Looking Beyond the Visible Scene

Aditya Khosla* Byoungkwon An* Joseph J. Lim Antonio Torralba
Massachusetts Institute of Technology
{khosla, dran, lim, torralba}@csail.mit.edu

Abstract

A common thread that ties together many prior works in scene understanding is their focus on the aspects directly present in a scene such as its categorical classification or the set of objects. In this work, we propose to look beyond the visible elements of a scene; we demonstrate that a scene is not just a collection of objects and their configuration or the labels assigned to its pixels - it is so much more. From a simple observation of a scene, we can tell a lot about the environment surrounding the scene such as the potential establishments near it, the potential crime rate in the area, or even the economic climate. Here, we explore several of these aspects from both the human perception and computer vision perspective. Specifically, we show that it is possible to predict the distance of surrounding establishments such as McDonald's or hospitals even by using scenes located far from them. We go a step further to show that both humans and computers perform well at navigating the environment based only on visual cues from scenes. Lastly, we show that it is possible to predict the crime rates in an area simply by looking at a scene without any real-time criminal activity. Simply put, here, we illustrate that it is possible to look beyond the visible scene.

1. Introduction

"Daddy, daddy, I want a Happy Meal!" says your son with a glimmer of hope in his eyes. Looking down at your phone, you realize it is fresh out of batteries, "how am I going to find McDonald's now?" you wonder. Looking left you see mountains and on the right some buildings. Right seems like the *right* way. Still no McDonald's in sight, you end up at a junction; the street on the right looks shady, its probably best to avoid it. As you walk towards the left, you are at a junction again; a residential estate on the left and some shops on the right. Right it is. Shortly thereafter, you have found your destination, all without a map or GPS!

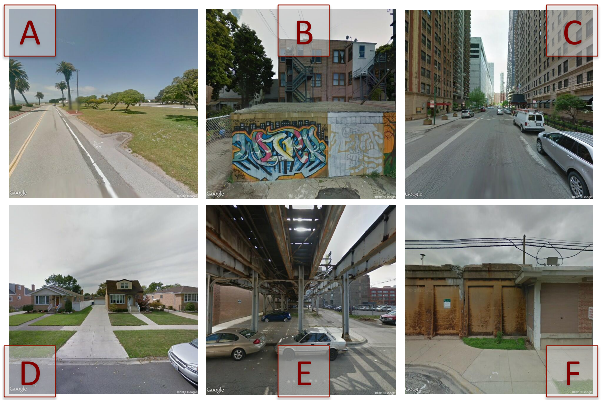


Figure 1. Can you rank the images by their distance to the closest McDonald's? What about ranking them based on the crime rate in the area? Check your answers below[1]. While not directly visible i.e. we do not see any McDonald's or crime in action, we can predict the possible actions or the type of surrounding establishments from just a small glimpse of our surroundings.

A common thread that ties together previous works in scene understanding is their focus on the aspects directly present in a scene. In this work, we propose to *look beyond* the visible elements of a scene; a scene is not just a collection of objects and their configuration or the labels assigned to the pixels - it is so much more. From a simple observation of a scene, one can tell a lot about the environment surrounding the scene such as the potential establishments near it, the potential crime rate in the area, or even the economic climate. See Fig. 1 for example. Can you rank the scenes based on their distance from the nearest McDonald's? What about ranking them by the crime rate in the area? You might be surprised by how well you did despite having none of this information readily available from the visual scene.

In our daily lives, we are constantly making decisions about our environment such as, is this location safe? Where can I find a parking spot? Where can I get a bite to eat? We do not need to observe a crime happening in real-time to guess that an area is unsafe. Even without a GPS, we can often navigate our environment to find the nearest restroom or a bench to sit on without performing a random

* - indicates equal contribution

[1] Answer key: crime rate (highest) B > E > C > F > D > A (lowest), distance to McDonald's: (farthest) A > F > D > E > C > B (closest)

walk. Essentially, we can look *beyond the visible scene* and infer properties about our environment using the visual cues present in the scene.

In this work, we explore the extent to which humans and computers are able to look beyond the immediately visible scene. To simulate the environment we observe around us, we propose to use Google Street View data that provides a panoramic view of the scene. Based on this, we show that it is possible to predict the distance of surrounding establishments such as McDonald's or hospitals even using scenes located far from them. We go a step further to show that both humans and computers perform reasonably well at navigating the environment based only on visual cues from scenes that contain no direct information about the target. Further, we show that it is possible to predict the crime rates in an area simply by looking at a scene without any current crimes in action. Last, we use deep learning and mid-level features to analyze the aspects of a scene that allow us to make these decisions.

We emphasize that the goal of this paper is not to propose complex mathematical equations; instead, using simple yet intuitive techniques, we demonstrate that humans and computers alike are able to understand their environment by just seeing a small glimpse of it in a single scene. Interestingly, we find that despite the relative simplicity of the proposed approaches, computers tend to perform on par, or even slightly outperform humans in some of the tasks. We believe that inference beyond the visible scene based on visual cues is an exciting avenue for future research in computer vision and this paper is merely a first step in this direction.

2. Related Work

Scene understanding is a fundamental problem in computer vision, one that has received a lot of attention. Despite its popularity, there is no single definition of scene understanding; as a field, we are still exploring what it really means to understand a scene. Recent works have explored scene understanding from a variety of perspectives - as a task of scene classification [19, 25, 30, 40] identifying the type of scene, semantic segmentation [2, 5, 14] involving the labeling of pixels as belonging to specific objects, 3D understanding [7, 8, 15, 16, 35] to obtain the 3D structure of a room or reason about the affordances of objects, contextual reasoning [6, 10, 31] involving the joint reasoning of the position of multiple objects in a scene, or a combination of these tasks [3, 27, 28, 32, 36, 41, 43].

There are some works that are similar to ours in flavor exploring features of the scene that may not be directly visible such as scene attributes [33]. In [33], Patterson and Hays explore various attributes such as indoor vs outdoor, and man-made vs natural. While these attributes may be hard to attribute to specific objects or pixels in a scene, they still tend to revolve around the visible elements of a scene. Another interesting line of work that deals with extending a scene in space is FrameBreak [42]. In this paper, the authors extend scenes using panoramic views of images, but their focus is on the local extension of a scene to generate a larger scene instead of exploring non-visual elements of a scene or the larger environment around it.

The work most related to ours is IM2GPS [17] by Hays and Efros. In that work, they explore the problem of obtaining the GPS coordinates of an image using a large dataset of images. While this problem deals with finding the specific GPS coordinates of an image, our work deals with more categorical classifications of the surrounding environment of an image such as finding instances of establishments such as McDonald's or Starbucks near it.

3. Dataset

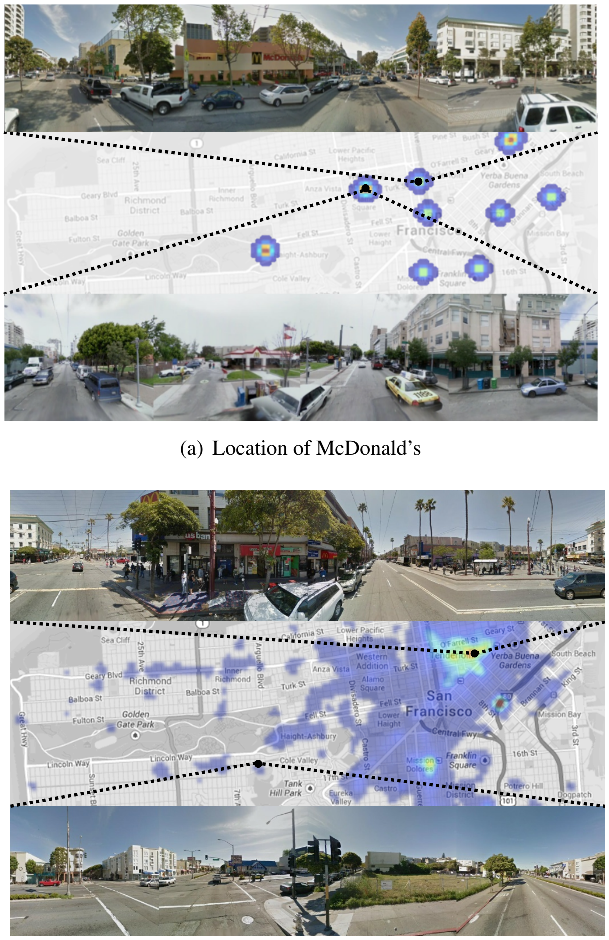
As described in Sec. 1, our goal is to look beyond the visible scene. One possible way of doing this is to download a random image from the internet and attempt to predict how far the nearest McDonald's or hospital might be. While geotagged images are commonly available, they tend to be spread unevenly across cities, and the GPS coordinates are often incorrect. To overcome this, we collect data from Google Street View where geotags are reliable and ground truth annotation can be easily obtained. In Fig. 2, we visualize some example images and annotation from Street View.

For our dataset, we pick 8 cities from around the world, namely Boston (Bo), Chicago (Ch), Hong Kong (HK), London (Lo), Los Angeles (LA), New York City (NYC), Paris (Pa) and San Francisco (SF). For each city, we manually define a polygon enclosing it, and sample points in a grid as illustrated in Fig. 3. The grid points are located 16m apart, and we obtain 4 images per point resulting in a total of ∼**8 million** images in our dataset. Each of the 4 images is taken at the same location but points in different directions, namely north, south, east and west.

From Google Places, we obtain the location of all the establishments of interest (i.e., McDonald's, Starbucks, hospitals) in the area and find their distance from our grid points. This allows us to build a dataset of street scene images where the establishments are largely not directly visible, but present in the surrounding area. In addition, we obtain longitude and latitude information related to crimes in San Francisco using CrimeSpotting [1], allowing us to build a *crime density map* as shown in Fig 2(b). We aggregate crime information over the past year related to aggravated assault and robbery.

Please refer to the supplemental material[1] for additional information such as the number of images per city, and the method for obtaining clean annotation from Google Places.

[1] Available at: http://mcdonalds.csail.mit.edu



(a) Location of McDonald's

(b) Crime density map

Figure 2. Sample images and maps from our dataset for the city of San Francisco. The map is overlayed with information related to (a) the location of McDonald's and (b) the crime rate in the area. Note that we obtain four images from Street View that have been layed out in this way to provide a panoramic view of the location.

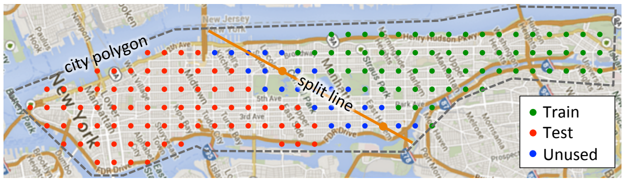


Figure 3. Illustration of grid where street view images are collected, and how train/test splits are obtained. The orange points are two randomly sampled points used to define the *split line* leading to the train/test splits as shown. Note that the actual density of the grid is significantly higher than shown here.

4. Where is McDonald's?

In this section, we investigate the ability of both humans and computers to find establishments such as McDonald's, Starbucks or even hospitals from images where only a generic scene, similar to the ones shown in Fig. 1 are available. In Sec. 4.1, we explore a comparatively simple question: given two panoramic images, can an observer tell which is closer to a particular establishment? We find that both humans and computers significantly outperform chance performance in this task. Based on this, an obvious question arises: can we then reliably find our way to the given establishment by moving around the environment based only on visual cues? In Sec. 4.2, we find that surprisingly, humans are quite adept at performing this task and significantly outperform doing a random walk of the environment.

4.1. Which is closer?

Here, our objective is to determine whether an observer (man or machine) can distinguish which of two scenes might be closer to a particular establishment. The details of the experimental setup are given in Sec. 4.1.1 and the results explained in Sec. 4.1.2.

4.1.1 Setup

For the experiments in this section, we subsample the city grid such that adjacent points are located 256m from each other. This reduces the chance of neighboring points looking too similar. Also, we conduct experiments on three establishments, namely McDonald's, Starbucks and hospitals.

Humans: For a given city, we first randomly sample a pair of unique grid points and obtain a set of panoramic images (e.g. Fig. 2). We show the pair of panoramic images to workers on Amazon's Mechanical Turk (AMT), a crowdsourcing platform, and instruct them to guess, to the best of their ability, which of the images is closer to a particular establishment of interest i.e. McDonald's, Starbucks or a hospital. After confirming the answer for an image pair, the worker receives feedback on whether the choice was correct or not. We found that providing feedback was essential to both improving the quality of the results, and keeping workers engaged in the task. We ensured a high quality of work by including 10% *obvious* pairs of images, where one image shows the city center, while the other shows a mountainous terrain. If a worker failed any of the obvious pairs, all their responses were discarded and they were blocked from doing further tasks.

After filtering the bad workers, we obtained approximately 5000 pairwise comparisons per city, per establishment (i.e. 120k in total). We compute performance in terms of accuracy: the percentage of correctly selected images from the pairwise comparisons.

Computers: Motivated by [21], we use various 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. For each feature, we describe our motivation and the extraction method below.

Gist: Various studies [34, 4] 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 [30] descriptor with a feature dimension of 512.

Texture: We often interact with various textures and materials in our surroundings both visually, and through touch. To encode this, we use the Local Binary Pattern (LBP) [29] feature. We use non-uniform LBP pooled in a 2-level spatial pyramid [25] resulting in a feature dimension of 1239.

Color: Colors are an important component of the human visual system for determining properties of objects, understanding scenes, etc. Various recent works have been devoted to developing robust color descriptors [38, 20], which have been shown to be valuable in computer vision for a variety of tasks. Here, we use the 50 colors proposed by [20], densely sampling 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 Locality-Constrained Linear Coding (LLC) [39] with max-pooling in a 2-level spatial pyramid [25] to obtain a final feature of dimension 4200.

Gradient: Much evidence suggests that, in the human visual system, retinal ganglion cells and cells in the visual cortex V1 are essentially gradient-based features. Further, gradient based features [9, 13] have also been successfully applied to various applications in computer vision. In this work, we use the powerful Histogram of Oriented Gradient (HOG) [9] features. We use dense sampling with a step size of 4 and apply K-means to build a dictionary of size 256. We then use LLC [39] to assign the descriptors to the dictionary, and finally apply a 2-level spatial pyramid [25] to obtain a final feature dimension of 5376.

Deep learning: Artificial neural networks are computational models inspired by neuronal structure in the brain. Recently, convolutional neural networks (CNNs) [26] have gained significant popularity as methods for learning image representations. In this work, we use the recently popular 'ImageNet network' [24] trained on 1.3 million images. Specifically, we use Caffe [18] to extract features from the layer just before the final classification layer (often referred to as fc7), resulting in a feature dimension of 4096.

Algorithm: For a given point on the street view grid point, we use the square-root[2] of the distance to the closest establishment (e.g. Starbucks) under consideration as labels, and train a linear support vector regression (SVR) machine [11, 12] on the image features described above. The four images from each point are treated as independent samples with the same label. The hyperparameter C was determined using 5-fold cross-validation.

(a) City-specific accuracy on finding McDonald's

	Human	*Computer*				
		Gist	Texture	Color	Gradient	Deep
Boston	**0.60**	0.54	0.57	0.54	0.58	0.55
Chicago	**0.56**	0.52	0.51	0.53	0.53	0.52
HK	0.70	0.71	0.71	0.69	**0.73**	0.72
LA	0.57	0.58	0.60	**0.62**	**0.62**	0.61
London	0.62	0.63	0.64	0.64	**0.65**	0.65
NYC	0.62	0.62	0.64	**0.66**	0.66	0.66
Paris	0.61	0.61	0.61	**0.62**	**0.62**	**0.62**
SF	**0.59**	0.53	0.53	0.54	0.53	0.54
Mean	0.60	0.59	0.60	**0.61**	**0.61**	**0.61**

(b) City-specific accuracy on finding Starbucks

	Human	*Computer*				
		Gist	Texture	Color	Gradient	Deep
Boston	**0.57**	0.53	0.54	0.53	0.55	0.54
Chicago	0.56	0.56	0.54	0.56	**0.58**	**0.58**
HK	0.64	0.66	0.67	0.67	**0.69**	0.67
LA	0.55	0.55	0.54	0.56	0.56	**0.57**
London	0.60	0.61	0.61	0.62	**0.63**	**0.63**
NYC	0.55	0.57	0.57	**0.59**	0.58	0.58
Paris	0.61	0.63	0.64	**0.66**	0.65	**0.66**
SF	**0.59**	**0.59**	0.58	0.58	**0.59**	0.58
Mean	0.58	0.59	0.59	**0.60**	**0.60**	**0.60**

(c) City-specific accuracy on finding Hospital

	Human	*Computer*				
		Gist	Texture	Color	Gradient	Deep
Boston	0.56	**0.57**	0.56	**0.57**	0.56	0.56
Chicago	0.56	0.59	0.58	0.60	**0.61**	0.59
HK	0.62	0.61	0.60	**0.62**	**0.62**	**0.62**
LA	**0.53**	0.49	0.51	0.50	0.50	0.50
London	0.59	0.59	0.59	**0.60**	**0.60**	**0.60**
NYC	0.54	0.59	0.59	**0.62**	0.61	0.61
Paris	**0.56**	0.55	0.54	0.54	0.54	0.54
SF	0.58	**0.59**	0.55	0.57	0.57	0.54
Mean	0.57	0.57	0.57	**0.58**	0.57	0.57

Table 1. Accuracy on various tasks on predicting the distance of an establishment given a pair of images, as described in Sec. 4.1.2

In order to prevent overfitting to a particular city, we generate reasonably challenging train/test splits of the data: as illustrated in Fig. 3, we randomly select two grid points in the city, and draw a line between them. Then we use data on one side of the split line for training, and the other for testing. We discard points that are near the dividing line from both the train and test splits. Through repeated sampling, we ensure that the size of the train split is fixed to at least 40% of the points, and at most 60% of them. If a split does not meet this criterion, it is discarded.

For prediction, we apply the learned model to each of the four images from a location, and use the minimum score as the predicted distance. Thus, the grid location receiving the lowest score was selected as the one closer to the establishment under consideration. Similar to the human experiments, we report accuracy on randomly sampled pairwise comparisons averaged over 5 train/test splits of the data. During testing, we obtain accuracy by testing on 100k random pairwise trials sampled from the test split of the data.

[2] The square-root transformation made the data distribution resemble a Gaussian, allowing us to learn more robust prediction models.

	Test							
Train	Bo	Ch	HK	LA	Lo	NY	Pa	SF
Boston	0.58	0.52	0.67	0.55	0.59	0.60	0.62	0.55
Chicago	0.57	0.53	0.66	0.55	0.59	0.61	0.60	0.53
HK	0.61	**0.55**	**0.73**	0.59	**0.65**	0.63	**0.63**	0.56
LA	0.59	0.54	0.68	**0.62**	0.62	0.61	0.61	0.53
London	**0.62**	0.54	0.71	0.60	**0.65**	0.64	**0.63**	**0.57**
NYC	**0.62**	**0.55**	0.71	0.59	0.64	**0.66**	0.62	0.56
Paris	0.61	0.52	0.68	0.58	0.61	0.61	0.62	0.55
SF	0.57	0.53	0.67	0.56	0.61	0.59	0.62	0.53

Table 2. Generalization accuracy from one city to another on finding McDonald's using Gradient features (Sec. 4.1.1).

4.1.2 Results

The results are summarized in Tbl. 1. Given a chance performance of 50%, we observe that humans tend to perform relatively well on the task with a mean accuracy of 59%. Human performance is largely consistent across different cities with the highest performance achieved in Hong Kong of 70% on the task of finding McDonald's. Interestingly, the relative ordering of the human performance closely resembles that of the computer vision algorithms.

Despite the challenging nature of the task, we observe that computer vision algorithms slightly outperform humans. To investigate whether this effect occurs only because we train on the same city as testing, we train on one city and test on another. The results are summarized in Tbl. 2. This also simulates the setting where workers on AMT may not originate from the locations being tested i.e. the worker might be from Paris while the task images are from Boston. This can also be thought of as a problem of dataset bias [23, 37] i.e., a sampling bias. Despite significant differences in the cities, we find that the features are able to generalize reasonably well across cities. Surprisingly, for four of the eight cities, training on a city different from the test city actually improves performance as compared to training on the same city. This might occur because of the difficult train/test splits used. Note that splitting the grid points randomly into train/test splits instead of using the proposed method increases average performance from 61% to 66% on the task of finding McDonald's. Similar improvement is observed for other tasks.

4.2. How do I get there?

Here, we explore the task of navigation based only on visual cues. We want to show that despite the lack of *visible* information available in the scene about where an establishment is, observers are actually able to navigate an environment effectively and locate instances of the required establishments. For this task, we use the full dataset, where adjacent points are separated by only 16m to provide continuity to people trying to navigate the environment visually. It is similar to using Google Street View to attempt to find a McDonald's from a random location in a city. Due to the high cost and tedious nature of obtaining human data, we focus our attention on four cities, namely Hong Kong, New York City, Paris and San Francisco, and only on the task of finding McDonald's. Thus, the question we hope to answer in this section is: if I drop you at a random location in a city, will you be able to find your way to a McDonald's without a GPS? Interestingly, we find that people and computers alike are reasonably adept at navigating the environment in this way, significantly outperforming a random walk, indicating the presence of visual cues that allow us to *look beyond the visible scene*. Below, we describe the setup for the human and computer experiments, and summarize the results obtained.

Humans: As done in Sec. 4.1.1, we conduct experiments on AMT. In this case, instead of the panoramic image, we show images arranged in a grid to indicate images pointing to north (top), east (right), south (bottom), and west (left) with the center being empty. Using the keyboard, workers can choose to go in any of the four directions allowing them to travel along the grid based on the visual cues provided by the scenes shown. We pick 8 random start locations around each city and collect data from 25 unique workers per starting point. We ensure that the start points are located at least 200m from the nearest McDonald's. We allow workers a maximum of 1000 steps from the start location, and workers are free to visit any grid point, even ones they have visited before. We record the path and number of steps taken to find McDonald's. Once a worker is within 40m of a McDonald's, the task ends successfully. To incentivize workers, we pay a significant bonus when they successfully locate McDonald's.

Note that a city occupies a finite grid, so some directions may be blocked if the users end up at the edge of the grid. They can always navigate back the way they came, and hence cannot get stuck at a particular point.

Computers: At each location, we want to predict which of the four images points towards the closest McDonald's (to identify the direction to move in). Thus, we obtain labels such that all four images at any given location have different distances; specifically, for each image at a particular location, we find the closest McDonald's from that location in the direction of the image. To allow for some slack, we consider an angle of 100° instead of 90° such that there is some overlap between the space, in case a McDonald's lies in the middle of two dividing lines. Thus, the regressor is trained to predict the distance to the nearest McDonald's in the direction of the image. We can now use this for navigation as described below.

We use a relatively naive method for navigation: given some start location, we find the predicted distance to the nearest McDonald's for each of the four images using the above regressor. Then, we move in the direction where the lowest distance is predicted. If there are blocked directions (e.g. edge of grid), we only consider the directions we can travel to. To prevent naive looping given the deterministic nature of the algorithm, we ensure that the computer cannot pick the same direction from a given location twice. Specifically, a single location can be visited multiple times, but once at that location, the directions picked in the previous visit cannot be picked again. This allows the algorithm to explore new areas if it gets stuck in a loop. If all paths from a location have been traversed, we move to a random unexplored point close to the current location. Additionally, we also have a variable step size (i.e. number of steps taken in a particular direction) that decays over time.

We use Gradient features for this task, as described in Sec. 4.1.1, and train the algorithm on London, and apply it to all the cities in the test set. Thus, our model is consistent across all cities, and this allows us to reduce biases caused by training and testing in the same city.

Results: The results are summarized in Tbl. 3. We observe that humans are able to navigate the environment with a reasonable success rate of about 65.2% with an average of 145.9 steps to find McDonald's when successful. Humans significantly outperform both random walk and our algorithm, succeeding more frequently in the limited number of steps, and also taking less steps to reach the destination. While humans outperform our algorithm, we find that our algorithm does considerably better as compared to doing a random walk, suggesting that the visual cues are helpful in navigating the space.

We also notice that humans tend to outperform our algorithm much more significantly when the start points are farther away. This is to be expected as our algorithm is largely local in nature and does not take global information into account, while humans do this naturally. For example, our algorithm optimizes locally even when the distance from the city center is fairly large while humans tend to follow the road into the city before doing a local search.

In Fig. 4, we investigate the path taken by humans starting from one common location. We observe that humans tend to be largely consistent at various locations and divergent at others when the signal is weak. When the visual cues are more obvious as shown in the images of the figure, humans tend to make similar decisions following a similar path, ending up at a nearby McDonald's. This shows that the visual cues do not arise from random noise but are instead largely consistent in the structure of the world.

5. Is it safe here?

Apart from predicting the location to nearby establishments, we also consider the problem of predicting crime rate from a visual scene. Our goal here is to find localized crime rate at particular streets, and explore whether humans and computers are capable of performing this task. We can think of this task as finding *latent scene affordances*. Note that the set of actions performed considered in this work i.e. crimes, may not be the most pleasant, but they are actions nonetheless. Without having people necessarily performing actions in scenes, we want to identify the type of scenes where people might perform certain actions. As in the previous task, here we show that people are able to predict to a reasonable accuracy the crime rate in an area.

City	Avg num steps			Success rate			Avg
	Human	Rand	Ours	Human	Rand	Ours	dist
HK	150.4	538.1	180.7	66.3%	27.2%	97.5%	450
NYC	72.8	483.0	300.7	91.7%	15.6%	67.5%	558
Paris	204.2	654.6	286.8	30.3%	2.9%	40.0%	910
SF	156.2	714.3	445.8	72.6%	1.1%	22.5%	1780
Mean	**145.9**	597.5	303.5	**65.2%**	11.7%	56.9%	925

Table 3. The average number of steps taken to find McDonald's when it is found successfully, and the success rate of the individual methods i.e. the percentage of trials where McDonald's was located in under 1000 steps. For random walk (Rand), we average the result of 500 trials from each of the 8 starting points in each city. 'Avg dist' refers to the average distance (in meters) of the randomly sampled starting points from the nearest McDonald's.

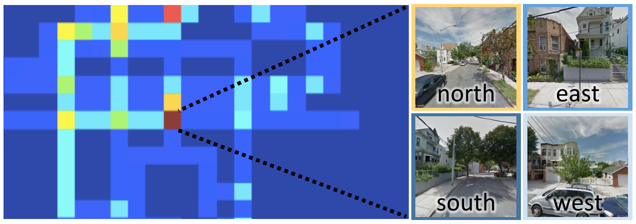


Figure 4. Results of human navigation starting from a particular start point. The above figure shows the number of times each location is visited by the 25 workers doing this task. The center of the grid (darkest red) shows the start location, and the map is color coded using the jet color scheme (i.e. dark red is highest, and blue is lowest). For the start location, we show the set of images a participant can choose from - the borders of the image indicate the frequency of selection by the participants (same color scheme as the map). It is interesting to observe that most participants chose to go north from the starting point given the potential appearance of a city in the distance. We observed that participants also tended to follow roads instead of going through buildings.

Below, we describe the experimental setup for both the human and computer experiments, and the results obtained. We only consider crime information from San Francisco as it is publicly available in a readily usable format. Similar to Sec. 4.1, we subsample the city grid such that adjacent points are located 256m from each other.

Humans: Similar to Sec. 4.1.1, we ask workers on AMT to perform comparisons between two pairs of locations and select the one that has a higher crime rate. As in Sec. 4.1.2, we report accuracy on the pairwise trials. We also follow a similar procedure to ensure high quality of work, and provide feedback on the task to keep workers engaged. In total, we collected annotation on 2000 pairs of panoramic images sampled randomly from San Francisco.

Computers: As before, we use Gradient features for predicting the crime rate. We train a SVR on the crime rates as shown in the map on Fig. 2(b). As the crime rate does not vary significantly throughout the city, except at few specific locations, we cannot divide the city using a split line (as done in Sec. 4.1.1) as either the train or test split may contain little to no variation in the crime rate. Instead, we randomly sample 5 train/test splits of equal size without taking location into account.

Results: The human accuracy on this task was 59.6%, and the accuracy of using Gradient based features was 72.5%, with chance performance being 50%. This indicates the presence of some visual cues that enable us to judge whether an area is safe or not. This is often associated with our intuition, where we choose to avoid certain areas because they may seem 'shady'. Another interesting thing to note is that computers significantly outperform humans, better being able to pick up on the visual cues that enable the prediction of crime rate in a given area. However, note that since the source of the data [1] is a self-reporting website, the signal may be extremely noisy as comprehensive crime information is unavailable; an area with more tech-savvy people might have a higher crime rate just because more people are using the particular application.

6. Analysis

In this section, we analyze some of the previously presented results in greater detail. Specifically, we aim to address the question: why do computers perform so well at this task? What visual cues are used for prediction? In order to do this, we approach the problem in two ways: (1) finding the set of objects that might lead us to believe a particular establishment is near (Sec. 6.1), and (2) using mid-level image patches to identify the importance of different image regions (Sec. 6.2).

6.1. Object importance

To analyze the importance of different objects in the images, significant effort would be required to label such a large-scale dataset manually. To overcome this, we use the ImageNet network [24], trained on 1000 object categories, to predict the set of objects in an image. As done in [21], we train SVR on the object feature. Now, we can analyze the weight vector to find the impact of different objects on the distance to McDonald's. Note that the smaller (or more negative) the weight, the more correlated an object is with close proximity to McDonald's. The resulting sets of objects with different impact on proximity are as follows:

- **High negative weight**: taxi, police van, prison house, cinema, fire engine, library, window screen
- **Neutral (close to zero)**: butterfly, golden retriever, tabby cat, gray whale, cheeseburger

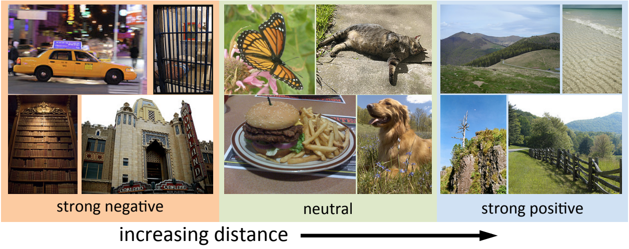


Figure 5. Visualizing object importance with respect to distance to McDonald's. The results are fairly intuitive - cars and structures looking like buildings tend to be located close to McDonald's, while alps and valleys are far away.

- **High positive weight**: alp, suspension bridge, headland, sandbar, worm fence, cliff, lakeside

We visualize the results in Fig. 5, and observe that they closely follow our intuition of where we might expect to find McDonald's. In the supplemental material, we provide additional analysis on the unique set of objects that may be correlated with particular establishments in different cities. For example, we find that in Paris, McDonald's tend to be closely located with cinemas/theaters and turnstiles while hospitals are not; instead, in Chicago, hospitals tend to be correlated with handrails while McDonald's are not.

6.2. Image region importance

To investigate the importance of image regions, we use a method similar to [22]: first we densely sample square image regions ranging from $40 * 40$ pixels to $80 * 80$ pixels. Representing each region with Gradient features (Sec. 4.1.1), we learn a dictionary of image regions using k-means clustering. Then we use LLC [39] to assign each image region to a dictionary element, and apply max-pooling to get the final image representation. Using this representation, we train SVR to predict the distance to the nearest establishment. Thus, the learned weights signify the importance of each region type - the results are shown in Fig. 6. Furthermore, we find that this feature representation is fairly effective for representing images, achieving a test accuracy of 0.64 on finding McDonald's in NYC (vs 0.66 for Gradient). As done in [22], this representation could also be combined with Gradient features to boost performance.

7. Conclusion

In this paper, we propose the problem of *looking beyond the visible scene* i.e. inferring properties of the environment instead of just identifying the objects present in a scene or assigning it a class label. We demonstrate a few possibilities for doing this, such as predicting the distance to nearby establishments, navigating to them based only on visual cues. In addition, we also show that we can predict the crime rate in an area without any actors performing crimes in real time. Interestingly, we find that computers are better at assimilat-

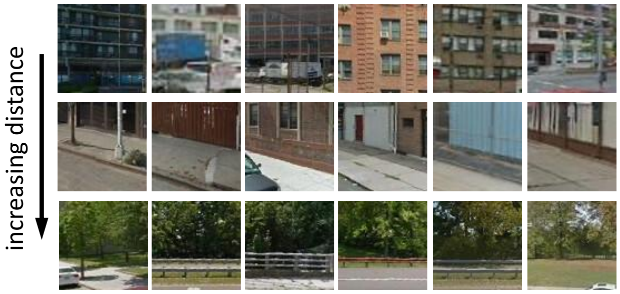


Figure 6. Importance of image regions with increasing distance from McDonald's - each row shows different images belonging to the same cluster. The first row shows regions that tend to be found close to McDonald's while the last row shows regions found far away. This matches our intuition as we expect to find more McDonald's in areas with higher population density (e.g. city center).

ing this information as compared to humans outperforming them for a variety of the tasks. We believe that this work just touches the surface of what is possible in this direction and there are many avenues for further exploration such as identifying scene attributes used for prediction, predicting crime in the future instead of over the same time period, or even predicting the socioeconomic or political climate of different locations using only visual cues.

References

[1] http://sanfrancisco.crimespotting.org/. 2, 7

[2] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. 2

[3] S. Y. Bao, M. Sun, and S. Savarese. Toward coherent object detection and scene layout understanding. *CVPR*, 2010. 2

[4] I. Biederman. Aspects and extensions of a theory of human image understanding. *Computational processes in human vision: An interdisciplinary perspective*, 1988. 4

[5] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*. 2012. 2

[6] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010. 2

[7] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. A discriminative model for learning semantic and geometric interactions in indoor scenes. *ECCV*, 2012. 2

[8] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013. 2

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 4

[10] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. *IJCV*, 2011. 2

[11] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *NIPS*, 1997. 4

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 2008. 4

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 4

[14] R. Guo and D. Hoiem. Beyond the line of sight: labeling the underlying surfaces. In *ECCV*. 2012. 2

[15] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*. 2010. 2

[16] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013. 2

[17] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008. 2

[18] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. *h ttp://caffe. berkeleyvision. org*, 2013. 4

[19] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013. 2

[20] R. Khan, J. Van de Weijer, F. S. Khan, D. Muselet, C. Ducottet, and C. Barat. Discriminative color descriptors. *CVPR*, 2013. 4

[21] A. Khosla, A. D. Sarma, and R. Hamid. What makes an image popular? In *WWW*, 2014. 3, 7

[22] A. Khosla, J. Xiao, A. Torralba, and A. Oliva. Memorability of image regions. In *NIPS*, 2012. 7

[23] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 5

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012. 4, 7

[25] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2, 4

[26] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361, 1995. 4

[27] C. Li, A. Kowdle, A. Saxena, and T. Chen. Toward holistic scene understanding: Feedback enabled cascaded classification models. *PAMI*, 2012. 2

[28] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009. 2

[29] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 2002. 4

[30] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001. 2, 4

[31] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 2007. 2

[32] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 2

[33] G. Patterson and J. Hays. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012. 2

[34] M. Potter. Meaning in visual search. *Science*, 1975. 4

[35] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*. 2012. 2

[36] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 2

[37] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 5

[38] J. Van De Weijer, C. Schmid, and J. Verbeek. Learning color names from real-world images. In *CVPR*, 2007. 4

[39] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *CVPR*, 2010. 4, 7

[40] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 2

[41] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012. 2

[42] Y. Zhang, J. Xiao, J. Hays, and P. Tan. Framebreak: dramatic image extrapolation by guided shift-maps. In *CVPR*, 2013. 2

[43] Y. Zhao and S.-C. Zhu. Scene parsing by integrating function, geometry and appearance models. CVPR, 2013. 2