INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 01, January 2024, pp : 400-407

e-ISSN : 2583-1062

Impact Factor : 5.725

A REVIEW ON SEMANTIC SEGMENTATION FOR AUTONOMOUS DRIVING OR VEHICLES

Mohammad Rafi[1], R R Anusha[2], Kavya B A[3]

[1,2,3]Department of Computer Science and Engineering, UBDTCE, India.

ABSTRACT

This review paper provides a thorough examination of the recent developments in semantic segmentation techniques tailored for autonomous driving applications. Focusing on the pivotal role of semantic segmentation in enhancing scene understanding for self-driving vehicles, we analyze state-of-the-art methods, benchmark datasets, and evaluate performance metrics. The paper delves into the challenges posed by diverse environmental conditions and presents innovative solutions proposed in literature. Additionally, it explores the integration of deep learning architectures, real-time processing considerations, and the impact of hardware advancements on semantic segmentation performance. The synthesis of these findings aims to offer a valuable resource for researchers, engineers, and practitioners involved in the evolution of autonomous driving technologies.

1. INTRODUCTION

Autonomous Vehicles have a variety of different sensor systems onboard to detect obstacles, lanes, free parking spaces, etc. [1], [2]. A frequently applied technique in this field is image segmentation, which uses camera images to classify each pixel. The predicted images can be used to plan the vehicle's behavior and avoid collisions [3], [4]. This work was conducted in the context of the Carolo-Cup, a student competition providing student teams with a platform for the design and implementation of autonomous Radio Controlled (RC) vehicles. They must accomplish various driving tasks such as parking or overtaking in an imitated environment containing obstacles, intersections, parking spaces, and more. Furthermore, RC vehicles use embedded hardware to run the sensing, planning, control algorithms, etc. The algorithms must therefore run in realtime so that the vehicle can drive smoothly and reliably. [5] To classify the pixels of the images delivered by the camera built on top of the vehicle, several image segmentation models. The advent of autonomous driving technology has propelled the need for robust perception systems capable of comprehending complex real-world environments. Among the critical components of such systems, semantic segmentation stands out as a key enabler for intelligent decision-making. Semantic segmentation involves classifying each pixel in an image into distinct categories, providing a detailed understanding of the scene. In the context of autonomous vehicles, this technology plays a pivotal role in enhancing situational awareness, enabling the vehicle to navigate safely through dynamic and varied surroundings.

This review paper aims to provide a comprehensive overview of the recent advancements in semantic segmentation techniques tailored specifically for autonomous driving applications. As self-driving vehicles move closer to real-world deployment, the accuracy and efficiency of their perception systems become paramount. Semantic segmentation not only aids in object recognition but also facilitates a nuanced understanding of the spatial relationships between different entities in the scene. This nuanced understanding is crucial for decision-making algorithms, allowing vehicles to navigate, plan trajectories, and interact with the environment in a manner that ensures both safety and efficiency.

In the following sections, we will delve into the evolution of semantic segmentation methodologies, exploring the transition from traditional computer vision approaches to the dominance of deep learning techniques. We will assess the challenges inherent in autonomous driving scenarios, such as varying lighting conditions, diverse landscapes, and the need for real-time processing. Furthermore, we will analyze benchmark datasets commonly used for evaluating segmentation algorithms, shedding light on the complexities of real-world scenarios. As we navigate through the intricacies of semantic segmentation in autonomous driving, this review aims to serve as a valuable resource for researchers, engineers, and practitioners involved in advancing the state-of-the-art in autonomous vehicle perception systems. By synthesizing key findings from recent literature, we seek to contribute to the ongoing dialogue that shapes the future of autonomous driving technologies. Environmental perception is an important aspect within the field of autonomous vehicles that provides crucial information about the driving domain, including but not limited to identifying clear driving areas and surrounding obstacles. Semantic segmentation is a widely used perception method for self-driving cars that associates each pixel of an image with a predefined class. In this context, several segmentation models are evaluated regarding accuracy and efficiency. Experimental results on the generated dataset confirm that the segmentation model FasterSeg is fast enough to be used in realtime on lowpower computational (embedded) devices in self-driving cars. A simple method is also introduced to generate synthetic training data for the model. Moreover, the accuracy of the first-person perspective and the bird's eye view perspective are compared. For a 320×256 input in the first-person perspective, FasterSeg achieves 65.44% mean Intersection over Union (mIoU), and for a 320×256 input from the bird's eye view perspective, FasterSeg achieves 64.08% mIoU. Both perspectives achieve a frame rate of 247.11 Frames per Second (FPS) on the NVIDIA Jetson AGX Xavier. Lastly, the frame rate and the accuracy with respect to the arithmetic 16-bit Floating Point (FP16) and 32-bit Floating Point (FP32) of both perspectives are measured and compared on the target hardware. were evaluated. A dataset representing the imitated environment is required to train the segmentation neural network. In this context, synthetic images generated with a simulation are combined with real images of the Carolo-Cup environment to compose the training dataset. Supervised learning is used in this work because each image of the dataset has its corresponding ground truth. The motivation of this work is to generate a dataset that mainly contains synthetic data to avoid high labeling effort. Thus, the routes can be generated in a simulation and must not be replicated. Moreover, a stateof-the-art image segmentation model is applied in realtime on a comparatively slow embedded hardware. Additionally, the potential of the bird’s eye view perspective is examined. Both the overall accuracy mean Intersection over Union (mIoU) and the accuracy Intersection over Union (IoU) of each class are then investigated more closely.

This paper attempts to answer four main questions:

- Which image segmentation model is fast and accurate enough for the Carolo-Cup?
- How to easily generate labeled synthetic data? • Is the bird’s eye view perspective a better alternative compared to the first-person perspective?
- What impact does the 16-bit Floating Point (FP16) and the 32-bit Floating Point (FP32) arithmetic have on the model accuracy and the real-time capability?
- This paper is organized as follows. First, different segmentation models are evaluated to find a suitable option for this work. Secondly, a method for generating labeled synthetic data is described. Lastly, two different experiments are conducted, using the selected segmentation model trained with the generated data. The first experiment examines the accuracy of two models trained with data from two different perspectives: the first-person and the bird’s eye view perspective. The second experiment explores the real-time capability and accuracy regarding the arithmetic FP16 and FP32 of both models. The intended contributions of this study are the following:
- The development of a simple, yet effective method to generate synthetic data representing an imitated environment for autonomous vehicles.
- Exploring the possibility of executing semantic segmentation on low-power embedded devices using images from the bird’s eye view perspective.

2. RELATED WORK T

he following section describes related work which is relevant. First, the fundamentals of synthetic data generation are introduced. In the second paragraph, an overview of real data sources is provided. Then, various image segmentation models are evaluated and compared in terms of accuracy and frame rate. Finally, the chosen image segmentation model is further described.

A. ROAD GENERATION AND SIMULATION

Gazebo is chosen as the simulation environment to replicate realistic driving scenes. The synthetic routes used in Gazebo can be generated as images using a road generator provided by [6]. These images can be directly rendered in the simulation environment. To create various routes, the road generator is extended with the objects listed in Fig. 2. Furthermore, the generator is customized to create equivalent annotated routes [7].

B. SOURCES OF REAL DATA

In addition to the generated synthetic data, images from real routes in imitated environments are used. These real images are provided by various research teams such as Spatzenhirn (University of Ulm) [8], ISF Lowen (Technical University of ¨ Braunschweig) [9], KITcar (Karlsruhe Institute of Technology) [10], and it:movES (Esslingen University of Applied Sciences) [11]. Different environments offer a relatively high diversity of real images which can be very useful for training and testing an image segmentation model. The images are recorded using different cameras.

C. EVALUATION OF STATE-OF-THE-ART IMAGE SEGMENTATION MODELS

Image segmentation is an important part of visual perception systems for autonomous vehicles. It can be described as separating an image into any segments. Image segmentation can be divided into semantic segmentation and instance segmentation. Semantic segmentation refers to the process of assigning a label to each pixel of a picture. Instance segmentation extends the semantic segmentation scope further by detecting each instance of the object within the image and delineating it with a bounding box or segmentation mask. [27] To interpret the images of the vehicle’s environment, different instance and semantic segmentation models were evaluated. The goal of the evaluation is to find a model that achieves a high inference frame rate measured in Frames per Second (FPS) while concurrently high detection accuracy. Table I lists the frame rate as well as the accuracy of some state-of-the-art instance segmentation models on different GPUs. The models were rated using the MS COCO benchmark dataset [12], and the Average Precision (AP) was used as an accuracy metric. Table II lists some state-of-theart semantic segmentation models rated with the Cityscapes benchmark dataset [13]. The mIoU is used to measure the models' accuracies. The mIoU is defined as follows:

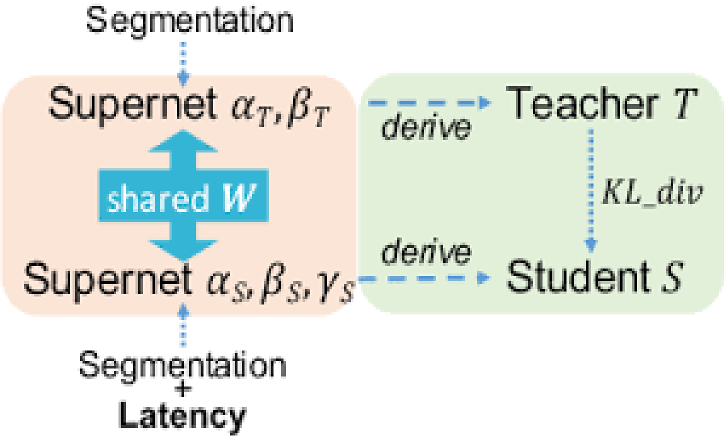


Fig. 1. During search (left), the co-searching framework optimizes two architectures, and during training from scratch (right), it distills from a complicated teacher to a light student using KL [22].

(αT , βT) and a lightweight student (αS, βS, γS). FasterSeg training can be broken down into four stages:

- Search the architecture
- Pre-train the supernet
- Pre-train the teacher network
- Pre-train the student network

In all experiments conducted the supernet is pre-trained for 20 epochs without changing the architecture parameters. Then the architecture search is done for 30 epochs. The epoch values are the same as used for the search experiments run by the FasterSeg developing team. [22]

3. PROPOSED APPROACH

The definitions of the labels with various scenarios of the imitated environment are shown in Fig. 2. In this section, the generation of the labeled data is described. Additionally, a method to transform the images into a bird's eye view perspective is presented.

A. SYNTHETIC DATA GENERATION

The process of synthetic data generation is shown in its entirety in Fig. 3. This process is divided into three fields: the road generator, the simulation, and the image processing. Furthermore, the automation level of each task within the fields is visualized with a corresponding color. In the following, the tasks of each main field of the figure are described.

1) Road Generator: Synthetic data generation starts with the creation of a route layout. High diversity and different constellations are essential for accurate predictions. Therefore, various configurations of parking zones, intersections, center line types, missing lines, objects, and curves with different radii and angles must be created. A raw and an annotated route are automatically generated using the designed layout. The road generator also creates x- and y-coordinates, as well as the yaw angle. This represents the spatial orientation for the trajectory of the simulated vehicle. The coordinates run along the center of the right lane. Special driving maneuvers, such as overtaking, parking, or crossing the intersection from different directions, must be added manually.

2) Simulation: As described in Section II-A, Gazebo is selected as the simulation environment to generate synthetic training data for this work. The simulator produces realistic first-person perspective image sequences that replicate real driving scenarios. To achieve this, two different virtual vehicles are rendered in the simulator, each one driving on a different route created by the road generator. The first vehicle is driving on the raw route, while the second one is assigned the colored route. Due to the fact Gazebo is based on Robotic Operating System (ROS), the architecture and therefore the behavior of both vehicles are similar. The virtual vehicles were built based on the RC vehicle used at Esslingen University. After rendering both vehicles and routes in the simulation environment, the trajectory generated by the road generator is published using ROS. The publishing of the trajectory is executed for both vehicles at the same time. The camera topics are then recorded to produce two synchronized sequences of raw and colored images. These images are finally sent to the next stage for processing. Fig. 4 shows a flow chart of the trajectory publishing process.

3) Image Processing: To process the annotations, FasterSeg requires an 8-bit grayscale image where each pixel contains the class ID. Therefore, each pixel of the colored image is replaced by the corresponding class ID using a lookup table. The result is an 8-bit grayscale image with IDs representing each class. Additionally, a Region of Interest (ROI) is set to exclude undetectable objects near the horizon. FasterSeg also requires the image height and width to be divisible by 64. The generated images are thus checked and eventually downscaled.

B. REAL DATA GENERATION

To cover all driving scenarios and achieve optimal predictions, synthetic data must be extended with real images. It is necessary to consider special features that are not included in the simulation like natural lightning, blurred scenes, and surroundings beyond the route, as illustrated in Fig. 5. Hence, real images containing these features are added to the dataset. The real images include different driving maneuvers as well as objects with various orientations and visibility. In addition, depending on the data source, various image resolutions, grayscale and colored images, and various RC vehicles are used (II-B).

C. BIRD'S EYE VIEW TRANSFORMATION

There are multiple ways to transform a first-person perspective image into a bird's eye view perspective. The warp perspective mapping method [26] is used for this work since no intrinsic nor extrinsic parameters of the cameras are available. This mapping method is suitable for several different camera models and does not require additional calibration. The mapping process consists of selecting four points Xego

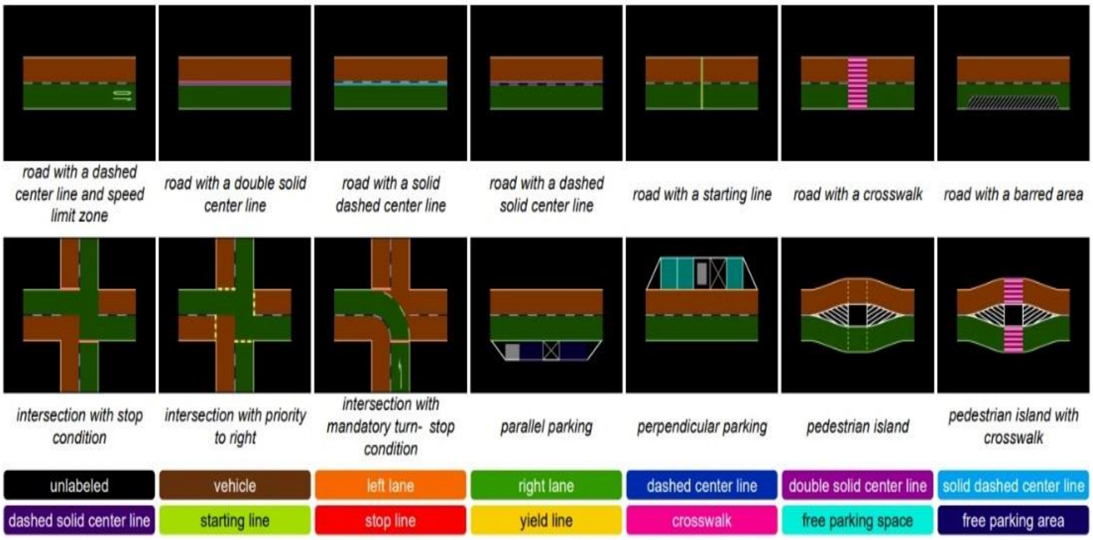

road with a dashed center line and speed limit zone
road with a double solid center line
road with a solid dashed center line
road with a dashed solid center line
road with a starting line
road with a crosswalk
road with a barred area
intersection with stop condition
intersection with priority to right
intersection with mandatory turn- stop condition
parallel parking
perpendicular parking
pedestrian island
pedestrian island with crosswalk
unlabeled
vehicle
left lane
right lane
dashed center line
double solid center line
solid dashed center line
dashed solid center line
starting line
stop line
yield line
crosswalk
free parking space
free parking area

Fig. 2. Definition of the existing objects. Each color represents the corresponding class, which should be recognized.

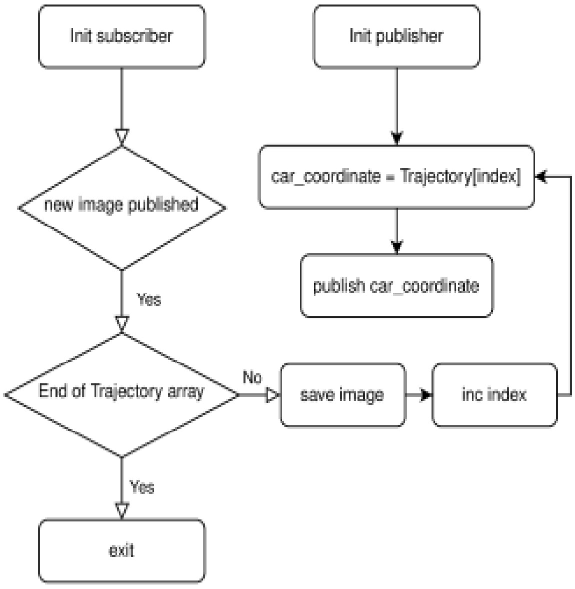


Fig.3

on the ground plane from the first-person perspective and their corresponding points X bird from the bird's eye view perspective as illustrated in Fig. 6. X ego of the input image will be viewed as X bird. The mapping from X ego to X bird can be expressed as:

The transformation matrix H can be calculated using the equation above. The matrix H is then used to map the input images from the first-person to the output of the bird's eye view perspective using a pixel-by-pixel process [25].

4. EXPERIMENTS

In this section, the conducted experiments to test the performance of the proposed model on the generated dataset as well as the respective results are described. In this context, the accuracy of the first-person and bird's eye view perspectives are compared. Also, the frame rate and the accuracy of both perspectives are measured and compared using different model arithmetic on the NVIDIA Jetson AGX Xavier board.

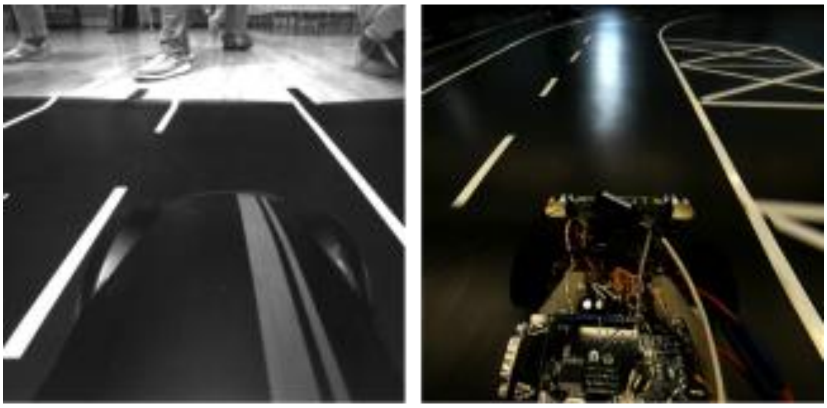


Fig. 4: In the left image [10], surroundings beyond the track like legs, feets and shoes are depicted in the background. The right image [11] shows a reflection on the track caused by natural lightning

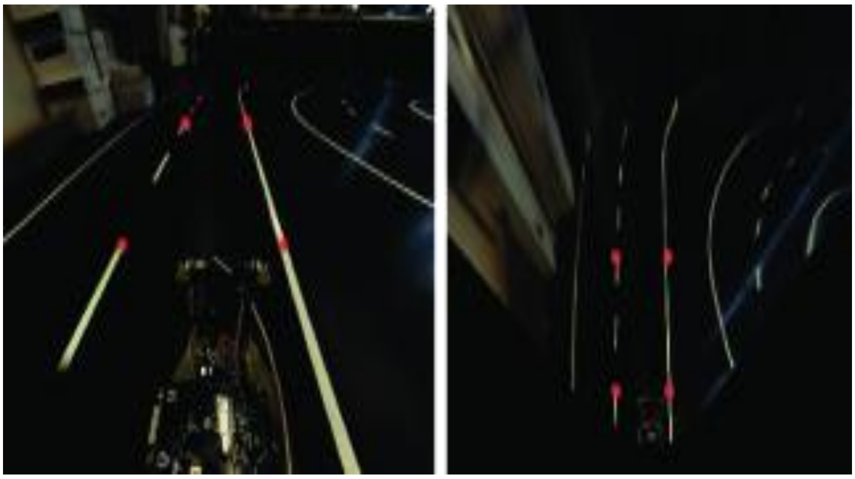


Fig.5: Bird's eye view transformation process using four mapping points illustrated in red.

A. COMPARISON OF FIRST-PERSON AND BIRD'S EYE VIEW PERSPECTIVE REGARDING ACCURACY

This experiment compares the accuracy of the semantic segmentation model regarding the first-person and the bird's eye view perspectives. For this purpose, two FasterSeg models are trained using a dataset from the first-person and the bird's eye view perspectives. In the following, the dataset and the hyperparameters, which were used to train the models, are described. Finally, the results of this experiment are presented.

1) DATASET:

Table III describes the dataset used for the training of the FasterSeg models. The dataset consists of synthetic and real images, received from the it:movES team. The images are divided into three sets: a training set (Train) containing 75 % of the images, a validation set (Val) consisting of 25 % of the images, and a test set (Test) composed of 20 real images used to measure the accuracy of the models. To train the bird's eye view model, all the images are transformed into the bird's eye view perspective using the method described in section III-C. Both FasterSeg models are trained using the same resolution (320 × 256) to objectively compare both perspectives. It is important to consider that this dataset contains only 11 objects instead of the initial 14 illustrated in Fig. 2.

2) SETTINGS:

The training process of FasterSeg is divided into four substeps [22], [24]. Note that only the teacher network is used in this experiment. The configured hyperparameters, such as the number of epochs and the number of iterations per epoch, are listed in Table IV. The training runs on an NVIDIA Tesla V100S-PCI GPU.

TABLE-1 The Hyperparameters Adjusted For The Perspective Comparison Experiment.

Substep	Epochs	Iterations	Batch size	Initial learning rate
Pretrain Supernet	20	400	3	2.10^{-2}
Search the Architecture	30	400	2	1.10^{-2}
Train the Teacher Network	249	1000	12	1.10^{-2}

3) RESULTS:

Table V lists the predictions' accuracies of the trained models. Strikingly, the bird's eye view perspective achieves an mIoU that is almost as good as the first-person perspective. Considering the IoU of each class, the double solid center line and the stop line reach a much higher IoU in the first-person perspective than the bird's eye view perspective. On the other hand, free parking space is predicted better in the bird's eye view perspective. Fig. 7 shows several test predictions and their corresponding ground truth images. Although some of the images have light reflections on the track, this has no apparent impact on the predictions' quality. Note that the predictions are less accurate when the vehicle changes lanes. The left and the right lanes are often confused during such maneuvers. Furthermore, both perspectives achieve an inference frame rate of 247.11 FPS on the NVIDIA Jetson AGX Xavier.

TABLE.2 - The Measured Accuracies Of The First-Person And Bird's Eye View Models Tested With Uniform Resolution.

Class	IOU[%] First-person perspective	IOU[%] bird eye view Perspective
unlabeled	92.74	96.40
left lane	74.94	78.32
right lane	83.97	79.43
dashed center line	51.57	53.84
double solid center line	89.01	53.32
starting line	65.71 65.71	60.51
stop line	50.48	27.37
crosswalk	76.26 76.26	88.89
free parking space	7.49	45.79
free parking area	62.27	56.93
mIoU	65.44	64.08

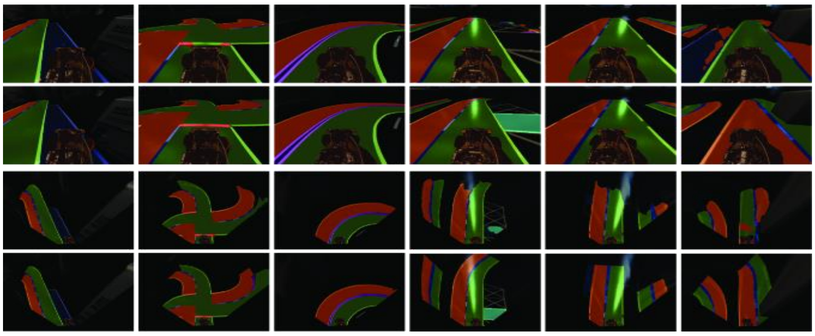


Fig. 6. Visual predictions of the trained FasterSeg models uzsing test dataset. First and second rows show the predictions and the corresponding ground truth images in the first-person perspective. Third and fourth rows illustrate the predictions and the corresponding ground truth images in the bird's eye view perspective. Each color corresponds to a predefined class.

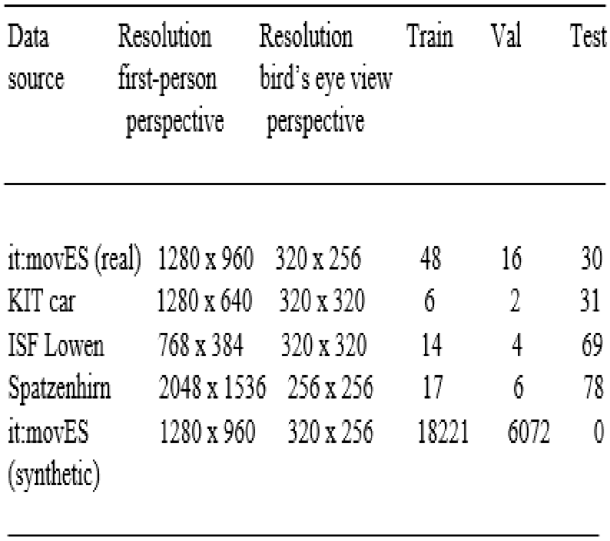
B. REAL-TIME CAPABILITY AND ACCURACY IN RELATION TO THE ARITHMETIC

This experiment can be divided into two parts. The first part deals with the analysis of the real-time capability of FasterSeg. The second part examines the accuracy of FasterSeg concerning the arithmetic. For this purpose, two models are trained using a dataset different from the experiment above. The dataset consists of various resolution images from the first-person and the bird's eye view perspectives. The inference is conducted on the NVIDIA Jetson AGX Xavier using the FP16 and the FP32 arithmetic. The TensorRT framework is used to perform the inference. In the following, the used dataset and hyperparameters are described. Finally, the results of the experiment are presented.

1) DATASET: The dataset used to train the models is listed in Table VI. It consists of synthetic and real images from different data sources with different resolutions. The dataset is also divided into three sets as described in IV-A1. The test set used to compute the accuracy consists of 208 real images. Note that the dataset is significantly larger than the dataset used in the previous experiment. The dataset is also transformed into the bird's eye view perspective to train the second Faster Seg model.

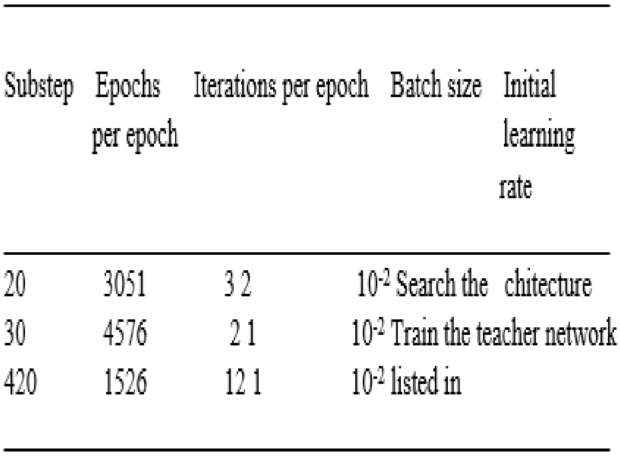
2) SETTINGS: The adjusted hyperparameters for the FasterSeg models are

TABLE- 3 The Dataset Generated For The Real-Time Capability And The Arithmetic Accuracy Comparison Experiment



Data source	Resolution first-person perspective	Resolution bird's eye view perspective	Train	Val	Test
it:movES (real)	1280 x 960	320 x 256	48	16	30
KIT car	1280 x 640	320 x 320	6	2	31
ISF Lowen	768 x 384	320 x 320	14	4	69
Spatzenhirn	2048 x 1536	256 x 256	17	6	78
it:movES (synthetic)	1280 x 960	320 x 256	18221	6072	0

TABLE- 4 The Hyperparameters Adjusted For The Real-Time Capability And The Arithmetic Accuracy Comparison Experiment.



Substep	Epochs per epoch	Iterations per epoch	Batch size	Initial learning rate
20	3051	3 2	10^{-2} Search the	chitecture
30	4576	2 1	10^{-2} Train the teacher network	
420	1526	12 1	10^{-2} listed in	

5. CONCLUSION

In this paper, the semantic segmentation model FasterSeg was investigated regarding the accuracy and the real-time capability in the Carolo-Cup environment on NVIDIA Jetson AGX Xavier embedded hardware.

Synthetic images, which were generated using a semi-automated process, as well as real images were used to train the FasterSeg model. The experimental evaluation demonstrated that FasterSeg model reaches an accuracy of over 64 % and a frame rate of 247.11 FPS in a Carolo-Cup environment.

6. REFERENCES

[1] Fridrich, A.J., Soukal, B.D., Lukas, A.J.: ‘Detection of copy–move forgery in digital images’. Proc. Digital Forensic Research Workshop, 2003

[2] Muhammad G, Hussain M, Bebisi G, ‘Passive copy–move image forgery detection using undecimated dyadic wavelet transform, Digit. Invest 2012.

[3] Caldelli, R, Amerini, I., Ballan, L., et al. ‘On the effectiveness of local warping against SIFT-based copy–move detection’. Communications Control and Signal Processing (ISCCSP), 2012

[4] Gonzalez R.C., Woods, R.E. ‘Digital image processing’ (Pearson Education India, 2009)

[5] Computer Vision Group, University of Granada Image Database, Computer Vision Group, University of Granada http://decsai.ugr.es/cvg, (accessed on 16th Feb, 2016)

[6] Farid, H. ‘Image forgery detection’, IEEE Signal Process. Mag., 2009

[7] Sencar H.T, Memon, N. ‘Overview of state-of-the-art in digital image forensics’, Algorithms Archit. Inf. Syst. Secur., 2008.

[8] Birajdara, G.K., Mankar, V.H. ‘Digital image forgery detection using passive techniques: a survey’, Digit. Invest., 2013.

[9] Savchenko V, Kojekine, N, Unno H. ‘A practical image retouching method’. Proc. First Int. Symp. Cyber Worlds, 2002.

[10] Redi J.A, Taktak, W, Dugelay, J. ‘Image splicing detection using 2D phase congruency and statistical moments of characteristic function’. Society of Photo-optical Instrumentation Engineers (SPIE) Conf. Series, 2007

[11] Yap, P.T., Raveendran, P. ‘Image focus measure based on Chebyshev moments’, IEE Proc., Vis. Image Signal Process., 2004.

[12] Klema, V.C, Laub, A.J. ‘The singular value decomposition and its computation and some applications’, IEEE Trans. Autom. Control, 1980

[13] Ting Z, Ding, W.R. ‘Copy–move forgery detection based on SVD in digital image’. Second Int. Congress on Image and Signal Processing, 2009

[14] Khotanzad A., Hong, Y.H. ‘Invariant image recognition by Zernike moments’, IEEE Trans. Pattern Anal. Mach. Intell., 2009

[15] Kingsbury N. ‘A dual-tree complex wavelet transform with improved orthogonality and symmetry properties’. Proc. Int. Conf. Image Processing, 2000, 2014

[16] Signal and Image Processing Institute, University of Southern California, Department of Electrical Engineering. http://sipi.usc.edu/database/database.php?volume=misc,

[17] (accessed on 16th Feb, 2016)