Comparison of Visual Datasets for Machine Learning

Kent Gauen*, Ryan Dailey*, John Laiman*, Yuxiang Zi*, Nirmal Asokan*, Yung-Hsiang Lu*,
George K. Thiruvathukal†,Mei-Ling Shyu‡,Shu-Ching Chen§

*School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN, USA
{gauenk, dailey1, jlaiman, zi, nasokan, yunglu}@purdue.edu
†Deparment of Computer Science
Loyola University Chicago, Chicago, IL, USA
gkt@cs.luc.edu
‡Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL, USA
shyu@miami.edu
§School of Computing and Information Sciences
Florida International University, Miami, FL, USA
chens@cs.fiu.edu

***Abstract*—One of the greatest technological improvements in recent years is the rapid progress using machine learning for processing visual data. Among all factors that contribute to this development, datasets with labels play crucial roles. Several datasets are widely reused for investigating and analyzing different solutions in machine learning. Many systems, such as autonomous vehicles, rely on components using machine learning for recognizing objects. This paper compares different visual datasets and frameworks for machine learning. The comparison is both qualitative and quantitative and investigates object detection labels with respect to size, location, and contextual information. This paper also presents a new approach creating datasets using real-time, geo-tagged visual data, greatly improving the contextual information of the data. The data could be automatically labeled by cross-referencing information from other sources (such as weather).**

I. Introduction

Creating machines that can solve complex problems has been the dream for humans. Movies such as *2001 Space Odyssey* depicted machines capable of understanding human speech. Such goals were unattainable until recently. Machine learning can be applied to analyze data with underlying patterns that are difficult to express by mathematical rules. The complexity of machine learning models often requires massive amounts of data. Among all successful stories of machine learning, the technologies for recognizing objects in images and videos are one of the most noticeable achievements. Many factors contribute to this; among them, large datasets play crucial roles. Visual datasets with labels are used to train and evaluate machine learning models and lead to success in computer vision with novel architectures, such as AlexNet [1], Faster-RCNN [2], and FCIS [3].

Many datasets are created by searching and downloading images from the Internet (such as Flickr), for example, ILSVRC [4], COCO [5], SUN [6], and PASCAL VOC [7]. Another source of images is gathered from driving a car with a dash-cam for creating KITTI [8] and the Caltech Pedestrian Datasets [9]. Prior work on machine learning often chooses one dataset and demonstrates that the proposed solution is better than the existing work for this particular dataset. The most difficult part of creating a dataset is not acquiring the data—this can be automated easily. Instead, it is labeling the data. The very fact that the datasets are used for training and evaluating machine learning models means that the existing computer solutions are inadequate and the labels must be created by human efforts. This laborious process significantly slows down the creation of a dataset and could also affect the selection of the data. Some researchers suggest using computer graphics to create labels [10], but graphics technologies do not always generate "photo-realistic" images and videos.

To make labeling easier, some existing datasets use images or videos in which the objects of interest stand out. In other words, many images in these datasets have few objects, each occupying many pixels in the images. COCO [5] and SUN [6] are examples of a conscious movement away from this image selection bias, but this appears to be an exception. There is a need for labeling massive amounts of diverse data quickly and accurately.

Despite the importance of the datasets, a comparison is made only when a new dataset is introduced, and the comparison is often focused on only two to four other datasets [4][5][7]. There are exceptions to this. In Dollàr et al. [11], they compare 13 datasets and 12 methods. However, to the authors' knowledge, there is not a comparison across the datasets used in their paper. On the contrary, this paper presents a qualitative and quantitative comparison of eight

datasets and introduces network camera data as a new source for image datasets. This paper focuses on the distribution of object locations in the image and the ratio of the object size to the image size. In this paper, only the *"person"* class is considered for two reasons: **(1)** *"the ability to interact with people is one of the most interesting and potentially useful challenges"* [11] and **(2)** limiting our scope to the people class allows comparison between datasets with an arbitrary number of classes.

Due to the challenges in creating labels, this paper presents a new method for creating datasets by using real-time geo-tagged visual data. This approach gives researchers the flexibility to create new datasets that meet their specific needs. Moreover, the time and location metadata can greatly improve the data's contextual information. For example, an image taken at a traffic intersection in the early morning of a holiday has fewer vehicles than another image taken during rush hour. As another example, an image taken in a national park sees trees and sky, without any skyscrapers. This paper explains how to construct a system that can create datasets by retrieving real-time geo-tagged data from network cameras.

This paper is organized as follows. In Section 2, several commonly used datasets are introduced. Section 3 explains how to discover network cameras that can provide real-time and geo-tagged data. Section 4 compares the datasets. In Section 5, potential improvements of the datasets for future machine learning research are discussed. This paper is concluded in Section 6.

(a) **(b)**

Fig. 1. PASCAL VOC Example Images. The red boxes indicate "*axis-align bounding-boxes*" [7] marked by two pairs of pixel coordinates to indicate an object's location within an image. While *every* person *should* be marked, some instances of small people are not marked **(a)** and some large ones are missed **(b)**.

(a) **(b)**

Fig. 2. ImageNet Example Images. Images are marked with bounding-box labels. Notice how the labels are large and centered in the images.

II. Dataset Summary

This section summarizes many different visual datasets, including ImageNet Large Scale Visual Recognition Challenge (ImageNet or ILSVRC) [4], Common Object in Context (COCO) [5], Scene UNderstanding (SUN) [6], Pattern Analysis, Statistical Modelling, and Computational Learning Visual Object Classes (PASCAL VOC) [7], Institut National de Recherche en Informatique et en Automatique Person Dataset (INRIA) [12], the Caltech Pedestrian Dataset (Caltech) [9], and Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago Object Detections (KITTI) [8]. Appendix Table I lists the ID's of the example images selected by this paper. All images below only visualize the people class labels.

A. PASCAL VOC

PASCAL VOC [7] started its first challenge in 2005 for object detection and classification of four classes. The motivation was that "*methods are now achieving such good performance that they have effectively saturated on these datasets.*" [13]. By 2008, PASCAL VOC introduced 20 classes, and in 2009 became a popular benchmark for object detection [7]. In 2012, the last year of the competition, the PASCAL VOC training and validation datasets consisted of $27,450$ detection objects in $11,530$ images with 20 different classes. For segmentation, VOC's training and validation dataset consists of $6,929$ segmented objects in $11,530$ images. Figure 1 gives two PASCAL VOC example images.

B. ImageNet

The ImageNet [4] competition started in 2010 and currently continues to be one the most popular machine learning competitions. Many successful classification and object detection models have resulted from this competition, including Krizhevsky's AlexNet [1]. For object detection, ImageNet consists of $465,567$ images for training and $20,121$ images for validation for 200 different classes including guacamole, neck brace, iPod, chime, etc. Two example images are shown in Figure 2. To label the dataset, ImageNet utilized Amazon Mechanical Turk. ImageNet has been used as the data for other competitions as well, such as the training data for the Low-Power Image Recognition Challenge [14].

C. SUN

The SUN [6] dataset was started to provide researchers with a comprehensive collection of annotated images covering a wide variety of scenes. It contains $4,479$ object categories and $313,884$ instance segmentation labels in $131,067$ images. For people alone, SUN has $6,202$ instances of people in $2,062$ images. Instance segmentations follow the contours of the objects of interest, and hence they create tighter containers for object detection labels (as shown in Figure 3). However,

(a) **(b)**

Fig. 3. SUN Example Images. In **(a)**, the large label covering the crowd is also a person label. This could be interpreted as a large label for the crowd of people, but is a different . Comparing **(a)** and **(b)** shows how the number of instances varies across images. The colors indicate a single instance's segmentation, which may consist of two or more disconnected polygons. The colors also repeat. It should be apparent from the context if two labels of the same color are distinct or meant to be shared.

(a) **(b)**

Fig. 4. INRIA Example Images. In these images, there are many people unlabeled. Despite the missing labels, INRIA continues to be a popular dataset for machine learning and has contributed greatly to the computer vision community.

instance segmentation labels take more time to annotate than bounding-boxes labels.

D. INRIA

INRIA [12] People Dataset was created in 2005 and is comprised of $1,237$ bounding-box labels for people in 614 positive images. A positive image means that people are labeled in the image. The dataset also includes $1,218$ negative images containing no labels. It has been reported that INRIA contains missing labels [15]. There does not seem to be a rational for the missing labels. In both images of Figure 4, there does not appear to be distinguishing features between the labeled people and the unlabeled people. Despite the missing labels, the original INRIA dataset is still popular and has made laudable contributions to pedestrian detection [11].

E. KITTI

The KITTI [8] Vision Benchmark Suite began in 2012 and contains a variety of labels for tracking, scene flow, odometry, etc. Since KITTI's images come from a video file, there is also a temporal relationship between images for object tracking. For object detection, KITTI provides stereo images, temporal frames, Velodyne point clouds, and the bounding-box labels.

(a) **(b)**

Fig. 5. KITTI Example Stereo Image Pair. **(a)** is the left image and **(b)** is the right image. For object detection, bounding-box labels only exist for the left image and only for people.

(a) **(b)**

Fig. 6. Caltech Example Images. The Caltech Pedestrian dataset contains bounding-box labels for "people" and person. "People" labels are used when there are many people grouped together, like in the top of **(a)**, and on the left and right in **(b)**.

There are $4,487$ people labeled in $7,480$ images. Figure 5 shows an example stereo image pair. KITTI was labeled by the KITTI team with help from a set of hired annotators.

F. Caltech Pedestrian Dataset (Caltech)

Introduced in 2012, The Caltech Pedestrian Dataset [9] consists of approximately ten hours of 600×400 taken at 30 frames per second video from a vehicle driving through regular urban traffic. The dataset provides bounding-box labels of pedestrians for every frame a person is visible in two formats: the *full* and *visible* bounding-box label. A *full* label marks a tight bounding-box region around the entire person. If there is occlusion, the hidden area is estimated. The *visible* label marks an label only around the visible portion of the person. The example images in Figure 6 have the *full* and *visible* labels. This is different from PASCAL VOC's [7] handling of occluded images, where only the visible portion of an object is marked. Caltech contains a total of $346,621$ bounding-box labels in about $250,000$ frames.

G. COCO

Introduced by Microsoft in 2015, Microsoft Common Object in Context (COCO) [5] is a dataset containing instance segmentation of 80 common objects in their natural context. The term "common" refers to the objects that can be *"easily recognizable by a four-year-old"* [5]. COCO's labels also include captioning, and keypoints were added in 2016. Figure 7 shows examples where the objects are centered in the images. The COCO dataset is comprised of 2.5 million labeled instances in $382,000$ images. To create the large-scale dataset, COCO was labeled with extensive use of Amazon Mechanical Turk.

(a) **(b)**

Fig. 7. COCO Example Images. COCO contains instance segmentations similar to SUN. In these images, the objects are centered in the images.

III. Network Camera Data

Millions of network cameras are deployed worldwide [16] [17]. Data from network cameras are different from other image sources such as search engines or publically available repositories such as Flickr. The objects in these images are generally smaller than those in other datasets. The small size of objects in network data is because network cameras are usually mounted in high-locations on buildings. Network data is often real-time. This is critical in some applications. In Figure 8, images from the 2016 Houston Flood show rescue workers, emergency vehicles, cars, and trucks stuck in the water from the flood. One application of real-time data is the detection of areas affected by natural disasters. This section describes a project called the Continuous Analysis of Many Cameras (CAM2) [18], which acquires and processes real-time data from network cameras.

(a) **(b)**

(c) **(d)**

Fig. 8. Real-time geo-tagged data gives data context. This data is from the flood in Houston, Texas in 2016. A possible use of the CAM2 system is to alert local authorities when natural disasters occur to the location most effected by the event.

A. Camera Discovery Procedure

The complete explanation of network camera discovery for CAM2 is in Dailey et al. [19], but a summary is provided here. Network cameras can be defined as cameras whose images are accessible through the network. Some network cameras may be available only through restricted accesses, but many publically available cameras can be viewed by anyone. There are two classifications of network cameras: IP (Internet Protocol) cameras and non-IP cameras. IP cameras have individual IP addresses, generally host their own web servers, and are accessible directly over the Internet. Notably, they respond to Hypertext Transfer Protocol (HTTP) GET requests. Non-IP cameras are not assigned individual IP addresses and hence are not directly accessible over the Internet. The data is often aggregated into file servers and accessible through websites which often include data from more than one camera.

For Non-IP camera discovery, aggregation websites are scraped using Selenium or BeautifulSoup4. Due to the variety of interfaces to websites, each website requires a new script to be written to scrape the camera data. The camera data and location information is commonly made available in three different formats: JSON or XML files, loaded into a JavaScript Applet, or loaded in the HTML page. On aggregated websites, the location of the camera is sometimes exact with the given longitude and latitude or more general like a street address.

The process for IP camera discovery is more automated. This process is outlined in Figure 9 and relies on issuing HTTP requests and detecting the responses. IP cameras are often hosted by an organization. Using data from Internet Assigned Numbers Authority (IANA), all valid IP addresses for an organization can be generated. Once a camera is discovered, it is added to the network camera database. If the download is successful, the camera's location is estimated using the Google Geolocation API.

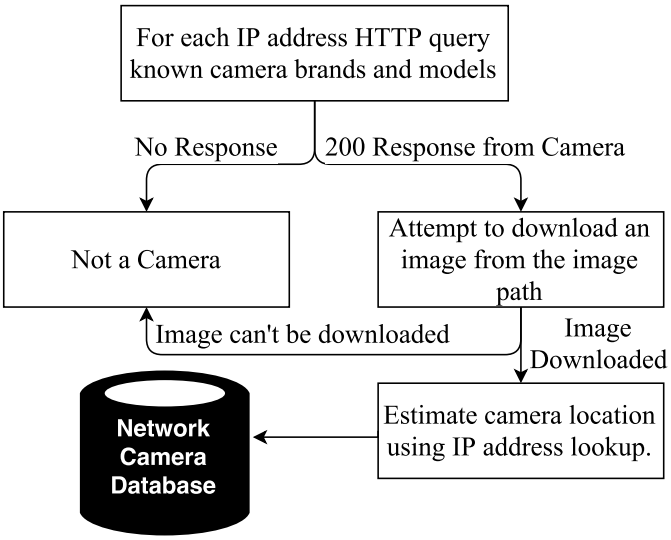


Fig. 9. IP Camera Discovery.

B. System Integration

The camera database is integrated into the CAM2 system, as seen in Figure 10. The CAM2 system provides users real-time data analysis tools which are run using the CAM2's Cloud Computing. CAM2 Cloud Computing is done using Amazon Web Services (AWS). Some of the current tools provided by CAM2 are edge detection, motion detection, and color quantization. Users can also upload custom modules. In Figure 10, the contents inside the blue square comprise the CAM2 system. A user interfaces the CAM2 system through the web portal and is authenticated using information stored in the user database. When the user chooses the cameras, the camera database provides the run-time system with the information to retrieve data from these cameras. The resource manager determines the most cost-efficient resource allocations for executing the analysis programs [20][21][22][23][24].

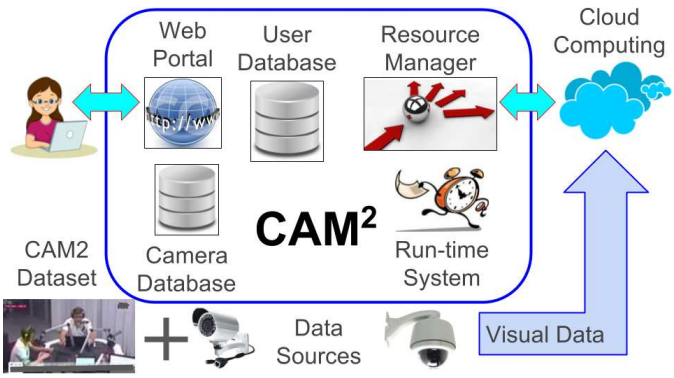


Fig. 10. CAM2 Architecture.

C. Creation of the CAM2 Sample Dataset

A small dataset has been created using the CAM2 system to compare network data to other datasets. This dataset consists of a modest 640 images with 3,322 bounding-box labels of people. Even though CAM2 has demonstrated the ability to retrieve and analyze 97 million unique images in 24 hours [22], the size of the dataset is initially small since object detection labeling is laborious to annotate. Aside from the bounding-box, each label also contains the date, time, and camera ID. The camera ID can be used in conjunction with the camera database to retrieve more meta-data about the image such as latitude, longitude, resolution, indoor/outdoor, and a frame-rate estimate. The data is taken from 111 different cameras with an effort to capture the diverse range of network camera quality.

D. Automatic Labeling

The CAM2 system provides a solution to unlabeled, continuous, live-feed network data. CAM2 can leverage the large repository of cameras to create a large dataset of automatically labeled images for image classification. Network cameras capture the same area under many different conditions such as daytime, nighttime, every season (shown in Figure 11), and holiday events. This can be cross-referenced with known events, like the weather, to create an automatic labeling platform. For example, a camera can be annotated with "hasTrees", "hasBuildings", and/or "hasStreet", each indicating that trees, buildings, or streets are visible in the camera view. While classification tasks require a single ground-truth label, using images with many labels gives the data more context. Furthermore, while large classification datasets exist, such as Places2 [25] with more than 10 million images and Tiny Images [26] containing 80 million image, the CAM2 system can retrieve more that 95 million images in a single day. Moreover, the data from network cameras can provide long-term observations. For example, Figure 11 shows a scene from a network camera over multiple years.

There are two known issues with an automatic labeling system for the CAM2 system. The first issue is that network cameras may scan an area, like a pan-tilt-zoom (PTZ) camera, or jump between different camera feeds. When the camera changes viewpoint, there may be categories marked as present for the camera which are not actually present in the current viewpoint. The second issue is that the redundancy in data pulled from the same camera (i.e., the background is the same) reduces the amount of new information contained in the data. However, the same camera's image can change dramatically over time, as can be seen from the Houston Flood images in Figure 8 and more subtly seen in the season changes in Figure 11. Further research is required to investigate the impact of these issues.

IV. Dataset Comparison

A. Real-time Geo-tagged Data

Network camera data offers both the geographic location and temporal relationships between frames. However, the CAM2 geographic location is the location of the IP address hosting the camera. Therefore, the accuracy of the identified location is challenging to evaluate. Some cameras' data contains indicators of the true camera location, such as a well-known landmark, while for other network cameras the ground-truth locations are more challenging to find. One method of determining the true location of cameras is to cross-reference the network camera data with current events. Figure 12 shows the locations of the network cameras which the CAM2 system can access.

The geographic location and the temporal relationship between images are features special to network data. These traits are desirable to give the data greater context. While one or the other is present in some datasets, the combination is unavailable in all of them.

B. Quantitative Measures

There are two quantitative measures to be compared between the datasets: **(a)** what is the distribution of the dataset labels and **(b)** what is the relative size of the dataset labels compared to the entire image? The distribution of the labels is analyzed through the people-density maps in Figure 13,

(a) 10/26/2011 **(b)** 12/31/2011 **(c)** 06/01/2012 **(d)** 10/10/2015 **(e)** 12/30/2016

Fig. 11. The data from Grand Teton (Wyoming's Yellowstone) changes over the years. CAM2 data can also be used to cross-reference the weather reports. Additionally, the variation of data from a single camera along with the weather and climate information can be a large resource of data for machine learning applications.

Fig. 12. The cameras of CAM2 are distributed across the world. The number of the map indicates the number of cameras in each location.

and the relative label size is analyzed through the plots in Figure 14.

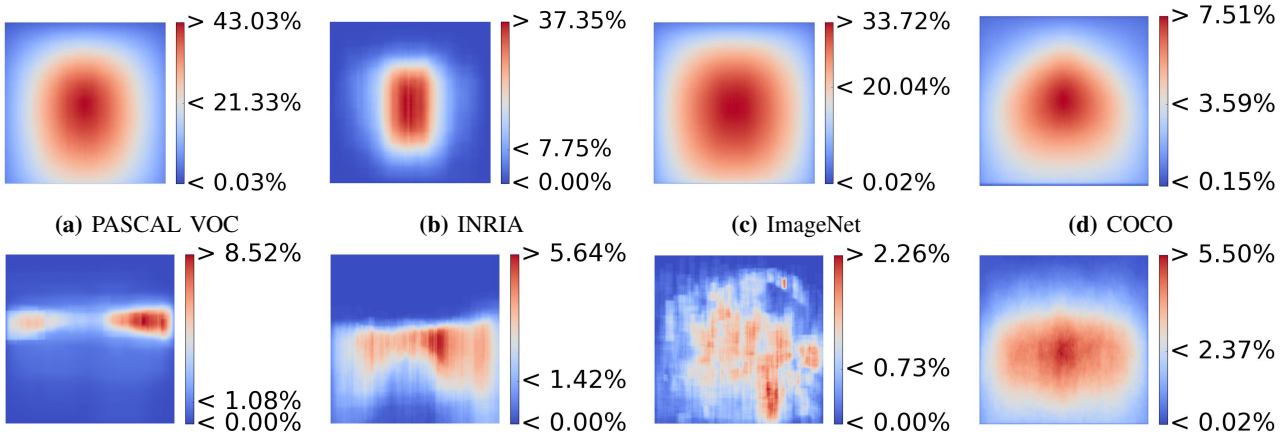
In this paper, people-density maps are defined as a square image that visualizes the distribution of a dataset's bounding-box or instance segmentation label locations. For each label, the polygon's location relative to the image dimensions is plotted onto the square grid. The color-mapping provides the distribution of the label locations so that label locations can be compared across datasets. The color-mapping range provides a reference to compare the intensities of different colors.

The process for creating a people-density map was completed by using the bounding-box or instance segmentation labels in each dataset. In order to standardize the results, each pixel coordinate (x, y) of an image size $w \times l$, for width and length, is represented as a percentage of the total image width and length: $\left(\frac{x}{w}, \frac{y}{l}\right)$. The percentage indicates the pixel's location on the fixed, square grid. When completed for each pixel in a label, this rescales the original label onto the square grid. The square grid begins with all zero values. A value of one is added to the area covered by the polygon. After all the labels are added, the square grid is divided by the total number of labels added. In Figure 13, each image uses a resolution of 500×500. The resolution determines the precision that the people-density map can capture. The precision determines the fidelity of the process to capture the label location. In this case, the figures provided use a precision of $\frac{1}{500}\%$ in both the x and y directions. Notably, the people-density map hardly changed from when the resolution was increased from 100×100 to 500×500. Therefore, a higher resolution was not computed.

Figure 13 shows the people-density map for each dataset. The minimum, mean, and maximum percents are marked on the vertical color bar from bottom to top. The density plots can be used to compare the concentration of labels across datasets. The coloring indicates the density of labels in that region - red indicates a high density of people labels and blue indicates a low density. However, the absolute coloring for each density plot should not be compared directly between the datasets. The range of the color bar, or vertical axis, must be considered as the maximum values of the color bars vary (the minimum is always 0%). For example, the maximum value of map **(a)** is 43.03%, while the maximum value of map **(e)** is only 8.52%. To compare the datasets' concentration intensities, one must consider that the deep red region's value in map **(e)** would appear as a blue-white color in map **(a)**. The variety of the color bar ranges is required so the distribution of locations in each plot can be seen.

The color-mapping also visualizes the label location across the datasets. In five datasets from Figures 13 **(a)**, **(b)**, **(c)**, **(d)**, and **(h)**, the labels are centered. The sharp gradient of the color in Figure 13 **(b)** indicates a high density of people focused in the center of the dataset, with a much lower, more even concentration of images outside of the center. In Figure 13 **(a)**, the gradient from red to blue is much smoother with many white pixels in between. This means that the labels in Figure 13 **(a)** are even more concentrated in the center of the image than in Figure 13 **(b)**, since there are fewer blue pixels and a much higher mean value: 21.33% versus 7.75%. A more evenly distributed density mask has a small color bar range, a smaller mean and the people-density map color is predominantly the color of the mean value.

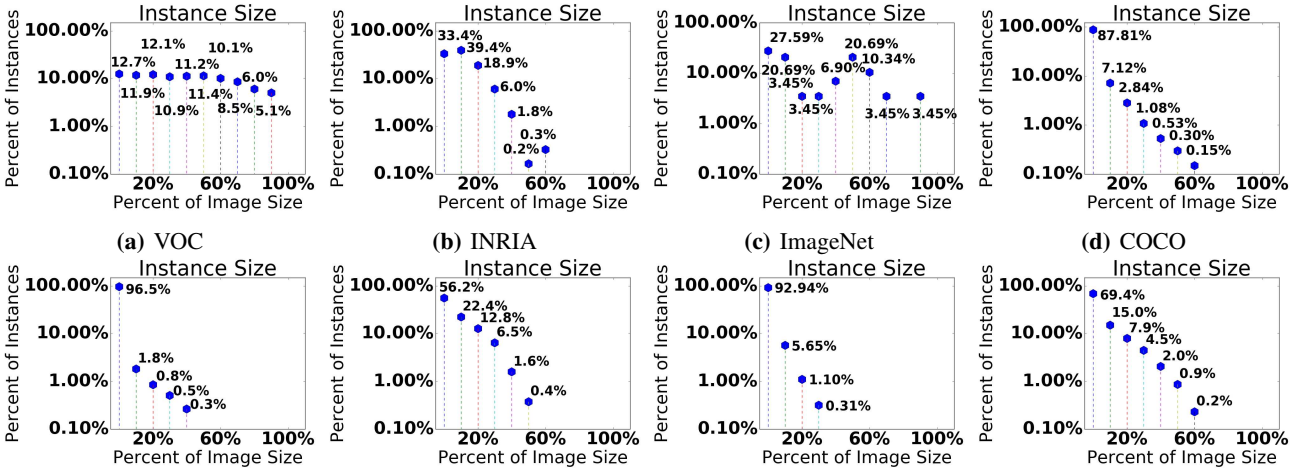
PASCAL VOC has the most centering effect of the objects in the image, with a range of $[0.03\%, 43.03\%]$. The larger range of PASCAL VOC means that more images are centered, and the smaller range of COCO implies the distribution is more even. The Caltech Pedestrian dataset, Figure 13 **(e)**, seems to have a concentrated number of detections in two locations on the sides of the image. This is reasonable since Caltech is taken from a dash-cam. It is likely that there are more people on either side of the car (on the sidewalks) than in front of the car (on the road). The KITTI dataset, Figure 13 **(f)**, contains very few detections across the top. This is reasonable



(a) PASCAL VOC **(b)** INRIA **(c)** ImageNet **(d)** COCO

(e) CALTECH **(f)** KITTI **(g)** CAM2 **(h)** SUN

Fig. 13. The people-density maps show the location and concentration of people bounding-box labels or instance segmentations in an image (better viewed in color). The images are each scaled from $[0.0\%, dataset\_max\%]$. The different ranges of the axes are required so that the characteristics of each distribution are visible.



(a) VOC **(b)** INRIA **(c)** ImageNet **(d)** COCO

(e) CALTECH **(f)** KITTI **(g)** CAM2 **(h)** SUN

Fig. 14. The ratio of people detections to the rest of the image. Note the log scale on the vertical axis. The domain is grouped into 10% sections. The first group is $[0\%, 10\%)$, the last group is $[90\%, 100\%)$, and there exists a final category for the complete 100% coverage.

since the data is collected from a camera mounted on the car. With the reason similar to Caltech, there are likely fewer people labeled far out in the middle of the road.

The network camera data in CAM2, Figure 13 **(g)**, seems to have scatter concentrations of detections across the people-density map. The network camera data also has the lowest mean pixel value of 0.73 and the smallest range. This indicates that network camera data has a more even distribution of the label locations than the other seven datasets.

The size of a label relative to the entire image size can help determine the difficulty of a label. If a dataset contains many large objects of interest, then the object detection task may be easier than if the dataset contains many small objects. In Figure 14, the plot represents the percent of labels within a range of label to image size ratios. The plots are created in two steps. First, the union of all binary mask labels is superimposed on a zero-valued image. The number of pixels contained in the binary mask is divided by the total number of pixels in the image. The percentage is assigned to one of the 11 ranges going from $[0\%, 10\%)$ by ten to the final bin of 100%.

As seen in Figure 14, the distribution of the relative object ratios follows a similar trend for each dataset except **(a)** and **(c)**. PASCAL VOC's and ImageNet's distributions in Figure 14 **(a)** and **(b)**, respectively, appear to be evenly distributed. The even distribution implies that most labels in the two datasets are large. Specifically, over 70% of labels accounts for 10% or more of the total image area. The Caltech dataset in Figure 14 **(e)** has the highest concentration of images in the first region, $[0, 10)$, with 96.5% of the dataset's labels. In Figure 14 **(g)**, the CAM2 network data follows with 92.94% of the labels in the first region. Overall, it appears that many datasets contain many small objects in their images.

V. Dataset Improvements

The datasets mentioned in this paper, especially large-scale datasets such as ILSVRC [4], COCO [5], and SUN [6], are major contributors to the recent, significant progress in computer vision. However, it is known that datasets such as these include issues [4][27][28][29] in terms of image selection bias and human labeling error.

There are two points worth mentioning about these potential dataset issues: **(1)** the image selection bias and **(2)** labeling quality. First, image selection bias appears in two ways: **(1a)** the resource of data and **(1b)** the selection of images within the resource. Exploring **(1a)**, PASCAL VOC [7], COCO [5], and ILSVRC [4], all collect images from Flickr, which introduces sampling bias. The samples used for current machine learning tasks are disproportionately sampled from a specific type of image, i.e., images that people take and upload to Flickr, instead of having a representational sample from the true distribution of possible images. Additional studies are needed to compare Flickr and network camera data.

Furthermore **(1b)**, datasets tend to select a specific type of image. In Khosla et al. [29], 300 randomly sampled images from PASCAL VOC's [7] and ILSVRC's [4] classification datasets were shown to be separable with 29% and 21% accuracy, respectively, against 12 other datasets using a histogram of gradients (HOG) detector followed by a linear support vector machine (SVM) [12]. In Tommasi et. al. [28], using the convolutional layers of AlexNet [1] followed by a 12-way linear SVM, the accuracy improved to about 50% for PASCAL [7] and maintained about 20% for ILSVRC [4]. Both of those accuracies are good. These two examples of separation serve as an attempt to quantify the difference between the two dataset image types. Further investigation is required to determine how significant these results are, but it provides a baseline understanding of a distinction between the datasets.

Another issue, **(2)**, is that due to the large number of images in both COCO [5] and ILSVRC [4], they utilize Amazon Mechanical Turk for labeling. This increases the chance for labeling error [30]. Some examples of a possible missing labels are shown in Figure 15 for COCO (left) and ILSVRC (right). Overall, it is difficult to measure the true number of missing labels in a dataset because marking the ground truth is laborious.

Network camera data may provide a partial solution to both problems. For **(1)**, the solution is obvious: network camera data is a completely new repository for datasets. For **(2)**, network cameras could be cross-referenced with events (such as weather) to automatically label the images for classification. While this does not yet solve the missing labels for object detection, perhaps the automatic classification of the images is only a first step.

VI. Conclusion

This paper describes and compares eight visual datasets and proposes a new method for creating a dataset using network cameras. This paper focuses on seven popular machine learning datasets for object detection, focusing exclusively on the "people" labels, and introduces a sample set of network camera data. The labels from each dataset are examined. First, we examine the distribution of label density for object detection datasets. We discover that many dataset labels are centered in the image. Labels for network camera data appear to be significantly less centered than other datasets. This paper also investigates the size of the objects (in number of pixels) compared to total image size. We find that while some datasets such as PASCAL VOC and ImageNet contain many objects which take up more than 10% of the total image size, other datasets contain mostly objects which are smaller. Finally, directions for future improvement on dataset creation are proposed and network camera data is offered as a possible solution.

Fig. 15. The COCO (left) and ImageNet (right) datasets contain missing people labels in the images. It seems that especially in crowds, more labels are missing.

Acknowledgments

The authors would like to thank Amazon for providing the cloud infrastructure, and the organizations that provide the camera data. A list of the data sources has been provided in the annex. This project is supported in part by National Science Foundation ACI-1535108, CNS-0958487, and OISE-1427808. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 91–99.

[3] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *CVPR*, 2017.

[4] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[5] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, Oral, Jan. 1, 2014.

[6] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010, pp. 3485–3492.

[7] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[8] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[9] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, 2009.

[10] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," *European Conference on Computer Vision*, 2016.

[11] P. Dollàr, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, 2012.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR - Volume 1 - Volume 01*, IEEE Computer Society, 2005, pp. 886–893.

[13] M. Everingham, A. Zisserman, C. K. I. Williams, *et al.*, "The 2005 PASCAL visual object classes challenge," in *First PASCAL Machine Learning Challenges Workshop*. 2006, pp. 117–176.

[14] K. Gauen, R. Rangan, A. Mohan, Y. H. Lu, W. Liu, and A. C. Berg, "Low-power image recognition challenge," in *Asia and South Pacific Design Automation Conference*, 2017, pp. 99–104.

[15] M. Taiana, J. C. Nascimento, and A. Bernardino, "An improved labelling for the inria person data set for pedestrian detection," in *Pattern Recognition and Image Analysis: Iberian Conference*. 2013, pp. 286–295.

[16] A. Doyle, R. K. Lippert, and D. Lyon, *Eyes everywhere : The global growth of camera surveillance*, English. Routledge London, 2012, xiv, 392 p.

[17] N. Jenkins, *245 million video surveillance cameras installed globally in 2014*, 2015.

[18] A. S. Kaseb, E. Berry, Y. Koh, *et al.*, "A system for large-scale analysis of distributed cameras," in *IEEE Global Conference on Signal and Information Processing*, 2014, pp. 340–344.

[19] R. Dailey, A. S. Kaseb, C. Brown, *et al.*, "Creating the world's largest real-time camera network," in *Imaging and Multimedia Analytics in a Web and Mobile World*, 2017.

[20] S. Nanda, T. J. Hacker, and Y.-H. Lu, "Predictive model for dynamically provisioning resources in multi-tier web applications," in *IEEE International Conference on Cloud Computing Technology and Science*, 2016.

[21] A. Mohan, A. Kaseb, Y.-H. Lu, and T. Hacker, "Location based cloud resource management for analyzing real-time video from globally distributed network cameras," in *IEEE International Conference on Cloud Computing Technology and Science*, 2016.

[22] A. S. Kaseb, A. Mohan, and Y.-H. Lu, "Cloud resource management for image and video analysis of big data from network cameras," in *International Conference on Cloud Computing and Big Data*, 2016.

[23] W. Chen, Y.-H. Lu, and T. Hacker, "Adaptive cloud resource allocation for analysing many video streams," in *IEEE International Conference on Cloud Computing Technology and Science*, 2015.

[24] A. S. Kaseb, E. Berry, E. Rozolis, *et al.*, "An interactive web-based system for large-scale analysis of distributed cameras," in *Imaging and Multimedia Analytics in a Web and Mobile World*, 2015.

[25] B. Zhou, A. Khosla, À. Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," *ArXiv preprint arXiv:1610.02055*, 2016.

[26] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.

[27] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011, pp. 1521–1528.

[28] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," in *Pattern Recognition: German Conference*. Springer International Publishing, 2015, pp. 504–516.

[29] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *European Conference on Computer Vision*, 2012.

[30] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *ACM SIGKDD Workshop on Human Computation*, 2010, pp. 64–67.

APPENDIX

TABLE I. Sample Image Sources

Image	**Source**	**Image ID**
Figure 1 (a)	Pascal VOC	003865
Figure 1 (b)	Pascal VOC	003856
Figure 2 (a)	ImageNet	1001
Figure 2 (b)	ImageNet	1008
Figure 3 (a)	SUN	a\airport\terminal \sun\acpxjhfbxfstfrtj
Figure 3 (b)	SUN	a\airfield \sun\bqrkjzaxxucgirds
Figure 4 (a)	INRIA	person_203
Figure 4 (b)	INRIA	crop001056
Figure 5 (a)	KITTI	00015 (left)
Figure 5 (b)	KITTI	00015 (right)
Figure 6 (a)	Caltech	set01\V001
Figure 6 (b)	Caltech	set03\V008
Figure 7 (a)	COCO	188592
Figure 7 (b)	COCO	197658
Figure 15 (a)	COCO	114907
Figure 15 (b)	ImageNet	1026
Figure 15 (c)	COCO	156071
Figure 15 (d)	ImageNet	1066
Figure 15 (e)	COCO	188465
Figure 15 (f)	ImageNet	1088