DualCam: A Novel Benchmark Dataset for Fine-grained Real-time Traffic Light Detection

Harindu Jayarathne, Tharindu Samarakoon, Hasara Koralege, Asitha Divisekara, Ranga Rodrigo and Peshala Jayasekara

Abstract—Traffic light detection is essential for self-driving cars to navigate safely in urban areas. Publicly available traffic light datasets are inadequate for the development of algorithms for detecting distant traffic lights that provide important navigation information. We introduce a novel benchmark traffic light dataset captured using a synchronized pair of narrow-angle and wide-angle cameras covering urban and semi-urban roads. We provide 1032 images for training and 813 synchronized image pairs for testing. Additionally, we provide synchronized test video pairs for qualitative analysis. The dataset includes images of resolution 1920 × 1080 covering 10 different classes. Furthermore, we propose a post-processing algorithm for combining outputs from the two cameras. Results show that our technique can strike a balance between speed and accuracy, compared to the conventional approach of using a single camera frame. The dataset and algorithm are available at https://github.com/harinduravin/DualCam

Index Terms—traffic light dataset, deep learning, multi-camera perception, object detection, planar homography

I. INTRODUCTION

For the rapid development of Advanced Driver Assistance Systems (ADAS) and Autonomous Driving Systems (ADS), traffic light and sign detection play a crucial role. Especially, traffic light detection and recognition are complex tasks due to the smaller object sizes, illumination variations, close resemblance with other objects and the dynamic nature at operating time. Detection in real-time can be even more challenging due to computational resource limitations.

Training and evaluation of detection algorithms on traffic light datasets is an important task. Earlier, traffic light detection algorithms mainly depended on traditional image processing techniques [1], [2], [3], [4]. Recent traffic light detectors [5], [6], [7], [8], [9] are based on state-of-the-art object detectors such as Faster R-CNN [10], YOLO [11] and SSD [12]. Several traffic light datasets are publicly available [1], [13], [14], [15], but some of them are limited by image resolution, number of classes, number of annotations, and quality. Most of the existing datasets tackle traffic lights in the short-range distance only, limiting the capability of identifying traffic lights at high speed.

To tackle this issue, we present DualCam, a novel traffic light benchmark dataset consisting of 2250 annotated images and 8321 object instances. The dataset covers 10 different traffic light classes. Our benchmark dataset consists of images generated from a pair of synchronized narrow-angle and wide-angle cameras. Such synchronization can assist traffic light detection from an extended range. This additional information will be valuable for the perception system of an ADAS/ADS to plan smooth navigation.

In order to combine detection results from the two camera frames, we propose a post processing algorithm that combines bounding boxes. The algorithm accurately combines all the detection results into the wide angle camera frame as shown in Figure 1. Additionally, it suppresses double detection caused by the same object being detected simultaneously in both the frames. We test the algorithm on DualCam dataset by employing the YOLOv5 [16] object detector. Class-wise accuracy values and detection speeds are provided for comparison. Additionally, we summarize the effectiveness of combining results from two camera-frame outputs, compared to detection using individual frames.

The use of multiple cameras for perception systems of self-driving cars is well-known. Stereo cameras are widely used to acquire depth information of driving scene. A traffic light dataset that is created using synchronized wide and narrow angle cameras will be a novel addition to the existing traffic light datasets.

Our contributions in this paper are as follows:

- We provide a novel traffic light benchmark dataset covering 10 object classes, captured using a pair of cam-

![Figure 1: A synchronized image pair from our dataset. (a) Narrow-angle camera frame. (b) Wide-angle camera frame. (c) Annotations from both frames integrated to wide-angle camera frame (Common ground truth).](image-url)
eras, one narrow-angled and the other wide-angled. A separate test set consisting of synchronized images and videos is provided for the purpose of testing algorithms with dual camera input.

- We propose a post-processing algorithm for real-time traffic light detection using a synchronized narrow-angle and wide-angle camera pair. We discuss challenges that might arise in such a system.
- We provide evaluation results of our benchmark dataset in terms of speed and accuracy using an object detection model. Results reveal that our approach strikes a balance between speed and accuracy.

This paper is organized as follows: First, we discuss related work in section II. We introduce our camera setup and the dataset in section III. A post processing algorithm for our dual camera setup is introduced in section IV. Section V includes the implementation details of our algorithm. The results of implementation are included in the Section VI.

II. RELATED WORK

In this section, we discuss several publicly available traffic light datasets and traffic light detection algorithms.

A. Datasets

After learning based detection methods became popular, the need for traffic light datasets increased. LARA dataset [1] provides frames of resolution 640 × 480 along with four different traffic light classes. The low resolution makes the dataset unsuitable for accurate detection. LISA dataset [15], which has been captured using stereo vision cameras, provides 14386 annotated daytime images with 7 traffic light classes. Even though the number of frames is high, the diversity of frames available is low. The frames have been captured at close time intervals, resulting in nearly identical frames.

More recently introduced Bosch Small Traffic Light Dataset (BSTLD) [13] provides 13334 frames of resolution 1280 × 720 with detailed annotations up to one pixel width bounding boxes. The dataset provides 13 traffic light classes for training and 4 traffic light classes for testing. The class frequency is skewed towards most common traffic light states of red, yellow, green and empty. The DriveU Traffic Light Dataset [14] is a large dataset that provides 230,000 annotations with wide range of traffic light classes. Due to their proposed class attribute structure, traffic lights can be divided into 344 unique classes. It provides an overview of existing datasets along with evaluation metrics for comparison purposes.

State-of-the-art datasets such as COCO [17] and Cityscapes [18] datasets include traffic lights as an annotation class. Even though traffic light detectors can be trained using these datasets, they cannot be trained for the classification task.

All of the above datasets provide images from a single wide angle camera. This method often leads to omission of detailed information of traffic lights situated at a slight distance from the vehicle.

B. Traffic light detection algorithms

Initially traffic light detection systems were mainly based on image processing and machine learning techniques. Under image processing techniques, transformation to different color spaces and subsequent thresholding [1], [2] were used for the identification of potential regions of traffic lights. Further filtering using shape filters [4] and template matching [1], [2] were also common among the detection algorithms. Machine learning based algorithms such as support vector machine (SVM) [3], tree-based models [15] use features such as histogram of gradients (HOG) for classification. Some methods such as [3] need prior knowledge about the locations of traffic lights. Although these methods are computationally less expensive, detection is not accurate in complex scenarios.

Recent deep learning object detection approaches outperform all classical methods. Most of the methods adopt and customize state-of-the-art object detection algorithms. The most initial work that uses deep learning approaches for traffic light detection is DeepTLR [5]. It introduces a convolutional neural network (CNN) that creates a pixel-wise probability map followed by a bounding box regressor. Traffic light detection algorithms introduced in [6], [19] consist of multi-camera systems for accurate detection from far distance. They have demonstrated the performance improvement by utilizing multiple-camera combinations. However, the algorithms presented are not compatible with modern object detectors.

A YOLO [11] based traffic light detection algorithm is introduced in [13]. A separate custom classifier is used for the purpose of identifying traffic light status. Additionally, a stereo vision based object tracker is introduced. Traffic light detection algorithms in [8], [9] are based on customized versions of Faster-RCNN [10] and an adapted version of SSD [12] is used in [7]. Even though these algorithms are robust towards detecting smaller traffic lights, the speed of the algorithms are not suitable for resource-constrained real-time applications.

III. DUALCAM TRAFFIC LIGHT DATASET

This section covers the details about the DualCam traffic light dataset including data collection details and dataset statistics.

A. Data collection and annotation

1) Camera details: Our camera system consists of two vision-grade Basler daA1920-30uc USB cameras with 1920×1080 resolution: a narrow-angle camera with 48° horizontal field of view (FoV) and a wide-angle camera with 125° horizontal FoV. They are vertically aligned in a single enclosure mounted behind the windshield of the vehicle (Figure 2). The baseline between two cameras is 42 mm. The narrow-angle camera is mainly used to detect the traffic lights located far away from the vehicle. The wide-angle camera is used to detect the traffic lights nearby the vehicle. The cameras provide synchronized image pairs.
The data collection was carried out in semi-urban and urban areas, where the density of traffic lights is high. Footage from both cameras are recorded at a frequency of 30 FPS while driving at regular speed. Image frames are extracted from the footage and manually annotated using CVAT [20] annotation tool. We present our dataset in both PASCAL VOC XML [21] and YOLO [11] annotation format.

B. Dataset statistics

DualCam dataset is divided into training and test sets. The training set contains 1032 images in total; out of them, 776 images are from the narrow-angle camera and the rest of the 256 images are from the wide-angle camera. The test set contains 1626 images. It consists of 813 image pairs captured simultaneously using the two cameras. The test set is larger compared to conventional datasets, since we provide additional synchronized image pairs for better evaluation. Additionally, 40 minutes of synchronized test video pairs from the two cameras are provided. The dataset contains 10 object classes related to traffic light detection. The frequency of each class is provided in the Table I.

| Traffic light class | Train set | Test set | Total |
|---------------------|-----------|----------|-------|
| Green (a)           | 1708      | 1251     | 2959  |
| Red (b)             | 565       | 901      | 1466  |
| Green-up (f)        | 426       | 495      | 921   |
| Empty-count-down (j)| 537       | 225      | 762   |
| Count-down (i)      | 346       | 396      | 742   |
| Yellow (c)          | 432       | 246      | 678   |
| Empty (e)           | 222       | 469      | 691   |
| Green-right (g)     | 115       | 171      | 286   |
| Green-left (h)      | 55        | 105      | 160   |
| Red-yellow (b)      | 66        | 80       | 146   |
| **Total**           | **3982**  | **4339** | **8321** |

IV. POST PROCESSING ALGORITHM

At a given moment the two cameras can produce two image frames simultaneously. These two frames are fed to the object detector as a single batch of size two. The bounding boxes of detected traffic lights in two frames should be concatenated to obtain useful results. As wide-angle camera frame has wider field of view, we transform detected bounding boxes in narrow-angle camera frame (Figure 1a) to match with the wide-angle camera frame. Then we concatenate all the detected bounding boxes together within wide-angle camera frame (Figure 1c). The bounding box transformation and concatenation are the major steps in this approach.

A. Bounding box transformation

Bounding box transformation is done in three steps as shown in Figure 4. First, the bounding boxes detected in original (distorted) narrow-angle camera frame are transformed into its undistorted camera frame using camera distortion parameters. Then these boxes are transformed into undistorted wide-angle camera frame using planar homography between the two camera frames. Finally, they are transformed on to original (distorted) wide-angle camera frame using camera distortion parameters. Details on homography matrix estimation can be found in section V-B. Compared to transforming full images, transforming the bounding box coordinates requires less computation power.

B. Bounding box concatenation while duplicate suppression

The bounding boxes of detected traffic lights in narrow-angle camera frame are transformed and concatenated with the bounding boxes of detected traffic lights in the wide-angle camera frame. Some traffic lights detected in the wide-angle camera frame might be detected fully or partially in the narrow-angle camera frame creating duplicate bounding boxes. We remove these duplicate bounding boxes as described below.

Let the sets of bounding boxes generated by narrow-angle camera and wide-angle camera be denoted as $\mathcal{N}$ and $\mathcal{W}$. Let the bounding boxes of $\mathcal{N}$ transformed using the transformation technique mentioned in the Figure 4 be denoted as $\mathcal{N}_0$. All $w \in \mathcal{W}$ that reside completely within the region where the narrow angle camera frame is mapped (called $R_0$) are removed from the set $\mathcal{W}$. The resulting set is $\mathcal{W}_0$.

To handle the bounding boxes generated from the wide-angle camera that reside partially inside $R_0$, let $q \in \mathcal{Q}$ denote the shapes generated by taking the intersection between $w$,
and $R_0$ for all $w_r \in W_r$. An associated bounding box in $N_0$ can exist for all $q \in Q$. The association is measured using the Intersection over Union (IoU) similar to the association metric used in SORT algorithm [22]. If the IoU calculated between each $q \in Q$ and $n_0 \in N_0$ exceeds a certain threshold $\zeta$, the corresponding bounding box is removed from $N_0$ to create the new set $N_r$. Finally, bounding boxes from $N_r \cup W_r$ are taken onto the wide-angle camera frame as the output. The steps from camera frame grabbing till result visualization is provided in algorithm 1.

**Algorithm 1 Overall algorithm**

**Require**: Synchronized narrow-angle and wide-angle camera frames, distortion parameters of the two cameras.

1. **Initialize**:
   2. Estimation of the planar homography $wH_n$ between the two frames after undistortion.
   3. Estimation of the region $R_0$ using the homography matrix and distortion parameters.

4. $N_r = N_0$
5. **repeat**
   6. Generate $N$ and $W$ using the object detector.
   7. Obtain $Q$ and $W_r$ from region $R_0$ and $W_r$.
   8. for $q \in Q$ do
   9. Calculate IoU values between $q$ and all $n_0 \in N_0$
10. if $3 n_0 \in N_0$: IoU $\geq \zeta$ then
11. $N_r = N_r \setminus \{n_0\}$
12. end if
13. end for
14. Output $N_r \cup W_r$
15. **until** End of camera frame extraction

**V. IMPLEMENTATION**

The end-to-end traffic light detection system is implemented as a Robot Operating System (ROS) [23] package. Synchronized image pair acquisition is carried out using a ROS node programmed using Pylon 6 C++ API [24]. For the training and testing of our proposed algorithms, we use a device comprising an Intel Core i9-9900K CPU and a Nvidia RTX-2080 Ti GPU. Additionally, for the purpose of testing the algorithm in a more resource constrained environment, we use Nvidia Jetson AGX Xavier platform.

**A. Object detector training and inferencing**

YOLOv5 [16] object detector is provided as 5 different models with varying sizes. We train and evaluate lightweight YOLOv5s [16] model and comparatively large YOLOv5l [16] model to examine how speed-accuracy trade-off affect our algorithm. YOLOv5s [16] is trained for 300 epochs using input size $448 \times 448$, batch size 64, learning rate 0.01 and SGD optimizer. YOLOv5l [16] is trained for 300 epochs using input size $640 \times 640$, batch size 16, learning rate 0.01 and SGD optimizer. After training we employ half-precision floating point (FP16) TensorRT optimization for batch size two. In algorithm 1, 0.5 is used for the value of $\zeta$.

**B. Homography matrix estimation**

Planar homography is a mapping between any two images of a planar surface. In order to calculate the homography between two camera frames, following closed form formula can be used by assuming pinhole camera model [25].

$$wH_n = k_w (wR_n - wt_n n^T / d) k_n^{-1}$$  \hspace{1cm} (1)

$wH_n$ is the homography that maps points in narrow-angle camera frame to wide-angle camera frame. $k_w$ and $k_n$ are the intrinsic camera matrices. $wR_n$, $wt_n$ are the rotation matrix and translation vector between two cameras. $n$, $d$ are the plane normal vector and distance to the chosen plane expressed in narrow-angle camera coordinates.

Due to practical implementation issues, above method does not give an accurate homography matrix. Least square error based estimation using multiple point correspondences is more suitable. Therefore, a set of corresponding points of interest in two frames is obtained using Scale Invariant Feature Transform (SIFT) [26]. These points belong to a chosen planar calibration surface which is parallel to the image plane.

Even though we use planar homography to match traffic lights in two frames, it can only be used to match objects on a single plane. Therefore, the traffic lights outside the plane that homography is calculated will not be properly matched. Still, this method gives a lower error rate as the area which maps the narrow-angle camera frame within the wide-angle camera frame $R_0$ can be approximated as a planar area. This is possible, because $R_0$ is quite small and far away with respect to the rest of the area in the wide-angle camera frame.
VI. RESULTS

A. Evaluation metrics

For the performance evaluation of object detectors on our dataset, we use $F_1$-score for each class. Recall is the proportion of correct predictions out of all ground truths. Precision is the proportion of correct predictions out of all predictions. The precision-recall curve is calculated from predictions ranked according to confidence score.

\[
Recall = \frac{TP}{TP + FN} \tag{2}
\]

\[
Precision = \frac{TP}{TP + FP} \tag{3}
\]

\[
F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}
\]

TP indicates the total number of detected traffic lights (true positives), FN indicates the total number of undetected traffic lights (false negatives) and FP indicates the total number of predictions that cannot be attributed to any ground truth (false positives). A prediction is considered as a true positive based on the IoU value across the ground truth and predicted bounding boxes.

B. Dataset evaluation

For the purpose of evaluating the performance of YOLOv5 [16] on our dataset, inferencing is carried out using batch size one over the test set. We evaluate YOLOv5s [16] and YOLOv5l [16] with input sizes of 448×448 and 640×640, respectively.

**TABLE II: Class-wise $F_1$-scores for evaluation on test set**

| Class            | YOLOv5s [16](FP16) | YOLOv5l [16](FP16) |
|------------------|---------------------|---------------------|
| Red              | 72.03               | 83.56               |
| Green-arrows     | 65.03               | 74.5                |
| Yellow           | 59.65               | 72.36               |
| Count-down       | 59.39               | 68.62               |
| Empty-count-down | 52.85               | 65.05               |
| Green            | 49.55               | 62.04               |
| Red-yellow       | 36.36               | 55.56               |
| Empty            | 38.25               | 48.59               |

Table II compares the class-wise $F_1$-scores obtained for the two models using 0.3 as the IoU threshold value. The ground-truth of the test set includes bounding boxes of size reaching a minimum of 6 pixels. YOLOv5l [16] outperforms YOLOv5s [16] in terms of accuracy. An important observation is that some classes seem to perform better compared to others. Count-down, Empty-count-down classes have lower $F_1$-scores due to their smaller size compared to traffic lights. Classes such as empty, red-yellow, green-left, green-right and green-up have low number of instances, resulting in lower $F_1$-scores. For testing, we combine Green-left, Green-right and Green-up classes to a super class named Green-arrows.

Figure 5 shows the class-wise precision-recall curves for the two models. In traffic light detection, even though precision can be improved to higher levels, recall has limited improvement capacity [13], [5], [6]. Clearly, using a larger model such as YOLOv5l [16] can improve predictions by shifting precision-recall curves to the right, compromising on detection speed.

**Fig. 5: Precision-recall curves of individual classes for YOLOv5s model (left) and YOLOv5l model (right)**

C. Algorithm evaluation

Our proposed algorithm 1 includes batch size 2 inferencing followed by a post processing algorithm. The speed of the combined frame approach is measured by considering the time spent on processing a pair of frames. This is compared with frame rate in individual frame approach. Table III shows the speed comparison for Nvidia RTX-2080 Ti GPU and Jetson AGX Xavier platform. Combined frame approach achieves more than half the speed achieved by the single frame approach. Even though Jetson AGX Xavier has constrained resources, it performs in real-time when YOLOv5s [16] is used. However, it only achieves a speed of 16 Hz for the case of YOLOv5l [16]. The post processing algorithm takes up additional 1ms computation time in RTX-2080 Ti GPU whereas it takes up 5ms in Jetson AGX Xavier with minimum impact on overall speed as shown in the Figure 6.

Reduction in small amount of speed leads to a significant improvement in accuracy by using our technique as shown in Figure 7. The precision-recall curves are generated using a common ground-truth prepared using synchronized images in the test set. This ground-truth contains bounding boxes of size reaching a minimum of 1 pixel. The resulting combined bounding boxes from the algorithm 1 achieves higher recall values compared to individual contributions from narrow-angle and wide-angle camera frames. The recall values reach higher values when the IoU threshold is 0.3 compared to the IoU threshold of 0.5. The contribution of narrow-angle

**TABLE III: Speed comparison of combined frame approach (pairs per second) with individual frame approach (frames per second)**

| Model        | RTX-2080 Ti                  | Jetson AGX Xavier                  |
|--------------|-------------------------------|-----------------------------------|
|              | Individual frames | Combined frames | Individual frames | Combined frames |
| YOLOv5s      | 416.7 Hz                   | 256.4 Hz                      | 57.3 Hz              | 42.8 Hz          |
| YOLOv5l      | 196.1 Hz                   | 117.6 Hz                      | 29.5 Hz              | 16.5 Hz          |
camera is higher compared to the wide-angle camera. This is due to the prevalence of higher number of instances in the narrow-angle camera frames.

Fig. 7: Overall precision-recall curves with respect to common ground truth. Detection using combination of both frames out-performs the detection using individual frames.

VII. CONCLUSION

This paper introduced DualCam, a novel traffic light benchmark dataset, addressing the need for synchronized images captured using a pair of cameras. The proposed post-processing algorithm can efficiently combine the detection output from the two cameras with minimum impact on the overall detection algorithm. Results show that our approach results in a considerable increase in recall, with a large contribution from the narrow-angle camera, compared to the conventionally used wide-angle cameras. Incorporating multiple cameras and assigning traffic lights to respective lanes are possible future extensions of the work presented in this paper.

REFERENCES

[1] R. de Charette and F. Rashashibi, “Real-time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates,” in IEEE Intelligent Vehicles Symposium, 2009, pp. 358–363.

[2] C. Wang, T. Jin, M. Yang, and B. Wang, “Robust and real-time traffic lights recognition in complex urban environments,” International Journal of Computational Intelligence Systems, pp. 1383–1390, 2011.

[3] N. Fairfield and C. Urmon, “Traffic light mapping and detection,” in 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 5421–5426.

[4] M. Diaz-Cabrera, P. Cerri, and P. Medici, “Robust real-time traffic light detection and distance estimation using a single camera,” Expert Systems with Applications, pp. 3911–3923, 2015.

[5] M. Weber, P. Wolf, and J. M. Zöllner, “DeepUtl: A deep convolutional network for detection and classification of traffic lights,” in 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 342–348.

[6] M. Bach, S. Reuter, and K. Dietmayer, “Multi-camera traffic light recognition using a classifying labeled multi-bernoulli filter,” in 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 1045–1051.

[7] J. Müller and K. Dietmayer, “Detecting traffic lights by single shot detection,” in International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 266–273.

[8] M. Bach, D. Stumper, and K. Dietmayer, “Deep convolutional traffic light recognition for automated driving,” in International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 851–858.

[9] A. Gupta and A. Choudhary, “A framework for traffic light detection and recognition using deep learning and grassmann manifolds,” in IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 600–605.

[10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” Advances in Neural Information Processing Systems, vol. 28, 2015.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, A. C. Berg, and A. Zisserman, “SSD: Single shot multibox detector,” in European Conference on Computer Vision. Springer, 2016, pp. 21–37.

[13] K. Behrendt, L. Novak, and R. Botros, “A deep learning approach to traffic lights: Detection, tracking, and classification,” in IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 1370–1377.

[14] A. Fregin, J. Muller, U. Krekel, and K. Dietmayer, “The driveu traffic light dataset: Introduction and comparison with existing datasets,” in IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 3376–3383.

[15] M. P. Philipsen, M. B. Jensen, A. Mogelmose, T. B. Moeslund, and M. M. Trivedi, “Traffic light detection: A learning algorithm and evaluations on challenging dataset,” in IEEE 18th International Conference on Intelligent Transportation Systems, 2015, pp. 2341–2345.

[16] G. Joscher. YOLOv5. [Online]. Available: https://www.github.com/ultralytics/yolov5

[17] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in European Conference on Computer Vision. Springer, 2014, pp. 740–755.

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.

[19] J. Müller, A. Fregin, and K. Dietmayer, “Multi-camera system for traffic light detection: About camera setup and mapping of detections,” in IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 165–172.

[20] Computer vision annotation tool. [Online]. Available: https://icvat.org

[21] M. Everingham, S. Eslami, L. Van Gool, C. K. 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.

[22] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464–3468.

[23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al., “ROS: an open-source robot operating system,” in ICRA Workshop on Open Source Software, vol. 3, no. 3.2, 2009, p. 5.

[24] Basler pylon camera software suite. [Online]. Available: https://www.baslerweb.com/en/products/software/basler-pylon-camera-software-suite/

[25] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, 2nd ed. Cambridge university press, 2003, ch. Scene planes and homographies.

[26] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.