Large-scale Privacy Protection in Google Street View

Andrea Frome[1], German Cheung[1], Ahmad Abdulkader[2], Marco Zennaro[1], Bo Wu[1],
Alessandro Bissacco[1], Hartwig Adam[1], Hartmut Neven[1], and Luc Vincent[1]
[1,2]Google, Inc, 1600 Amphitheatre Pkwy, Mountain View, CA 94043
[1]{afrome,gcheung,zennaro,bowu,bissacco,hadam,neven,luc}@google.com
[2]ahmad@abdulkader.org

Abstract

The last two years have witnessed the introduction and rapid expansion of products based upon large, systematically-gathered, street-level image collections, such as Google Street View, EveryScape, and Mapjack. In the process of gathering images of public spaces, these projects also capture license plates, faces, and other information considered sensitive from a privacy standpoint. In this work, we present a system that addresses the challenge of automatically detecting and blurring faces and license plates for the purpose of privacy protection in Google Street View. Though some in the field would claim face detection is "solved", we show that state-of-the-art face detectors alone are not sufficient to achieve the recall desired for large-scale privacy protection. In this paper we present a system that combines a standard sliding-window detector tuned for a high recall, low-precision operating point with a fast post-processing stage that is able to remove additional false positives by incorporating domain-specific information not available to the sliding-window detector. Using a completely automatic system, we are able to sufficiently blur more than 89% *of faces and* $94 - 96\%$ *of license plates in evaluation sets sampled from Google Street View imagery.*

1. Introduction

In the last two years, there has been a rapid expansion of systematically-gathered street-level imagery available on the web. The largest and probably most well-known collection to date is Google Street View[1] [13]. Street View launched as part of Google Maps in May 2007 and has expanded rapidly since, at last count providing imagery from twelve countries on four continents. Other smaller products have found their niches around the world, including Mapjack[2], Everyscape[3], and Daum's Road View[4]. What makes these products truly unprecedented is the amount and density of consistent, geo-positioned imagery they make available to users. This combination of scale and accurate location allows users to effectively search and find specific points of interest, while also making it possible to virtually wander through the street-level environment, thus enabling a wide range of uses including real estate search, virtual tourism, travel planning, enhanced driving directions, and business search.

As these products expand, they become more useful, but a major challenge has emerged in demonstrating that this does not have to come at the price of individual privacy. Primary among privacy concerns is the publication of potentially personally-identifiable information such as a person's face or license plate captured as a side-effect of gathering the target imagery. In this paper we address the challenge of automatically removing faces and license plates from street-level imagery. This is a formidable challenge for four main reasons. First, the scale is large, which requires fully-automatic, optimized algorithms and a large amount of computing resources. Second, there is little control over the conditions of capture, and the appearance of objects can vary widely: people with a variety of physical appearances are captured close to the camera, in the distance, in shadow, behind car windows, at a wide range of angles, at a variety of scales, on cell phones, wearing hats and sunglasses, occluded, cut off at the edge of the image, and distorted by image compression (Figure 4). In many of these cases it could be argued that the person is still identifiable. License plates are challenging due to the large variation in viewing angle, shadows, occlusions, and the variation among plates within and across geographic locations. Third, and most importantly, in order to protect individuals' privacy at the moment the imagery becomes public, the recall of faces and license

[1] http://maps.google.com/help/maps/streetview

[2] http://www.mapjack.com

[3] http://www.everyscape.com

[4] http://local.daum.net/map

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

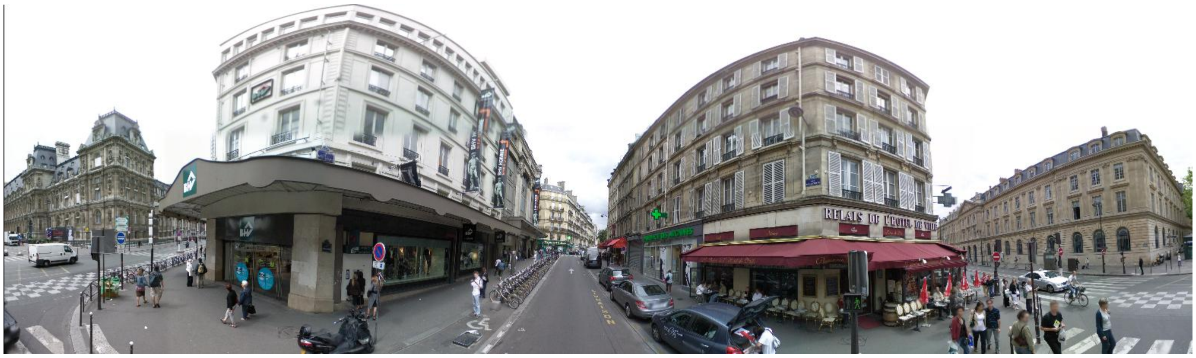


Figure 1. The end product: a stitched, redacted panorama (at reduced resolution).

plates would ideally be 100%. This is beyond the reach of state-of-the-art automatic methods, and as we show in this paper, one of the best detectors in the world achieves less than 78% on our in-the-wild face data set. So while face detection is considered "solved" by some, we found out-of-the-box face detectors alone to be surprisingly inadequate for this problem. Lastly, we need to preserve the quality of the images while still achieving high recall. This requires us to control our false-positive rate and to obscure faces and license plates in an manner unobtrusive enough so that a viewer's eye is not drawn to erroneously blurred regions.

In order to address these challenges we combine existing computer vision and machine learning techniques in a way specially suited to our domain and to faces (Section 3) and license plates (Section 4). For faces, we first trained and tuned a fast integral image-based sliding window face detector to achieve a high-recall, low-precision operating point (see Section 3.2.1). From the detected boxes, we compute sets of features specialized to our setting, and pass those feature vectors through a neural network-based post-processor that is tuned to remove as many false positives as possible while keeping almost all the true positives (Section 3.2.2). For license plate detection, we applied the same basic components, with a few modifications, which we discuss in Section 4.2. To produce the final panorama, we blur the detected boxes and publish the imagery in Street View (Section 5).

The goal of this paper is to describe in detail the system for detecting and blurring faces and license plates in Google Street View imagery, and to provide qualitative and quantitative results demonstrating our performance on a large data set from around the world. We face a challenge in this work in that, for the sake of protecting privacy, we cannot publish raw images that contain non-redacted faces or license plates. The purpose of this paper is not to explore in depth alternate design options, so in order to provide a forum for other academic work within this setting, we are releasing a special data set in conjunction with this paper, and report our performance separately on that data set (Section 3.1). Also, in order to provide a context for our face detection performance, we compare to a small set of state-of-the-art industrial and academic face detectors, though this is not intended as a critical evaluation of those detectors as they have not been tuned or trained for our setting.

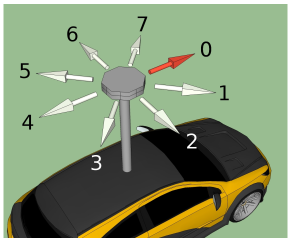


Figure 2. A diagram showing the car-mounted camera rosette and the positions of the 8 radially-arranged cameras. Fish-eye camera not shown.

2. Image Capture

The imagery with which we are working was collected from a moving vehicle using a custom camera system mounted on a rig on the roof of the car. The rig consists of nine 5-megapixel cameras arranged into a rosette, with eight of the cameras aimed roughly parallel to the ground in a radial configuration at an even spacing and one fish-eye lens pointed upwards (Figure 2). For purposes of this work we ignore the upward-facing camera because any faces that may be captured will be too low-resolution to be identifiable. The cameras are synchronized, and the set of images taken from the nine cameras at a given time step can be stitched together to form a panorama (Figure 1). In the rest of the paper we will only work with images that came from the individual cameras before they are stitched into a panorama (Figure 3). We will refer to the individual images as "camera images" and the set of images that are taken together as a "panorama set". We can assume that identical car and camera configurations are used to capture imagery gathered in different geographic regions and across time.

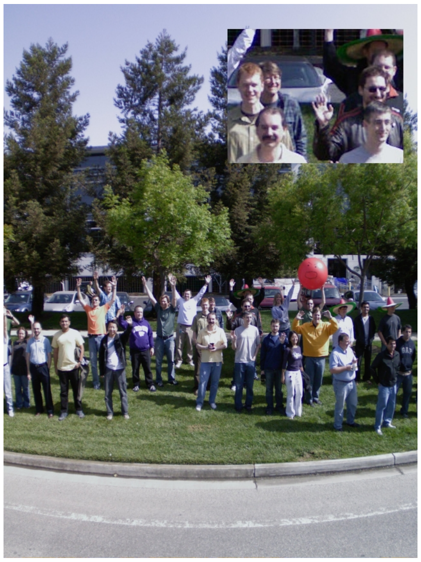


Figure 3. A Campus Face Set image from camera 2, at reduced resolution, with a full-resolution inset.

3. Faces

3.1. Evaluation data sets

We present face detection results on two hand-labeled evaluation data sets. We built the first of the face data sets from 29,106 camera images sampled from three major world cities (one each from the United States, northern Europe, and southern Europe) on 162 different days. The faces in each camera image were marked by a person who we instructed to label only faces of real people that they judged to be possibly identifiable, and not to mark faces on billboards or other signs. This data set, which we refer to as the “Cities Face Set” contains 1,614 labeled faces.

Due to the same privacy concerns we are aiming to address, we cannot release the Cities Face Set in its unblurred form or publish examples without redaction. For this reason, a second data set was created for academic purposes[5]. We refer to this set as the “Campus Face Set”, and it consists of 19,187 images with 15,075 faces of consenting individuals in 2,176 of the images (Figure 3). This data set is somewhat artificial in that the participants knew the vehicle would be driving by, so the large majority of people are looking at the camera, thus skewing the set toward frontal face views, which are typically easier to detect. Nonetheless, this set still presents a challenge and has a much higher face density.

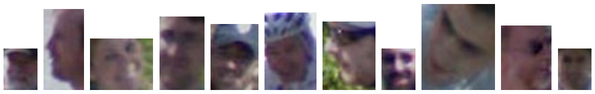


Figure 4. A sample of difficult ground-truth faces from the Campus set that we consider “identifiable” (upscaled by 2×).

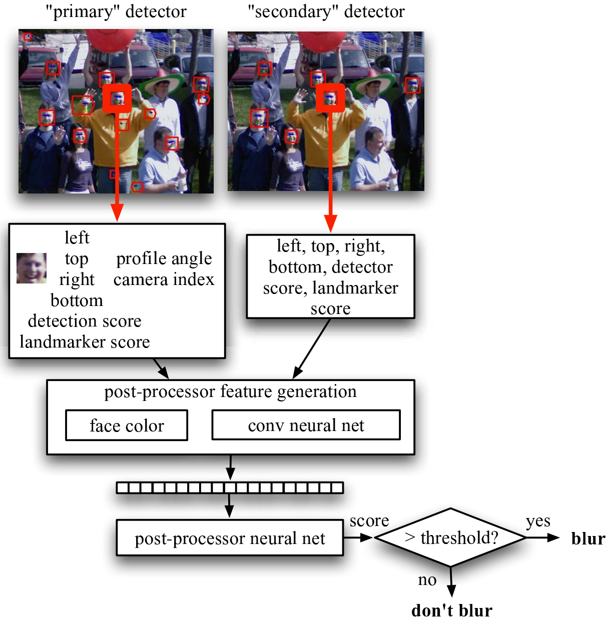


Figure 5. Data flow diagram for the face detection system.

3.2. Algorithms and system

The face detection system is built from a *primary* high-recall sliding-window detector, a *secondary* high-precision, low-sensitivity detector, and a fast post-processing stage that generates a feature vector from the detector outputs and scores the box (Figure 5). Here we describe the components of the system at a high level, and drill into the components in more detail below.

The top left box in Figure 5 shows the output of the “primary” high-recall detector for part of a Campus Set image. For a given image, we run the primary and secondary detectors in parallel and store the outputs. In Section 3.2.1 below we describe the detectors in more detail. Each box from the primary detector is passed to the post-processor, where a feature vector is created from the outputs of the detectors, the index of the camera image in the rosette, and the pixels, and the feature vector is passed through a neural network. If the network output meets a given threshold, the box is blurred in the final image. See Section 3.2.2 for details.

Our two-stage system is close in spirit to the 2006 paper by Hoiem, et al. [4] that improved the performance of an existing pedestrian detector by setting the threshold of the SVM-based sliding window detector to increase the recall, and then removing false positives with a Bayesian network that incorporated 3D scene structure, estimated horizon, and camera viewpoint. Our post-processor has the advantage of

[5] To obtain a copy of the data set for academic use, please send an e-mail to `iccv2009_face_data@google.com`.

being fast, which is possible in large part because we can leverage the consistencies in our image set.

3.2.1 Primary and secondary face detectors

At first, we tried an early version of the face detector used in Google Picasa. This detector is tuned for high precision, but we found its recall to be too low for our application, so we retrained and tuned the detector to increase the sensitivity. This gave us two detectors at complementary operating points: a high-recall primary detector, and a high-precision, "low-sensitivity" secondary detector. Both use the same features and algorithm. The Picasa detector has since been retrained with additional data, and we upgraded our secondary detector to the newer version (though not yet our primary).

The detector uses a fast sliding-window approach over a range of window sizes. It employs a linear combination of a heterogeneous set of feature detectors, which are based on families of features of varying complexity encompassing (1) simple but fast features such as bit features, as well as (2) more expensive but more informative features such as Gabor wavelets. The detector is trained by minimizing an objective function that employs a logistic loss term and L1 regularization. The output is a score assigned to each window in the range $[0, 1]$. When all scales are processed, the remaining windows are filtered and merged according to their scores and overlap across scales. We run three separate detectors using the same aspect ratio, covering profile angles (pan or yaw) of 0-30, 30-60, and 60-90 degrees, and the output scores for the different angles are combined. The detector covers a tilt (pitch) angle of ± 30 degrees.

We refine the face detection score further by adding a module we call the landmarker which pinpoints facial feature locations within a face bounding box. Features extracted at those locations are then used to obtain a refined score that indicates the probability of a face being present.

We made several adjustments to arrive at our high-recall detector. First, we decreased the minimum box width from 20 pixels for the secondary detector to 12 pixels in the primary detector[6]. Second, we increased the contrast sensitivity which helps us, for example, detect faces behind glass. We also added about 200 faces gathered from low-resolution street-level imagery to the training set, decreased the stride of the sliding windows, and retrained with a target recall-precision trade-off tuned heavily to favor recall.

Our detection algorithm belongs to a large family of sliding window detectors, including such seminal detectors as those by Rowley[11], Schneiderman and Kanade[12], and Viola and Jones[14] and more recent detectors with very impressive performance, such as the Tsinghua University detector[5]. Our system as a whole is not tied to the choice of detector, but could be built upon any detector that can be retrained to achieve a different recall/precision trade-off. In Section 3.3 we compare the performance of our final system to our base detector as well as the Tsinghua detector.

features	face	plates
left, top	2	2
width (right - left), height (bottom - top)	2	2
height / bottom	1	1
width $*$ height	1	1
camera index (categorical)	8	8
angle (categorical)	3	2
detector score	1	1
landmarker score	1	
mean hue, saturation, value	3	3
secondary detector overlap	-	
average face color probability	1	
car model agreement		2
conv. neural net output	1	1

Table 1. The features we use for post-processing with the number of elements in the feature vector for each in the right column. Some features are "categorical" in that they are expanded into a binary vector where just the entry in the position corresponding to the active value is one. For example, if the box is from camera zero and there are eight cameras, then the entries for the camera index are $\{10000000\}$.

3.2.2 Post-processor

We perform post-processing using a neural network with features derived from the box pixels and the detector outputs (see Table 1). We did early experiments that showed that a neural network with one or two hidden layers achieves better results than logistic regression. The face post-processor is a fully-connected neural network with a total of 58 nodes: 24 input, 2 output, and two hidden layers of 16 nodes each. The final value from the neural network is read from the second of the two output nodes. The features were designed to capture information that was not available to the detectors. Most of them are self-explanatory, but the secondary detector overlap, the height-to-bottom ratio, the color model probability, and the convolutional neural net output require explanation.

The ratio of the box height to box bottom is intended to capture whether the real-world size of the object indicated by a candidate box is reasonable, given the examples seen in the training set. This was inspired in part by the work of [4] which infers the 3D height of objects from image information in order to reject false positives from a high-recall detector. In that work, they use $\frac{\hat{v}_1}{\hat{v}_1 - \hat{v}_2} = \frac{y_c}{y}$ to relate the bottom edge of the object in the image ($\hat{v}_1$), the height of the object in the image ($\hat{v}_2$ is the object's top edge), the camera height (y_c), and the height in 3D of the plane at

[6] We determined, from a survey of our coworkers, that a 12-pixel-wide box was the smallest box from our imagery in which a face is still identifiable. Both detectors use a maximum with of 1,000 pixels, which we determined from our ground truth data to be sufficient to cover all faces.

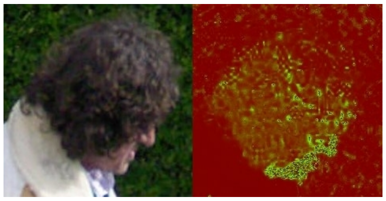


Figure 6. Face color model response. Low probability is red, high probability is green. Note the high response to face and hair pixels but not the background.

the bottom of the object (y). If we can assume that our camera height is always the same and that peoples' heights are consistent between our training and test sets, then we can use the ratio $\frac{\hat{v}_1}{\hat{v}_1 - \hat{v}_2}$ to implicitly capture the 3D height of the object.

We use the boxes from the secondary detector to "overrule" the decision of the neural network. The boxes returned from the secondary detector are almost always a subset of the boxes from the high-recall primary detector (with some minor differences due to final box merging), and the precision of the secondary detector is very good compared to the precision of the primary detector plus post-processor, so when a box is returned from that detector, we skip running the post-processor neural network and return a "perfect score" from the post-processor for the corresponding primary box. This protects us from mistakes that the post-processor might make on easily-identifiable faces due to a lack of representation in the training set, such as rejecting a face on a balcony because the height-to-bottom ratio lies outside the range found in the training data. In Section 3.3, we compare the recall-FPR curves for the primary and secondary detectors.

Included in our feature vector is an average face color probability, computed for a candidate box from a per-pixel face color probability measure. We built a 128-by-128-bin color histogram in hue-saturation space from the pixels of primary detection boxes that overlap at least 10% with a ground-truth face box. We did the same with detection boxes that were not labeled as faces to create a color histogram of non-face pixels. Note that we do not label skin versus non-skin pixels, so our model learns all the colors that occur within our detection boxes. At run-time, we compute $p(\mathbf{face}|color)$ for each detected pixel using the two histograms, and use the average for the box as a post-processing feature (Figure 6). The use of skin color in face detection has been widely explored in the literature ([3]); our approach is related to [7], and our choice of hue-saturation space is supported by the findings in [6].

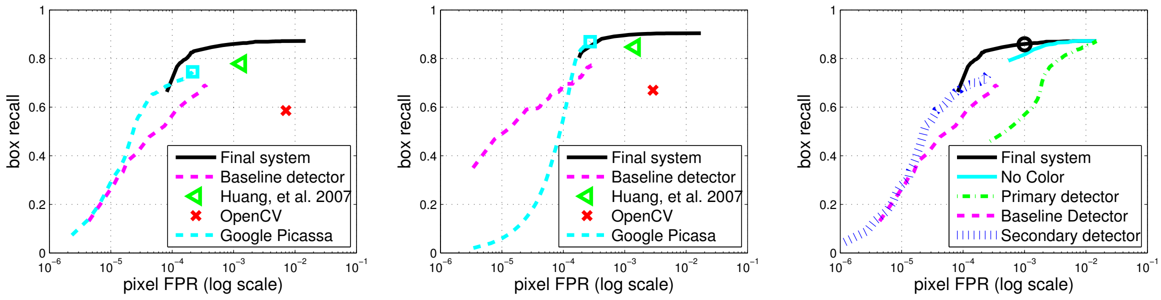
We also include in the feature vector the output of a convolutional neural network that acts on the contents of the candidate box. The box is converted to grayscale, rescaled to 64-by-64, and converted to a feature vector with a floating point value for each pixel. The network has 4,096 inputs, uses local receptive fields and shared weights, and has two convolutional layers and two sub-sampling layers. This architecture is considered a deep architecture[1] and training requires a special technique, *constructive layer addition*[9], which is commonly used to train deep architectures.

Training data for the post-processors was generated by first running the primary detector and secondary detectors, generating the feature vectors as described above, and then assigning a positive value to the example if and only if we had a hand-marked ground-truth box that overlapped the detected box by at least 10%. The target output nodes were $\{0, 1\}$ for positive examples and $\{1, 0\}$ for negative examples. We used stochastic gradient descent to minimize a cross entropy energy function[1] defined over all the boxes in the training set. Weight decay was applied during training for regularization[1]. In addition, we seek to maximize box precision and recall on per-pixel basis, so the cross entropy energy function was modified from its original form to reflect the area of each box. In this manner, a larger box contributes more to the overall energy function than a smaller box. This adjustment results in a better pixel FPR (pixel false positive rate) but slightly worse box precision.

The training set used for the post-processor consists of about 71,000 faces hand-labeled in images sampled from a city not represented in our evaluation set. This set was split into different subsets, which were used to train and validate the components of the post-processor (face color model, convolutional neural net, and post-processor neural network).

3.3. Face results

We evaluate our performance using three metrics: (1) automatically-generated per-box recall, (2) an exhaustive hand-counted per-box recall, and (3) pixel false positive rate (pixel FPR). For purposes of privacy protection, the hand-counted per-box recall is the most important number. The hand count is performed by looking at all the ground-truth boxes, blurred as they will be in the final product, and counting the number that are not "sufficiently blurred", where a sufficiently blurred face is one where the face has been blurred enough to obscure the facial features (see Figure 7). The hand count is necessary for an accurate number because the ground truth boxes are often larger than the boxes from the automatic system. The hand count is time-consuming, however, so to generate recall-FPR curves and to do relative comparisons, we use an automatic box recall where a ground-truth box is counted as recalled if the mask of detection boxes that overlap it cover at least 50% of the box. While privacy is the primary goal, we also must maintain the quality of the imagery in the final product, which means that we must consider the impact of our precision. We measure our precision using pixel FPR because it is the best measure of unnecessary image degradation: it is the percentage of all pixels that have been blurred that lie outside any ground truth box.



(a) Cities Faces (b) Campus Faces (c) Cities Faces (system)

Figure 8. Face results showing (a) comparison of our final system and our baseline with other detectors on the Cities Set, (b) same comparison on the Campus Set, and (c) comparison between versions of our detector and the full system on the Cities Set. The circle marker in (c) denotes the operating point of our system at our chosen threshold. The Tsinghua and OpenCV detectors are shown by single points because the code provided does not produce a recall-precision curve. Recall that our system keeps boxes from the secondary detector with a score of zero, so our curve does not extend into the lower-recall regime, though the performance overlaps strongly with the Picasa detector, which is the same as our secondary, but run at a 12-pixel box width instead of 20.

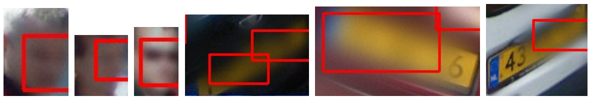


Figure 7. Face and license plate boxes from our evaluation set, blurred by our system with detection boxes added in red. These examples illustrate that a detection box can cover a small area of the ground-truth box, but we still "sufficiently blur" the face or license plate. This causes the gap between our hand-counted recall and automatic recall, which is largest with EU license plates. Faces upscaled by $1.5\times$.

On the Cities Data Set, we are able to achieve a hand-counted recall of 89.0%, which we feel is a remarkable achievement considering the very difficult faces we aim to detect. On the Campus Face Set, which is an easier set because it is biased toward frontal faces, we achieve a hand-counted recall of 90.7%. Our ground-truth boxes were selected using a conservative notion of identifiable which causes our recall to be low. Identifiability exists on a continuum and is highly subjective; in Figure 9 we show our recall at varying levels of identifiability.

We use the automatic box recall and FPR to compare different versions of our own detector and to provide some context for our performance by comparing to the Tsinghua University detector[5], the detector used in Google's Picasa, and the OpenCV detector[7]. It is important to emphasize that those comparisons do not serve to show that our detector is superior in general; those detectors do not have available to them the signals that we leverage to achieve our performance, and in fact, we could base our system on any of those detectors. We ran the Picasa, OpenCV, and Tsinghua detectors with their default settings, except to set the minimum face size to the same size we use with our primary detector (12-pixel box width). We also show comparisons to our "baseline" detector, which is our primary detector before it was retrained for high recall, but using a 12-pixel minimum box width to allow for a fair comparison. In Figure 8(a) we show that on our Cities set, we dominate the other detectors in the high recall regime. In 8(b) we show

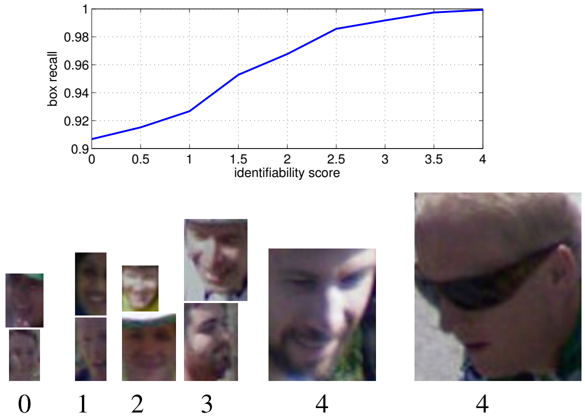


Figure 9. Over 20 of our fellow engineers rated the identifiability of the faces we did not sufficiently blur in the Campus Set on a scale from zero to four, where zero is almost unidentifiable and four is easily identifiable. The authors did not participate. We gathered three scores for each face and took the mean to get an identifiability score for each face. We show two randomly-selected faces from each level above (at half-resolution to save space and preserve relative size). A plurality (42%) of the missed faces were rated as a one, with 32% rated as a two. Most of the ratings were within one level of the mean, indicating a high level of agreement, with the most disagreement on faces in $[1, 3]$. Above we show a graph of what our hand-counted recall would be if we took each of the identifiability levels as our threshold for "identifiability".

[7] http://opencv.willowgarage.com

the same results for the easier Campus set. In 8(c), we show the relative performance of different versions of our detector, including our baseline detector, our primary detector without post-processing, our secondary detector, and our full system.

In addition to having high recall and a relatively low pixel FPR, our system is also fast. To process a 5-megapixel image, our primary detector takes an average of 6-8s, our secondary detector takes 1-2s, and the post-processor takes on the order of 100ms, for a total of 7-10 seconds, depending upon the image. Compare this to about 30s for the Tsinghua detector using the same minimum face size.

4. License plates

4.1. Evaluation data sets

We separate our license plate evaluation ground truth into two sets, one for the United States and one for the European Union. EU license plates are longer and thinner than US license plates, thus changing the aspect ratio, and EU plates are different than US plates in their font, text size, and coloring. Most EU plates vary little in appearance from one another, and while US license plates can vary between and within states, we have found it sufficient to use a single detection system for all US plates. The US data set includes 17,680 plates from 120,448 images sampled from three metropolitan areas from different geographical regions, and the EU set includes 12,768 plates from 40,295 images sampled from three EU countries on 168 different driving days. For privacy reasons, we cannot release these data sets and must redact any plates shown in the paper.

4.2. Algorithms and system

The license plate system has the same overall structure as the face system and uses the same basic algorithms, with these exceptions: (1) the feature set is reduced (Table 1), (2) there is no secondary detector, and (3) instead of a color model, it makes use of a simplified "car detector" to provide context. We directly adapted the face detector from Section 3.2.1 to the task of license plate detection, and to our surprise, it provided a high recall. The ground truth boxes are rectilinear, and were drawn as tightly as possible around the license plates such that all four corners of the plates were touching some side of the box (see Figure 10). We do not provide context outside the plate box to the license plate detector because this would require the detector to generalize over the variation in cars in addition to the variation in plates. Instead we leverage the surrounding context in our post-processor.

Because plates captured at an angle result in ground truth boxes that are more square and because they have a characteristic pattern in the corners, we use two channels in the detector for frontal and slanted plates, each with one aspect ratio. To prepare for training, we hand-separated the plates with the aid of a simple classifier. Otherwise, the algorithm and training for the license plate detector follows the same methodology as in 3.2.1. We trained separated detectors for US and EU plates to accommodate the different aspect ratio.

The post-processor neural network has a total of 24 input nodes, two fully-connected hidden layers of 16 nodes each, and two output nodes. Again we use the output of a convolutional neural network as input to the post-processor neural network, and use the same 64-by-64 pixel box size, architecture, and methodology as with faces (Section 3.2.2).

We also provide to the post-processor network two features from a sliding window-based car detector, which acts as a "context" signal for license plates. Figure 11 shows the output of the car detector for one image. The car detector uses a variety of features, including (1) Haar features computed over pixel intensity, Harris corner responses, and gradient responses, and (2) gradient and intensity histograms computed over the full box. The car boxes used as training data are automatically generated by selecting an expanded image region around the ground-truth license plate boxes (Figure 11). The outputs of the detector are not merged, providing multiple overlapping car box candidates. To generate the post-processor context features, we compute an ideal car box for the license plate according to the same proportions used in training, and find the "best" detected car box, which is the the one that best overlaps the ideal car box. The two features provided to the post-processor are (1) the overlap (intersection over union) of the ideal and best detected car boxes, and (2) the detection score of the best box.

We used about 10,000 hand-marked boxes to train the detectors, about 36,000 boxes to train the car context detectors, and about 18,900 boxes to train the post-processors. The training data was sampled from US and EU major metropolitan areas which are not represented in our evaluation set.

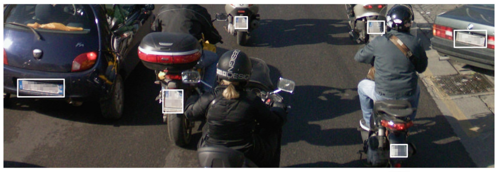


Figure 10. Examples of how ground-truth license plates were marked by humans (in white). Image hand-redacted.

There has been less work in the academic literature on license plate detection than on faces. Porikli and Kocak [10] train a covariance descriptor-based neural network classifier for license plates; and Dlagnekov [2] applied the Viola and Jones's method directly to license plate detection. There has also been work incorporating local context for detection, for example the work of Kruppa et al. [8] that uses an upper-

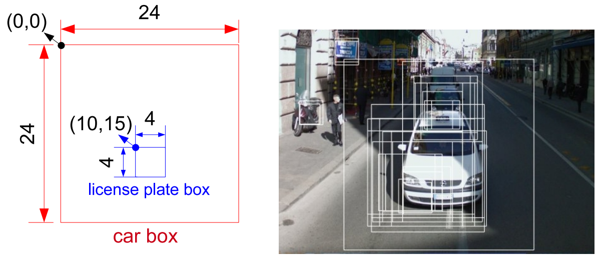


Figure 11. Left: the geometry of a context box relative to a ground-truth or detected US license plate. Right: example car detector output.

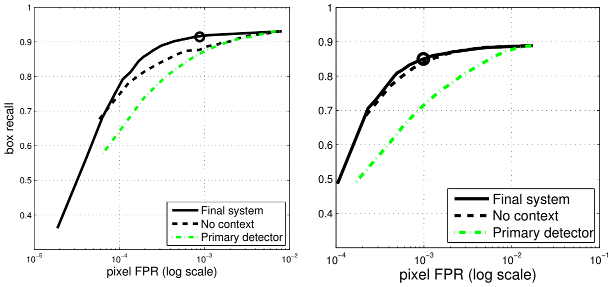


Figure 12. License plate results for the US (left) and EU (right). The circle shows our system's operating point, chosen using a held-out validation set.

body classifier to improve face detection.

4.3. License plate results

We use the same evaluation methodology as for faces, but for plates, we consider a license plate "identifiable" if three or more characters of the plate number are legible and consider it "sufficiently blurred" if we blur it such that it is no longer identifiable. Our hand-counted recall at our chosen operating point is 96.5% for US and 93.6% for EU.

With license plates, there is a wider gap between the hand-counted and automatic recall because even fewer pixels need to be detected in order to sufficiently blur a plate (Figure 7), especially for EU plates. For this reason we use an automatic box recall threshold of 0.3, though the automatic recall still significantly under counts our performance. In Figure 12 we show a comparison between our final system, our detectors before post-processing, and a version of our system without the car context features.

5. Publication

Using the final boxes, we redact the camera images, which are stitched to give the panorama. We need a redaction method that: (1) is irreversible, (2) is not too strange-looking on faces, (3) hides some of our many false positives, and (4) makes it obvious to the public that redaction has occurred. (Note that the fourth requirement eliminates algorithms that cleanly swap faces or facial features.) We chose to apply a combination of noise and aggressive Gaussian blur that we alpha-blend smoothly with the background starting at the edge of the box.

Lastly, while this system is state-of-the-art, the privacy protections in Google Street View don't end here. Our users help us to continually narrow the gap between our automatic performance and 100% recall by reporting unblurred faces and license plates which we then blur in the live product.

Acknowledgments

Many thanks to those that contributed to the privacy effort and helped make this paper possible: Raquel Romano, Dennis Strelow, Daniel Filip, Bradley Bossard, Henry Rowley, Sergey Ioffe, Elaine Filadelfo, Gavin McGinty, Stephen Chau, Tim Porter, David Whealan, Ben Lee, and Peter Fleischer. We would especially like to thank Johannes Steffens for providing the baseline training and detection framework.

References

[1] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, 2007.

[2] L. Dlagnekov. Car license plate, make, and model recognition. In *CVPR*, 2005.

[3] E. Hjelmas and B. K. Low. Face detection: a survey. *Computer Vision and Image Understanding*, 83:236–274, 2001.

[4] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

[5] C. Huang, H. Ai, Y. Li, and S. Lao. High performance rotation invariant multiview face detection. *PAMI*, 29(4):671–686, April 2007.

[6] H. F. J. C. Terrillon, M. N. Shirazi and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Proc. IEEE Automatic face and gesture recognition*, 2000.

[7] M. J. Jones and J. M. Rehg. In *Statistical color models with application to skin detection*, pages 274–280, 1999.

[8] H. Kruppa, M. Castrillon-Santana, and B. Schiele. Fast and robust face finding via local context. In *Joint IEEE International Workshop on VS-PETS*, 2003.

[9] R. Lengellé and T. Denœux. Training mlps layer by layer using an objective function for internal represeztations. *Neural Netw.*, 9(1):83–97, 1996.

[10] F. Porikli and T. Kocak. Robust license plate detection using covariance descriptor in a neural network framework. *AVSS*, 2006.

[11] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *PAMI*, (1):23–38, January 1998.

[12] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *CVPR*, 2000.

[13] L. Vincent. Taking online maps down to street level. *IEEE Computer*, 40(12):118–120, December 2007.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.