MOHE-NET: MONOCULAR OBJECT HEIGHT ESTIMATION NETWORK USING DEEP LEARNING AND SCENE GEOMETRY

Jianli Wei[1], Jinwei Jiang[2], Alper Yilmaz[1]

[1]Photogrammetric Computer Vision Lab., The Ohio State University, Columbus, OH, USA - (wei.909, alper.15)@osu.edu
[2]Ford Motor Company - jjiang30@ford.com

Commission II, WG 7

KEY WORDS: Object Detection, Height Estimation, Moving Camera, Convolutional Neural Networks, linear MLP.

ABSTRACT:

Estimating the heights of objects in the field of view has applications in many tasks such as robotics, autonomous platforms and video surveillance. Object height is a concrete and indispensable characteristic people or machine could learn and capture. Many actions such as vehicle avoiding obstacles will be taken based on it. Traditionally, object height can be estimated using laser ranging, radar or stereo camera. Depending on the application, cost of these techniques may inhibit their use, especially in autonomous platforms. Use of available sensors with lower cost would make the adoption of such techniques at higher rates. Our approach to height estimation requires only a single 2D image. To solve this problem we introduce the Monocular Object Height Estimation Network (MOHE-Net) that includes a cascade of two networks. The first network performs the object detection task. This network detects the bounding box of objects of interest. This information is then input to a second network to estimate the object height and is a linear Multi-layer Perceptron (MLP). The linear MLP model models the camera-scene geometry and does not require training or contain activation function as normal MLP did. The developed approach works for static camera set up as well as moving platform. The proposed approach performs state-of-the-art and can be deployed for obstacle avoidance on autonomous platforms. Our code is available at `https://github.com/OSUPCVLab/Ford2019/tree/master/Moving%20Object%20Height%20Estimation%20Network`

1. INTRODUCTION

Object height has applications in a number of problem domains including but not limited to autonomous driving, robotics and visual surveillance. Once the object height information estimated, this information can be used, for instance, to avoid obstacles in autonomous driving scenarios to ensure the safety. Deploying a height estimation system requires two an object detection module and a height estimation module , both of which are required to perform in real-time processing for time constraint problems.

Arguably object detection can be considered a mid-level perception problem required by many higher level tasks (Zou et al., 2019). It has been an activate area of research for several decades. The goal of object detection is to determine whether instances of an object, such as person, car, truck, exists in the image and return its location as an enclosing mask or bounding box (Liu et al., 2020). Recently, deep learning techniques, including but not limited to faster r-cnn, yolo series (Ren et al., 2016, Redmon et al., 2016, Redmon and Farhadi, 2017, Redmon and Farhadi, 2018), have been shown to work comparatively far more accurate and faster than other traditional approaches based on such as SIFT, SURF and BRIEF (Lindeberg, 2012, Calonder et al., 2010, Bay et al., 2006)). Our proposed object height estimation system adopts an existing pretrained deep convolutional neural networks (Ren et al., 2016, Jocher et al., 2020) to detect object instances with bounding boxes recorded in monocular cameras.

We refer to the height estimation problem as the metric estimation of the object height from the 2D bounding boxes. In particular, this step uses, backprojections of the pixel coordinates to 3D camera coordinates using view geometry modeled as MLP. Considering 2D to 3D relation is projective and the object scale is unknown, we introduce additional geometric constraints to solve the problem. The first of these is the assumption that the camera looks at piece-wise planar scene, such that normalized Direct Linear Transformation (DLT) (Hartley and Zisserman, 2003) applies as shown in Fig. 1. Second, we assume the objects and image plane are standing upright and vertical to the ground which generates a special geometry that will be discussed later in text.

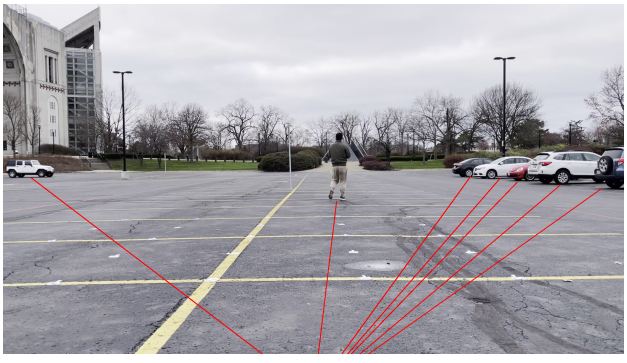

Figure 1. An example image acquired from a monocular camera mounted on a vehicle showing the road from the vehicle's perspective. Red lines indicate projective physical distance of embedded camera and objects. Each red line has two ending points. One is from camera outside the image. The other is located from the object pixel coordinate. DLT utilized relationships between points in projective coordinates and geometry coordinates.

Our main contributions to height estimation can be summarized as follows:

- MOHE-Net requires only a monocular camera.
- It can estimate object height from both stationary and moving platform.
- The geometry is represented as an MLP to generate the network cascade.
- It generates accurate results on the collected dataset. More specifically, it achieves 5.08 cm mean error and 26.4 fps speed.

The rest of this paper is organized as follows. Section 2 reviews recent related work on object detection as well as height estimation. Section 3 describes the problem and provides details of proposed MOHE-Net. Section 4 introduces our collected data and experimental implementation. Section 5 provides details on the results.

2. RELATED WORK

Object height has been considered an important piece of information for autonomous systems and can be directly solved using range systems estimation such as LIDAR or stereo camera. Its importance stems from the fact that avoiding high or low lying obstacles will reduce defects to the vehicle while ensuring the safety of the passengers and the objects around the vehicle. Below we will discuss the the two modules required for an end to end system: object detection and height estimation.

Considering that the amount of work published on object detection is vast, we will only consider more recent studies that uses deep learning. With the introduction of regions to the CNN architectures (R-CNN) (Girshick et al., 2014) object detection methods have started to produce results significantly better than traditional approaches. These developments can be divided into two categories. The first category is a two-stage approach which starts from a region proposal followed by classification and bounding box regression. The approaches in the first category include R-CNN and its improved version, fast R-CNN (Girshick, 2015) and faster R-CNN (Ren et al., 2016). Fast R-CNN performs feature extraction as a whole, avoiding independent feature extraction of each proposed region. Faster R-CNN replaces selective search with a region proposal network to generate proposed regions. The second category is a one-stage approach which performs the classification and localization steps simultaneously via grid regression. Arguably the most representative model for this category is the You only look once (YOLO) variations (Redmon et al., 2016, Redmon and Farhadi, 2017, Redmon and Farhadi, 2018, Bochkovskiy et al., 2020, Jocher et al., 2020). Those models, typically, take Pascal VOC (Everingham et al., 2010) and MS COCO (Lin et al., 2014) for training and evaluation purposes. We adopted the second category of approaches, and observed that using existing their pre-trained networks provided good accuracy and speed for the object detection task in our approach.

Height estimation task estimates the height information by transforming the image pixel coordinates to 3D coordinates (Hartley and Zisserman, 2003). When an image of a scene is captured, the depth information has lost. The estimation of the 3D object characteristics, one has to backproject the image into the 3D space. Godard *et al.* (Godard et al., 2017) proposed an unsupervised monocular depth estimation approach to predict depth using a single camera. Zhou *et al.* (Zhou et al., 2017) proposed an approach to recover depth information from 2D motion information providing disparity. These methods, while can be used for height estimation, cannot be used due to high computational cost that recovers depth information for all the pixels. The planarity condition of the scene also makes these approaches impractical for height estimation. Our approach in contrast uses the planarity condition directly (Abdel-Aziz et al., 2015), to estimate heights of upright objects.

Many recent approaches (Mousavian et al., 2017, Wu et al., 2019, Ke et al., 2020, Kundu et al., 2018) were proposed to estimate vehicle size (length, width and height) and 6-DoF. But without exception, those approaches require estimation of camera rotation and translation even though they use a monocular likewise. What distinguish our approach from the others is that we doesn't require any estimation of camera pose but can estimate over 80 object classes height accurately achieving real-time.

3. METHODOLOGY

For metric height estimation, the proposed MOHE-Net cascades two neural networks as shown in Fig. 2. The first neural network (OD-Net) detects object-instances among a set of objects of interest. The second network (HE-Net) estimates metric height, and is a designed multilayer perceptron (MLP) which contains the geometry weights. HE-Net estimates the heights of all object instances detected by OD-Net. The final output of MOHE-Net contains metric locations and estimated heights of object instances.

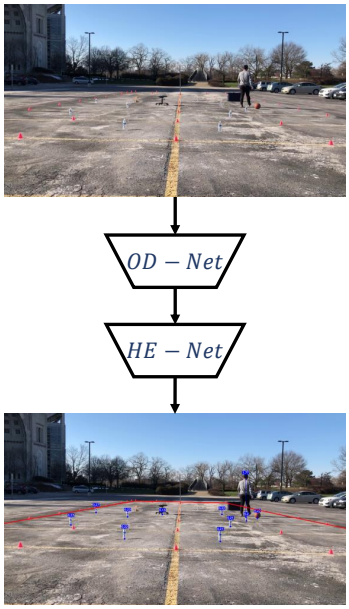

Figure 2. Object detection and height estimation pipeline.

3.1 Problem formulation

Let 4x1 tuple A = ($\mathbf{I}$, S_0, L, c_0) defines imaged scene for the i^{th} frame, where $\mathbf{I} \in \mathbb{R}^{H*W*3}$ is the image frame with width W and height H, S_0 represents the ROI, $L_0 = (l_1, l_2, ..., l_n)$ is the classes of objects of interest (COI), and c_0 is the confidence threshold of object detector. Objects not a member of COI are ignored by MOHE-Net as well as objects with confidences lower than c_0.

3.2 Object Detection Network

Object detection network is a pretrained CNN model with objects within the COI: $O_i = f_{OD-Net}(I_i)$. This network maps the input image, I_i, to output O_i, where O_i is a 6xn tuple, $O_i = (b_{i,j,1}, b_{i,j,2}, b_{i,j,3}, b_{i,j,4}, l_{i,j}, c_{i,j})$, where $b_{i,j,1} \sim b_{i,j,4}$ are the upper left and lower right bounding box coordinates of the detected objects, shown in Fig. 3. In this equation, the first subscript *i* denotes the frame index, the second subscript *j* indicates j^{th} object-instance, the last subscript represents one of the four sides of j^{th} instance, $l_{i,j}$ is OD-Net predicted label of j^{th} instance and $c_{i,j}$ represents its corresponding confidence. Height estimation network only activates when $l_{i,j} \in L$ and $c_{i,j} \geq c_0$. The implementation details of f_{OD-Net} is given in Section 4.

Figure 3. Object detection output generated by the detection network in the form of a bounding box: $b_{i,j,1}, b_{i,j,2}, b_{i,j,3}, b_{i,j,4}$. Marked two points (middle bottom and middle above) of the bounding box represent object bottom and top, such that the object height is the the length of the line connecting these two points.

To represent object height in the image domain, the algorithm selects two points, middle bottom point and middle above point of the rectangular bounding box, referred to as the bottom point and top point as shown in Fig. 3.

$$P_{i,j,bottom} = (\frac{b_{i,j,1} + b_{i,j,3}}{2}, b_{i,j,2}) \quad (1)$$

$$P_{i,j,top} = (\frac{b_{i,j,1} + b_{i,j,3}}{2}, b_{i,j,4}) \quad (2)$$

3.3 Height Estimation Network

To estimate the height of an object instance, we designed a task oriented multi-layer perceptron (MLP) referred to as the HE-Net that inversely project the bottom and top points from image coordinates to the real world coordinates, $H_i = f_{HE-Net}(O_i)$, where O_i is OD-Net output, f_{HE-Net} mapped input O_i into 3xm ($m \leq n$) tuple H_i, $H_i = (h_{i,j}, l_{i,j}, c_{i,j})$, satisfying $l_{i,j} \in L$ and $c_{i,j} \geq c_0$, and subscripts *i,j* are same as in Section 3.2.

The object bottom and top points are coplanar and lie on a plane perpendicular to the horizontal road plane as shown in Fig. 4. This coplanarity condition is denoted by the orange line and is vertical to the road plane (black canvas). The two dotted lines originating from the camera center (black point) represent the backprojection from the image plane to object space. Distance between two parallel planes is referred to as parameter *z* on camera *z* direction shown in Fig. 5, also called object depth estimation. In our data collection, a single camera is mounted on the vehicle for collecting dataset. We defined spatial *xyz* axis to be right, down and forward direction with respect to vehicle moving forward.

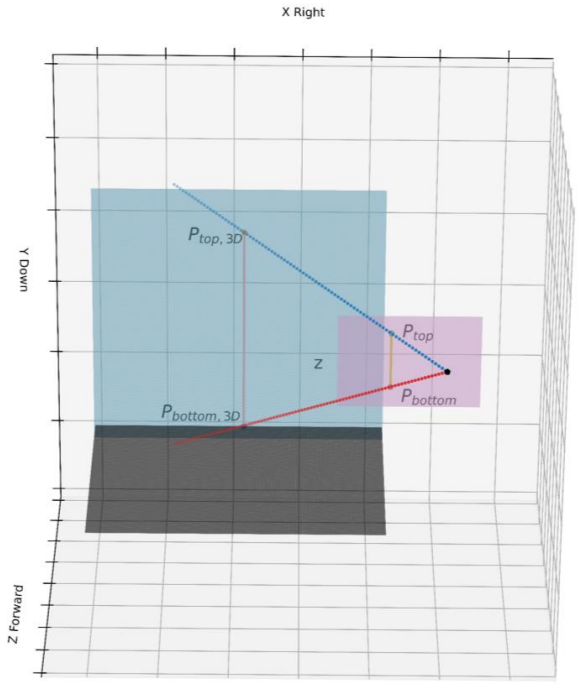

Figure 4. A diagram demonstrates HE-Net inverse projection process. Lilac and blue canvases respectively represent image plane and object plane vertical to the horizontal road plane (black canvas). Two dotted inverse projection lines originating from camera pinhole (black point) project object through image plane back to object plane at a distance of *z*.

Let $(u, v, 1)$ be the homogeneous coordinates of x in image coordinates and $X = (X_W, Y_W, Z_W, 1)$ be the corresponding homogeneous coordinate in the real world. Given the camera intrinsic matrix *K* and the pose of camera in world coordinate frame in i^{th} video frame $(R, T) \in SE(3)$, the geometric relationship between x and X is:

$$x = K[R, T]X \quad (3)$$

In fact, object height is the same in both camera frame and world frame. The proposed MOHE-Net estimates object height in camera coordinate to avoid estimating the motion of camera, which is (R, T) of equation (3). Given $(X_C, Y_C, Z_C, 1)$ the camera frame homogeneous coordinate, the projection from 3D

Figure 5. Dataset collection platform with predefined camera coordinates with respect to vehicle moving forward direction, x, y and z axis point at right, down and forward directions.

point in camera coordinate frame to image plane is:

$$z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{pmatrix} \tag{4}$$

and the inverse projection from image plane to camera coordinate frame is:

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{pmatrix} = z \underbrace{\begin{pmatrix} 1/f_x & 0 & -c_x/f_x \\ 0 & 1/f_y & -c_y/f_y \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}}_{K_{inv}} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{5}$$

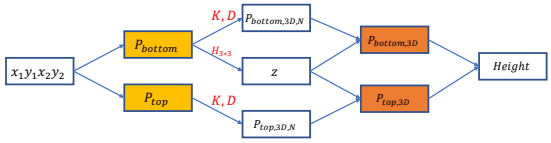

Figure 6. HE-Net flowchart. $x_1y_1x_2y_2$ is bounding box location in image. P_{bottom} and P_{top} represent object bottom and top pixel location. Respectively, $P_{bottom,3D,N}$ and $P_{top,3D,N}$ are depth-normalized 3D points in camera coordinate frame. z is object estimated depth. $P_{bottom,3D}$ and $P_{top,3D}$ are estimated inverse-projected points in camera frame.

where K_{inv} is the inverse camera intrinsic matrix. We took a short video with a 6×9 chessboard, to calibrate camera intrinsic matrix *K* and distortion coefficients *D* using chessboard calibration algorithm developed by OpenCV library (Bradski, 2000). Apart from K_{inv}, object depth, parameter z denoted as the distance between parallel image planes and object plane in Fig. 4, is also needed for image point inverse projection. In our proposed HE-Net, we implement DLT for estimating object depth, parameter z. This will be discussed later in Section 3.4.

Overall HE-Net is a handcrafted linear MLP, whose flowchart is shown in Fig. 6. The MLP has *K,D* as its pre-trained weights, inversely projects object bottom point and top point on image plane back to 3D depth-normalized camera coordinate frame, denoted as $P_{bottom,3D,N}$ and $P_{bottom,3D,N}$. The subscript *N* indicates point depth-normalized. With estimated object depth, parameter z in equation (5), those two points are denormalized back to 3D camera frame. Object height is the distance in the vertical direction from the bottom point ($P_{bottom,3D}$) to top point ($P_{top,3D}$), also shown in Fig. 4.

3.4 Object Depth Estimation

In our above mentioned HE-Net, we applied DLT to estimate parameter z. In Fig. 7, there are 36 cone markers on the ground within ROI or on its margins. Those markers on the ground plane have 2 degree of freedom, x and z. When collecting dataset, we firstly measure markers coordinates with respect to the mounted camera as the origin projecting on the road plane and defined as $P_C = (p_{C,1}, p_{C,2}, ..., p_{C,36})$, where $p_{C,i} = (x_i, z_i, 1)$. Then, we manually picked up those makers on image plane and recorded their homogeneous coordinates as $P_I = (p_{I,1}, p_{I,2}, ..., p_{I,36})$, where $p_{I,i} = (u_i, v_i, 1)$.

Normalization is basically a preconditioning to decrease condition number of the matrix P_C and P_I. Assuming T_C and T_I are normalization matrix to normalize P_C and P_I respectively to $\hat{P}_C$ and $\hat{P}_I$ in camera coordinate frame and image coordinate frame with mean 0 and standard deviation $\sqrt{2}$, we estimate homography matrix $h_{3\times3}$ as:

$$\hat{P}_C = h_{3\times3} * \hat{P}_I \tag{6}$$

Matrix $h_{3\times3}$ is estimated based on normalized points, $\hat{P}_C$ and $\hat{P}_I$. Taking matrix denormalization, $H_{3\times3}$ will be:

$$P_C = \underbrace{{T_C}^{-1} * h_{3\times3} * T_I}_{H_{3\times3}} * P_I \tag{7}$$

Given object bottom point $P_{i,j,bottom}$ = (u,v,1) in Section 3.2, we are able to estimate parameter z as:

$$\begin{pmatrix} x \\ z \\ 1 \end{pmatrix} = H_{3\times3} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{8}$$

where z in equation (8) would be applied into inverse projection equation (5) in Section 3.3 as object depth in camera frame.

4. EXPERIMENT DESIGN

To evaluate the performance of the MOHE-Net, we conducted threee case studies. In case I, objects and the platform are both stationary. The relative distance between objects and platform remains the same. We measure ground truth height for all objects within ROI, which range from 20 cm inches to 180 cm. In case II, we keep the platform motionless and a person 183cm high is walking within ROI. The person walks from the left to the right, from the close to the distant. In case III, platform as vehicle is moving forward so that more object instances are on camera.

Figure 7. Camera view from monocular camera mounted on the vehicle. Red polyline is the margin of ROI. HE-Net will be activated only objects whose bottom points are within ROI. Cones on the ground are markers for homography estimation.

For collecting dataset, we mounted a monocular camera on top of a moving vehicle. The z-direction of the camera in Fig. 5 aligned with vehicle moving forward direction. Its front view is shown in Fig 7 without any occlusion. The MOHE-Net design requires calibration of the vehicle mounted camera to estimate homography matrix. In order to calibrate, we uniformly placed 36 red cones at measured locations and use the red polylines to denote the ROI margins. The homography estimation is then achieved using Random Sample Consensus (RANSAC) (Fischler and Bolles, 1981) that minimizes overall geometric error. The coordinates of these points are also marked in images. Vehicle

mounted camera recorded image sequence with 20 frame per second (fps) and the resolution of images are 1280×720.

The proposed MOHE-NET pipeline consists of object detection and height estimation. For object detection, we adopted Faster r-cnn (Ren et al., 2016) and YOLOv5 (Jocher et al., 2020). Fig. 3 shows as example object detection via YOLOv5. Both detectors could detect and classify up to 80 object classes achieving real-time performance. In the experiment, we set the confidence threshold to $c_0 = 0.25$. The second component of the pipeline, HE-Net, uses the estimated camera intrinsic matrix *K*, distortion coefficients *D* and $H_{3\times3}$ matrix to assign the MLP weights. We additionally introduce the ROI map shown in Fig. 7 in the HE-Net to reduce computation time and increase inference speed.

5. RESULTS

We discuss two different aspects of results generated by MOHE-Net, the accuracy and speed. In either case the output of MOHE-Net is predicted objects heights within ROI shown in Fig. 8. We note that all experiments are conducted on NVIDIA Titan V.

In the first case study dataset, objects (such as bottles, chair, sports ball) are statically placed on the ground (see Fig. 8). Object height predictions and ground truths are summarized in Table 1. As can be observed the errors for different objects range from 0 to within 6 centimeters. We observed that, the MOHE-Net with YOLOv5+OST as its object detection backbone has accurately estimated object heights no matter they are tall or short.

Category	Predicted Height[1]	Ground Truth[1]	Error[1]
Bottles[2]	22.4	20.3	+2.1
Chair	44	40	+4.0
Suitcase	53	48	+5.0
Sports Ball	25	25	0
Person	177	183	-6.0

[1] In centimeter(s)
[2] Averaging all bottles' estimated heights.

Table 1. Object height predictions and ground truths

In case study II, the height estimation is performed sequentially. Fig. 9 shows predictions registered on the ground truth for one of the backbones used in the study. The statistical analysis for all other backbones is tabulated in Table 2. We observe that as the gait of the person is moving up and down as the person walks our approach provided a range of height estimation with a 5.09 cm mean error. The gait change is manifested in the plot as a sinusoidal variation as shown in Fig. 10. Among the sequential predictions, there are several errors which we point out with arrows in Fig. 10. The main reason for these errors is that our approach relies on monocular camera, such that appearance changes affect height estimation. Fig. 11 respectively shows MOHE-Net estimated person's heights. Low contrast between person's appearance and background and occlusion negatively result into those abnormal predictions.

Category	Mean[1]	Mean(%)	Std[1]	Error Min/Max[1]
Person[2]	5.09	2.78	5.87	-19.0/14.0

[1] Centimeters
[2] The person in the dataset is 183 cm tall.

Table 2. Statistic Analysis

In case study III, vehicle mounted camera moves with the platform. Region of interest simultaneously changes as platform moving forward. Many objects comes in and out ROI, as shown in Fig. 13. In each row, there are several vehicles within ROI are estimated height. For instance, totally four vehicles within red polygons are estimated height. From our perspective, in the last row, the silver wagon looks close to its right one but higher than the other two. Predicted heights displayed on in the blue box in meters match our judgement.

The computational bandwidth for autonomous vehicles is consumed by many tasks the vehicles is performing every second. Hence, the speed of object height estimation is a key factor in algorithm evaluation. Aside from the quantitative comparisons, we also compare the speed of the entire architecture when the object detector is changed in the MOHE-Net pipeline. The results for Faster r-cnn, YOLOv5 and its variants are shown in Table 3. The table shows total parameter count, height estimation error and speed of the pipeline respectively. The results in the table are also ploted in In Fig. 12. The architecture with lower error and faster speed is observed for MOHE-Net with YOLOv5x+OST as object detection backbone and has real-time performance.

Model	Mean Error (cm)	Speed (ms)	FPS	Params
Faster r-cnn	7.52	83.45	11.98	-
YOLOv5s[1]	8.82	14.93	66.97	7.3M
YOLOv5m[1]	7.82	18.68	53.52	21.4M
YOLOv5l[1]	7.70	21.88	45.71	47.0M
YOLOv5x[1]	7.06	25.29	39.54	87.7M
YOLOv5x+TTA[2]	5.87	68.36	14.63	87.7M
YOLOv5x+OST[3]	5.08	37.87	26.40	87.7M

[1] Normal test take image size 640 pixels.
[2] Test Time Augmentation (TTA) increases the image size by about 30%. It typically takes about 2-3X the time of normal inference.
[3] Original Size Test (OST) increase the image size to its original size, which is 1280 pixels in our experiment.

Table 3. Speed Analysis

CONCLUSIONS AND FUTURE WORK

In this paper, we achieved object height estimation from monocular image sequence using a cascade of neural networks that encodes the view geometry. The cascade architecture referred to as the MOHE-Net is evaluated for its accuracy and speed in autonomous vehicle setting and is observed to achieve state-of-the-art accuracy. The proposed MOHE-Net cascade contains an object detector network and a height estimator network and perform real time estimation of height of all objects in the field of view.

ACKNOWLEDGEMENTS

This project was funded by the Ford Motor Company.

REFERENCES

Abdel-Aziz, Y., Karara, H., Hauck, M., 2015. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogrammetric Engineering & Remote Sensing*, 81(2), 103–107.

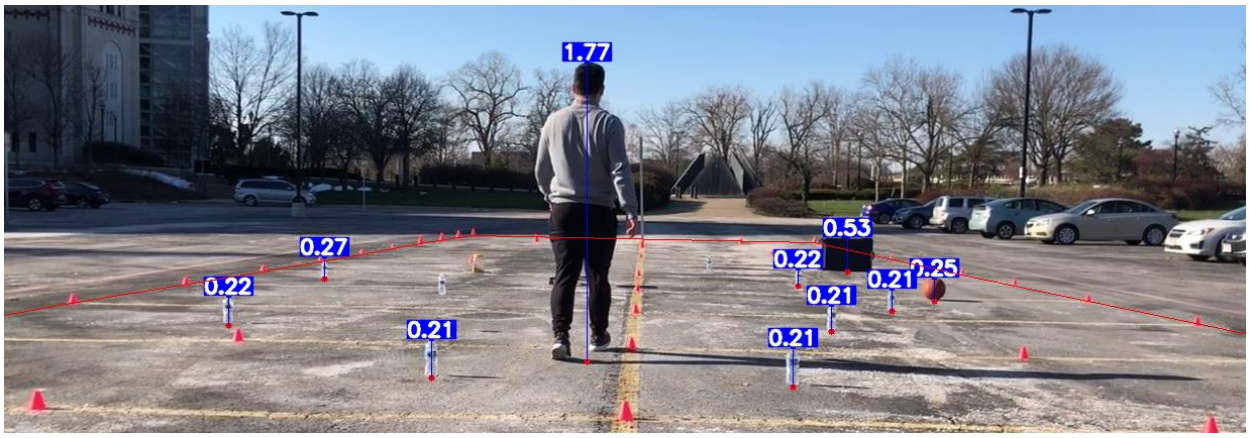

Figure 8. 2D image with bottles, sports ball, suitcase, person on the ground. MOHE-Net estimated object heights are displayed inside blue boxes. All values are reported in meters.

Figure 9. Person's estimated height over frame. Texts above the person are heights in meters.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features. *European conference on computer vision*, Springer, 404–417.

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*.

Bradski, G., 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. Brief: Binary robust independent elementary features. *European conference on computer vision*, Springer, 778–792.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303–338.

Fischler, M. A., Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.

Girshick, R., 2015. Fast r-cnn. *International Conference on Computer Vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

Godard, C., Mac Aodha, O., Brostow, G. J., 2017. Unsupervised monocular depth estimation with left-right consistency. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 270–279.

Hartley, R., Zisserman, A., 2003. *Multiple View Geometry in Computer Vision*. 2 edn, Cambridge University Press, USA.

Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, Hogan, A., lorenzomammana, yxNONG, AlexWang1900, Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Ingham, F., Frederik, Guilhen, Hatovix, Poznanski, J., Fang, J., 于力军, L. Y., changyu98, Wang, M., Gupta, N., Akhtar, O., PetrDvoracek, Rai, P., 2020. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements.

Ke, L., Li, S., Sun, Y., Tai, Y.-W., Tang, C.-K., 2020. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. *European Conference on Computer Vision*, Springer, 515–532.

Kundu, A., Li, Y., Rehg, J. M., 2018. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft coco: Common objects in context. *European conference on computer vision*, Springer, 740–755.

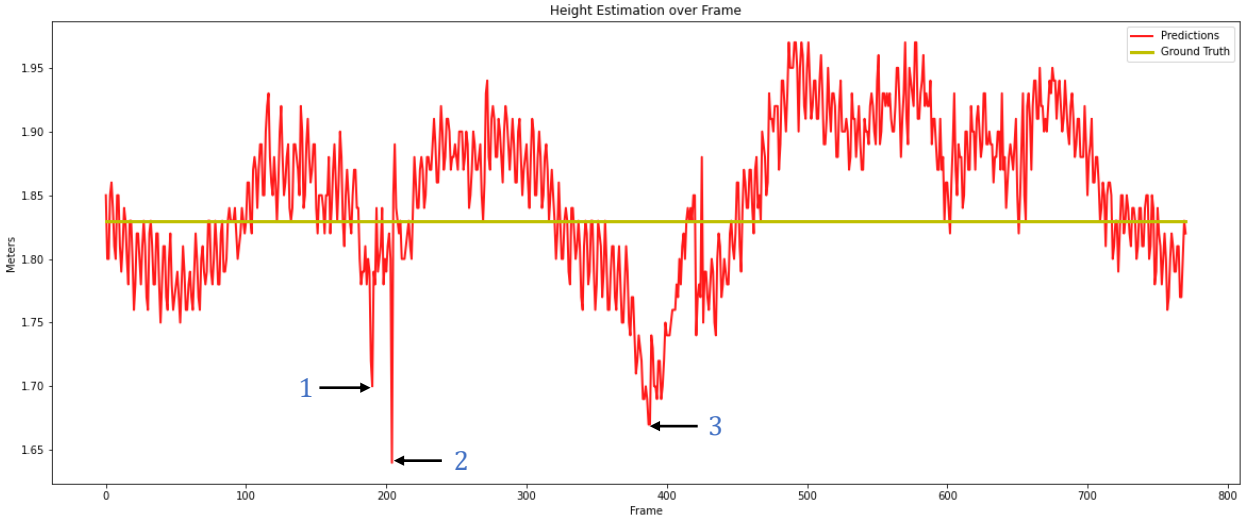

Figure 10. Person's estimated height over frame. Yellow line is ground truth.

Figure 11. Three predictions scenes corresponding to the pointing out abnormal results in Fig. 10. Person's appearance looks dark similar to the background in the first and second columns. Besides, person in column *2* was also occluded by a sign post. In third column, person is partially outside camera view, resulting in occlusion.

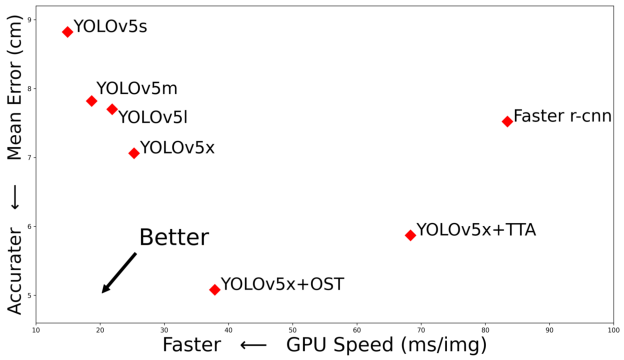

Figure 12. The speed and accuracy of seven variations MOHE-Net based on the object detection backbone are plotted in mean error (centimeters) and GPU speed (milliseconds). Black arrow indicates the faster speed and less error is better meeting our expectation.

Lindeberg, T., 2012. Scale invariant feature transform.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M., 2020. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2), 261–318.

Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J., 2017. 3d bounding box estimation using deep learning and geometry. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7074–7082.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Redmon, J., Farhadi, A., 2017. Yolo9000: better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263–7271.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), 1137–1149.

Wu, D., Zhuang, Z., Xiang, C., Zou, W., Li, X., 2019. 6d-vnet: End-to-end 6-dof vehicle pose estimation from monocular rgb images. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Zhou, T., Brown, M., Snavely, N., Lowe, D. G., 2017. Unsupervised learning of depth and ego-motion from video. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zou, Z., Shi, Z., Guo, Y., Ye, J., 2019. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*.

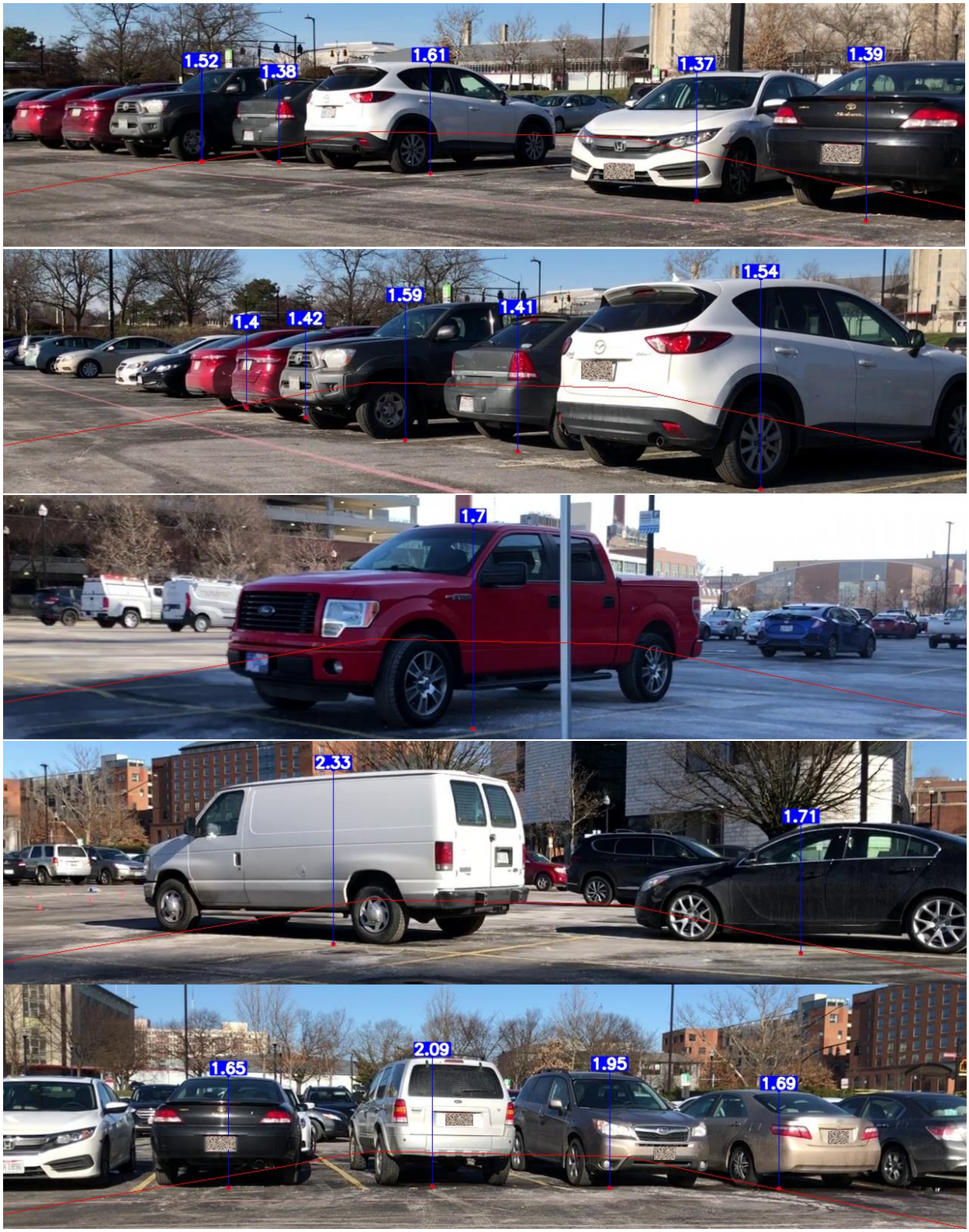

Figure 13. Platform moves and objects within the red polygons are estimated height. Heights are also displayed in the blue boxes in meters, matching judgements we made on vehicles appearance.