual processing. In this work, we consider five such features namely gist, texture, color, gradient and deep learning features. For each of the features, we describe our motivation and the method used for extraction below.

Gist: Various experimental studies [124, 9] have suggested that the recognition of scenes is initiated from the encoding of the global configuration, or spatial envelope of the scene, overlooking all of the objects and details in the process. Essentially, humans can recognize scenes just by looking at their 'gist'. To encode this, we use the popular GIST [114] descriptor with a feature dimension of 512.

Texture: We routinely interact with various textures and materials in our surroundings both visually, and through touch. To test the importance of this type of feature in predicting image popularity, we use the popular Local Binary Pattern (LBP) [112] feature. We use non-uniform LBP pooled in a 2-level spatial pyramid [90] resulting in a feature of 1239 dimensions.

Color patches: Colors are a very important component of human visual system for determining properties of objects, understanding scenes, etc. The space of colors tends to have large variations by changes in illumination, shadows, etc, and these variations make the task of robust color identification difficult. While difficult to work with, various works have been devoted to developing robust color descriptors [158, 74], which have been proven to be valuable in computer vision for various tasks including

image classification [73]. In this paper, we use the 50 colors proposed by [74] in a bag-of-words representation. We densely sample them in a grid with a spacing of 6 pixels, at multiple patch sizes (6, 10 and 16). Then we learn a dictionary of size 200 and apply LLC [161] together with max-pooling in a 2-level spatial pyramid [90] to obtain a final feature vector of 4200 dimensions.

Gradient: In the human visual system, much evidence suggests that retinal ganglion cells and cells in the visual cortex V1 are essentially gradient-based features. Furthermore, gradient based features have been successfully applied to various applications in computer vision [24, 42]. In this work, we use the powerful Histogram of Oriented Gradient (HOG) [24] features combined with a bag-of-words representation for popularity prediction. We sample them in a dense grid with a spacing of 4 pixels for adjacent descriptors. Then we learn a dictionary of size 256, and apply Locality-Constrained Linear Coding (LLC) [161] to assign the descriptors to the dictionary. We finally concatenate descriptors from multiple image regions (max-pooling + 2-level spatial pyramid) as described in [90] to obtain a final feature of $10,752$ dimensions.

Deep learning: Deep learning algorithms such as convolutional neural networks (CNNs) [92] have recently become popular as methods for learning image representations [87]. CNNs are inspired by biological processes as a method to model the neurons in the brain, and have proven to generate effective representation of images. In this paper, we use the recently popular 'ImageNet network' [87] trained on 1.3 million images from the ImageNet [27] challenge 2012. Specifically, we use Decaf [31] to extract features from the layer just before the final 1000 class classification layer, resulting in a feature of 4096 dimensions.

Results: As described in Section 3.3.1, we train a linear SVR to perform popularity prediction on the different datasets. The results averaged over 10 random splits are summarized in Table 3.1. As can be seen from any of the columns, the rank correlation performance is best for the user-specific dataset, and next for user-mix, followed by one-per-user. However, it is important to note that the prediction accuracy when all the features are combined (as seen in the last column in Table 3.1) is at least 0.30 in all three cases. Given that these results only leverage image-content fea-

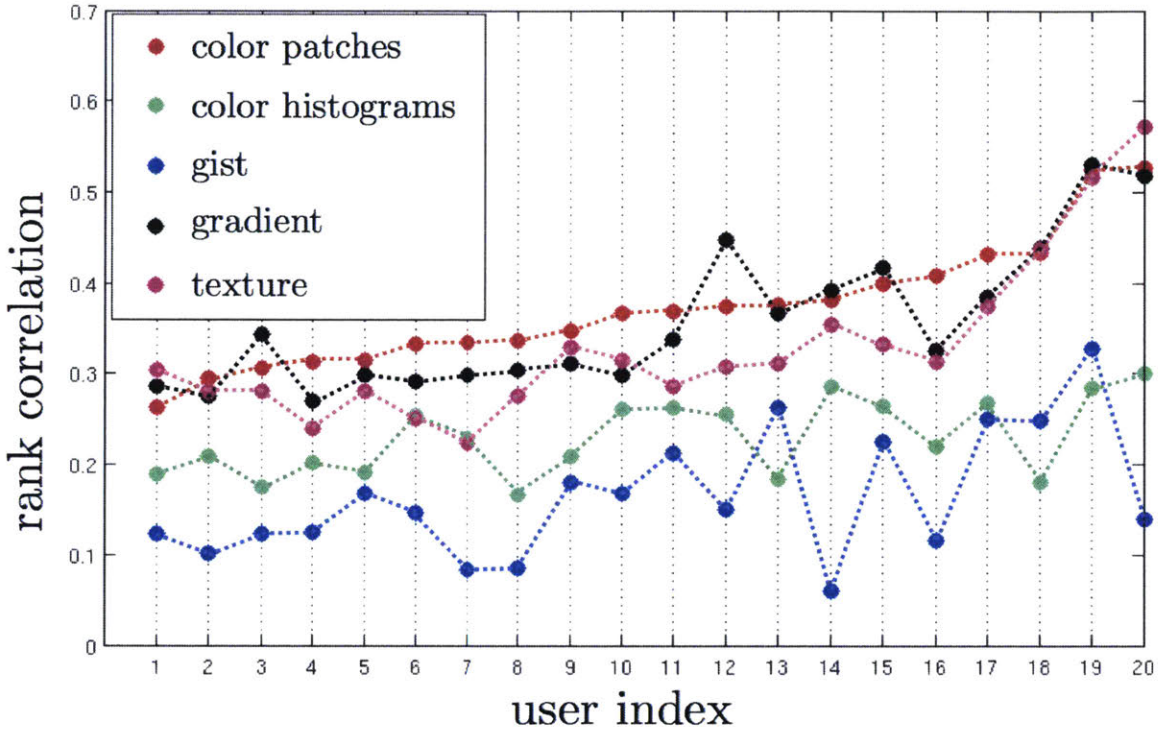


Figure 3-5: Prediction performance of different features for 20 random users from the *user-specific* setting. This figure is best viewed in color.

Dataset	Mean Views	Photo Count	Contacts	Groups	Group members	Member duration
One-per-user	0.75	0.07	0.46	0.27	0.27	-0.08
User-mix	0.62	-0.05	0.26	0.24	0.06	-0.01
User-specific	n/a	n/a	n/a	n/a	n/a	n/a

Dataset	Is pro?	Tags	Title length	Desc. length	All
One-per-user	0.22	0.52	0.21	0.45	**0.77**
User-mix	0.07	0.40	0.29	0.35	**0.66**
User-specific	n/a	0.19	0.16	0.19	**0.21**

Table 3.2: Prediction results using social content only as described in Section 3.4.

tures (therefore ignore any aspects of the social platform or user-specific attributes), the correlation is very significant. While most of the current research focuses solely on the social network aspect when predicting popularity, this result suggests that the content also plays a crucial role, and may provide complementary information to the social cues.

On exploring the columns individually in Table 3.1, we notice that the color histogram alone gives a fairly low rank correlation (ranging between 0.12 and 0.23 across the three datasets), but texture, and gradient features perform significantly better (improving the performance ranges to 0.20 to 0.32 and 0.26 to 0.34 respectively). The deep learning features outperform other features for the *one-per-user* and *user-mix* settings but not the *user-specific* setting. We then combine features by training a SVR on the output of the SVR trained on the individual features. This finally pushes the range of the rank correlations to 0.31 to 0.40.

In conjunction with the aggregate evaluation metrics described above and presented in Table 3.1, it is informative to look at the specific performance across 20 randomly selected users from the *user-specific* setting for each of these features - this is presented in a rank correlation vs. user index scatter plot in Figure 3-5. The performance for all features combined is not displayed here to demonstrate the specific variance across the image cues. As can be seen here, the color patches (red dots) performs best nearly consistently across all the users. For a couple of users, the gradient (black dots) performs better, and for others performs nearly as well as the color patches. In both these features, the prediction accuracy is fairly consistent. On the other hand, gist has a larger variance and the lowest average rank correlation. From the figure, it can be seen that texture (pink dots) also plays a vital role in predicting rank correlation accuracy, as it spans the performance range of roughly 0.22 to 0.58 across the randomly sampled users.

Another point of Figure 3-5 is to show that for personalization (i.e. user specific experiments), different features may be indicative of what a user's social network likes (because that is essentially what the model will capture by learning a user-specific model based on the current set of images). It is therefore interesting to note some

variation in importance or relative ranking of features across the sampled users.

3.3.3 High-level features: objects in images

In this section, we explore some high-level or semantically meaningful features of images, namely the objects in an image. We want to evaluate whether the presence or absence of certain objects affects popularity e.g. having people in an image might make it more popular as compared to having a bottle. Given the scale of the problem, it is difficult and costly to annotate the dataset manually. Instead, we can use a computer vision system to roughly estimate the set of objects present in an image. Specifically, we use the deep learning classifier [87] described in the previous subsection that distinguishes between 1000 object categories ranging from lipstick to dogs to castle. In addition, this method achieved state-of-the-art classification results in the ImageNet classification challenge demonstrating its effectiveness in this task. We treat the output of this 1000 object classifier as features, and train a SVR to predict log-normalized image popularity as done in the previous sections. The results are summarized in Table 3.1.

We observe that the presence or absence of objects is a fairly effective feature for predicting popularity for all three settings with the best performance achieved in *user-specific* case. Further, we investigate the type of objects leading to image popularity. On the *one-per-user* dataset, we find that some of the most common objects present in images are: seashore, lakeside, sandbar, valley, volcano. In order to evaluate the correlation of objects with popularity, we compute the mean of the SVR weights across the 10 train/test splits of the data, and sort them. The resulting set of objects with different impact on popularity is as follows:

- **Strong positive impact:** miniskirt, maillot, bikini, cup, brassiere, perfume, revolver
- **Medium positive impact:** cheetah, giant panda, basketball, llama, plow, ladybug
- **Low positive impact:** wild boar, solar dish, horse cart, guacamole, catamaran

- **Negative impact:** spatula, plunger, laptop, golfcart, space heater

It is interesting to observe that this is similar to what we might expect. It is important to note that this object classifier is not perfect, and may often wrongly classify images to contain certain objects that they do not. Furthermore, there may be certain object categories that are present in images but not in the 1000 objects the classifier recognizes. This object-popularity correlation might therefore not pick up on some important object factors. However in general, our analysis is still informative and intuitive about what type of objects might play a role in a picture's popularity

3.4 Predicting popularity using social cues

While image content is useful to predict image popularity to some extent, social cues play a significant role in the number of views an image will receive. A person with a larger number of contacts would naturally be expected to receive a higher number of average views. Similarly, we would expect that an image with more tags shows up in search results more often (assuming each tag is equally likely). Here, we attempt to quantify the extent to which the different social cues impact the popularity of an image.

For this purpose, we consider several user-specific or social context specific features. We refer to user features as ones that are shared by all images of a single user. The user features that we use in our analysis are listed and described below. Note that this work investigates the relative merits of social and image features, and the goal isn't to heavily exploit one. Thus, we use relatively simple features for social cues that could likely be improved by using more sophisticated approaches.

- **Mean Views**: mean of number of normalized views of all public images of the given user
- **Photo count**: number of public images uploaded by the given user
- **Contacts**: number of contacts of the given user

- **Groups**: number of groups the given user belongs to
- **Group members**: average number of members in the groups a given user belongs to
- **Member duration**: the amount of time since the given user joined Flickr
- **Is pro**: whether the given user has a Pro Flickr account or not

We further subdivide some of the above features such as 'groups' into number of groups a given user is an administrator of, and the number of groups that are 'invite only'. We also include some image specific features that refer to the context (i.e. supporting information associated with the image as entered by the user but not its pixel content, as explored in Section 3.3). These are listed below.

- **Tags**: number of tags of the image
- **Title length**: length of the image title
- **Desc. length**: length of the image description

For each of the above features, we find its rank correlation with log-normalized image popularity. The results are shown in Table 3.2. Note that the user features such as Mean Views and Contacts would have the same value for all images by the particular user in the dataset. Not surprisingly, in the *one-per-user* dataset, the Mean Views feature performs extremely well in predicting the popularity of a new image with a rank correlation of 0.75. However, the rank correlation drops to 0.62 on the *user-mix* setting because there is no differentiation between a user's photos.

That said, 0.75 rank correlation is still a significantly better performance than we had expected since we note that the mean of views was taken over *all* public photos of the given user, not just the ones in our dataset. The mean number of public photos per user is over 1000, and of these, typically 1-2 are in our dataset, so this is a fairly interesting observation.

Another noteworthy feature is contacts - we see a rank correlation for these two datasets to be 0.46 and 0.26 respectively i.e. the more contacts a user has, the

higher the popularity of their images. This is to be expected as their photos would tend to be highlighted for a larger number of people (i.e. their social network). Further, we observe that the image-specific social features such as tags, title length, and description length are also good predictors for popularity. As we can see in Table 3.2, their performance across the three dataset types range from 0.19 to 0.52 for tags, and 0.19 to 0.45 for description length. This is again to be expected as having more tags or a longer description/title increases the likelihood of these images appearing in the search results.

Further, we combine all the social features by training a SVR with all the social features as input. The results are shown in the rightmost column of Table 3.2. In this case, we see a rank correlation of 0.21, 0.66, and 0.77 for the *user-specific*, *user-mix*, and *one-per-user* datasets respectively. Thus, it is helpful to combine the social features, but we observe that the performance does not increase very significantly as compared to the most dominant feature. This suggests that many of these features are highly correlated and do not provide complementary information.

To contrast the results of social features from Table 3.2 with the image content features presented in the previous section in Table 3.1, we observe that the social features tend to perform better in the *one-per-user* and *user-mix* dataset types, while the image content features perform better in the *user-specific* dataset type. We suspect that in the user-specific dataset, where each user has thousands of images, the importance of personalized social cues becomes less relevant (perhaps due to the widespread range of images uploaded by them) and so the image content features become particularly relevant.

One thing that is evident from these results is that the social features and image content features are both necessary, and offer individual insights that are not subsumed by each other i.e. the choice of features for different applications largely depends on the data distribution. In the following section, we investigate techniques combining both of these features that turn out to be more powerful and perform well across the spectrum of datasets.

Dataset	Content only	Social only	Content + Social
One-per-user	0.31	0.77	**0.81**
User-mix	0.36	0.66	**0.72**
User-specific	0.40	0.21	**0.48**

Table 3.3: Prediction results using image content and social cues as described in Section 3.5.1.

3.5 Analysis

In this section, we further analyze some of our results from the previous sections. First, we combine the signal provided by image content and social cues in Section 3.5.1. We observe that both of these modalities provide some complementary signal and can be used together to improve popularity prediction further. In Section 3.5.2 we visualize the good and bad predictions made by our regressors in an attempt to better understand the underlying model. Last, in Section 3.5.3 we show some preliminary visualizations of the image regions that make images popular by reversing the learned weights and applying them to image regions.

3.5.1 Combining image content and social cues

We combine the output of the image content and social cues using a SVR trained on the outputs of the most basic features, commonly referred to as late fusion. Table 3.3 shows the resulting performance. We observe that the performance improves significantly for all 3 datasets as compared to using either sets of features independently. The smallest increase of 0.04 rank correlation is observed for the *one-per-user* dataset as it already has a fairly high rank correlation largely contributed by the social features. This makes it difficult to improve performance further by using content features, but it is interesting to observe that we can predict the number of views in this setting with a fairly high rank correlation of 0.81. We observe the largest gains in the *user-specific* dataset, of 0.08 rank correlation, where image content features play a much bigger role as compared to social features.

3.5.2 Visualizing results

In Figure 3-6, we visualize some of the good and bad predictions made using our regressors. We show the four main quadrants: two with green background where the high or low popularity prediction matches the ground truth, and two with green background where the prediction is either too high or too low. We observe that images with low predicted scores (bottom half) tend to be less 'busy' and possibly lack interesting features. They tend to contain clean backgrounds with little to no salient foreground objects, as compared to the high popularity images. Further, we observe that the images with low popularity but predicted to have high popularity (top-left quadrant) tend to resemble the highly popular images (top-right quadrant) but may not be popular due to the social network effects of the user. In general, our method tends to do relatively well in picking images that could potentially have a high number of views regardless of social context.

3.5.3 Visualizing what makes an image popular

To better understand what makes an image popular, we attempt to attribute the popularity of an image to its regions (similar to [80]). Being able to visualize the regions that make an image popular can be extremely useful in a variety of applications e.g. we could teach users to take better photographs by highlighting the important regions, or modify images automatically to make them more popular by replacing the regions with low impact on popularity.

In Figure 3-7, we visualize the contribution of different image regions to the popularity of an image by reversing the contribution of the learned weights to image descriptors. Since we use a bag-of-words descriptor, it can be difficult to identify exactly which descriptors in the image are contributing positively or negatively to the popularity score. Since max-pooling is used over spatial regions, we can carefully record the descriptors and their locations that led to the maximum value for each bag-of-words dictionary element. Then, we can combine this with the weights learned by SVR to generate a 'heatmap' of the regions that make an image popular. Note that

this is a rather coarse heatmap because there can be image descriptors that have high values for certain dictionary elements, but not the highest, and their contribution is not considered in this max-pooling scenario. Thus, we end up with heatmaps that do not look semantically pleasing but indicate regions of high or low interest rather coarsely. This representation could be improved by using the recently popular mid-level features [139, 69] which encode more semantically meaningful structure.

From Figure 3-7, we can see that semantically meaningful objects such as people tend to contribute positively to the popularity of an image (first row right, and second row). Further we note that open scenes with little activity tend to be unpopular (with many exceptions of course). We observe that the number of high-scoring red/yellow regions decrease as the popularity of an image decreases (bottom row). Further, we observe that several semantically meaningful objects in the images are highlighted such as the train or different body parts, but due to the shortcoming described earlier, the regions are incoherent and broken up into several parts. Overall, popularity is a difficult metric to understand precisely based on image content alone because social cues have a large influence on the popularity of images.

3.6 Discussion

In this chapter, we explored what makes images uploaded by users popular among online social media. Specifically, we explored millions of images on Flickr uploaded by tens of thousands of users and studied the problem of predicting the popularity of of the uploaded images. While some images get millions of views, others go completely unnoticed. This variation is noticed even among images uploaded by the same user, or images from the same genre. We designed an approach that leverages social cues as well as image content features to come up with a prediction technique for overall popularity. We also show key insights from our method that suggest crucial aspects of the image that determine or influence popularity. We extensively test our methodology across different dataset types that have a variable distribution of images per user, as well as explore prediction models that are focused on certain user groups

Figure 3-6: Predictions on some images from our dataset using Gradient based image predictor. We show four quadrants of ground truth popularity and predicted popularity. The green and red background colors represent correct and false predictions respectively.

or independent users. The results show interesting variation in importance of social cues such as number of photos uploaded or number of contacts, and contrast it with image cues such as color or gradients, depending on the dataset types.

Several directions remain for future exploration. An interesting question is predicting *shareability* as opposed to *popularity*. Are these different traits? There might be some images that are viewed/consumed, but not necessarily shared with friends. Does this perhaps have a connection with the emotion that is elicited? For example, peaceful images may get liked, funny images may get shared, and scary/disturbing images may get viewed but not publicly broadcasted. It would be interesting to understand the features/traits that distinguish the kinds of interaction they elicit from users. vs. those that elicit more uniform responses.

On a more open-ended note, do the influence of social context and image content spill across their boundaries? It is conceivable that a user who uploads refined photographs, over time, accumulates a larger number of followers. This could garner a

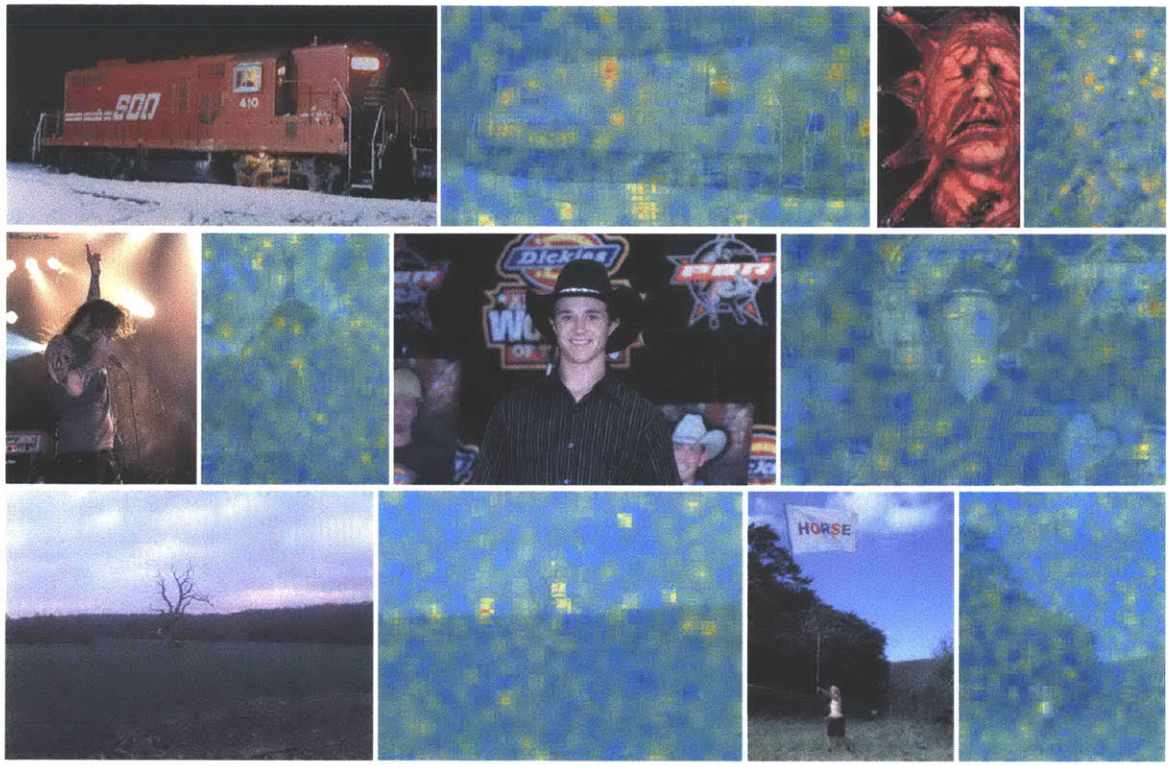


Figure 3-7: Popularity score of different image regions for images at different levels of popularity: high (top row), medium (middle row) and low (bottom row). The importance of the regions decreases in the order red > green > blue.

stronger influence through the network and thereby result in increased popularity of photos uploaded by this user. Attribution might be inaccurate as a consequence - the resulting popularity may be ascribed to the user's social context and miss the image content. Popularity *prediction* as such may not be adversely affected, but what is the right causality here? Disentangling these features promises for an exciting direction to take this research forward. Being able to disentangle these factors may also result in improved content based popularity prediction by removing *noise* from the labels caused by social factors. Another specific question, for which data is unfortunately unavailable, is understanding time series of popularity for images: rather than simply looking at total popularity (normalized or unnormalized), can one investigate temporal gradients as well? For example, the total popularity of two images (or classes of images) may be the same, yet, one may have rapidly gained popularity and then sharply fallen, while another might have slowly and constantly retained popularity.

These could perhaps exhibit fundamentally different photograph-types, perhaps the former being due to a sudden news or attention on an event, figure, or location, while the latter due to some intrinsic lasting value. Such exploration would be really valuable and exciting if the time series data were available for uploaded images.

Finally, from an application standpoint, is there a photography popularity tool that could be built here? Can photographers be aided with suggestions on how to modify their pictures for broad appeal vs artistic appeal? This could be an interesting research direction as well as a promising product. This is to be contrasted with some recent work[3] on aided movie script writing tools, where machine learning is potentially used to predict the likelihood of viewers enjoying the movie plot.

[3]http://www.nytimes.com/2013/05/06/business/media/solving-equation-of-a-hit-film-script-with-data.html

Chapter 4

Predicting Gaze

In this chapter, I tackle the problem of predicting gaze from two perspectives, first, in Section 4.1, from a third person perspective, known as gaze-following, and second, in Section 4.2, from the perspective of a device, known as eye tracking. In gaze-following, the goal is to identify where an individual in an image is looking in their environment, while in eye tracking, the goal is to identify where a user interacting with a device is looking within the screen of the device. I describe the approach to tackling each of these problems in detail below.

4.1 Gaze-Following

You step out of your house and notice a group of people looking up. You look up and realize they are looking at an aeroplane in the sky. Despite the object being far

Figure 4-1: **Gaze-following:** We present a model that learns to predict where people in images are looking. We also introduce GazeFollow, a new large-scale annotated dataset for gaze-following.

away, humans have the remarkable ability to precisely follow the gaze direction of another person, a task commonly referred to as *gaze-following* (see [35] for a review). Such an ability is a key element to understanding what people are doing in a scene and their intentions. Similarly, it is crucial for a computer vision system to have this ability to better understand and interpret people. For instance, a person might be holding a book but looking at the television, or a group of people might be looking at the same object which can indicate that they are collaborating at some task, or they might be looking at different places which can indicate that they are not familiar with each other or that they are performing unrelated tasks (see Figure 1). Gaze-following has applications in robotics and human interaction interfaces where it is important to understand the object of interest of a person. Gaze-following can also be used to predict what a person will do next as people tend to attend to objects they are planning to interact with even before they start an action.

Despite the importance of this topic, only a few works in computer vision have explored gaze-following [40, 119, 105, 118, 141]. Previous work on gaze-following addresses the problem by limiting the scope (e.g., people looking at each other only [105]), by restricting the situations (e.g., scenes with multiple people only or synthetic scenarios [66, 55]), or by using complex inputs (multiple images [40, 118, 141] or eye-tracking data [41]). Only [119] tackles the unrestricted gaze-following scenario but relies on face detectors (therefore can not handle situations such as people looking away from the camera) and is not evaluated on a gaze-following task. Our goal is to perform gaze-following in natural settings without making restrictive assumptions and when only a single view is available. We want to address the general gaze-following problem to be able to handle situations in which several people are looking at each other, and one or more people are interacting with one or more objects.

In this paper, we formulate the problem of gaze-following as: given a single picture containing one or more people, the task is to the predict the location that each person in a scene is looking at. To address this problem, we introduce a deep architecture that learns to combine information about the head orientation and head location with the scene content in order to follow the gaze of a person inside the picture. The input

to our model is a picture and the location of the person for who we want to follow the gaze, and the output is a distribution over possible locations that the selected person might be looking at. This output distribution can be seen as a saliency map from the point of view of the person inside the picture. To train and evaluate our model, we also introduce GazeFollow, a large-scale benchmark dataset for gaze-following. Our model, code and dataset are available for download at http://gazefollow.csail.mit.edu.

4.1.1 Related work

Saliency: Although strongly related, there are a number of important distinctions between gaze-following [35] and saliency models of attention [64]. In traditional models of visual attention, the goal is to predict the eye fixations of an observer *looking at a picture*, while in gaze-following the goal is to estimate what is being looked at by a person *inside a picture*. Most saliency models focus on predicting fixations while an observer is free-viewing an image [64, 68] (see [13] for a review). However, in gaze-following, the people in the picture are generally engaged in a task or navigating an environment and, therefore, are not free-viewing and might fixate on objects even when they are not the most salient. A model for gaze-following has to be able to follow the line of sight and then select, among all possible elements that cross the line of sight, which objects are likely to be the center of attention. Both tasks (gaze-following and saliency modeling) are related in several interesting ways. For instance, [12] showed that gaze-following of people inside a picture can influence the fixations of an observer looking at the picture as the object being fixated by the people inside the picture will attract the attention of the observer of the picture.

Gaze: The work on gaze-following in computer vision is very limited. Gaze-following is used in [119] to improve models of free-viewing saliency prediction. However, they only estimate the gaze direction without identifying the object being attended. Further, their reliance on a face detector [175] prevents them from being able to estimate gaze for people looking away from the camera. Another way of approaching gaze-following is using a wearable eye-tracker to precisely measure the gaze of several people in a scene. For instance, [41] used an eye tracker to predict the next

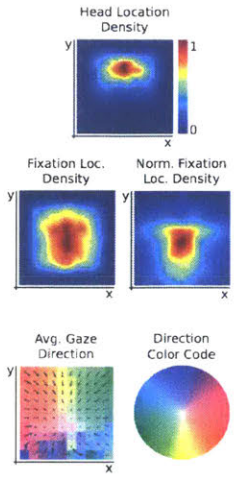
object the user will interact with, and to improve action recognition in egocentric vision. In [105] they propose detecting people looking at each other in a movie in order to better identify interactions between people. As in [119], this work only relies on the direction of gaze without estimating the object being attended, and, therefore, cannot address the general problem of gaze-following, in which a person is interacting with an object. In [40], they perform gaze-following in scenes with multiple observers in an image by finding the regions in which multiple lines of sight intersect. Their method needs multiple people in the scene, each with an egocentric camera, used to get 3D head location, as the model only uses head orientation information and does not incorporate knowledge about the content of the scene. In [118, 141], the authors propose a system to infer the region attracting the attention of a group of people (social saliency prediction). As in [40] their method takes as input a set of pictures taken from the viewpoint of each of the people present in the image and it does not perform gaze-following. Our method only uses a single third-person view of the scene to infer gaze.

4.1.2 GazeFollow: A Large-Scale Gaze-Following Dataset

In order to both train and evaluate models, we built GazeFollow, a large-scale dataset annotated with the location of where people in images are looking. We used several major datasets that contain people as a source of images: $1,548$ images from SUN [164], $33,790$ images from MS COCO [96], $9,135$ images from Actions 40 [166], $7,791$ images from PASCAL [36], 508 images from the ImageNet detection challenge [134] and $198,097$ images from the Places dataset [174]. This concatenation results in a challenging and large image collection of people performing diverse activities in many everyday scenarios.

Since the source datasets do not have gaze ground-truth, we annotated it using Amazon's Mechanical Turk (AMT). Workers used our online tool to mark the center of a person's eyes and where the worker believed the person was looking. Workers could indicate if the person was looking outside the image or if the person's head was not visible. To control quality, we included images with known ground-truth, and

(a) Example test images and annotations



(b) Test set statistics

Figure 4-2: **GazeFollow Dataset:** We introduce a new dataset for gaze-following in natural images. On the left, we show several example annotations and images. In the graphs on the right, we summarize a few statistics about test partition of the dataset. The top three heat maps show the probability density for the location of the head, the fixation location, and the fixation location normalized with respect to the head position. The bottom shows the average gaze direction for various head positions.

we used these to detect and discard poor annotations. Finally, we obtained $130,339$ people in $122,143$ images, with gaze locations inside the image.

We use about $4,782$ people of our dataset for testing and the rest for training. We ensured that every person in an image is part of the same split, and to avoid bias, we picked images for testing such that the fixation locations were uniformly distributed across the image. Further, to evaluate human consistency on gaze-following, we collected 10 gaze annotations per person for the test set.

We show some example annotations and statistics of the dataset in Figure 4-2. We designed our dataset to capture various fixation scenarios. For example, some images contain several people with joint attention while others contain people looking at each other. The number of people in the image can vary, ranging from a single person to a crowd of people. Moreover, we observed that while some people have consistent fixation locations others have bimodal or largely inconsistent distributions, suggesting that solutions to the gaze-following problem could be multimodal.

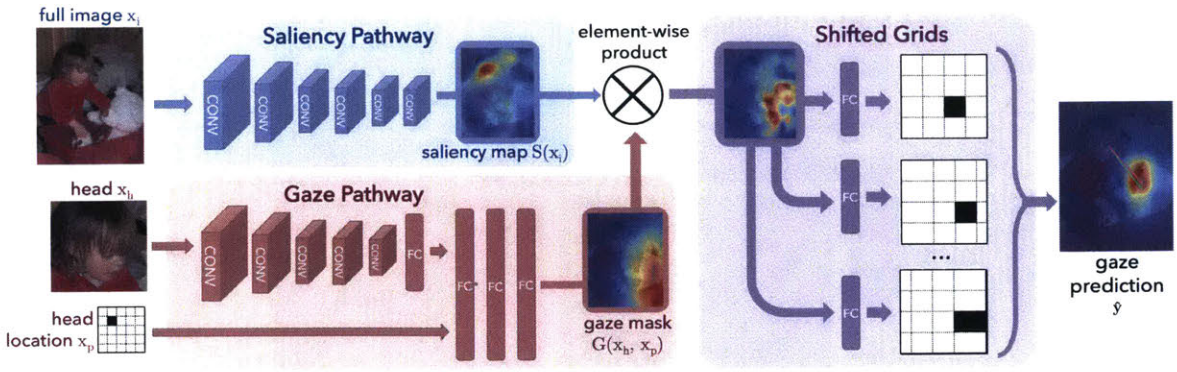


Figure 4-3: **Network architecture:** We show the architecture of our deep network for gaze-following. Our network has two main components: the saliency pathway (top) to estimate saliency and the gaze pathway (bottom) to estimate gaze direction. See Section 4.1.3 for details.

4.1.3 Learning to Follow Gaze

At a high level, our model is inspired by how humans tend to follow gaze. When people infer where another person is looking, they often first look at the person's head and eyes to estimate their field of view, and subsequently reason about salient objects in their perspective to predict where they are looking. In this section, we present a model that emulates this approach.

Gaze and Saliency Pathways

Suppose we have an image x_i and a person for whom we want to predict gaze. We parameterize this person with a quantized spatial location of the person's head x_p and a cropped, close-up image of their head x_h. Given x, we seek to predict the spatial location of the person's fixation y. Encouraged by progress in deep learning, we also use deep networks to predict a person's fixation.

Keeping the motivation from Section 4.1.3 in mind, we design our network to have two separate pathways for gaze and saliency. The gaze pathway only has access to the closeup image of the person's head and their location, and produces a spatial map, $G(x_h, x_p)$, of size $D \times D$. The saliency pathway sees the full image but not the person's location, and produces another spatial map, $S(x_i)$, of the same size $D \times D$.

We then combine the pathways with an element-wise product:

$$\hat{y} = F\left(G(x_h, x_p) \otimes S(x_i)\right) \tag{4.1}$$

where $\otimes$ represents the element-wise product. $F(\cdot)$ is a fully connected layer that uses the multiplied pathways to predict where the person is looking, $\hat{y}$.

Since the two network pathways only receive a subset of the inputs, they cannot themselves solve the full problem during training, and instead are forced to solve subproblems. Our intention is that, since the gaze pathway only has access to the person's head, x_h and location, x_p, we expect it will learn to predict the direction of gaze. Likewise, since the saliency pathway does not know which person to follow, we hope it learns to find objects that are salient, independent of the person's viewpoint. The element-wise product allows these two pathways to interact in a way that is similar to how humans approach this task. In order for a location in the element-wise product to be activated, both the gaze and saliency pathways must have large activations.

Saliency map: To form the saliency pathway, we use a convolutional network on the full image to produce a hidden representation of size $D \times D \times K$. Since [173] shows that objects tend to emerge in these deep representations, we can create a gaze-following saliency map by learning the importance of these objects. To do this, we add a convolutional layer that convolves the hidden representation with a $w \in \mathbb{R}^{1\times1\times K}$ filter, which produces the $D \times D$ saliency map. Here, the sign and magnitude of w can be interpreted as weights indicating an object's importance for gaze-following saliency.

Gaze mask: In the gaze pathway, we use a convolutional network on the head image. We concatenate its output with the head position and use several fully connected layers and a final sigmoid to predict the $D \times D$ gaze mask.

Pathway visualization: Figure 4-4 shows examples of the (a) gaze masks and (b) saliency maps learned by our network. Figure 4-4(b) also compares the saliency maps of our network with the saliency computed using a state of the art saliency

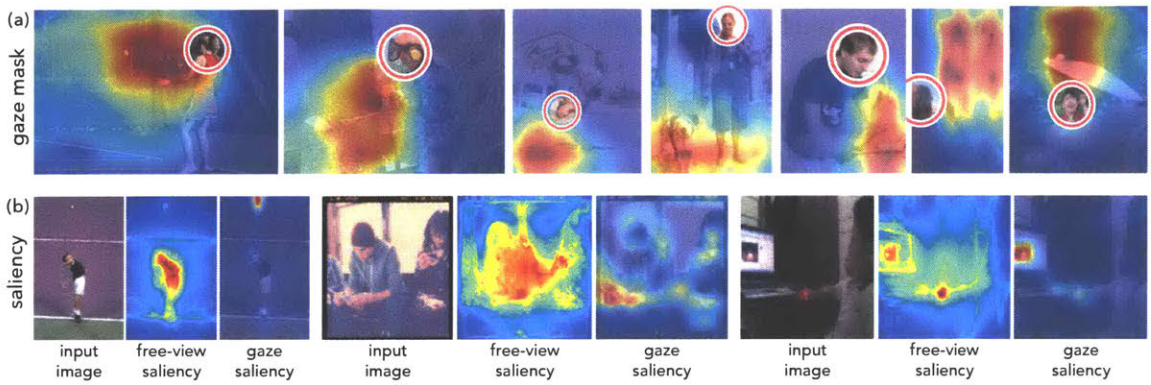


Figure 4-4: **Pathway visualization:** (a) The gaze mask output by our network for various head poses. (b) Each triplet of images show, from left to right, the input image, its free-viewing saliency estimated using [68], and the gaze-following saliency estimated using our network. These examples clearly illustrate the differences between free-viewing saliency [68] and gaze-following saliency.

model [68]. Note that our model learns a notion of saliency that is relevant for the gaze-following task and places emphasis on certain objects that people tend to look at (e.g., balls and televisions). In the third example, the red light coming from the computer mouse is salient in the Judd et al [68] model but that object is not relevant in a gaze-following task as the computer monitor is more likely to be the target of attention of the person inside the picture.

Multimodal Predictions

Although humans can often follow gaze reliably, predicting gaze is sometimes ambiguous. If there are several salient objects in the image, or the eye pose cannot be accurately perceived, then humans may disagree when predicting gaze. We can observe this for several examples in Figure 4-2. Consequently, we want to design our model to support multimodal predictions.

We could formulate our problem as a regression task (i.e., regress the Cartesian coordinates of fixations) but then our predictions would be unimodal. Instead, we can formulate our problem as a classification task, which naturally supports multimodal outputs because each category has a confidence value. To do this, we quantize the fixation location y into a $N \times N$ grid. Then, the job of the network is to classify the inputs x into one of N^2 classes. The model output $\hat{y} \in \mathbb{R}^{N \times N}$ is the confidence that

the person is fixating in each grid cell.

Shifted grids: For classification, we must choose the number of grid cells, N. If we pick a small N, our predictions will suffer from poor precision. If we pick a large N, there will be more precision, but the learning problem becomes harder because standard classification losses do not gradually penalize spatial categories – a misclassification that is off by just one cell should be penalized less than errors multiple cells away. To alleviate this trade-off, we propose the use of *shifted grids*, as illustrated in Figure 4-3, where the network solves several overlapping classification problems. The network predicts locations in multiple grids where each grid is shifted such that cells in one grid overlap with cells in other grids. We then average the shifted outputs to produce the final prediction.

Training

We train our network end-to-end using backpropagation. We use a softmax loss for each shifted grid and average their losses. Since we only supervise the network with gaze fixations, we do not enforce that the gaze and saliency pathways solve their respective subproblems. Rather, we expect that the proposed network structure encourages these roles to emerge automatically (which they do, as shown in Figure 4-6).

Implementation details: We implemented the network using Caffe [67]. The convolutional layers in both the gaze and saliency pathways follow the architecture of the first five layers of the AlexNet architecture [87]. In our experiments, we initialize these convolutional layers of the saliency pathway with the Places-CNN [174] and those of the gaze pathway with ImageNet-CNN [87]. The last convolutional layer of the saliency pathway has a $1 \times 1 \times 256$ convolution kernel (i.e., $K = 256$). The remaining fully connected layers in the gaze pathway are of sizes 100, 400, 200, and 169 respectively. The saliency map and gaze mask are 13×13 in size (i.e., $D = 13$), and we use 5 shifted grids of size 5×5 each (i.e., $N = 5$). For learning, we augment our training data with flips and random crops with the fixation locations adjusted accordingly.

Figure 4-5: **Qualitative results:** We show several examples of successes and failures of our model. The red lines indicate ground truth gaze, and the yellow lines indicate our predicted gaze.

4.1.4 Experiments

Setup

We evaluate the ability of our model to predict where people in images are looking. We use the disjoint train and test sets from GazeFollow, as described in Section 4.1.2, to train and evaluate our model. The test set was randomly sampled such that the fixation location was approximately uniform, and ignored people who were looking outside the picture or at the camera. Similar to PASCAL VOC Action Recognition [36] where ground-truth person bounding boxes are available both during train-

Model	AUC	Distance	Minimum Distance	Angular Error
Our	**0.878**	**0.190**	**0.113**	**24°**
SVM+shift grid	0.788	0.268	0.186	40°
SVM+one grid	0.758	0.276	0.193	43°
Judd [68]	0.711	0.337	0.250	54°
Fixed bias	0.674	0.306	0.219	48°
Center	0.633	0.313	0.230	49°
Random	0.504	0.484	0.391	69°
One human	0.924	0.096	0.040	11°

(a) Main Evaluation

Model	AUC	Distance	Minimum Distance	Angular Error
No image	0.821	0.221	0.142	27°
No position	0.837	0.238	0.158	32°
No head	0.822	0.264	0.179	41°
No eltwise	0.876	0.193	0.117	25°
5×5 grid	0.839	0.245	0.164	36°
10×10 grid	0.873	0.218	0.138	30°
L2 loss	0.768	0.245	0.169	34°
Our full	0.878	0.190	0.113	24°

(b) Model Diagnostics

Table 4.1: **Evaluation:** (a) We evaluate our model against baselines and (b) analyze how it performances with some components disabled. *AUC* refers to the area under the ROC curve (higher is better). *Distance* refers to the L_2 distance to the average of ground truth fixation, while *Minimum Distance* refers to the L_2 distance to the nearest ground truth fixation (lower is better). *Angular Error* is the error of predicted gaze in degrees (lower is better). See Section 4.1.4 for details.

ing and testing, we assume that we are given the head location at both train and test time. This allows us to focus our attention on the primary task of gaze-following. In Section 4.1.4, we show that our method performs well even when using a simple head detector.

Our primary evaluation metric compares the ground truth annotations[1] against the distribution predicted by our model. We use the **Area Under Curve (AUC)** criteria from [68] where the predicted heatmap is used as confidences to produce an ROC curve. The AUC is the area under this ROC curve. If our model behaves

[1]Note that, as mentioned in Section 4.1.2, we obtain 10 annotations per person in the test set.

perfectly, the AUC will be 1 while chance performance is 0.5. **L_2 distance:** We evaluate the Euclidean distance between our prediction and the average of ground truth annotations. We assume each image is of size 1×1 when computing the L_2 distance. Additionally, as the ground truth may be multimodal, we also report the minimum L_2 distance between our prediction and all ground truth annotations. **Angular error:** Using the ground truth eye position from the annotation we compute the gaze vectors for the average ground truth fixations and our prediction, and report the angular difference between them.

We compare our approach against several baselines ranging from simple (center, fixed bias) to more complex (SVM, free-viewing saliency) as described below. **Center:** The prediction is always the center of the image. **Fixed bias:** The prediction is given by the average of fixations from the training set for heads in similar locations as the test image. **SVM:** We generate features by concatenating the quantized eye position with `pool5` of the ImageNet-CNN [87] for both the full image and the head image. We train a SVM on these features to predict gaze using a similar classification grid setup as our model. We evaluate this approach for both, a single grid and shifted grids. **Free-viewing saliency:** We use a state-of-the-art free-viewing saliency model [68] as a predictor of gaze. Although free-viewing saliency models ignore head orientation and location, they may still identify important objects in the image.

Results

We compare our model against baselines in Table 4.1(a). Our method archives an AUC of 0.878 and a mean Euclidean error of 0.190, outperforming all baselines significantly in all the evaluation metrics. The SVM model using shifted grids shows the best baseline performance, surpassing the one grid baseline by a reasonable margin. This verifies the effectiveness of the shifted grids approach proposed in this work.

Figure4-5 shows some example outputs of our method. These qualitative results show that our method is able to distinguish people in the image by using the gaze pathway to model a person's point of view, as it produces different outputs for different people in the same image. Furthermore, it is also able to find salient objects in images,

such as balls or food. However, the method still has certain limitations. The lack of 3D understanding generates some wrong predictions, as illustrated by the 1st image in the 2nd row of Figure 4-5, where one of the predictions is in a different plane of depth.

To obtain an approximate upper bound on prediction performance, we evaluate human performance on this task. Since we annotated our test set 10 times, we can quantify how well one annotation predicts the mean of the remaining 9 annotations. A single human is able to achieve an AUC of 0.924 and a mean Euclidean error of 0.096. While our approach outperforms all baselines, it is still far from reaching human performance. We hope that the availability of GazeFollow will motivate further research in this direction, allowing machines to reach human level performance.

Analysis

Ablation study: In Table 4.1(b), we report the performance after removing different components of our model, one at a time, to better understand their significance. In general, all three of inputs (image, position and head) contribute to the performance of our model. Interestingly, the model with only the head and its position achieves comparable *angular* error to our full method, suggesting that the gaze pathway is largely responsible for estimating the gaze direction. Further, we show the results of our model with single output grids (5×5 and 10×10). Removing shifted grids hurts performance significantly as shifted grids have a spatially graded loss function, which is important for learning.

Internal representation: In Figure 4-6, we visualize the various stages of our network. We show the output of each of the pathways as well as the element wise product. For example, in the second row we have two different girls writing on the blackboard. The gaze mask effectively creates a heat map of the field of view for the girl in the right, while the saliency map identifies the salient spots in the image. The element-wise multiplication of the saliency map and gaze mask removes the responses of the girl on the left and attenuates the saliency of the right girl's head. Finally, our shifted grids approach accurately predicts where the girl is looking.

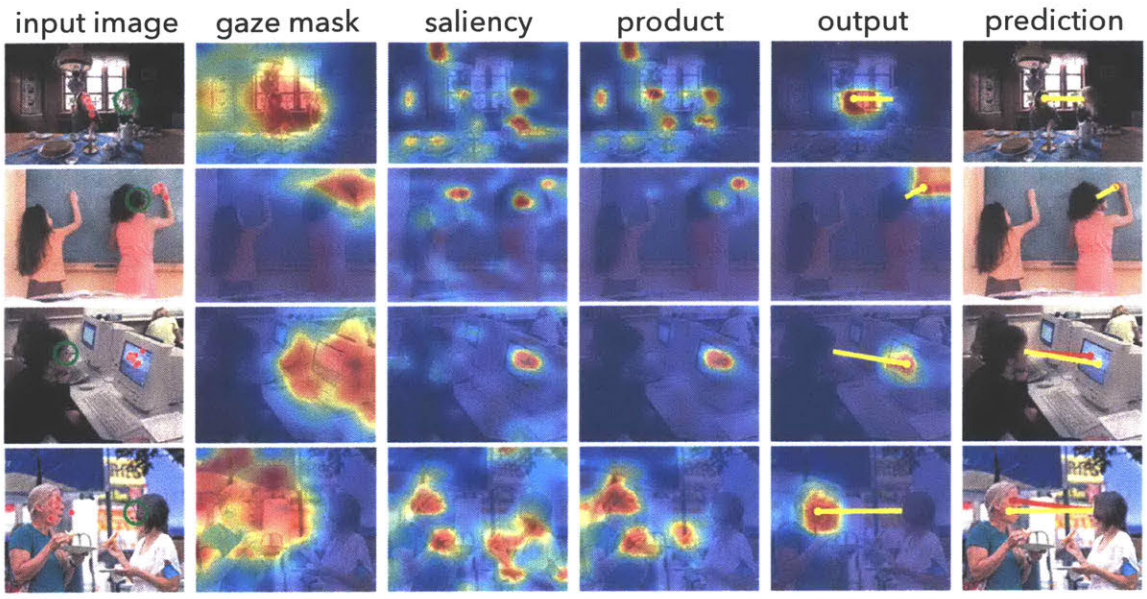


Figure 4-6: **Visualization of internal representations:** We visualize the output of different components of our model. The green circle indicates the person whose gaze we are trying to predict, the red dots/lines show the ground truth gaze, and the yellow line is our predicted gaze.

Further, we apply the technique from [173] to visualize the top activations for different units in the fifth convolutional layer of the saliency pathway. We use filter weights from the sixth convolutional layer to rank their contribution to the saliency map. Figure 4-7 shows four units with positive (left) and negative (right) contributions to the saliency map. Interestingly, w learns positive weights for salient objects such as *switched on TV monitors* and balls, and negative weights for non-salient objects.

Automatic head detection: To evaluate the impact of imperfect head locations on our system, we built a simple head detector, and input its detections into our model. For detections surpassing the intersection over union threshold of 0.5, our model achieved an AUC of 0.868, as compared to an AUC of 0.878 when using ground-truth head locations. This demonstrates that our model is robust to inaccurate head detections, and can easily be made fully-automatic.

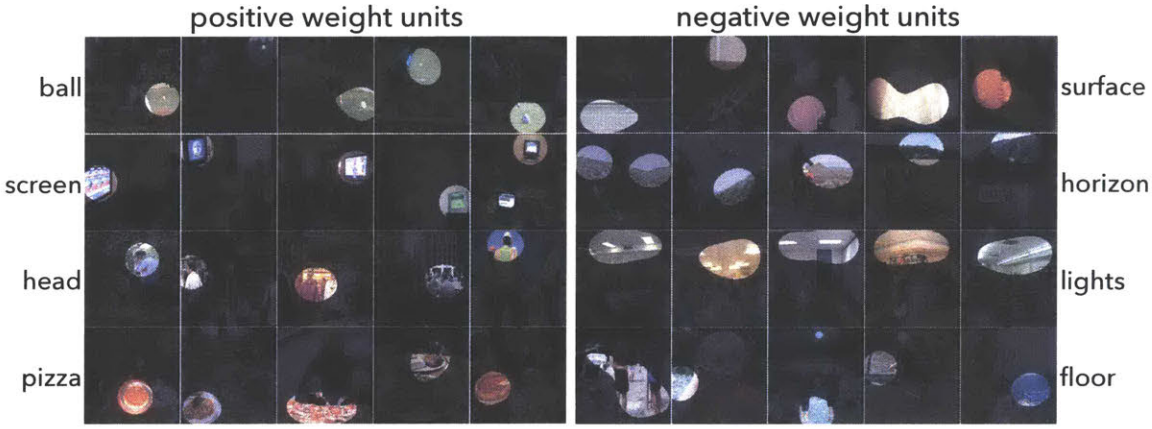


Figure 4-7: **Visualization of saliency units:** We visualize several units in our saliency pathway by finding images with high scoring activations, similar to [173]. We sort the units by w, the weights of the sixth convolutional layer (See Section 4.1.3 for more details). Positive weights tend to correspond to salient everyday objects, while negative weights tend to correspond to background objects.

4.1.5 Summary

Accurate gaze-following achieving human-level performance will be an important tool to enable systems that can interpret human behavior and social situations. In this section, we introduced a model that learns to do gaze-following using GazeFollow, a large-scale dataset of human annotated gaze. Our model automatically learns to extract the line of sight from heads, without using any supervision on head pose, and to detect salient objects that people are likely to interact with, without requiring object-level annotations during training. We hope that our model and dataset will serve as important resources to facilitate further research in this direction.

4.2 Eye Tracking

From human–computer interaction techniques [65, 103, 108] to medical diagnoses [56] to psychological studies [127] to computer vision [11, 71], eye tracking has applications in many areas [34]. Gaze is the externally-observable indicator of human visual attention, and many have attempted to record it, dating back to the late eighteenth century [59]. Today, a variety of solutions exist (many of them commercial) but all suffer from one or more of the following: high cost (*e.g.*, Tobii X2-60), custom

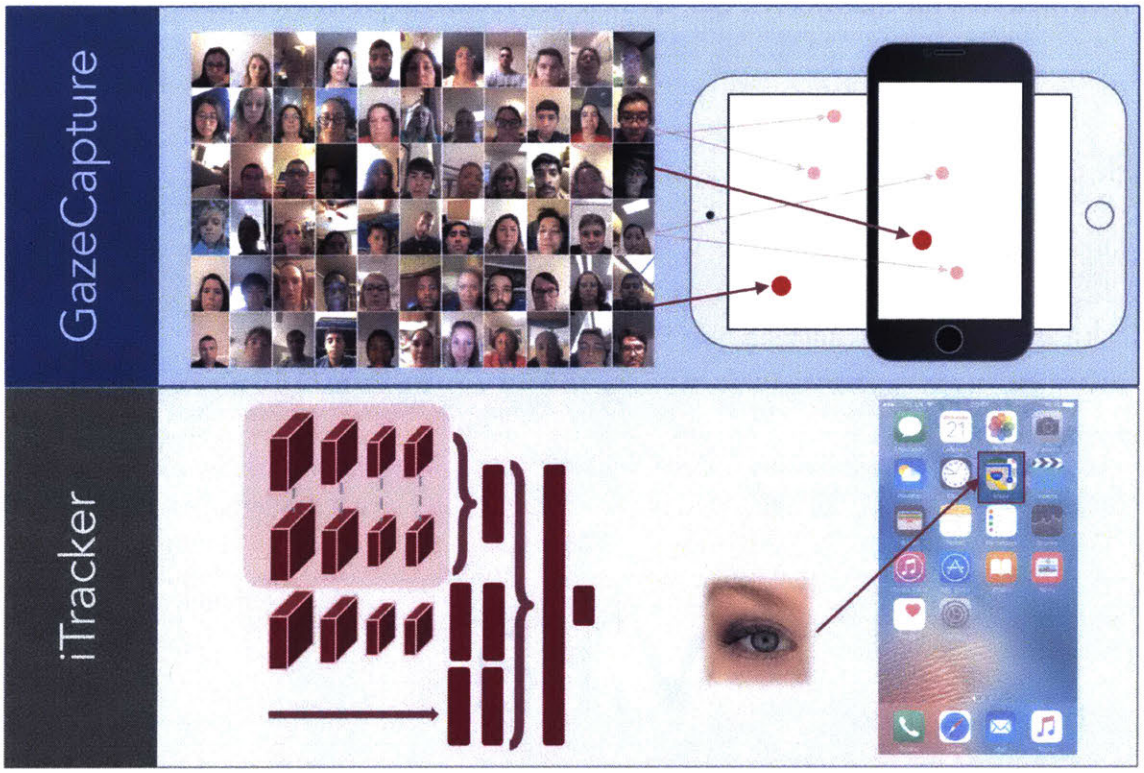


Figure 4-8: In this work, we develop GazeCapture, the first large-scale eye tracking dataset captured via crowdsourcing. Using GazeCapture, we train iTracker, a convolutional neural network for robust gaze prediction.

or invasive hardware (*e.g.*, Eye Tribe, Tobii EyeX) or inaccuracy under real-world conditions (*e.g.*, [107, 145, 171]). These factors prevent eye tracking from becoming a pervasive technology that should be available to anyone with a reasonable camera (*e.g.*, a smartphone or a webcam). In this work, our goal is to overcome these challenges to bring eye tracking to everyone.

We believe that this goal can be achieved by developing systems that work reliably on mobile devices such as smartphones and tablets, without the need for any external attachments (Figure 4-8). Mobile devices offer several benefits over other platforms: (1) widespread use—more than a third of the world's population is estimated to have smartphones by 2019 [142], far exceeding the number of desktop/laptop users; (2) high adoption rate of technology upgrades—a large proportion of people have the latest hardware allowing for the use of computationally expensive methods, such as convolutional neural networks (CNNs), in real-time; (3) the heavy usage of cameras on

mobile devices has lead to rapid development and deployment of camera technology, and (4) the fixed position of the camera relative to the screen reduces the number of unknown parameters, potentially allowing for the development of high-accuracy calibration-free tracking.

The recent success of deep learning has been apparent in a variety of domains in computer vision [87, 47, 149, 129, 78], but its impact on improving the performance of eye tracking has been rather limited [171]. We believe that this is due to the lack of availability of large-scale data, with the largest datasets having ~50 subjects [58, 145]. In this work, using crowdsourcing, we build *GazeCapture*, a mobile-based eye tracking dataset containing almost 1500 subjects from a wide variety of backgrounds, recorded under variable lighting conditions and unconstrained head motion.

Using GazeCapture, we train *iTracker*, a convolutional neural network (CNN) learned end-to-end for gaze prediction. iTracker does not rely on any preexisting systems for head pose estimation or other manually-engineered features for prediction. Training the network with just crops of both eyes and the face, we outperform existing eye tracking approaches in this domain by a significant margin. While our network achieves state-of-the-art performance in terms of accuracy, the size of the inputs and number of parameters make it difficult to use in real-time on a mobile device. To address this we apply ideas from the work on dark knowledge by Hinton *et al.* [54] to train a smaller and faster network that achieves real-time performance on mobile devices with a minimal loss in accuracy.

Overall, we take a significant step towards putting the power of eye tracking in everyone's palm.

4.2.1 Related Work

There has been a plethora of work on predicting gaze. Here, we give a brief overview of some of the existing gaze estimation methods and urge the reader to look at this excellent survey paper [51] for a more complete picture. We also discuss the differences between GazeCapture and other popular gaze estimation datasets.

Gaze estimation: Gaze estimation methods can be divided into model-based or

appearance-based [51]. Model-based approaches use a geometric model of an eye and can be subdivided into corneal-reflection-based and shape-based methods. Corneal-reflection-based methods [167, 176, 177, 53] rely on external light sources to detect eye features. On the other hand, shape-based methods [61, 18, 156, 52] infer gaze direction from observed eye shapes, such as pupil centers and iris edges. These approaches tend to suffer with low image quality and variable lighting conditions, as in our scenario. Appearance-based methods [150, 136, 101, 100, 154, 6] directly use eyes as input and can potentially work on low-resolution images. Appearance-based methods are believed [171] to require larger amounts of user-specific training data as compared to model-based methods. However, we show that our model is able to generalize well to novel faces without needing user-specific data. While calibration is helpful, its impact is not as significant as in other approaches given our model's inherent generalization ability achieved through the use of deep learning and large-scale data. Thus, our model does not have to rely on visual saliency maps [19, 144] or key presses [143] to achieve accurate calibration-free gaze estimation. Overall, iTracker is a data-driven appearance-based model learned end-to-end without using any hand-engineered features such as head pose or eye center location. We also demonstrate that our trained networks can produce excellent features for gaze prediction (that outperform hand-engineered features) on other datasets despite not having been trained on them.

Gaze datasets: There are a number of publicly available gaze datasets in the community [106, 162, 140, 107, 145, 171, 58]. We summarize the distinctions from these datasets in Table 4.2. Many of the earlier datasets [106, 162, 140] do not contain significant variation in head pose or have a coarse gaze point sampling density. We overcome this by encouraging participants to move their head while recording and generating a random distribution of gaze points for each participant. While some of the modern datasets follow a similar approach [145, 107, 171, 58], their scale—especially in the number of participants—is rather limited. We overcome this through the use of crowdsourcing, allowing us to build a dataset with $\sim$30 times as many participants as the current largest dataset. Further, unlike [171], given our recording

	# People	Poses	Targets	Illumination	Images
[106]	20	1	16	1	videos
[162]	20	19	2–9	1	1,236
[140]	56	5	21	1	5,880
[107]	16	cont.	continuous	2	videos
[145]	50	8+synthesized	160	1	64,000
[171]	15	continuous	continuous	cont.	213,659
[58]	51	cont.	35	continuous	videos
Ours	**1474**	**continuous**	**13+continuous**	**continuous**	**2,445,504**

Table 4.2: Comparison of our GazeCapture dataset with popular publicly available datasets. GazeCapture has approximately 30 times as many participants and 10 times as many frames as the largest datasets and contains a significant amount of variation in pose and illumination, as it was recorded using crowdsourcing.

permissions, we can release the complete images without post-processing. We believe that GazeCapture will serve as an invaluable resource for future work in this domain.

4.2.2 GazeCapture: A Large-Scale Eye Tracking Dataset

In this section, we describe how we achieve our goal of scaling up the collection of eye tracking data. We find that most existing eye tracking datasets have been collected by researchers inviting participants to the lab, a process that leads to a lack of variation in the data and is costly and inefficient to scale up. We overcome these limitations through the use of crowdsourcing, a popular approach for collecting large-scale datasets [134, 78, 172, 129]. In Section 4.2.2, we describe the process of obtaining reliable data via crowdsourcing and in Section 4.2.2, we compare the characteristics of GazeCapture with existing datasets.

Collecting Eye Tracking Data

Our goal here is to develop an approach for collecting eye tracking data on mobile devices that is (1) scalable, (2) reliable, and (3) produces large variability. Below, we describe, in detail, how we achieve each of these three goals.

Scalability: In order for our approach to be scalable, we must design an automated mechanism for gathering data and reaching participants. Crowdsourcing

is a popular technique researchers use to achieve scalability. The primary difficulty with this approach is that most crowdsourcing platforms are designed to be used on laptops/desktops and provide limited flexibility required to design the desired user experience. Thus, we decided to use a hybrid approach, combining the scalable workforce of crowdsourcing platforms together with the design freedom provided by building custom mobile applications. Specifically, we built an iOS application, also named GazeCapture[2], capable of recording and uploading gaze tracking data, and used Amazon Mechanical Turk (AMT) as a platform for recruiting people to use our application. On AMT, the workers were provided detailed instructions on how to download the application from Apple's App Store and complete the task.

We chose to build the GazeCapture application for Apple's iOS because of the large-scale adoption of latest Apple devices, and the ease of deployment across multiple device types such as iPhones and iPads using a common code base. Further, the lack of fragmentation in the versions of the operating system (as compared to other platforms) significantly simplified the development process. Additionally, we released the application publicly to the App Store (as opposed to a beta release with limited reach) simplifying installation of our application, thereby further aiding the scalability of our approach.

Reliability: The simplest rendition of our GazeCapture application could involve showing workers dots on a screen at random locations and recording their gaze using the front-facing camera. While this approach may work well when calling individual participants to the lab, it is not likely to produce reliable results without human supervision. Thus, we must design an automatic mechanism that ensures workers are paying attention and fixating directly on the dots shown on the screen.

First, to avoid distraction from notifications, we ensure that the worker uses *Airplane Mode* with no network connection throughout the task, until the task is complete and ready to be uploaded. Second, instead of showing a plain dot, we show a pulsating red circle around the dot, as shown in Figure 4-9, that directs the fixation of the eye to lie in the middle of that circle. This pulsating dot is shown for approx-

[2]http://apple.co/1q1Ozsg

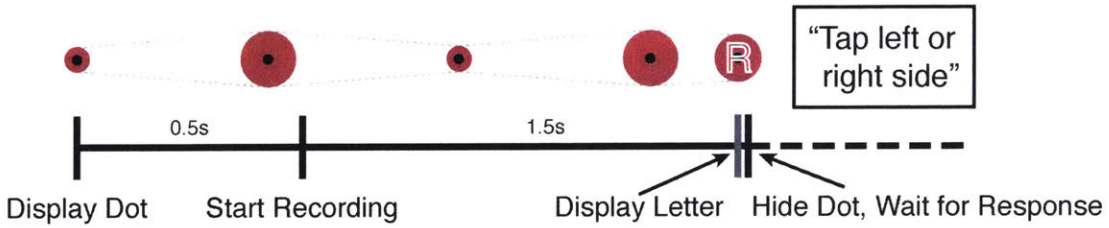


Figure 4-9: The timeline of the display of an individual dot. Dotted gray lines indicate how the dot changes size over time to keep attention.

imately 2s and we start the recording 0.5s after the dot moves to a new location to allow enough time for the worker to fixate at the dot location. Third, towards the end of the 2s window, a small letter, L or R is displayed for 0.05s—based on this letter, the worker is required to tap either the left (L) or right (R) side of the screen. This serves as a means to monitor the worker's attention and provide engagement with the application. If the worker taps the wrong side, they are warned and must repeat the dot again. Last, we use the real-time face detector built into iOS to ensure that the worker's face is visible in a large proportion of the recorded frames. This is critical as we cannot hope to track where someone is looking without a picture of their eyes.

Variability: In order to learn a robust eye tracking model, significant variability in the data is important. We believe that this variability is critical to achieving high-accuracy calibration-free eye tracking. Thus, we designed our setup to explicitly encourage high variability.

First, given our use of crowdsourcing, we expect to have a large variability in pose, appearance, and illumination. Second, to encourage further variability in pose, we tell the workers to continuously move their head and the distance of the phone relative to them by showing them an instructional video with a person doing the same. Last, we force workers to change the orientation of their mobile device after every 60 dots. This change can be detected using the built-in sensors on the device. This changes the relative position of the camera and the screen providing further variability.

Implementation details: Here, we provide some implementation details that

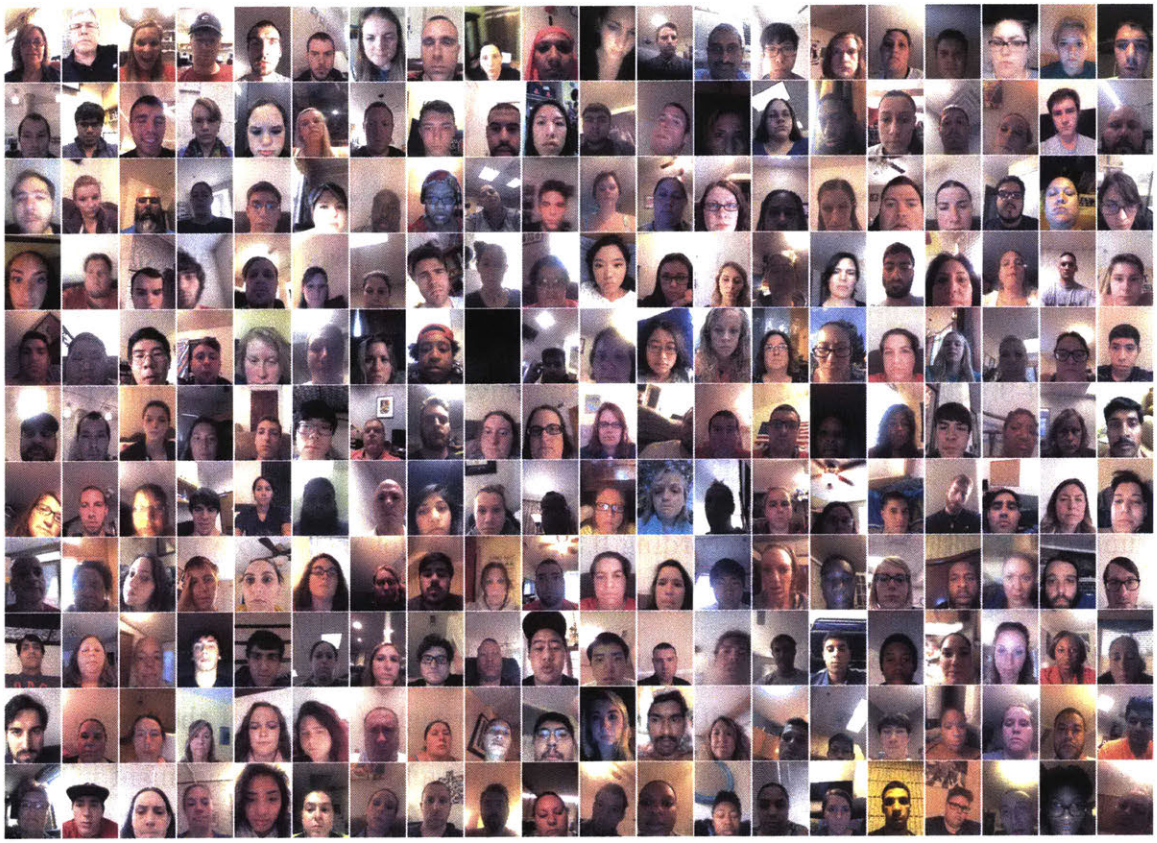


Figure 4-10: Sample frames from our GazeCapture dataset. Note the significant variation in illumination, head pose, appearance, and background. This variation allows us to learn robust models that generalize well to novel faces.

may be helpful for other researchers conducting similar studies. In order to associate each mobile device with an AMT task, we provided each worker with a unique code in AMT that they subsequently typed into their mobile application. The dot locations were both random and from 13 fixed locations (same locations as Figure 3 of [165])—we use the fixed locations to study the effect of calibration (Section 4.2.4). We displayed a total of 60 dots[3] for each orientation of the device[4] leading to a task duration of ~10min. Each worker was only allowed to complete the task once and we paid them $1–$1.50. We uploaded the data as individual frames rather than a video to avoid compression artifacts. Further, while we did not use it in this work, we also

[3]This was the number of dots displayed when the user entered a code provided via AMT. When the user did not enter a code (typical case when the application is downloaded directly from the App Store), they were shown 8 dots per orientation to keep them engaged.

[4]Three orientations for iPhones and four orientations for iPads following their natural use cases.

recorded device motion sensor data. We believe that this could be a useful resource for other researchers in the future.

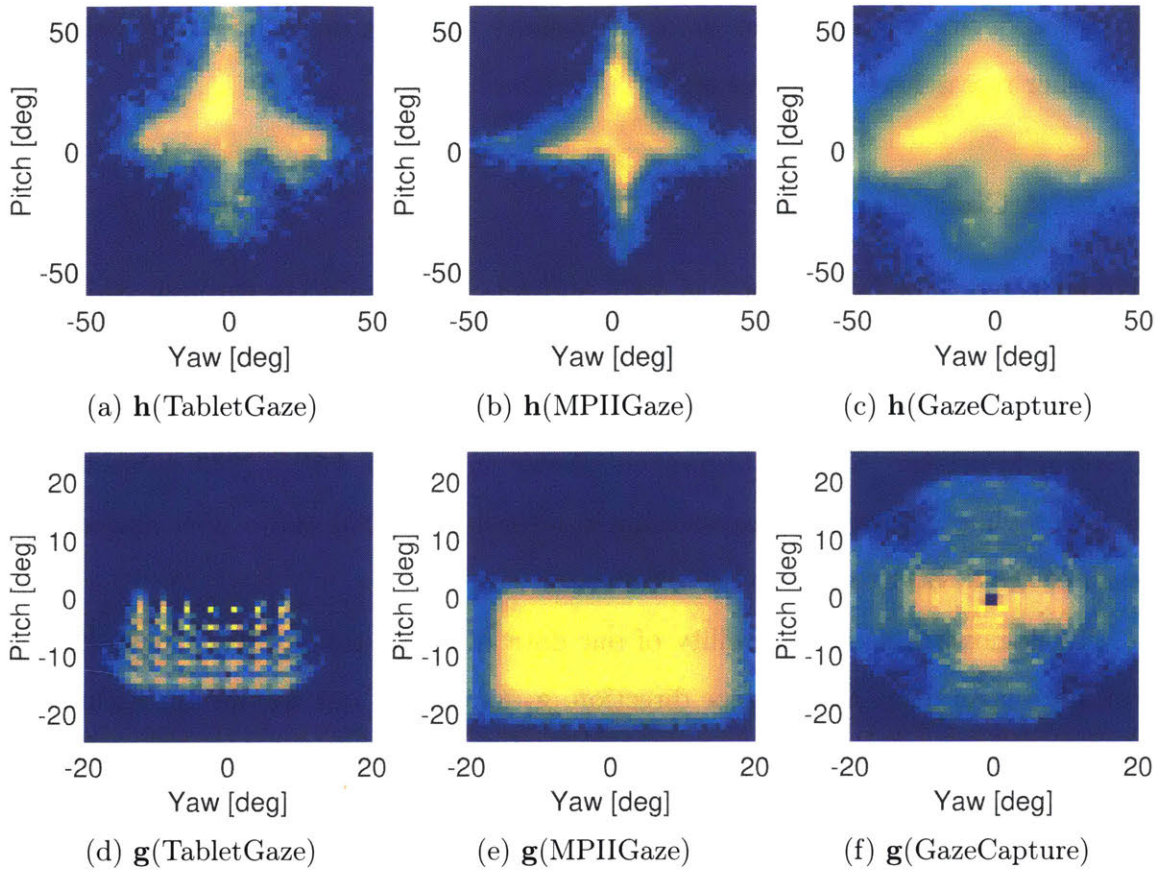
Dataset Characteristics

We collected data from a total of 1474 subjects: 1103 subjects through AMT, 230 subjects through in-class recruitment at UGA, and 141 subjects through other various App Store downloads. This resulted in a total of 2,445,504 frames with corresponding fixation locations. Sample frames are shown in Figure 4-10. 1249 subjects used iPhones while 225 used iPads, resulting in a total of $\sim$ 2.1M and $\sim$ 360k frames from each of the devices respectively.

To demonstrate the variability of our data, we used the approach from [171] to estimate head pose, $\mathbf{h}$, and gaze direction, $\mathbf{g}$, for each of our frames. In Figure 4-11 we plot the distribution of $\mathbf{h}$ and $\mathbf{g}$ on GazeCapture as well as existing state-of-the-art datasets, MPIIGaze [171] and TabletGaze [58]. We find that while our dataset contains a similar overall distribution of $\mathbf{h}$ there is a significantly larger proportion of outliers as compared to existing datasets. Further, we observe that our data capture technique from Section 4.2.2 introduces significant variation in the relative position of the camera to the user as compared to other datasets; *e.g.*, we have frames where the camera is mounted below the screen (*i.e.*, when the device is turned upside down) as well as above. These variations can be helpful for training and evaluating eye tracking approaches.

4.2.3 iTracker: A Deep Network for Eye Tracking

In this section, we describe our approach for building a robust eye tracker using our large-scale dataset, GazeCapture. Given the recent success of convolutional neural networks (CNNs) in computer vision, we use this approach to tackle the problem of eye tracking. We believe that, given enough data, we can learn eye tracking end-to-end without the need to include any manually engineered features, such as head pose [171]. In Section 4.2.3, we describe how we design an end-to-end CNN for robust



(a) **h**(TabletGaze) (b) **h**(MPIIGaze) (c) **h**(GazeCapture)

(d) **g**(TabletGaze) (e) **g**(MPIIGaze) (f) **g**(GazeCapture)

Figure 4-11: Distribution of head pose **h** (1st row) and gaze direction **g** relative to the head pose (2nd row) for datasets TabletGaze, MPIIGaze, and GazeCapture (ours). All intensities are logarithmic.

eye tracking. Then, in Section 4.2.3 we use the concept of *dark knowledge* [54] to learn a smaller network that achieves a similar performance while running at 10–15fps on a modern mobile device.

Learning an End-to-End Model

Our goal is to design an approach that can use the information from a single image to robustly predict gaze. We choose to use deep convolutional neural networks (CNNs) to make effective use of our large-scale dataset. Specifically, we provide the following as input to the model: (1) the image of the face together with its location in the image (termed *face grid*), and (2) the image of the eyes. We believe that using the model can (1) infer the head pose relative to the camera, and (2) infer the pose of

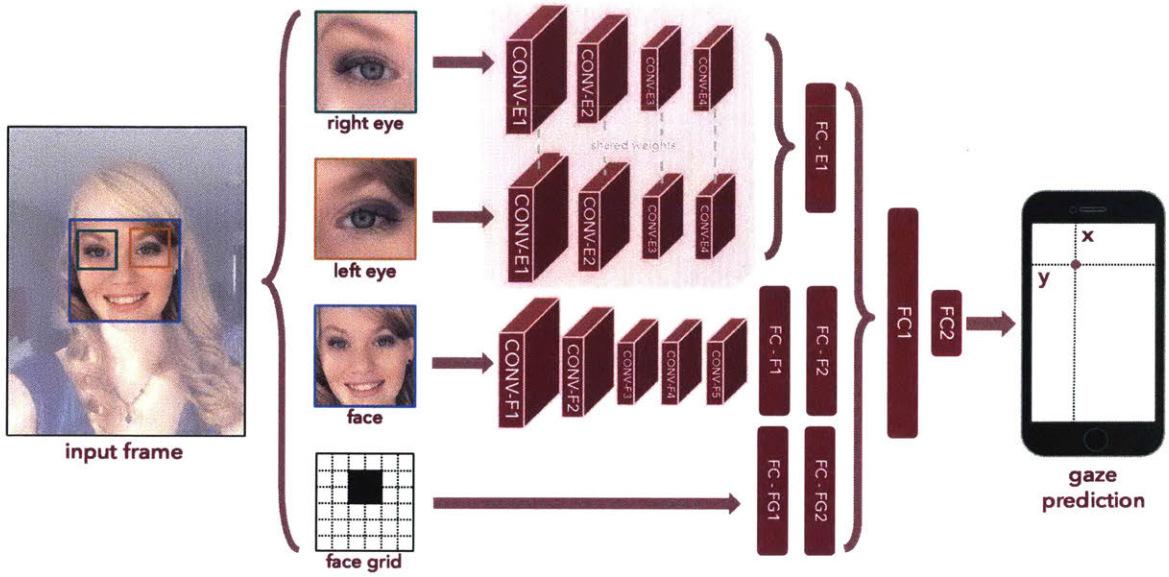


Figure 4-12: Overview of iTracker, our eye tracking CNN. Inputs include left eye, right eye, and face images detected and cropped from the original frame (all of size 224×224). The face grid input is a binary mask used to indicate the location and size of the head within the frame (of size 25×25). The output is the distance, in centimeters, from the camera. `CONV` represents convolutional layers (with filter size/number of kernels: `CONV-E1`,`CONV-F1`: $11 \times 11/96$, `CONV-E2`,`CONV-F2`: $5 \times 5/256$, `CONV-E3`,`CONV-F3`: $3 \times 3/384$, `CONV-E4`,`CONV-F4`: $1 \times 1/64$) while `FC` represents fully-connected layers (with sizes: `FC-E1`: 128, `FC-F1`: 128, `FC-F2`: 64, `FC-FG1`: 256, `FC-FG2`: 128, `FC1`: 128, `FC2`: 2). The exact model configuration is available on the project website.

the eyes relative to the head. By combining this information, the model can infer the location of gaze. Based on this information, we design the overall architecture of our iTracker network, as shown in Figure 4-12. The size of the various layers is similar to those of AlexNet [87]. Note that we include the eyes as individual inputs into the network (even though the face already contains them) to provide the network with a higher resolution image of the eye to allow it to identify subtle changes.

In order to best leverage the power of our large-scale dataset, we design a unified prediction space that allows us to train a single model using all the data. Note that this is not trivial since our data was collected using multiple devices at various orientations. Directly predicting screen coordinates would not be meaningful beyond a single device in a single orientation since the input could change significantly. Instead,

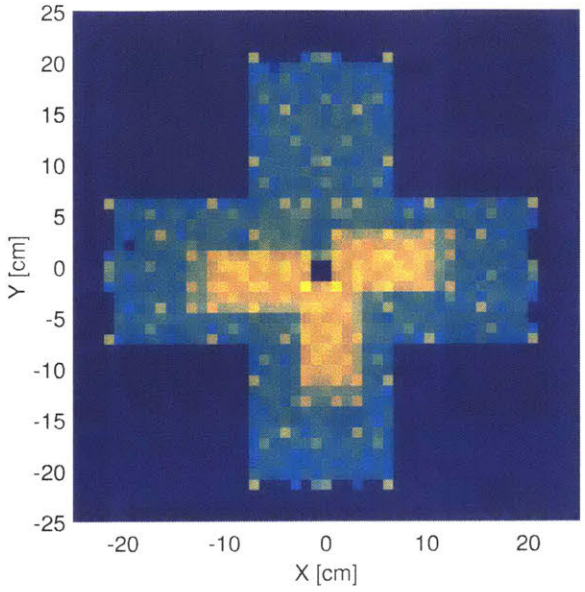


Figure 4-13: Our unified prediction space. The plot above shows the distribution of all dots in our dataset mapped to the prediction space. Axes denote centimeters from the camera; *i.e.*, all dots on the screen are projected to this space where the camera is at $(0, 0)$.

we leverage the fact that the front-facing camera is typically on the same plane as, and angled perpendicular to, the screen. As shown in Figure 4-13, we predict the dot location relative to the camera (in centimeters in the x and y direction). We obtain this through precise measurements of device screen sizes and camera placement. Finally, we train the model using a Euclidean loss on the x and y gaze position. The training parameters are provided in Section 4.2.4.

Further, after training the joint network, we found fine-tuning the network to each device and orientation helpful. This was particularly useful in dealing with the unbalanced data distribution between mobile phones and tablets. We denote this model as iTracker*.

Real-Time Inference

As our goal is to build an eye tracker that is practically useful, we provide evidence that our model can be applied on resource-constrained mobile devices. Encouraged by the work of Hinton *et al.* [54], we apply dark knowledge to reduce model complexity

and thus, computation time and memory footprint. First, while we designed the iTracker network to be robust to poor-quality eye detections, we use tighter crops (of size 80×80) produced by facial landmark eye detections [5] for the smaller network. These tighter crops focus the attention of the network on the more discriminative regions of the image, while also being faster due to the reduced image size. Then, we fine-tune the architecture configuration using the validation set to optimize efficiency without sacrificing much accuracy. Specifically, we have a combined loss on the ground truth, the predictions from our full model, as well as the features from the penultimate layer to assist the network in producing quality results. We implemented this model on an iPhone using Jetpac's Deep Belief SDK[5]. We found that the reduced version of the model took about 0.05s. to run on a iPhone 6s. Combining this with Apple's face detection pipeline, we can expect to achieve an overall detection rate of 10–15fps on a typical mobile device.

4.2.4 Experiments

In this section, we thoroughly evaluate the performance of iTracker using our large-scale GazeCapture dataset. Overall, we significantly outperform state-of-the-art approaches, achieving an average error of $\sim$ 2cm without calibration and are able to reduce this further to 1.8cm through calibration. Further, we demonstrate the importance of having a large-scale dataset as well as having variety in the data in terms of number of subjects rather than number of examples per subject. Then, we apply the features learned by iTracker to an existing dataset, TabletGaze [58], to demonstrate the generalization ability of our model.

Setup

Data preparation: First, from the 2,445,504 frames in GazeCapture, we select 1,490,959 frames that have both face and eye detections. These detections serve as important inputs to the model, as described in Section 4.2.3. This leads to a total

[5]https://github.com/jetpacapp/DeepBeliefSDK

of 1471 subjects being selected where each person has at least one frame with a valid detection. Then, we divide the dataset into train, validation, and test splits consisting of 1271, 50, and 150 subjects[6], respectively. For the validation and test splits, we only select subjects who looked at the full set of points. This ensures a uniform data distribution in the validation/test sets and allows us to perform a thorough evaluation on the impact of calibration across these subjects. Further, we evaluate the performance of our approach by augmenting the training and test set 25-fold by shifting the eyes and the face, changing face grid appropriately. For training, each of the augmented samples is treated independently while for testing, we average the predictions of the augmented samples to obtain the prediction on the original test sample (similar to [87]).

Implementation details: The model was implemented using Caffe [67]. It was trained from scratch on the GazeCapture dataset for 150,000 iterations with a batch size of 256. An initial learning rate of 0.001 was used, and after 75,000 iterations, it was reduced to 0.0001. Further, similar to AlexNet [87], we used a momentum of 0.9 and weight decay of 0.0005 throughout the training procedure. Further, we truncate the predictions based on the size of the device.

Evaluation metric: Similar to [58], we report the error in terms of average Euclidean distance (in centimeters) from the location of the true fixation. Further, given the different screen sizes, and hence usage distances of phones and tablets, we provide performance for both of these devices (even though the models used are exactly the same for both devices, unless otherwise specified). Lastly, to simulate a realistic use case where a stream of frames is processed for each given fixation rather than just a single frame, we report a value called *dot error*. In this case, the output of the classifier is given as the average prediction of all the frames corresponding to a gaze point at a certain location.

[6]Train, validation and test splits contain 1,251,983, 59,480, and 179,496 frames, respectively.

Unconstrained Eye Tracking

Here, our goal is to evaluate the generalization ability of iTracker to novel faces by evaluating it on unconstrained (calibration-free) eye tracking. As described in Section 4.2.4, we train and test iTracker on the appropriate splits of the data. To demonstrate the impact of performing data augmentation during train and test, we include the performance with and without train/test augmentation. As baseline, we apply the best performing approach (pre-trained ImageNet model) on TabletGaze (Section 4.2.4) to GazeCapture. The results are summarized in the top half of Table 4.3 and the error distribution is plotted in Figure 4-14.

We observe that our model consistently outperforms the baseline approach by a large margin, achieving an error as low as 1.53cm and 2.38cm on mobile phones and tablets respectively. Further, we find that the *dot error* is consistently lower than the *error* demonstrating the advantage of using temporal averaging in real-world eye tracking applications. Also note that both train and test augmentation are helpful for reducing the prediction error. While test augmentation may not allow for real-time performance, train augmentation can be used to learn a more robust model. Last, we observe that fine-tuning the general iTracker model to each device and orientation (iTracker*) is helpful for further reducing errors, especially for tablets. This is to be expected, given the large proportion of samples from mobile phones (85%) as compared to tablets (15%) in GazeCapture.

Eye Tracking with Calibration

As mentioned in Section 4.2.2, we collect data from 13 fixed dot locations (per device orientation) for each subject. We use these locations to simulate the process of calibration. For each subject in the test set, we use frames from these 13 fixed locations for training, and evaluate on the remaining locations. Specifically, we extract features from the `fc1` layer of iTracker and train a model using SVR to predict each subject's gaze locations. The results are summarized in Table 4.4. We observe that the performance decreases slightly when given few points for calibration. This likely

Model	Augmentation	Mobile phone		Tablet	
		error	dot error	error	dot error
Baseline	train + test	2.99	2.40	5.13	4.54
iTracker	none	2.04	1.62	3.32	2.82
iTracker	test	1.84	1.58	3.21	2.90
iTracker	train	1.86	1.57	2.81	2.47
iTracker	train + test	1.77	**1.53**	2.83	2.53
iTracker*	train + test	**1.71**	**1.53**	**2.53**	**2.38**
iTracker (no eyes)	none	2.11	1.72	3.40	2.93
iTracker (no face)	none	2.15	1.69	3.45	2.92
iTracker (no fg.)	none	2.23	1.81	3.90	3.36

Table 4.3: Unconstrained eye tracking results (top half) and ablation study (bottom half). The error and dot error values are reported in centimeters (see Section 4.2.4 for details); lower is better. *Baseline* refers to applying support vector regression (SVR) on features from a pre-trained ImageNet network, as done in Section 4.2.4. We found that this method outperformed all existing approaches. For the ablation study (Section 4.2.4), we removed each critical input to our model, namely eyes, face and face grid (*fg.*), one at a time and evaluated its performance.

occurs due to overfitting when training the SVR. However, when using the full set of 13 points for calibration, the performance improves significantly, achieving an error of 1.34cm and 2.12cm on mobile phones and tablets, respectively.

Cross-Dataset Generalization

We evaluate the generalization ability of the features learned by iTracker by applying them to another dataset, TabletGaze [58]. TabletGaze contains recordings from a total of 51 subjects and a sub-dataset of 40 usable subjects[7]. We split this set of 40 subjects into 32 for training and 8 for testing. We apply support vector regression (SVR) to the features extracted using iTracker to predict the gaze locations in this dataset, and apply this trained classifier to the test set. The results are shown in Table 4.5. We report the performance of applying various state-of-the-art approaches (TabletGaze [58], TurkerGaze [165] and MPIIGaze [171]) and other baseline methods for comparison. We propose two simple baseline methods: (1) center prediction (*i.e.*, always predicting the center of the screen regardless of the data) and (2) apply-

[7] [58] mentions 41 usable subjects but at the time of the experiments, only data from 40 of them was released.

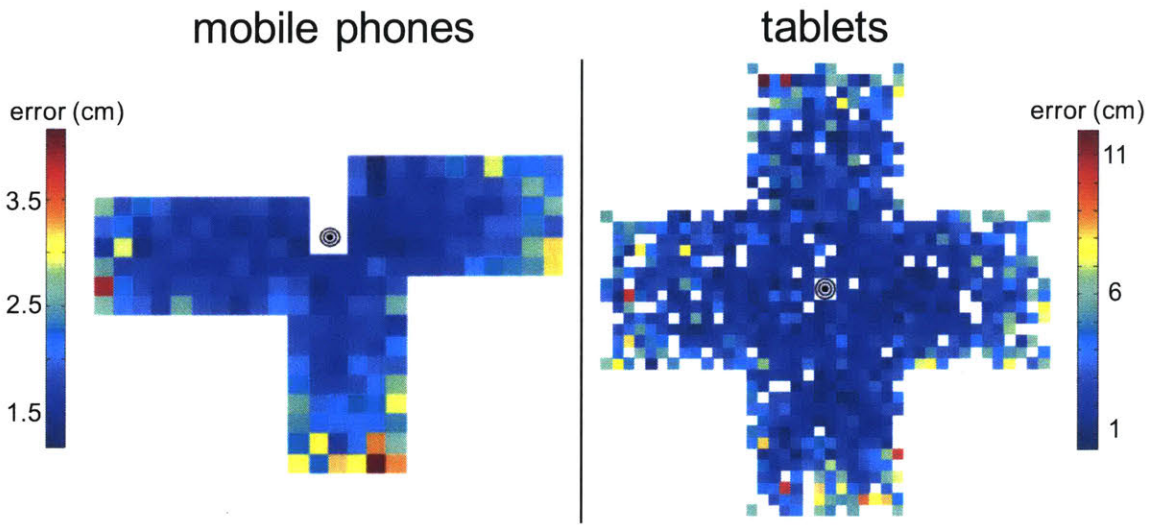


Figure 4-14: Distribution of error for iTracker (with train and test augmentation) across the prediction space, plotted at ground truth location. The black and white circles represent the location of the camera. We observe that the error near the camera tends to be lower.

ing support vector regression (SVR) to image features extracted using AlexNet [87] pre-trained on ImageNet [134]. Interestingly, we find that the AlexNet + SVR approach outperforms all existing state-of-the-art approaches despite the features being trained for a completely different task. Importantly, we find that the features from iTracker significantly outperform all existing approaches to achieve an error of 2.58cm demonstrating the generalization ability of our features.

Analysis

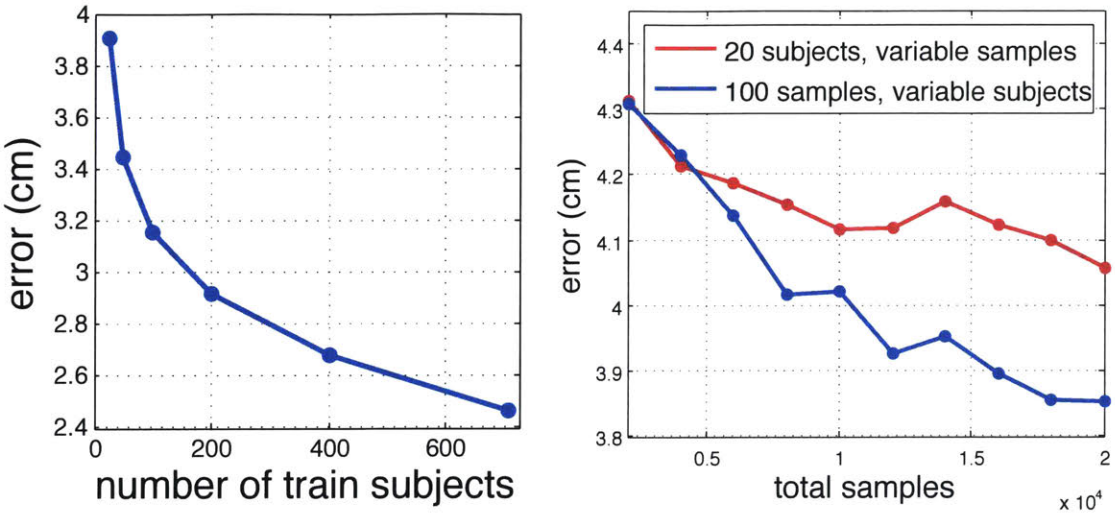
Ablation study: In the bottom half of Table 4.3 we report the performance after removing different components of our model, one at a time, to better understand their significance. In general, all three inputs (eyes, face, and face grid) contribute to the performance of our model. Interestingly, the mode with face but no eyes achieves comparable performance to our full model suggesting that we may be able to design a more efficient approach that requires only the face and face grid as input. We believe the large-scale data allows the CNN to effectively identify the fine-grained differences across people's faces (their eyes) and hence make accurate predictions.

Model	# calibration points	Mobile phone		Tablet	
		error	dot error	error	dot error
iTracker	0	1.77	1.53	2.83	2.53
	4	1.92	1.71	4.41	4.11
	5	1.76	1.50	3.50	3.13
	9	1.64	1.33	3.04	2.59
	13	**1.56**	**1.26**	**2.81**	**2.38**
iTracker*	0	1.71	1.53	2.53	2.38
	4	1.65	1.42	3.12	2.96
	5	1.52	1.22	2.56	2.30
	9	1.41	1.10	2.29	1.87
	13	**1.34**	**1.04**	**2.12**	**1.69**

Table 4.4: Performance of iTracker using different numbers of points for calibration (error and dot error in centimeters; lower is better). Calibration significantly improves performance.

Method	Error (cm)	Description
Center	7.54	Simple baseline
TurkerGaze [165]	4.77	pixel features + SVR
TabletGaze	4.04	Our implementation of [58]
MPIIGaze [171]	3.63	CNN + head pose
TabletGaze[58]	3.17	Random forest + mHoG
AlexNet [87]	3.09	eyes (`conv3`) + face (`fc6`) + fg.
iTracker (ours)	**2.58**	`fc1` of iTracker + SVR

Table 4.5: Result of applying various state-of-the-art approaches to TabletGaze [58] dataset. For the AlexNet + SVR approach, we train a SVR on the concatenation of features from various layers of AlexNet (`conv3` for eyes and `fc6` for face) and a binary face grid (*fg.*).



(a) No. of subjects *vs.*error (b) Subjects *vs.*samples

Figure 4-15: Dataset size is important for achieving low error. Specifically, growing the number of subjects in a dataset is more important than the number of samples, which further motivates the use of crowdsourcing.

Importance of large-scale data: In Figure 4-15a we plot the performance of iTracker as we increase the total number of train subjects. We find that the error decreases significantly as the number of subjects is increased, illustrating the importance of gathering a large-scale dataset. Further, to illustrate the importance of having variability in the data, in Figure 4-15b, we plot the performance of iTracker as (1) the number of subjects is increased while keeping the number of samples per subject constant (in blue), and (2) the number of samples per subject is increased while keeping the number of subjects constant (in red). In both cases the total number of samples is kept constant to ensure the results are comparable. We find that the error decreases significantly more quickly as the number of subjects is increased indicating the importance of having variability in the data.

4.2.5 Summary

In this section, we introduced an end-to-end eye tracking solution targeting mobile devices. First, we introduced GazeCapture, the first large-scale mobile eye tracking dataset. We demonstrated the power of crowdsourcing to collect gaze data, a method

unexplored by prior works. We demonstrated the importance of both having a large-scale dataset, as well as having a large variety of data to be able to train robust models for eye tracking. Then, using GazeCapture we trained iTracker, a deep convolutional neural network for predicting gaze. Through careful evaluation, we show that iTracker is capable of robustly predicting gaze, achieving an error as low as 1.04cm and 1.69cm on mobile phones and tablets respectively. Further, we demonstrate that the features learned by our model generalize well to existing datasets, outperforming state-of-the-art approaches by a large margin. Though eye tracking has been around for centuries, we believe that this work will serve as a key benchmark for the next generation of eye tracking solutions. We hope that through this work, we can bring the power of eye tracking to everyone.

Chapter 5

Visualizing and Understanding Convolutional Neural Networks

Current deep neural networks achieve remarkable performance at a number of vision tasks surpassing techniques based on hand-crafted features. However, while the structure of the representation in hand-crafted features is often clear and interpretable, in the case of deep networks it remains unclear what the nature of the learned representation is and why it works so well. A convolutional neural network (CNN) trained on ImageNet [27] significantly outperforms the best hand crafted features on the ImageNet challenge [134]. But more surprisingly, the same network, when used as a generic feature extractor, is also very successful at other tasks like object detection on the PASCAL VOC dataset [36].

A number of works have focused on understanding the representation learned by CNNs. The work by [169] introduces a procedure to visualize what activates each unit. Recently [168] use transfer learning to measure how generic/specific the learned features are. In [1] and [148], they suggest that the CNN for ImageNet learns a distributed code for objects. They all use ImageNet, an object-centric dataset, as a training set.

When training a CNN to distinguish different object classes, it is unclear what the underlying representation should be. Objects have often been described using part-based representations where parts can be shared across objects, forming a distributed

code. However, what those parts should be is unclear. For instance, one would think that the meaningful parts of a face are the mouth, the two eyes, and the nose. However, those are simply functional parts, with words associated with them; the object parts that are important for visual recognition might be different from these semantic parts, making it difficult to evaluate how efficient a representation is. In fact, the strong internal configuration of objects makes the definition of what is a useful part poorly constrained: an algorithm can find different and arbitrary part configurations, all giving similar recognition performance.

Learning to classify scenes (i.e., classifying an image as being an office, a restaurant, a street, etc) using the Places dataset [174] gives the opportunity to study the internal representation learned by a CNN on a task other than object recognition.

In the case of scenes, the representation is clearer. Scene categories are defined by the objects they contain and, to some extent, by the spatial configuration of those objects. For instance, the important parts of a bedroom are the bed, a side table, a lamp, a cabinet, as well as the walls, floor and ceiling. Objects represent therefore a distributed code for scenes (i.e., object classes are shared across different scene categories). Importantly, in scenes, the spatial configuration of objects, although compact, has a much larger degree of freedom. It is this loose spatial dependency that, we believe, makes scene representation different from most object classes (most object classes do not have a loose interaction between parts). In addition to objects, other feature regularities of scene categories allow for other representations to emerge, such as textures [130], GIST [113], bag-of-words [90], part-based models [117], and ObjectBank [95]. While a CNN has enough flexibility to learn any of those representations, if meaningful objects emerge without supervision inside the inner layers of the CNN, there will be little ambiguity as to which type of representation these networks are learning.

The main contribution of this chapter is to show that object detection emerges inside a CNN trained to recognize scenes, even more than when trained with ImageNet. This is surprising because our results demonstrate that reliable object detectors are found even though, unlike ImageNet, no supervision is provided for objects. Although

object discovery with deep neural networks has been shown before in an unsupervised setting [91], here we find that many more objects can be naturally discovered, in a supervised setting tuned to scene classification rather than object classification.

Importantly, the emergence of object detectors inside the CNN suggests that a single network can support recognition at several levels of abstraction (e.g., edges, texture, objects, and scenes) without needing multiple outputs or a collection of networks. Whereas other works have shown that one can detect objects by applying the network multiple times in different locations [47], or focusing attention [152], or by doing segmentation [49, 38], here we show that the same network can do both object localization and scene recognition in a single forward-pass. Another set of recent works [115, 8] demonstrate the ability of deep networks trained on object classification to do localization without bounding box supervision. However, unlike our work, these require object-level supervision while we only use scenes.

5.1 ImageNet-CNN and Places-CNN

Convolutional neural networks have recently obtained astonishing performance on object classification [87] and scene classification [174]. The ImageNet-CNN from [67] is trained on 1.3 million images from 1000 object categories of ImageNet (ILSVRC 2012) and achieves a top-1 accuracy of 57.4%. With the same network architecture, Places-CNN is trained on 2.4 million images from 205 scene categories of Places Database [174], and achieves a top-1 accuracy of 50.0%. The network architecture used for both CNNs, as proposed in [87], is summarized in Table 5.1[1]. Both networks are trained from scratch using only the specified dataset.

The deep features from Places-CNN tend to perform better on scene-related recognition tasks compared to the features from ImageNet-CNN. For example, as compared to the Places-CNN that achieves 50.0% on scene classification, the ImageNet-CNN combined with a linear SVM only achieves 40.8% on the same test set[2] illustrating

[1]We use *unit* to refer to neurons in the various layers and *features* to refer to their activations.

[2]Scene recognition demo of Places-CNN is available at `http://places.csail.mit.edu/demo.html`. The demo has 77.3% top-5 recognition rate in the wild estimated from 968 anonymous user

Layer	conv1	pool1	conv2	pool2	conv3	conv4	conv5	pool5	fc6	fc7
Units	96	96	256	256	384	384	256	256	4096	4096
Feature	55×55	27×27	27×27	13×13	13×13	13×13	13×13	6×6	1	1

Table 5.1: The parameters of the network architecture used for ImageNet-CNN and Places-CNN.

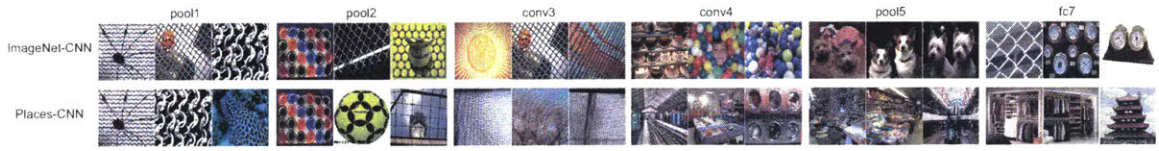


Figure 5-1: Top 3 images producing the largest activation of units in each layer of ImageNet-CNN (top) and Places-CNN (bottom).

the importance of having scene-centric data.

To further highlight the difference in representations, we conduct a simple experiment to identify the differences in the type of images preferred at the different layers of each network: we create a set of 200k images with an approximately equal distribution of scene-centric and object-centric images[3], and run them through both networks, recording the activations at each layer. For each layer, we obtain the top 100 images that have the largest average activation (sum over all spatial locations for a given layer). Figure 5-1 shows the top 3 images for each layer. We observe that the earlier layers such as `pool1` and `pool2` prefer similar images for both networks while the later layers tend to be more specialized to the specific task of scene or object categorization. For layer `pool2`, 55% and 47% of the top-100 images belong to the ImageNet dataset for ImageNet-CNN and Places-CNN. Starting from layer conv4, we observe a significant difference in the number of top-100 belonging to each dataset corresponding to each network. For `fc7`, we observe that 78% and 24% of the top-100 images belong to the ImageNet dataset for the ImageNet-CNN and Places-CNN respectively, illustrating a clear bias in each network.

In the following sections, we further investigate the differences between these networks, and focus on better understanding the nature of the representation learned by

responses.

[3]100k object-centric images from the test set of ImageNet LSVRC2012 and 108k scene-centric images from the SUN dataset [164].

Places-CNN when doing scene classification in order to clarify some part of the secret to their great performance.

5.2 Uncovering the CNN representation

The performance of scene recognition using Places-CNN is quite impressive given the difficulty of the task. In this section, our goal is to understand the nature of the representation that the network is learning.

5.2.1 Simplifying the input images

Simplifying images is a well known strategy to test human recognition. For example, one can remove information from the image to test if it is diagnostic or not of a particular object or scene (for a review see [10]). A similar procedure was also used by [151] to understand the receptive fields of complex cells in the inferior temporal cortex (IT).

Inspired by these approaches, our idea is the following: given an image that is correctly classified by the network, we want to simplify this image such that it keeps as little visual information as possible while still having a high classification score for the same category. This simplified image (named minimal image representation) will allow us to highlight the elements that lead to the high classification score. In order to do this, we manipulate images in the gradient space as typically done in computer graphics [121]. We investigate two different approaches described below.

In the first approach, given an image, we create a segmentation of edges and regions and remove segments from the image iteratively. At each iteration we remove the segment that produces the smallest decrease of the correct classification score and we do this until the image is incorrectly classified. At the end, we get a representation of the original image that contains, approximately, the minimal amount of information needed by the network to correctly recognize the scene category. In Figure 5-2 we show some examples of these minimal image representations. Notice that objects seem to contribute important information for the network to recognize the scene. For

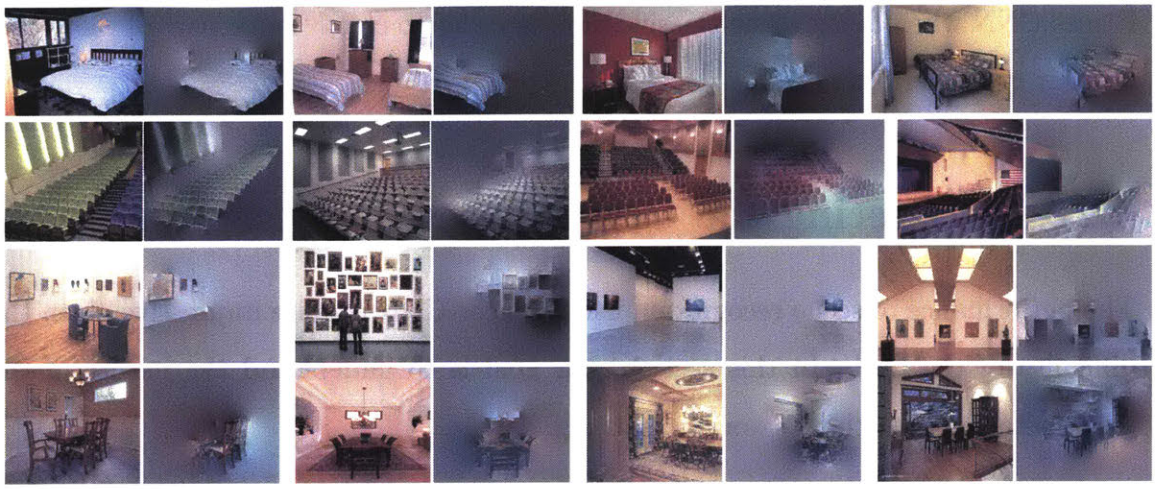


Figure 5-2: Each pair of images shows the original image (left) and a simplified image (right) that gets classified by the Places-CNN as the same scene category as the original image. From top to bottom, the four rows show different scene categories: bedroom, auditorium, art gallery, and dining room.

instance, in the case of bedrooms these minimal image representations usually contain the region of the bed, or in the art gallery category, the regions of the paintings on the walls.

Based on the previous results, we hypothesized that for the Places-CNN, some objects were crucial for recognizing scenes. This inspired our second approach: we generate the minimal image representations using the fully annotated image set of SUN Database [164] (see Section 5.3.1 for details on this dataset) instead of performing automatic segmentation. We follow the same procedure as the first approach using the ground-truth object segments provided in the database.

This led to some interesting observations: for bedrooms, the minimal representations retained the bed in 87% of the cases. Other objects kept in bedrooms were wall (28%) and window (21%). For art gallery the minimal image representations contained paintings (81%) and pictures (58%); in amusement parks, carousel (75%), ride (64%), and roller coaster (50%); in bookstore, bookcase (96%), books (68%), and shelves (67%). These results suggest that object detection is an important part of the representation built by the network to obtain discriminative information for scene classification.

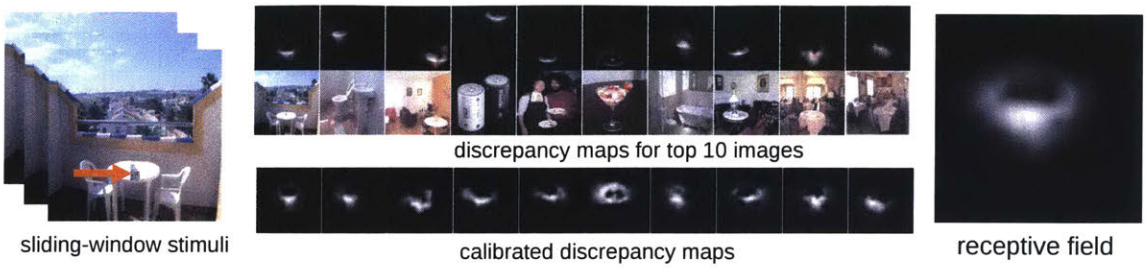


Figure 5-3: The pipeline for estimating the RF of each unit. Each sliding-window stimuli contains a small randomized patch (example indicated by red arrow) at different spatial locations. By comparing the activation response of the sliding-window stimuli with the activation response of the original image, we obtain a discrepancy map for each image (middle top). By summing up the calibrated discrepancy maps (middle bottom) for the top ranked images, we obtain the actual RF of that unit (right).

5.2.2 Visualizing the receptive fields of units and their activation patterns

In this section, we investigate the shape and size of the receptive fields (RFs) of the various units in the CNNs. While theoretical RF sizes can be computed given the network architecture [98], we are interested in the actual, or *empirical* size of the RFs. We expect the empirical RFs to be better localized and more representative of the information they capture than the theoretical ones, allowing us to better understand what is learned by each unit of the CNN.

Thus, we propose a data-driven approach to estimate the learned RF of each unit in each layer. It is simpler than the deconvolutional network visualization method [169] and can be easily extended to visualize any learned CNNs[4].

The procedure for estimating a given unit's RF, as illustrated in Figure 5-3, is as follows. As input, we use an image set of 200k images with a roughly equal distribution of scenes and objects (similar to Section 5.1). Then, we select the top K images with the highest activations for the given unit.

For each of the K images, we now want to identify exactly which regions of the image lead to the high unit activations. To do this, we replicate each image many times with small random occluders (image patches of size 11×11) at different locations

[4]More visualizations are available at `http://places.csail.mit.edu/visualization`

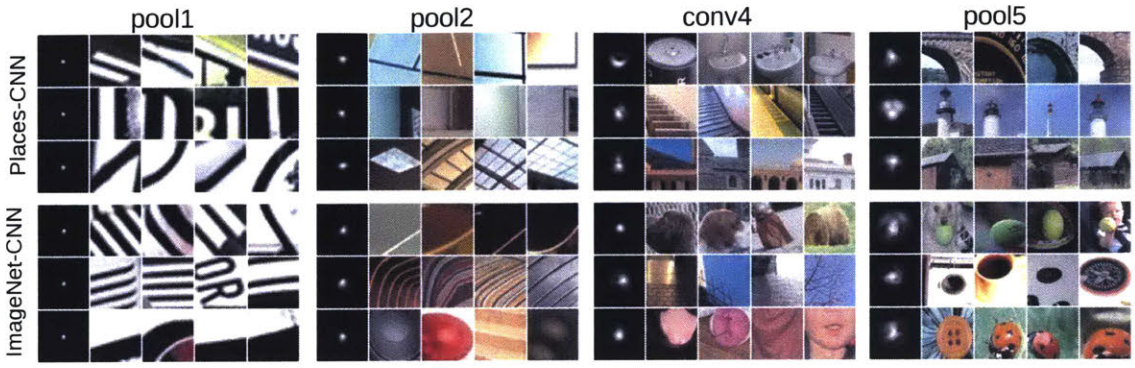


Figure 5-4: The RFs of 3 units of `pool1`, `pool2`, `conv4`, and `pool5` layers respectively for ImageNet- and Places-CNNs, along with the image patches corresponding to the top activation regions inside the RFs.

in the image. Specifically, we generate occluders in a dense grid with a stride of 3. This results in about 5000 occluded images per original image. We now feed all the occluded images into the same network and record the change in activation as compared to using the original image. If there is a large discrepancy, we know that the given patch is important and vice versa. This allows us to build a discrepancy map for each image.

Finally, to consolidate the information from the K images, we center the discrepancy map around the spatial location of the unit that caused the maximum activation for the given image. Then we average the re-centered discrepancy maps to generate the final RF.

In Figure 5-4 we visualize the RFs for units from 4 different layers of the Places-CNN and ImageNet-CNN, along with their highest scoring activation regions inside the RF. We observe that, as the layers go deeper, the RF size gradually increases and the activation regions become more semantically meaningful. Further, as shown in Figure 5-5, we use the RFs to segment images using the feature maps of different units. Lastly, in Table 5.2, we compare the theoretical and empirical size of the RFs at different layers. As expected, the actual size of the RF is much smaller than the theoretical size, especially in the later layers. Overall, this analysis allows us to better understand each unit by focusing precisely on the important regions of each image.

	pool1	pool2	conv3	conv4	pool5
Theoretical size	19	67	99	131	195
Places-CNN	17.8± 1.6	37.4± 5.9	52.1±10.6	60.0± 13.7	72.0± 20.0
ImageNet-CNN	17.9± 1.6	36.7± 5.4	51.1±9.9	60.4± 16.0	70.3± 21.6

Table 5.2: Comparison of the theoretical and empirical sizes of the RFs for Places-CNN and ImageNet-CNN at different layers. Note that the RFs are assumed to be square shaped, and the sizes reported below are the length of each side of this square, in pixels.

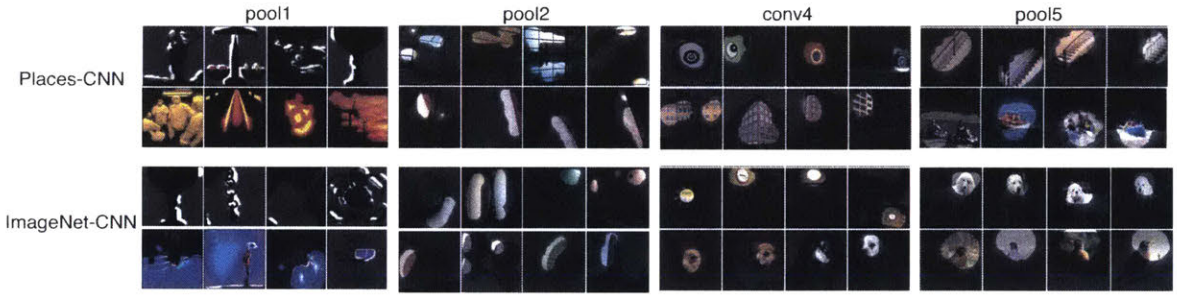


Figure 5-5: Segmentation based on RFs. Each row shows the 4 most confident images for some unit.

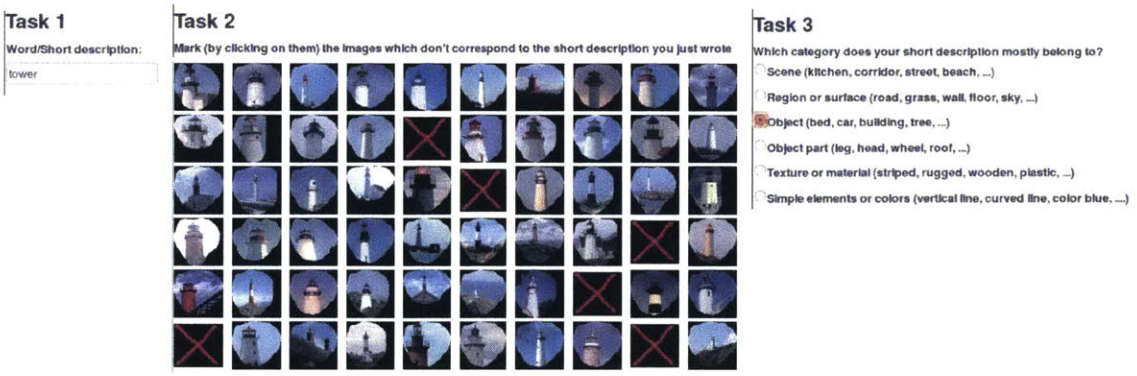


Figure 5-6: AMT interface for unit concept annotation. There are three tasks in each annotation.

5.2.3 Identifying the semantics of internal units

In Section 5.2.2, we found the exact RFs of units and observed that activation regions tended to become more semantically meaningful with increasing depth of layers. In this section, our goal is to understand and quantify the precise semantics learned by each unit.

In order to do this, we ask workers on Amazon Mechanical Turk (AMT) to identify the common theme or *concept* that exists between the top scoring segmentations for each unit. We expect the tags provided by naive annotators to reduce biases. Workers provide tags without being constrained to a dictionary of terms that could bias or limit the identification of interesting properties.

Specifically, we divide the task into three main steps as shown in Figure 5-6. We show workers the top 60 segmented images that most strongly activate one unit and we ask them to (1) identify the concept, or semantic theme given by the set of 60 images e.g., car, blue, vertical lines, etc, (2) mark the set of images that do not fall into this theme, and (3) categorize the concept provided in (1) to one of 6 semantic groups ranging from low-level to high-level: simple elements and colors (e.g., horizontal lines, blue), materials and textures (e.g., wood, square grid), regions ans surfaces (e.g., road, grass), object parts (e.g., head, leg), objects (e.g., car, person), and scenes (e.g., kitchen, corridor). This allows us to obtain both the semantic information for each unit, as well as the level of abstraction provided by the labeled concept.

To ensure high quality of annotation, we included 3 images with high negative scores that the workers were required to identify as negatives in order to submit the task. Figure 5-7 shows some example annotations by workers. For each unit, we measure its precision as the percentage of images that were selected as fitting the labeled concept. In Figure 5-8.(a) we plot the average precision for ImageNet-CNN and Places-CNN for each layer.

In Figure 5-8.(b-c) we plot the distribution of concept categories for ImageNet-CNN and Places-CNN at each layer. For this plot we consider only units that had

a precision above 75% as provided by the AMT workers. Around 60% of the units on each layer where above that threshold. For both networks, units at the early layers (pool1, pool2) have more units responsive to simple elements and colors, while those at later layers (conv4, pool5) have more high-level semantics (responsive more to objects and scenes). Furthermore, we observe that conv4 and pool5 units in Places-CNN have higher ratios of high-level semantics as compared to the units in ImageNet-CNN.

Figure 5-9 provides a different visualization of the same data as in Figure 5-8.(b-c). This plot better reveals how different levels of abstraction emerge in different layers of both networks. The vertical axis indicates the percentage of units in each layer assigned to each concept category. ImageNet-CNN has more units tuned to simple elements and colors than Places-CNN while Places-CNN has more objects and scenes. ImageNet-CNN has more units tuned to object parts (with the maximum around conv4). It is interesting to note that Places-CNN discovers more objects than ImageNet-CNN despite having no object-level supervision.

5.3 Emergence of objects as the internal representation

As shown before, a large number of units in pool5 are devoted to detecting objects and scene-regions (Figure 5-9). But what categories are found? Is each category mapped to a single unit or are there multiple units for each object class? Can we actually use this information to segment a scene?

5.3.1 What object classes emerge?

To answer the question of why certain objects emerge from pool5, we tested ImageNet-CNN and Places-CNN on fully annotated images from the SUN database [164]. The SUN database contains 8220 fully annotated images from the same 205 place categories used to train Places-CNN. There are no duplicate images between SUN and

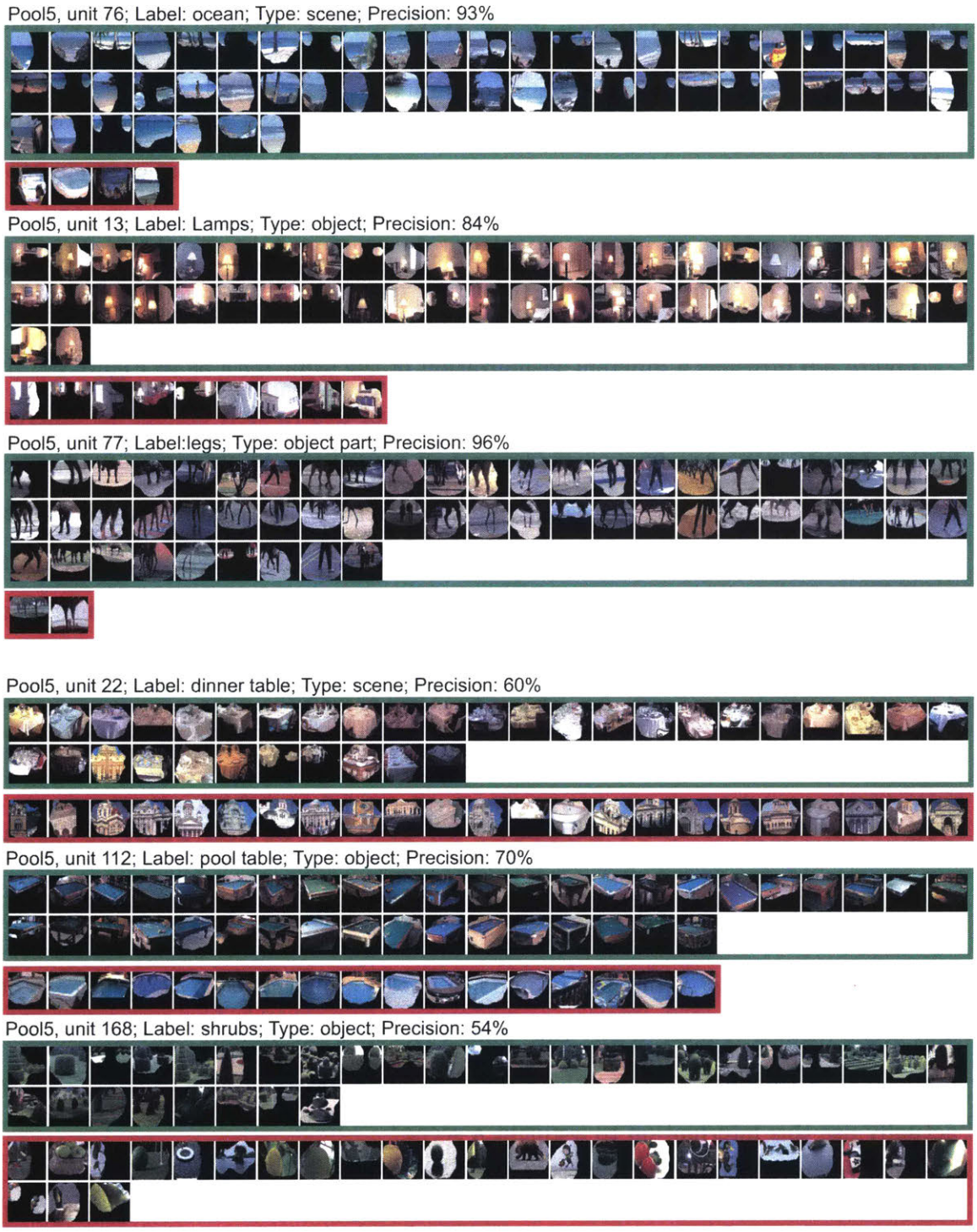


Figure 5-7: Examples of unit annotations provided by AMT workers for 6 units from `pool5` in Places-CNN. For each unit the figure shows the label provided by the worker, the type of label, the images selected as corresponding to the concept (green box) and the images marked as incorrect (red box). The precision is the percentage of correct images. The top three units have high performance while the bottom three have low performance ($< 75\%$).

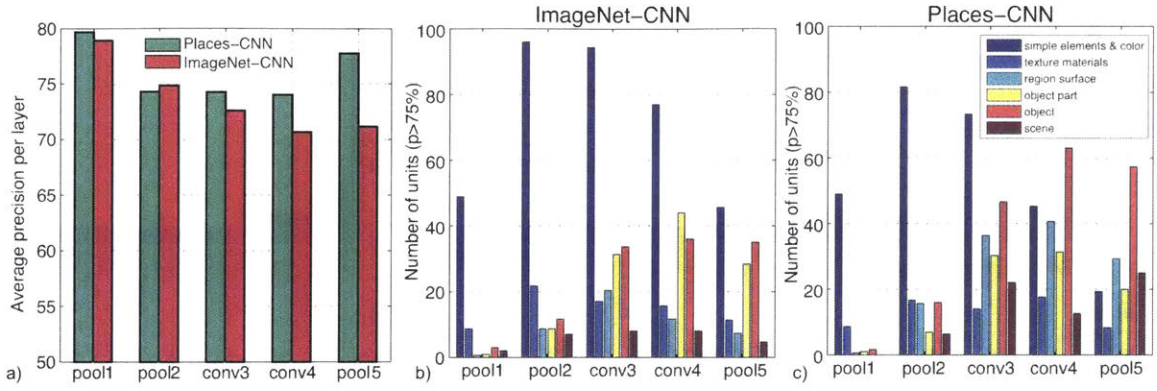


Figure 5-8: (a) Average precision of all the units in each layer for both networks as reported by AMT workers. (b) and (c) show the number of units providing different levels of semantics for ImageNet-CNN and Places-CNN respectively.

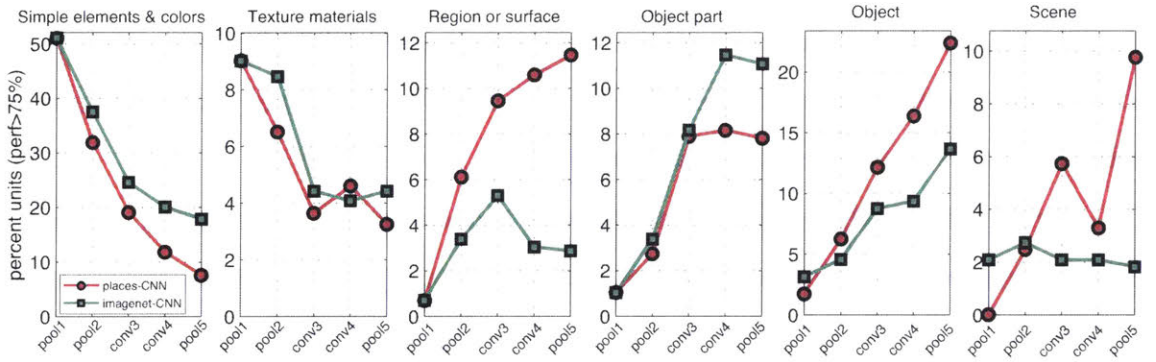


Figure 5-9: Distribution of semantic types found for all the units in both networks. From left to right, each plot corresponds to the distribution of units in each layer assigned to simple elements or colors, textures or materials, regions or surfaces, object parts, objects, and scenes. The vertical axis is the percentage of units with each layer assigned to each type of concept.

Places. We use SUN instead of COCO [96] as we need dense object annotations to study what the most informative object classes for scene categorization are, and what the natural object frequencies in scene images are. For this study, we manually mapped the tags given by AMT workers to the SUN categories.

Figure 5-10(a) shows the distribution of objects found in `pool5` of Places-CNN. Some objects are detected by several units. For instance, there are 15 units that detect buildings. Figure 5-11 shows some units from the Places-CNN grouped by the type of object class they seem to be detecting. Each row shows the top five images for a particular unit that produce the strongest activations. The segmentation shows

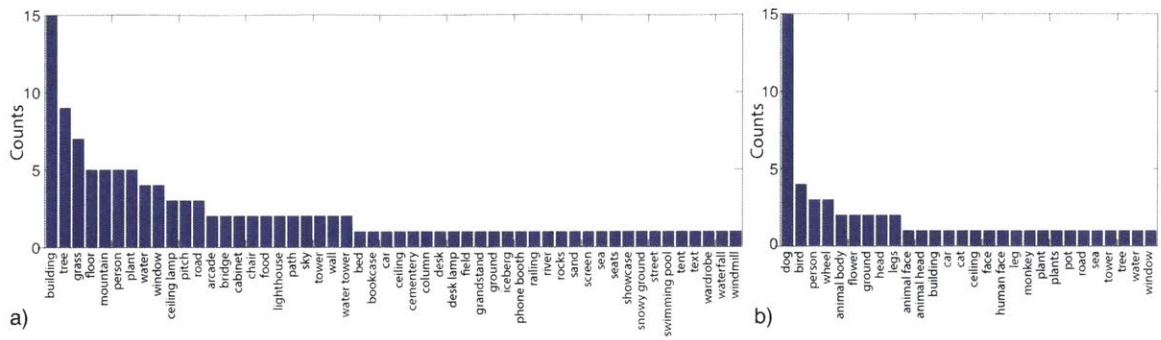


Figure 5-10: Object counts of CNN units discovering each object class for (a) Places-CNN and (b) ImageNet-CNN.

the regions of the image for which the unit is above a certain threshold. Each unit seems to be selective to a particular appearance of the object. For instance, there are 6 units that detect lamps, each unit detecting a particular type of lamp providing finer-grained discrimination; there are 9 units selective to people, each one tuned to different scales or people doing different tasks.

Figure 5-10(b) shows the distribition of objects found in `pool5` of ImageNet-CNN. ImageNet has an abundance of animals among the categories present: in the ImageNet-CNN, out of the 256 units in `pool5`, there are 15 units devoted to detecting dogs and several more detecting parts of dogs (body, legs, ...). The categories found in `pool5` tend to follow the target categories in ImageNet.

Why do those objects emerge? One possibility is that the objects that emerge in `pool5` correspond to the most frequent ones in the database. Figure 5-12(a) shows the sorted distribution of object counts in the SUN database which follows Zipf's law. Figure 5-12(b) shows the counts of units found in `pool5` for each object class (same sorting as in Figure 5-12(a)). The correlation between object frequency in the database and object frequency discovered by the units in `pool5` is 0.54. Another possibility is that the objects that emerge are the objects that allow discriminating among scene categories. To measure the set of discriminant objects we used the ground truth in the SUN database to measure the classification performance achieved by each object class for scene classification. Then we count how many times each object class appears as the most informative one. This measures the number of scene categories a particular object class is the most useful for. The counts are shown in

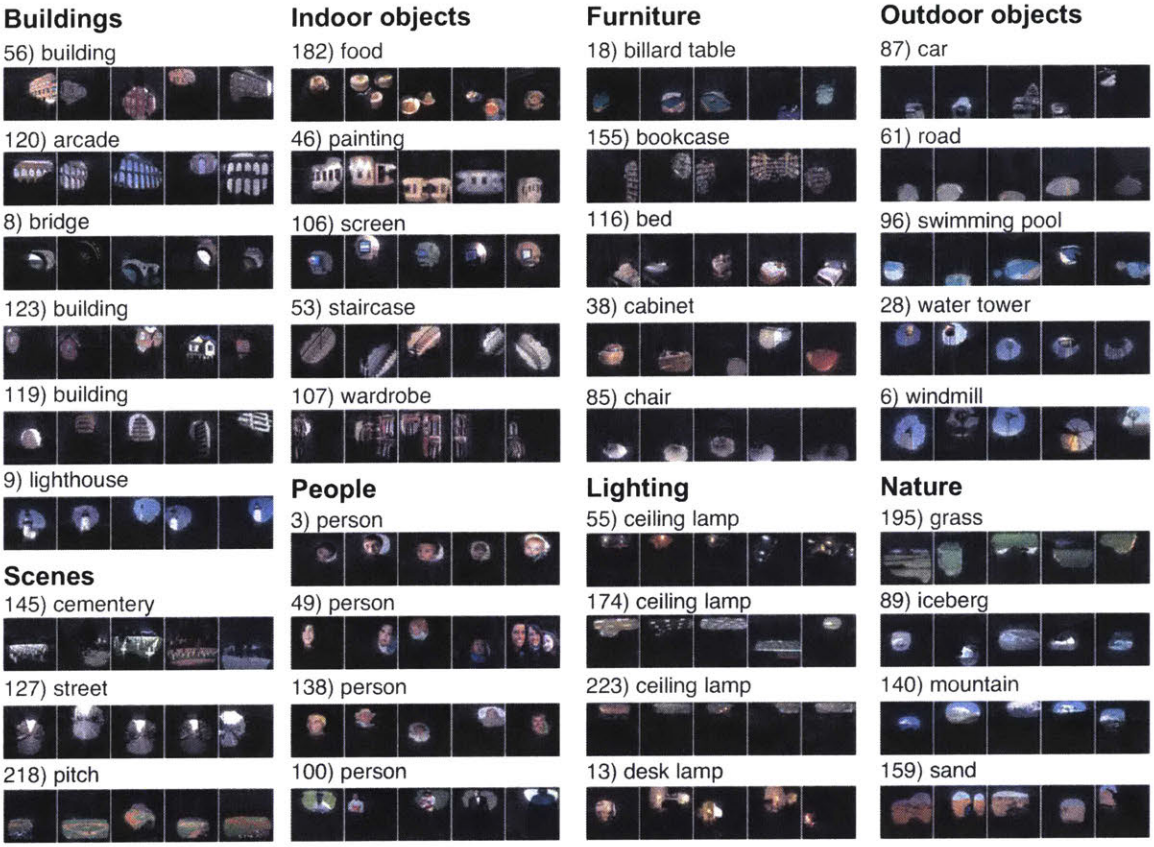


Figure 5-11: Segmentations using `pool5` units from Places-CNN. Many classes are encoded by several units covering different object appearances. Each row shows the 5 most confident images for each unit. The number represents the unit number in `pool5`.

Figure 5-12(c). Note the similarity between Figure 5-12(b) and Figure 5-12(c). The correlation is 0.84 indicating that the network is automatically identifying the most discriminative object categories to a large extent.

Note that there are 115 units in `pool5` of Places-CNN not detecting objects. This could be due to incomplete learning or a complementary texture-based or part-based representation of the scenes. Therefore, although objects seem to be a key part of the representation learned by the network, we cannot rule out other representations being used in combination with objects.

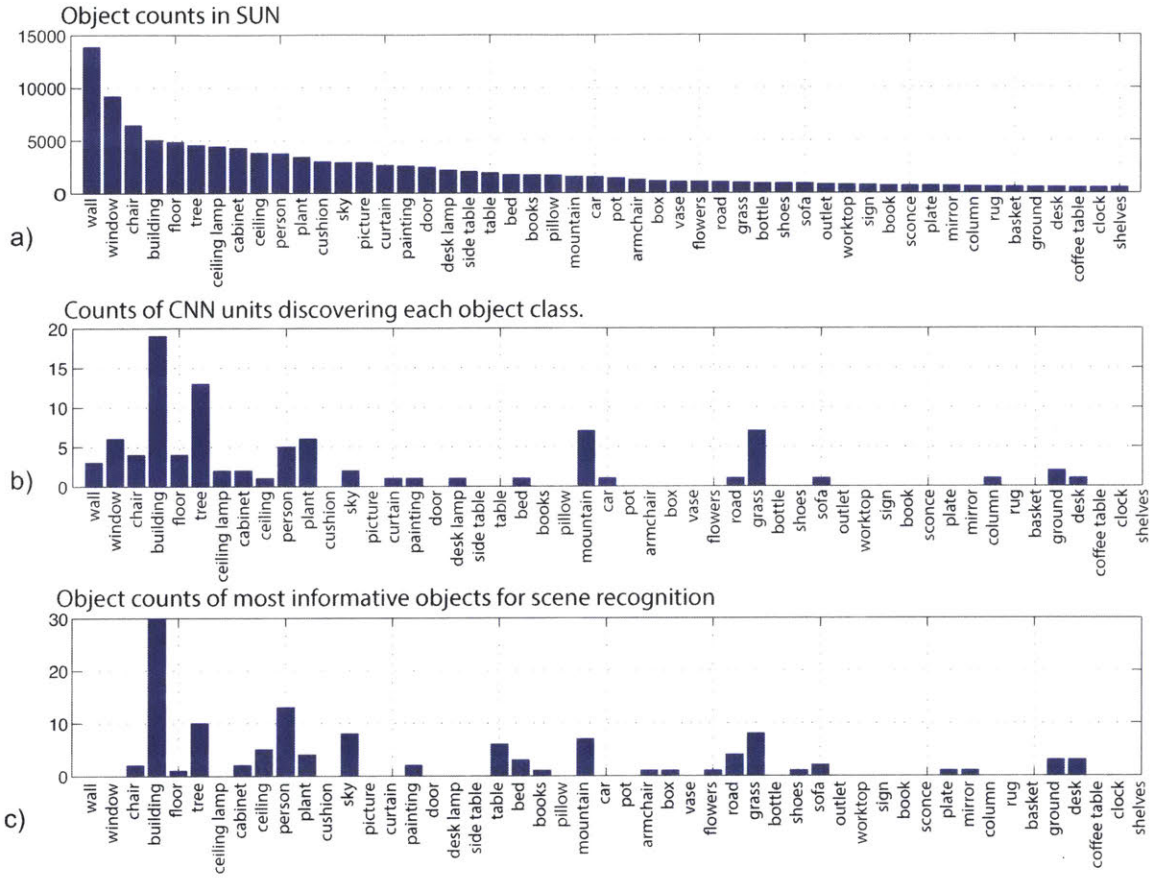


Figure 5-12: (a) Object frequency in SUN (only top 50 objects shown), (b) Counts of objects discovered by `pool5` in Places-CNN. (c) Frequency of most informative objects for scene classification.

5.3.2 Object Localization within the inner Layers

Places-CNN is trained to do scene classification using the output of the final layer of logistic regression and achieves state-of-the-art performance. From our analysis above, many of the units in the inner layers could perform interpretable object localization. Thus we could use this single Places-CNN with the annotation of units to do both scene recognition and object localization in a single forward-pass. Figure 5-13 shows an example of the output of different layers of the Places-CNN using the tags provided by AMT workers. Bounding boxes are shown around the areas where each unit is activated within its RF above a certain threshold.

In Figure 5-14 we provide the segmentation performance of the objects discovered in `pool5` using the SUN database. The performance of many units is very high which

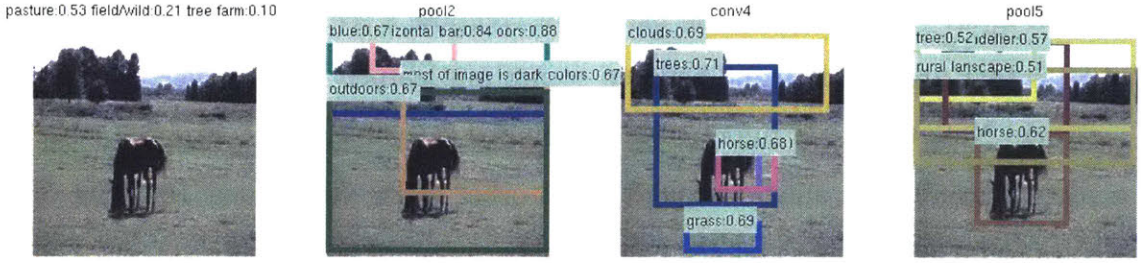


Figure 5-13: Interpretation of a picture by different layers of the Places-CNN using the tags provided by AMT workers. The first shows the final layer output of Places-CNN. The other three show detection results along with the confidence based on the units' activation and the semantic tags.

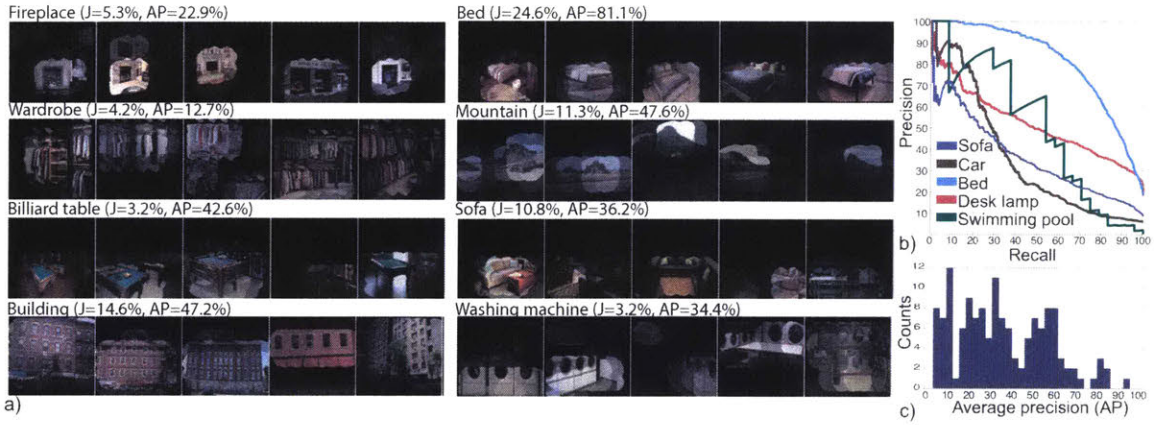


Figure 5-14: (a) Segmentation of images from the SUN database using `pool5` of Places-CNN (J = Jaccard segmentation index, AP = average precision-recall.) (b) Precision-recall curves for some discovered objects. (c) Histogram of AP for all discovered object classes.

provides strong evidence that they are indeed detecting those object classes despite being trained for scene classification.

5.4 Summary

We find that object detectors emerge as a result of learning to classify scene categories, showing that a single network can support recognition at several levels of abstraction (e.g., edges, textures, objects, and scenes) without needing multiple outputs or networks. While it is common to train a network to do several tasks and to use the final layer as the output, here we show that reliable outputs can be extracted

at each layer. As objects are the parts that compose a scene, detectors tuned to the objects that are discriminant between scenes are learned in the inner layers of the network. Note that only informative objects for specific scene recognition tasks will emerge. Future work should explore which other tasks would allow for other object classes to be learned without the explicit supervision of object labels.

Chapter 6

Conclusion

In this thesis, I have demonstrated various ways of using visual media to understand human behavior. I have made major contributions to computational cognition by providing the first model that robustly predicts which images people will remember or forget. This work, published in the top venues of computer vision and widely publicized in the popular press. Currently, more than 80% of internet traffic is digital visual information, producing billions of materials that artificial systems will have to make sense of. In 2014, I published a model able to determine, pre-emptively, which images people will prefer, a first attempt at understanding people's collective theory of mind. Predicting what will become popular opens the door to predicting many societal and economic behaviors, from advertising to consumer and political preferences. Predicting the state of mind of agents is at the core of social and economic sciences. Now, it is within the realm of computer science: I developed novel approaches to automatically find and follow people's gaze (from images and from using their own cell phone), and can potentially use this information to infer where they are heading, and what they may do next. Finally, I developed approaches for visualizing and understanding convolutional neural networks allowing us, in the future, to learn more about human behaviors through the use of algorithms capable of predicting behavior better than humans.

My work has just begun to touch the surface of what is possible in this domain, and I hope it inspires others to pursue a similar line of research. Below, I present

some ideas for future work that I did not get a chance to pursue during the course of my PhD. I hope that these ideas will help guide some of the future work in this domain.

6.1 Ideas for Future Work

Predicting individual memory: Memorability reveals the commonality in memory across individuals. However, each person is unique and potentially, so is their memory. While memorability explains a large fraction of the variance in memory across individuals, the rest can be attributed to individual differences in memory. Future research would allow us to explore these differences to answer questions such as: Do different individuals remember different objects? For example, do you remember cats better while your friend remembers dogs? Can social network photos be used as a sampling of an individual's visual experiences allowing us to predict their memory? We could build a personalized model for each individual that accurately predicts their memory. Combining memorability with individual memory differences will allow us to comprehensively understand and explain human visual memory.

Learning everything about everyone: In my current works, I have studied a number of human behavioral traits independently. However, each individual is a combination of each of these traits to different extents. I believe that there are interesting correlations between these traits (e.g., personality may be an indicator of memory) but the limited availability of data makes analysis difficult. To address this, one could develop a mobile application that can simultaneously capture a variety of traits about each individual. This application could be marketed through gamification and providing novel insights into people's memory using my current work on memorability. By connecting with users' social networks we can gather a tremendous amount of background information and through games related to memory, personality and IQ, we can understand their behavioral traits. The unique combination of information here will serve as an invaluable resource for tackling a variety of previously unapproachable research questions for many years to come. Finally, this will enable

the development of computational models of individuals that can accurately predict their memory, decisions and behavior.

Cognitive neuroscience: The human brain is essentially the source of all human cognition. As such, it is one of the most important pieces of the puzzle that is human behavior. My recent work [21, 20] in this domain has revealed a correspondence in the hierarchy of the visual brain and the hierarchy in neural networks. I believe that this relationship further motivates the use of deep networks in understanding human behavior. Future research could explore how we can directly use the brain signals to develop better models of human behavior, a process I call *brainpropagation* i.e., backpropagating signals from the human brain into deep learning algorithms.

Bibliography

[1] Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Analyzing the performance of multilayer neural networks for object recognition. *ECCV*, 2014.

[2] Brian Amberg, Pascal Paysan, and Thomas Vetter. Weight, sex, and facial expressions: On the manipulation of attributes in generative 3D face models. In *Advances in Visual Computing*. 2009.

[3] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06*, pages 44–54, 2006.

[4] Wilma Alice Bainbridge, Phillip Isola, and Aude Oliva. The intrinsic memorability of face photographs. In *Journal of Experimental Psychology: General*, 2013.

[5] Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 354–361. IEEE, 2013.

[6] Shumeet Baluja and Dean Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical report, 1994.

[7] Philip J Benson and David I Perrett. Perception and recognition of photographic quality facial caricatures: Implications for the recognition of natural images. *EJCP*, 1991.

[8] Alessandro Bergamo, Loris Bazzani, Dragomir Anguelov, and Lorenzo Torresani. Self-taught object localization with deep networks. *arXiv preprint arXiv:1409.3964*, 2014.

[9] Irving Biederman. Aspects and extensions of a theory of human image understanding. *Computational processes in human vision: An interdisciplinary perspective*, pages 370–428, 1988.

[10] Irving Biederman. *Visual object recognition*, volume 2. MIT press Cambridge, 1995.

[11] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *PAMI*, 2013.

[12] Ali Borji, Daniel Parks, and Laurent Itti. Complementary effects of gaze direction and early saliency in guiding fixations during free viewing. *Journal of vision*, 14(13):3, 2014.

[13] Ali Borji, Dicky N Sihite, and Laurent Itti. Salient object detection: A benchmark. In *ECCV*. 2012.

[14] Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *ACM MM*, 2013.

[15] Timothy F. Brady, T. Konkle, George A. Alvarez, and A. Oliva. Visual long-term memory has a massive storage capacity for object details. *Proc Natl Acad Sci, USA*, 105(38), 2008.

[16] Thomas A Busey. Formal models of familiarity and memorability in face recognition. *Computational, geometric, and process perspectives on facial cognition: Contexts and challenges*, pages 147–191, 2001.

[17] Bora Celikkale, A Tanju Erdem, and Esra Erdem. Visual attention-driven spatial pooling for image memorability. In *CVPR Workshop*. IEEE, 2013.

[18] Jixu Chen and Qiang Ji. 3d gaze estimation with a single camera without ir illumination. In *ICPR*, 2008.

[19] Jixu Chen and Qiang Ji. Probabilistic gaze estimation without active personal calibration. In *CVPR*, 2011.

[20] R. M. Cichy, **A. Khosla**, D. Pantazis, and A. Oliva. Neural dynamics of the cortical representation of scenes: Evidence from magnetoencephalography and deep neural networks. *Under review for NeuroImage*, 2015.

[21] R. M. Cichy, **A. Khosla**, D. Pantazis, A. Torralba, and A. Oliva. Deep neural network models predict spatio-temporal cortical dynamics of visual object recognition. *Under review for Neuron*, 2015.

[22] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *PAMI*, 23(6):681–685, 2001.

[23] David J. Crandall, Dan Cosley, Daniel P. Huttenlocher, Jon M. Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *KDD*, pages 160–168, 2008.

[24] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[25] Ritendra Datta, Jia Li, and James Ze Wang. Algorithmic inferencing of aesthetics and emotion in natural images: An exposition. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 105–108. IEEE, 2008.

[26] Doug DeCarlo and Anthony Santella. Stylization and abstraction of photographs. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 769–776. ACM, 2002.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[28] Sagnik Dhar, Vicente Ordonez, and Tamara L Berg. High level describable attributes for predicting aesthetics and interestingness. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1657–1664. IEEE, 2011.

[29] P. S. Dodds and D. J. Watts. A generalized model of social and biological contagion. *Journal of Theoretical Biology*, 2005.

[30] Pedro Domingos and Matthew Richardson. Mining the network value of customers. In *KDD*, pages 57–66, 2001.

[31] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[32] Harris Drucker, Chris JC Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *NIPS*, pages 155–161, 1997.

[33] Rachit Dubey, Joshua Peterson, Aditya Khosla, Ming-Hsuan Yang, and Bernard Ghanem. What makes an object memorable? In *International Conference on Computer Vision (ICCV)*, 2015.

[34] Andrew Duchowski. *Eye tracking methodology: Theory and practice.* Springer Science & Business Media, 2007.

[35] NJ Emery. The eyes have it: the neuroethology, function and evolution of social gaze. *Neuroscience & Biobehavioral Reviews*, 2000.

[36] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

[37] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[38] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2013.

[39] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[40] Alireza Fathi, Jessica K Hodgins, and James M Rehg. Social interactions: A first-person perspective. In *CVPR*, 2012.

[41] Alireza Fathi, Yin Li, and James M Rehg. Learning to recognize daily actions using gaze. In *ECCV*. 2012.

[42] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[43] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[44] Flavio Figueiredo. On the prediction of popularity of trends and hits for user generated videos. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 741–746. ACM, 2013.

[45] Flavio Figueiredo, Fabrício Benevenuto, and Jussara M Almeida. The tube over time: characterizing popularity growth of youtube videos. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 745–754. ACM, 2011.

[46] Juliet Fiss, Aseem Agarwala, and Brian Curless. Candid portrait selection from video. In *ACM TOG*, volume 30, page 128. ACM, 2011.

[47] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[48] Malcolm Gladwell. *The Tipping Point: How Little Things Can Make a Big Difference*. Back Bay Books, 2002.

[49] D. Grangier, L. Bottou, and R. Collobert. Deep convolutional networks for scene parsing. *TPAMI*, 2009.

[50] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, Fabian Nater, and Luc Van Gool. The interestingness of images. In *IEEE ICCV*, 2013.

[51] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *PAMI*, 2010.

[52] Dan Witzner Hansen and Arthur EC Pece. Eye tracking in the wild. *CVIU*, 2005.

[53] Craig Hennessey, Borna Noureddin, and Peter Lawrence. A single camera eye-gaze tracking system with free head motion. In *ETRA*, 2006.

[54] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

[55] Matthew W Hoffman, David B Grimes, Aaron P Shon, and Rajesh PN Rao. A probabilistic model of gaze imitation and shared attention. *Neural Networks*, 2006.

[56] Philip S Holzman, Leonard R Proctor, Deborah L Levy, Nicholas J Yasillo, Herbert Y Meltzer, and Stephen W Hurt. Eye-tracking dysfunctions in schizophrenic patients and their relatives. *Archives of general psychiatry*, 1974.

[57] Liangjie Hong, Ovidiu Dan, and Brian D. Davison. Predicting popular messages in twitter. In *WWW (Companion Volume)*, pages 57–58, 2011.

[58] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. TabletGaze: A dataset and baseline algorithms for unconstrained appearance-based gaze estimation in mobile tablets. *arXiv:1508.01244*, 2015.

[59] Edmund Burke Huey. *The psychology and pedagogy of reading.* The Macmillan Company, 1908.

[60] Mark J Huiskes and Michael S Lew. The MIR Flickr retrieval evaluation. In *ACM MIR*, 2008.

[61] Takahiro Ishikawa. Passive driver gaze tracking with active appearance models. 2004.

[62] Phillip Isola, Devi Parikh, Antonio Torralba, and Aude Oliva. Understanding the instrinsic memorability of images. In *NIPS*, 2011.

[63] Phillip Isola, Jianxiong Xiao, Devi Parikh, Antonio Torralba, and Aude Oliva. What makes a photograph memorable? *IEEE PAMI*, 2014.

[64] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2001.

[65] RJ Jacob and Keith S Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2003.

[66] Hector Jasso, Jochen Triesch, and GO Deák. Using eye direction cues for gaze following–a developmental model. In *ICDL*, 2006.

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

[68] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *CVPR*, 2009.

[69] Mayank Juneja, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman. Blocks that shout: Distinctive parts for scene classification. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2013.

[70] Sepandar D. Kamvar and Jonathan Harris. We feel fine and searching the emotional web. In *WSDM*, pages 117–126, 2011.

[71] S Karthikeyan, Vignesh Jagadeesh, Renuka Shenoy, Miguel Ecksteinz, and BS Manjunath. From where and how to what we see. In *ICCV*, 2013.

[72] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[73] Fahad S Khan, Joost Weijer, Andrew D Bagdanov, and Maria Vanrell. Portmanteau vocabularies for multi-cue image representation. In *Advances in neural information processing systems*, pages 1323–1331, 2011.

[74] Rahat Khan, Joost Van de Weijer, Fahad Shahbaz Khan, Damien Muselet, Christophe Ducottet, and Cecile Barat. Discriminative color descriptors. *CVPR*, 2013.

[75] Aditya Khosla, Wilma A. Bainbridge, Antonio Torralba, and Aude Oliva. Modifying the memorability of face photographs. In *International Conference on Computer Vision (ICCV)*, 2013.

[76] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What makes an image popular? In *WWW*, 2014.

[77] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, USA, June 2013.

[78] Aditya Khosla, Akhil S. Raju, Antonio Torralba, and Aude Oliva. Understanding and predicting image memorability at a large scale. In *ICCV*, 2015.

[79] Aditya Khosla, Jianxiong Xiao, Phillip Isola, Antonio Torralba, and Aude Oliva. Image memorability and visual inception. In *SIGGRAPH Asia 2012 Technical Briefs*, page 35. ACM, 2012.

[80] Aditya Khosla, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Memorability of image regions. In *Advances in Neural Information Processing Systems*, pages 305–313, 2012.

[81] Jongpil Kim, Sejong Yoon, and Vladimir Pavlovic. Relative spatial features for image memorability. In *ACM MM*, 2013.

[82] Suin Kim, JinYeong Bak, and Alice Haeyun Oh. Do you feel what i feel? social aspects of emotions in twitter conversations. In *ICWSM*, 2012.

[83] Barbara Knappmeyer, Ian M Thornton, and Heinrich H Bülthoff. The use of facial motion and facial form during the processing of identity. *Vision research*, 43(18):1921–1936, 2003.

[84] T. Konkle, Timothy F. Brady, George A. Alvarez, and A. Oliva. Scene memory is more detailed than you think: the role of categories in visual long-term memory. *Psych Science*, 2010.

[85] Gueorgi Kossinets and Duncan J Watts. Origins of homophily in an evolving social network1. *American Journal of Sociology*, 115(2):405–450, 2009.

[86] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[87] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[88] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE, 2009.

[89] Andreas Lanitis, Christopher J. Taylor, and Timothy F. Cootes. Toward automatic simulation of aging effects on face images. *PAMI*, 2002.

[90] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[91] Quoc V Le. Building high-level features using large scale unsupervised learning. In *ICASSP*, 2013.

[92] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361, 1995.

[93] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), 2007.

[94] T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski. Data-driven enhancement of facial attractiveness. *SIGGRAPH*, 2008.

[95] Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, pages 1378–1386, 2010.

[96] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[97] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, 2015.

[98] Jonathan Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *NIPS*, 2014.

[99] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[100] Feng Lu, Takahiro Okabe, Yusuke Sugano, and Yoichi Sato. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing*, 2014.

[101] Feng Lu, Yusuke Sugano, Toshiya Okabe, and Yuuki Sato. Adaptive linear regression for appearance-based gaze estimation. *PAMI*, 2014.

[102] Jana Machajdik and Allan Hanbury. Affective image classification using features inspired by psychology and art theory. In *ACM MM*, 2010.

[103] Päivi Majaranta and Andreas Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in Physiological Computing*. Springer, 2014.

[104] Matei Mancas and Olivier Le Meur. Memorability of natural scenes: The role of attention. In *ICIP*. IEEE, 2013.

[105] Manuel Jesús Marin-Jimenez, Andrew Zisserman, Marcin Eichner, and Vittorio Ferrari. Detecting people looking at each other in videos. *IJCV*, 2014.

[106] Christopher D McMurrough, Vangelis Metsis, Jonathan Rich, and Fillia Makedon. An eye tracking dataset for point of gaze detection. In *ETRA*, 2012.

[107] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. *ETRA*, 2014.

[108] Carlos H Morimoto and Marcio RM Mimica. Eye gaze tracking techniques for interactive applications. *CVIU*, 2005.

[109] Naila Murray, Luca Marchesotti, and Florent Perronnin. AVA: A large-scale database for aesthetic visual analysis. *CVPR*, 2012.

[110] M. E. J. Newman. Spread of epidemic disease on networks. *Physical Review E*, 2002.

[111] Amandianeze O. Nwana, Salman Avestimehr, and Tsuhan Chen. A latent social approach to youtube popularity prediction. *CoRR*, abs/1308.1418, abs/1308.1418, 2013.

[112] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[113] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 2006.

[114] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[115] Maxime Oquab, Léon Bottou, Ivan Laptev, Josef Sivic, et al. Weakly supervised object recognition with convolutional neural networks. In *NIPS*. 2014.

[116] Alice J O'Toole, Thomas Vetter, Harald Volz, Elizabeth M Salter, et al. Three-dimensional caricatures of human heads: distinctiveness and the perception of facial age. *Perception*, 26:719–732, 1997.

[117] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011.

[118] Hyun Soo Park, Eakta Jain, and Yaser Sheikh. Predicting primary gaze behavior using social saliency fields. In *ICCV*, 2013.

[119] Daniel Parks, Ali Borji, and Laurent Itti. Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes. *Vision Research*, 2014.

[120] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2751–2758. IEEE, 2012.

[121] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 2003.

[122] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM*, 2011.

[123] Henrique Pinto, Jussara M. Almeida, and Marcos André Gonçalves. Using early view patterns to predict the popularity of youtube videos. In *WSDM*, pages 365–374, 2013.

[124] M.C. Potter. Meaning in visual search. *Science*, 1975.

[125] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[126] Subramanian Ramanathan, Harish Katti, Nicu Sebe, Mohan Kankanhalli, and Tat-Seng Chua. An eye fixation database for saliency detection in images. In *ECCV*. 2010.

[127] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 1998.

[128] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014.

[129] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? In *NIPS*, 2015.

[130] Laura Walker Renninger and Jitendra Malik. When is scene identification just texture recognition? *Vision research*, 44(19):2301–2311, 2004.

[131] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

[132] Everrett M. Rogers. *Diffusion of Innovations*. Simon and Schuster, 2003.

[133] Daniel M. Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion. In *WWW '11*, 2011.

[134] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[135] Babak Saleh, Ali Farhadi, and Ahmed Elgammal. Object-centric anomaly detection by attribute-based reasoning. In *CVPR*, 2013.

[136] Weston Sewell and Oleg Komogortsev. Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network. In *SIGCHI*, 2010.

[137] David A. Shamma, Jude Yew, Lyndon Kennedy, and Elizabeth F. Churchill. Viral actions: Predicting video view counts using synchronous sharing behaviors. In *ICWSM*, 2011.

[138] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. Ieee, 2007.

[139] Saurabh Singh, Abhinav Gupta, and Alexei A Efros. Unsupervised discovery of mid-level discriminative patches. In *Computer Vision–ECCV 2012*, pages 73–86. Springer, 2012.

[140] Brian A Smith, Qi Yin, Steven K Feiner, and Shree K Nayar. Gaze locking: Passive eye contact detection for human-object interaction. In *UIST*, 2013.

[141] Hyun Soo Park and Jianbo Shi. Social saliency prediction. In *CVPR*, 2015.

[142] Statista. Global smartphone user penetration 2014 - 2019. `http://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/`, 2015.

[143] Yusuke Sugano, Yasuyuki Matsushita, Yoichi Sato, and Hideki Koike. An incremental learning method for unconstrained gaze estimation. In *Computer Vision–ECCV 2008*, pages 656–667. Springer, 2008.

[144] Yusuke Sugano, Yuki Matsushita, and Yuuki Sato. Appearance-based gaze estimation using visual saliency. *PAMI*, 2013.

[145] Yusuke Sugano, Yuki Matsushita, and Yuuki Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *CVPR*, 2014.

[146] Jinli Suo, Song-Chun Zhu, Shiguang Shan, and Xilin Chen. A compositional and dynamic model for face aging. *PAMI*, 2010.

[147] Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.

[148] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[149] Yaniv Taigman, Ming Yang, M Ranzato, and L Wolf. Deep-face: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[150] Kar-Han Tan, David J Kriegman, and Narendra Ahuja. Appearance-based eye gaze estimation. In *WACV*, 2002.

[151] Keiji Tanaka. Neuronal mechanisms of object recognition. *Science*, 262(5134):685–688, 1993.

[152] Yichuan Tang, Nitish Srivastava, and Ruslan R Salakhutdinov. Learning generative models with visual attention. In *NIPS*. 2014.

[153] Antonio Torralba, Kevin P Murphy, William T Freeman, and Mark A Rubin. Context-based vision system for place and object recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 273–280. IEEE, 2003.

[154] Diego Torricelli, Silvia Conforto, Maurizio Schmid, and Tommaso D'Alessio. A neural-based remote eye gaze tracker under natural head motion. *Computer methods and programs in biomedicine*, 2008.

[155] Naman Turakhia and Devi Parikh. Attribute dominance: What pops out? In *IEEE ICCV*, 2013.

[156] Roberto Valenti, Nicu Sebe, and Theo Gevers. Combining head pose and eye location information for gaze estimation. *TIP*, 2012.

[157] Tim Valentine. A unified account of the effects of distinctiveness, inversion, and race in face recognition. *QJEP*, 1991.

[158] Joost Van De Weijer, Cordelia Schmid, and Jakob Verbeek. Learning color names from real-world images. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[159] John R Vokey and J Don Read. Familiarity, memorability, and the effect of typicality on the recognition of faces. *Memory & Cognition*, 20(3):291–302, 1992.

[160] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: Visualizing Object Detection Features. *ICCV*, 2013.

[161] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[162] U Weidenbacher, G Layher, P-M Strauss, and H Neumann. A comprehensive head pose and gaze database. 2007.

[163] Janine Willis and Alexander Todorov. First impressions making up your mind after a 100-ms exposure to a face. *Psychological science*, 2006.

[164] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[165] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv:1504.06755*, 2015.

[166] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011.

[167] Dong Hyun Yoo and Myung Jin Chung. A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. *CVIU*, 2005.

[168] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.

[169] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[170] Ning Zhang, Manohar Paluri, Marc'Aurelio Ranzato, Trevor Darrell, and Lubomir Bourdev. Panda: Pose aligned networks for deep attribute modeling. *CVPR*, 2014.

[171] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *CVPR*, 2015.

[172] Bolei Zhou, Aditya Khosla, Agata Lapedriz, Antonio Torralba, and Aude Oliva. Places2: A large-scale database for scene understanding. *arXiv*, 2016.

[173] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.

[174] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.

[175] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.

[176] Zhiwei Zhu and Qiang Ji. Eye gaze tracking under natural head movements. In *CVPR*, 2005.

[177] Zhiwei Zhu, Qiang Ji, and Kristin P Bennett. Nonlinear eye gaze mapping function estimation via support vector regression. In *Pattern Recognition*, 2006.