Performance Evaluation of Object Detection Algorithms

Vladimir Y. Mariano[1] Junghye Min[1] Jin-Hyeong Park[1] Rangachar Kasturi[1]
David Mihalcik[2] Huiping Li[2] David Doermann[2] Thomas Drayer[3]

[1]Pennsylvania State University, University Park, PA 16802 USA
[2]University of Maryland, College Park, MD 20742 USA
[3]Department of Defense, Fort Meade, MD 20755 USA
E-mail: {mariano,min,jhpark,kasturi}@cse.psu.edu,tdrayer@afterlife.ncsc.mil
{davidm,li,doermann}@cfar.umd.edu

Abstract

The continuous development of object detection algorithms is ushering in the need for evaluation tools to quantify algorithm performance. In this paper, a set of seven metrics are proposed for quantifying different aspects of a detection algorithm's performance. The strengths and weaknesses of these metrics are described. They are implemented in the Video Performance Evaluation Resource (ViPER) system and will be used to evaluate algorithms for detecting text, faces, moving people and vehicles. Results for running two previous text-detection algorithms on a common data set are presented.

1. Introduction

Many algorithms have been developed for detecting objects in images and video. These include methods for detecting text [5, 4], faces [6, 7], vehicles [7] and moving people [2]. With many new algorithms emerging, rules for quantitative measurement of performance are necessary.

The performance of a detection algorithm can be measured by comparing its output with the ground-truth, where a human operator has marked the boundaries of the target objects. Tools for ground-truthing video sequences have been developed in [3] and [8].

In this paper, we propose several metrics which can be used to determine how well the output of object detection algorithms matches the ground truth. Different metrics capture different aspects of performance. The objects considered here are compact objects – those that can be covered by simple bounding shapes. For other applications, pixel-based representation of the data may be more appropriate.

The use of simple bounding shapes allows inexpensive ground-truthing. This would enable a large volume and variety of video data to be ground-truthed using an intuitive interface such as ViPER [3]. In contrast, ground-truthing of indivudual pixels (e.g. text pixels) is expensive which makes it prohibitive for large and diverse data sets. Furthermore, the output of most detection algorithms are presented as simple bounding shapes (e.g. boxes) which are consistent with the ground-truth. Beyond bounding shapes, criteria for detection might include heuristics on specific object class features. For example, an algorithm's coverage of two eyes and a mouth (ground-truthed features in addition to the face bounding box) could be used as a criteria for succesful face detection [6]. This paper covers only the use of bounding shapes which is applicable to many classes of objects in video.

Using the proposed metrics, we can do the following:

- Algorithm parameters can be optimized for a particular set of metrics.
- The performance of an algorithm for different kinds of data can be compared.
- Quantitative comparison of different detection algorithms is possible.
- In the course of an algorithm's development, any performance improvement can be measured.
- Tradeoffs between performance aspects can be determined.

We believe no single metric can measure all the different aspects of performance. Furthermore, developers may be interested in optimizing on a small set of performance aspects, thus a single universal metric would not be suitable.

The paper is organized as follows: Section 2 briefly describes recent work on evaluation of detection algorithms. In Section 3, seven proposed performance metrics are described. Finally, Section 4 describes their implementation

1051-4651/02 $17.00 © 2002 IEEE

in the ViPER Evaluation tool and results on previous text-detection algorithms are presented.

2. Previous Work

ViPER [3] was developed as a tool for ground-truthing video sequences. Objects are marked by a bounding box and object detection algorithms are evaluated using temporal and spatial metrics. In [8], text objects are assigned a value for *detection importance* and *detection difficulty* which are factored in the computation of detection rate. In [1], five algorithms for text detection were compared. The ground-truth and the algorithm output are considered as binary pixel maps (text = 1, non-text = 0). Recall and precision are computed using the intersection of the binary images.

3. Performance Metrics

Our proposed metrics are described below along with their advantages and disadvantages. Table 1 illustrates different cases and the computed metric values. All the metrics' values range from zero to one (perfect). The last two metrics are expressed as ratios in Table 1 since they are based on counts of boxes.

The metrics are defined first for a single frame and then extended to an entire video sequence. We use the term "ground truth object" to denote the ground-truth bounding box marked around an image of the object. Detections are marked by an algorithm using "output boxes". The bounding shapes can also be extended to ellipses (suitable for faces) or arbitrary polygons.

The first two metrics (3.1 and 3.2) treats each pixel in the ground-truth as object/non-object and the output pixels as detected/non-detected. Next, we focus on each ground-truth object and compute how the covering output boxes are fragmented (3.3). Metrics 3.4 and 3.5 measure the recall and precision for each ground-truth object and output box respectively. Finally, metrics 3.6 and 3.7 impose a threshold ($OverlapMin$) on the measured recall and precision areas to declare whether a ground-truth object was detected and whether an output box significatly covers the ground-truth.

Let $G^{(t)}$ be the set of ground truth objects in a single frame t and let $D^{(t)}$ be the set of output boxes produced by the algorithm. $N_{G^{(t)}}$ and $N_{D^{(t)}}$ are their respective values in frame t.

3.1. Area-Based Recall for Frame

This metric is a pixel-count-based metric that measures how well the algorithm covers the pixel regions of the ground-truth. Initially it is computed for each frame, and it is the weighted average for the whole data set.

Let $U_{G^{(t)}}$ and $U_{D^{(t)}}$ be the spatial union of the boxes in $G^{(t)}$ and $D^{(t)}$:

$$U_{G^{(t)}} = \bigcup_{i=1}^{N_{G^{(t)}}} G_i^{(t)} \qquad U_{D^{(t)}} = \bigcup_{i=1}^{N_{D^{(t)}}} D_i^{(t)}$$

For a single frame t, we define $Rec(t)$ as the ratio of the detected areas in the ground truth with the total ground truth area:

$$Rec(t) = \begin{cases} \text{undefined} & \text{if } U_{G^{(t)}} = \emptyset \\ \frac{|U_{D^{(t)}} \cap U_{G^{(t)}}|}{|U_{G^{(t)}}|} & \text{otherwise} \end{cases}$$

$OverallRec$ is the weighted average recall of all the frames.

$$OverallRec = \begin{cases} \text{undefined} & \text{if } \sum_{t=1}^{N_f} |U_{G^{(t)}}| = 0 \\ \frac{\sum_{t=1}^{N_f} |U_{G^{(t)}}| \times Rec(t)}{\sum_{t=1}^{N_f} |U_{G^{(t)}}|} & \text{otherwise} \end{cases}$$

where N_f is the number of frames in the ground-truth data set and the $||$ operator denotes the number of pixels in the area.

This metric treats the frame *not* as collection of objects but as a binary pixel map (object/non-object; output-covered/not-output-covered). The metric provides a fairly good recall measure for comparing different algorithms but the ground-truth is no longer treated as a collection of individual objects. Furthermore, this metric is biased towards large ground-truth objects.

3.2. Area-Based Precision for Frame

This metric is a pixel-count-based metric that measures how well the algorithm minimized false alarms. Initially it is computed for each frame, and it is the weighted average for the whole data set.

Let $U_{G^{(t)}}$ and $U_{D^{(t)}}$ be the spatial union of the boxes in $G^{(t)}$ and $D^{(t)}$:

$$U_{G^{(t)}} = \bigcup_{i=1}^{N_{G^{(t)}}} G_i^{(t)} \qquad U_{D^{(t)}} = \bigcup_{i=1}^{N_{D^{(t)}}} D_i^{(t)}$$

For a single frame t, we define $Prec(t)$ as the ratio of the detected areas in the ground truth with the total detection:

$$Prec(t) = \begin{cases} \text{undefined} & \text{if } U_{D^{(t)}} = \emptyset \\ 1 - \frac{|U_{D^{(t)}} \cap U_{G^{(t)}}|}{|U_{D^{(t)}}|} & \text{otherwise} \end{cases}$$

$OverallPrec$ is the weighted average precision of all the frames.

$$OverallPrec = \begin{cases} \text{undefined} & \text{if } \sum_{t=1}^{N_f} |U_{D^{(t)}}| = 0 \\ \frac{\sum_{t=1}^{N_f} |U_{D^{(t)}}| \times Prec(t)}{\sum_{t=1}^{N_f} |U_{D^{(t)}}|} & \text{otherwise} \end{cases}$$

where N_f is the number of frames in the ground-truth data set and the || operator denotes the number of pixels in the area.

This metric treats the frame *not* as collection of objects but as a binary pixel map (object/non-object; output-covered/not-output-covered). A similar precision metric was used in [1] for comparing five text-detection algorithms.

3.3. Average Fragmentation

Detection of objects is usually not the final step in a vision system. For example, extracted text from video will go through enhancement, binarization and finally recognition by an OCR system. Ideally, the extracted text should be in one piece, but a detection algorithm could produce several boxes (e.g. one for each word or character) or multiple overlapping boxes which could increase the difficulty for the next processing step.

This metric is intended to penalize an algorithm for multiple output boxes covering a ground-truth object. Multiple detections include overlapping and non-overlapping boxes.

For a ground-truth object $G_i^{(t)}$ in frame t, the fragmentation of the output boxes overlapping the object $G_i^{(t)}$ is measured by:

$$Frag(G_i^{(t)}) = \begin{cases} \text{undefined} & \text{if } N_{D^{(t)} \cap G_i^{(t)}} = 0 \\ \frac{1}{1+log_{10}(N_{D^{(t)} \cap G_i^{(t)}})} & otherwise \end{cases}$$

where $N_{D^{(t)} \cap G_i^{(t)}}$ is the number of output boxes in $D^{(t)}$ that overlap with the ground-truth object $G_i^{(t)}$. Figure 1 shows the function $Frag(N) = \frac{1}{1+log_{10}N}$.

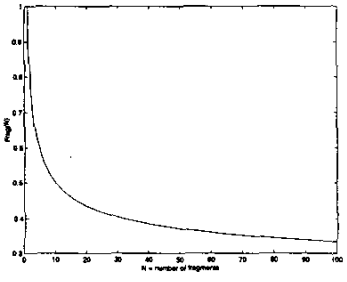

Figure 1. Fragmentation metric.

For a single frame t, $Frag(t)$ is simply the average fragmentation of all ground truth objects in frame t where $Frag(G_i^{(t)})$ is defined.

$OverallFragmentation$ is defined as average fragmentation for all ground-truth objects in the entire data set.

This metric measures how well the ground-truth object is *not* broken into pieces. Every output box that overlaps the ground-truth object is counted as a fragment for that object, even if the overlap is a single pixel. This disadvantage is evident for a large output box that "accidentally" overlapped a ground-truth object by a few pixels.

3.4. Average Object Area Recall

This metric is intended to measure the average area recall of all the ground-truth objects in the data set. The recall for an object is the proportion of its area that is covered by the algorithm's output boxes. The objects are treated equally regardless of size.

For a single frame t, we define $Recall(t)$ as the average recall for all the objects in the ground truth $G^{(t)}$:

$$Recall(t) = \frac{\sum_{\forall G_i^{(t)}} ObjectRecall(G_i^{(t)})}{N_{G^{(t)}}}$$

$$\text{where} \quad ObjectRecall(G_i^{(t)}) = \frac{|G_i^{(t)} \cap U_{D^{(t)}}|}{|G_i^{(t)}|}$$

and the || operator denotes the number of pixels in the area. Finally, $OverallRecall$ is the weighted average recall of all the frames.

$$OverallRecall = \frac{\sum_{t=1}^{N_f} N_{G^{(t)}} \times Recall(t)}{\sum_{t=1}^{N_f} N_{G^{(t)}}}$$

The metric gives credit to any portion of a ground-truth object that is covered by the algorithm output no matter how small that portion is. This can be thought of as the "degree" of detection for that object. This is in contrast with metric 3.6 where the detection criteria is a hard threshold applied to the proportion of the covered area. All the ground-truth objects contribute equally to the metric, regardless of their size. On one extreme, if a frame t contains two objects – a large object that was completely detected and a very small object that was missed, then $Recall(t)$ would be 50%.

3.5. Average Detected Box Area Precision

This metrics is a counterpart of the previous metric 3.4 where the output boxes are examined instead of the ground-truth objects. Precision is computed for each output box and averaged for the whole frame. The precision of a box is the proportion of its area that covers the ground truth objects.

For a single frame t, we define $Precision(t)$ as the average precision of the algorithm's output boxes $D^{(t)}$:

$$Precision(t) = \frac{\sum_{\forall D_i^{(t)}} BoxPrecision(D_i^{(t)})}{N_{D^{(t)}}}$$

where $BoxPrecision(D_i^{(t)}) = \frac{|D_i^{(t)} \cap U_{G^{(t)}}|}{|D_i^{(t)}|}$

and the || operator denotes the number of pixels in the area. $OverallPrecision$ is the weighted average precision of all the frames.

$$OverallPrecision = \frac{\sum_{t=1}^{N_f} N_{D^{(t)}} \times Precision(t)}{\sum_{t=1}^{N_f} N_{D^{(t)}}}$$

In this metric the output boxes are treated equally regardless of size. That is, the metric is not easily skewed by large (or small) output boxes.

3.6. Localized Object Count Recall

In this metric, a ground-truth object is considered detected if a minimum proportion of its area is covered by the output boxes. Recall is computed as the ratio of the number of detected objects with the total number of ground-truth objects.

Define $Loc\_Obj\_Recall(t)$ to be the number of detected objects in frame t:

$$Loc\_Obj\_Recall(t) = \sum_{\forall G_i^{(t)}} ObjDetect(G_i^{(t)}) \quad \text{where}$$

$$ObjDetect(G_i^{(t)}) = \begin{cases} 1 & \text{if } \frac{|G_i^{(t)} \cap U_{D^{(t)}}|}{|G_i^{(t)}|} > OverlapMin \\ 0 & \text{otherwise} \end{cases}$$

$OverlapMin$ is the minimum proportion of the ground-truth object's area that should be overlapped by the output boxes in order to say that it is correctly detected by the algorithm.

$Overall\_Loc\_Obj\_Recall$ is the ratio of detected objects to the total number of objects in the ground-truth:

$$Overall\_Loc\_Obj\_Recall = \frac{\sum_{f=1}^{N_f} Loc\_Obj\_Recall(t)}{\sum_{f=1}^{N_f} N_{G^{(t)}}}$$

Unlike 3.4, this metric makes a hard decision for each ground-truth object: it is either detected or missed. Intuitively, this tells whether the algorithm detected the object (or it didn't). The threshold on the overlapped area determines the detection criteria for each object.

Again, the ground-truth objects are treated equally regardless of size.

3.7. Localized Output Box Count Precision

This is a counterpart of metric 3.6. The metric counts the number of output boxes that significantly covered the ground truth. An output box $D_i^{(t)}$ significantly covers the ground-truth if a minimum proportion of its area overlaps with $U_{G^{(t)}}$.

Define $Loc\_Box\_Count(t)$ to be the number of output boxes that significantly overlap with the ground-truth objects in frame t:

$$Loc\_Box\_Count(t) = \sum_{\forall D_i^{(t)}} BoxPrec(D_i^{(t)}) \quad \text{where}$$

$$BoxPrec(D_i^{(t)}) = \begin{cases} 1 & \text{if } \frac{|D_i^{(t)} \cap U_{G^{(t)}}|}{|D_i^{(t)}|} > OverlapMin \\ 0 & \text{otherwise} \end{cases}$$

$OverlapMin$ is the minimum proportion of the output box's area that should be overlapped by the ground truth in order to say that the output box is precise.

$Overall\_Output\_Box\_Prec$ is the ratio of precise output boxes to the total number of output boxes produced by the algorithm:

$$Overall\_Output\_Box\_Prec = \frac{\sum_{f=1}^{N_f} Loc\_Box\_Count(t)}{\sum_{f=1}^{N_f} N_{D^{(t)}}}$$

Again, in this metric the output boxes are treated equally regardless of size. This precision metric examines each output box and judges whether or not it significantly covered the ground truth. The metric tells whether the algorithm does a good job of creating accurate output boxes. The disavantage is, for a given output box, the covered ground-truth regions is not necessarily a single ground-truth object (but a union of pixel regions). Thus the metric does consider the correct associations between output boxes and ground-truth objects.

4. Implementation and Future Work

The metrics were implemented in ViPER [3] and will be used in the development and evaluation of algorithms for detecting text, faces, moving people and vehicles. In the development stages, the metrics are used to optimize algorithm parameters for a training data set. Evaluation on a test data set is then performed using the same set of metrics. Periodic evaluation is done to monitor the progress of algorithm development. The detection algorithms currently considered for evaluation look for object instances in single images. The evaluation techniques need to be extended to algorithms that detect objects that persist in time and space (i.e. objects in video).

We used the metrics to evaluate and compare two text-detection algorithms, one from Penn State University (PSU) [5] and the other from the University of Maryland (UMD) [4]. These were run on 1291 video key frames ground-truthed at UMD using the ViPER Ground-Truthing Tool. The following overall results were obtained:

METRIC	PSU	UMD
Area-Based Recall for Frame	0.59	0.56
Area-Based Precision for Frame	0.29	0.32
Average Fragmentation	0.97	0.98
Average Object Area Recall	0.58	0.57
Average Detected Box Area Precision	0.27	0.31
Localized Object Count Recall	0.58	0.57
Localized Output Box Count Precision	0.22	0.21

The ViPER software can be downloaded from the website http://lamp.cfar.umd.edu/Media/Projects/ViPER. This work was supported by the Advanced Research and Development Activity (ARDA) under contract number MDA904-98-C-B294 and MDA0949-6C-1250.

References

[1] Sameer Antani, David Crandall, Anand Narasimhamurthy, Vladimir Y. Mariano, and Rangachar Kasturi. Evaluation of methods for detection and localization of text in video. In *Proc. International Workshop on Document Analysis Systems*, 2000.

[2] Larry Davis, V. Philomin, and R. Duraiswami. Tracking humans from a moving platform. In *Proc. International Conference on Pattern Recognition*, 2000.

[3] David Doermann and David Mihalcik. Tools and techniques for video performance evaluation. In *Proc. International Conference on Pattern Recognition*, volume 4, pages 167–170, 2000.

[4] Huiping Li, David Doermann, and Omid Kia. Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing*, 9(1):147–156, January 2000.

[5] Vladimir Y. Mariano and Rangachar Kasturi. Locating uniform-colored text in video frames. In *Proc. International Conference on Pattern Recognition*, volume 4, pages 539–542, 2000.

[6] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[7] Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 746–751, 2000.

[8] Xian-Sheng Hua, Liu Wenyin, and Hong-Jiang Zhang. Automatic performance evaluation for video text detection. In *Proc. International Conference on Document Analysis and Recognition*, 2001.

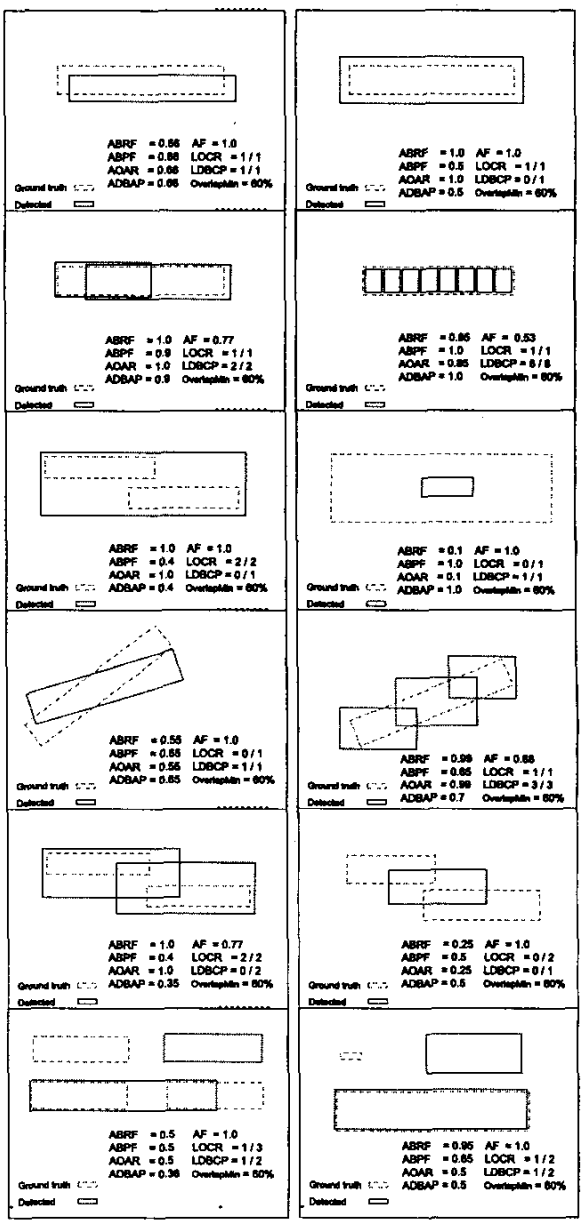

Table 1. Metrics computed on different cases. Metric names are abbreviated: ABRF (Sec 3.1), ABPF (Sec 3.2), AOAR (Sec 3.4), ADBAP (Sec 3.5), AF (Sec 3.3), LOCR (Sec 3.6), LOBCP (Sec 3.7)