3D Perception for Autonomous Mobile Robots Navigation Using Deep Learning for Safe Zones Detection: A Comparative Study

Felipe Manfio Barbosa
Universidade de São Paulo
felipe.manfio.barbosa@usp.br

Fernando Santos Osório
Universidade de São Paulo
fosorio@usp.br

ABSTRACT

Computer vision plays an important role in intelligent systems, particularly for autonomous mobile robots and intelligent vehicles. It is essential to the correct operation of such systems, increasing safety for users/passengers and also for other people in the environment. One of its many levels of analysis is semantic segmentation, which provides powerful insights in scene understanding, a task of utmost importance in autonomous navigation. Recent developments have shown the power of deep learning models applied to semantic segmentation. Besides, 3D data shows up as a richer representation of the world. Although there are many studies comparing the performances of several semantic segmentation models, they mostly consider the task over 2D images and none of them include the recent GAN models in the analysis. In this paper, we carry out the study, implementation and comparison of recent deep learning models for 3D semantic image segmentation. We consider the FCN, SegNet and Pix2Pix models. The 3D images are captured indoors and gathered in a dataset created for the scope of this project. Our main objective is to evaluate and compare the models' performances and efficiency in detecting obstacles, safe and unsafe zones for autonomous mobile robots navigation. Considering as metrics the mean IoU values, number of parameters and inference time, our experiments show that Pix2Pix, a recent Conditional Generative Adversarial Network, outperforms the FCN and SegNet models in the considered task.

KEYWORDS

Computer Vision, Deep Semantic Segmentation, RGB-D Images, Autonomous Mobile Robots' Navigation

1 INTRODUCTION

The rapid progress of technical and scientific knowledge related to artificial intelligence and robotics allowed the development of disruptive technologies like autonomous vehicles. This technology, although very practical and with many promising applications, still faces some limitations that can lead to errors in operation, with severe consequences, also, related to people safety.

In 2018, a self-driving Uber car hit and killed a cyclist because of an error in its vision system, which couldn't recognize that object as a cyclist or pedestrian, and incorrectly calculated its trajectory and the time for activating the brakes. The authorities concluded that the error occurred because the car "lacked the capability to classify an object as a pedestrian unless that object was near a crosswalk" [1]. Another event, also occurred in 2018, involved a Tesla semi-autonomous vehicle in auto-pilot mode, which accelerated directly to a barrier in the highway. The driver suffered fatal consequences. The authorities agreed that the accident occurred because "the collision avoidance system was not designed to detect the crash [barrier]" [2].

Motivated by reducing the chances of events like these to happen and inspired by the current effort in study and development related to autonomous vehicles, which can also be classified as autonomous mobile robots, we seek to simulate and compare, on a reduced scale, the operation of such systems, concerning the role of computer vision methods in scene understanding.

Since one of the main characteristics of such systems is the interaction with the environment, computer vision is of utmost importance for their correct and safe operation, as it provides an interface between the robot and the world. Defined as the transformation of data from a still or video camera into either a decision or a new representation [3], computer vision provides us with many methods, or levels, of image analysis (Fig. 1). When considering scene understanding, a particularly powerful method is semantic segmentation. Defined as the process of associating each pixel of an image with a class label [4], semantic segmentation allows to figure out what is in the image at pixel level [5].

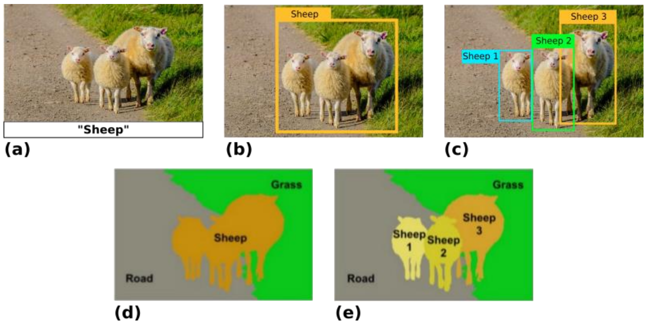

Figure 1: Levels of analysis in computer vision: classification (a), classification with localization (b), detection (c), semantic segmentation (d) and instance segmentation (e). Adapted from: <https://bit.ly/2Cg4mAn>.

Classical approaches of semantic segmentation demanded complex and time-consuming hand-engineered pipelines for feature extraction and classification – thresholding, edge detection and the K-means algorithms are some examples [6]. However, the advent of deep learning and, more precisely, the convolutional neural networks have permitted the automation of that pipeline, naturally performing hierarchical feature extraction and classification through end-to-end learning. This advance allowed semantic segmentation to achieve great progress in recent years. The first important work in deep semantic segmentation was the Fully Convolutional Networks (FCNs) [7], which proposed an end-to-end approach to pixel-wise classification by replacing fully connected layers by their equivalent convolutional ones. Another prominent work in this field is SegNet [8], an encoder-decoder architecture, primarily motivated by road scene understanding applications, which introduced the concept of pooling indices to encapsulate boundary information throughout the network.

These models are often used as baselines in comparative studies on deep semantic segmentation available in the literature [9-11]. One common aspect of these works is that they compare variations of well-known and accepted architectures, such as VGG16 [12] and UNet [13].

Currently, though, another breakthrough architecture emerged: the Generative Adversarial Networks (GANs) [14]. With applications ranging from high resolution image synthesis [15-16] to image segmentation [17], they represent a promising base for future developments in the field of computer vision. Throughout the years, several GAN based architectures were proposed [15, 16, 18]. Pix2Pix [19], which has particularly attracted a lot of attention, is an example of a Conditional GAN. Its main characteristic is that it considers the input as part of the loss calculation, allowing it to fit to a wide range of image-to-image applications – reason why this type of loss is known as adaptive loss.

Therefore, one of the main contributions of this paper is our analysis of promising and currently widely used GANs, in addition to famous deep semantic segmentation models, such as FCN and SegNet. This is done in order to evaluate the impact of the advances in architectural designs in the performance of deep semantic segmentation tasks.

Another point to note is that all the works aforementioned conduct the analysis based on datasets composed by 2D images. In fact, great part of the progress made in deep semantic segmentation is due to the proposition of various large annotated datasets for 2D semantic image segmentation [20-21], some of them specifically created for autonomous driving use cases [22-23]. The choice to use 2D data was mainly driven by the inaccessibility of sensors for 3D data acquisition years ago. However, cheaper sensors have permitted a growing accessibility to different data representation types, other than 2D images. One example is the Microsoft Kinect [24] sensor, which permits the capture of 3D data and other types of image representation by an accessible cost.

As a richer representation of the world, 3D data (RGB-D images, point clouds and depth-maps) describe the environment with the additional information of depth. A particular type of 3D data is the RGB-D – sometimes called 2.5D - representation, which incorporates scene depth into the image structure, i.e. encoded in the color channels. With this approach, models originally engineered to work with 2D RGB images can leverage the additional depth information of the RGB-D data to improve their performances with no need for modifications [25].

Hence, another important contribution of this work is the analysis of the models on 3D data, more precisely RGB-D images. Instead of using one of the publicly available RGB-D datasets, frequently aimed at indoor object segmentation [26-27], we choose to create our own dataset. This was done because we wanted to, instead of just detecting objects, add context information by simulating a navigation path for an autonomous mobile robot, with the safe zone, the borders and the obstacles.

We analyze the performance of each model in the task of segmenting safe and unsafe zones for navigation, borders and obstacles, considering the 3D data (new approach). We compare them with regards to mean IoU value, memory/disk space demand and inference time, critical issues when choosing robust and precise models to embed in autonomous mobile robots. We also evaluate the results qualitatively, through visual inspection of the resultant segmentations.

The remainder of the paper is organized as follows. In Section 2, we review related recent works. We detail the methodology adopted in Section 3. In Section 4, we describe our experiments. We present the results and evaluate the performances of the models on our RGB-D indoor simulated road scene dataset in Section 5. Finally, we conclude the paper and present a discussion regarding our approach and alternative directions to future work in Section 6.

2 RELATED WORK

Semantic segmentation is an active topic of research and has been widely applied to many different fields, ranging from medicine [28-29] to autonomous navigation [30-31], a topic of research and development with great popularity nowadays.

Fueled by diverse large datasets of 2D annotated images [22, 23, 32, 33], the research on deep convolutional models for semantic segmentation (deep semantic segmentation) has achieved significant results in recent years [7-8].

These developments paved the road for many other works in the field, so that the literature currently presents several deep learning architectures, models and approaches to semantic segmentation [25, 34].

Given this context, several studies have explored the problem of comparing different deep learning algorithms available for 2D semantic image segmentation. In [9], a comparative study of FCN model and its variants is performed, based on accuracy and training time metrics. In [10], a real-time semantic segmentation benchmarking framework with a decoupled design for feature extraction and decoding methods is presented. The authors conduct experiments with different combinations of feature extractors and decoders, both composed by well-known deep learning architectures like VGG16, MobileNet [35] and UNet. Finally, they present a comparative analysis of the different combinations based on IoU and efficiency with respect to the computational cost (number of operations) of running the models. In [11], a similar analysis is performed, but using the running time and IoU values as metrics.

Additionally, the advent of low cost RGB-D sensors, like the Microsoft Kinect, permitted to incorporate depth information in RGB images as a means of improving performance of models originally designed for 2D segmentation and detection tasks. In [36-37], the authors show that RGB-D images represent up to 6% improvement in comparison with RGB-based approaches for semantic segmentation. In [38] is proposed a method based on RGB-D images that achieves 59% improvement over the SegNet model on the SUN RGB-D dataset [26]. This stimulated the proposition of various RGB-D datasets. Some examples are [26, 39].

In [36] an extensive survey on indoor RGB-D semantic segmentation is presented. The authors evaluate the performance of different approaches, ranging from hand-crafted feature analysis to deep convolutional models. The evaluation is performed taking into account the pixel accuracy and the mean intersection over union value over different RGB-D datasets [26, 39, 40].

All the aforementioned comparative works focuses on famous and well accepted deep convolutional approaches. However, recent developments in Generative Adversarial Neural Networks [14] supplied the research community with a powerful tool for further developments in image synthesis [15, 16, 18], autonomous driving [41] and semantic segmentation [17, 42, 43], to mention some. A particularly distinct work applied a conditional term to the GAN architecture in order to create the Pix2Pix model, which has as its main characteristic the adaptability to a wide range of image-to-image translation scenarios [19].

Inspired by the aforementioned works and the current developments and popularity of autonomous driving, we conduct a comparative analysis on deep semantic segmentation methods applied to autonomous navigation. Like the previous studies, we analyze the models in terms of mean IoU values and inference time; we additionally consider the efficiency with regards to the model size (number of parameters). Similarly to [36], we evaluate the models in the task of RGB-D semantic image segmentation; unlike it, though, we construct our own dataset, comprised of 562 RGB-D images from indoor scenes, gathered with a Kinect sensor. This project choice was made in order to simulate, in reduced scale, the elements of an urban driving context - safe zone/unsafe zones, border and obstacles. Finally, unlike previous works, and as our main contribution, we add a GAN-based model (Pix2Pix [19]) to the set of evaluated methods.

We hope that the use of depth information can help learning the segmentations for the elements in the scene - road, borders and obstacles. Besides, the addition of a GAN-based model was intended in order to evaluate the performance of a new, powerful and extremely adaptable architecture front state-of-the-art methods in deep semantic segmentation.

3 METHOD

The models were trained end-to-end through supervised learning. To this end, we first created an entirely new dataset, comprising RGB-D images and its correspondent annotations. We after implemented and trained the models to perform semantic segmentation on this brand new dataset. Finally, we evaluated and compared the results with regards to precision and robustness based on different performance metrics.

3.1 Data Acquisition

We used the Microsoft Kinect V2 sensor (Fig. 2) in order to obtain the data. Using its "Kinect for Windows Software Development Kit" (SDK), we captured different scenario settings.

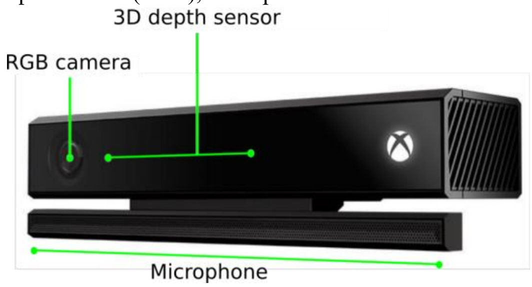

Figure 2: Microsoft Kinect V2 sensor and its components.

Constructed in indoor environment, these scenarios simulated, in reduced scale, elements of an urban driving context, like a street (plane ground), its borders (PVC tubes) and possible obstacles (objects available in the laboratory). As the main objective of this work is to study the suitability of different semantic segmentation approaches to autonomous mobile robots, we simulated its navigation by placing the Kinect sensor on top of a mobile robot, to capture its perspective. We then simulated several setups for navigation, by changing the angles and heights of the sensor, as well as building different "street" configurations. Fig. 3 shows an example of the pipeline used for data acquisition.

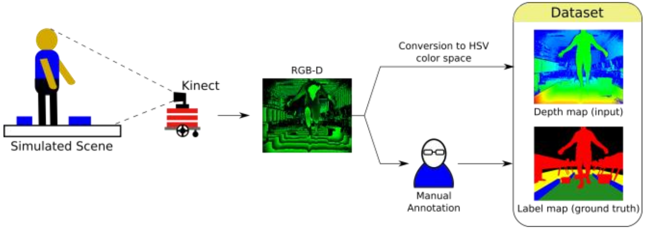

Figure 3: Pipeline used for data acquisition.

We collected 562 RGB-D images (Fig. 4 (a)). In this type of data representation, the depth information is structurally stored in one of the color channels. To better visualize the depth information of the scene, the images were converted to the HSV color space, which gives us the correspondent depth map representation (Fig. 4 (b)).

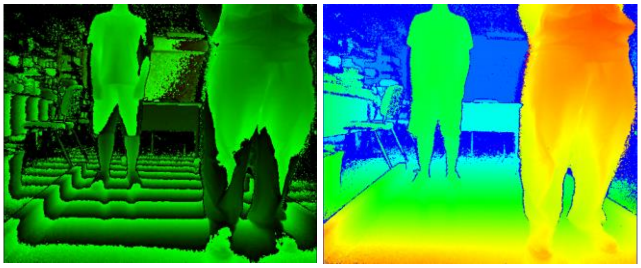

Figure 4: Example of image captured with the Kinect sensor and its correspondent depth map in the HSV color space.

3.2 Data Annotation

In supervised learning, there are two main elements: the input and the expected output, also known as ground truth. The ground truth is a translated representation of the input, i.e. the target, which depends on the application. In our case, it consists of label maps, or annotations, of the inputs.

The process of annotating or labeling an image consists in the association of each pixel to a class. To this end, we’ve used an annotation tool called LabelMe [44], a free software that implements a friendly interface for image annotating (manual annotation step – Fig. 3). The original image superimposed with the labels and also the final label map used as ground truth in the dataset are depicted in Fig. 5.

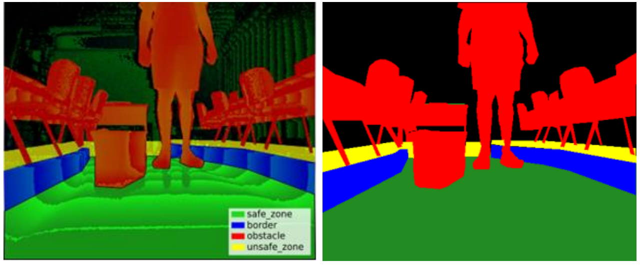

Figure 5: Overlapping of labels on the original image, with description of the classes considered, and map of labels to be used when training the models.

3.3 Model Implementations

The models were implemented using Python [45], Tensorflow [46] and the Keras API [47].

3.3.1 Fully Convolutional Networks (FCNs)

Proposed in [7], this architecture has as its main characteristic the absence of densely connected layers. This is achieved by converting all the original dense layers of a base model into their equivalent convolutional ones, in a process called “convolutionalization”. Therefore, instead of an array of probabilities, it outputs dense predictions composed by matrices, called heatmaps. Each heatmap is related to a class and contains the probabilities of each pixel to belong to that class – for instance, in an application with 21 possible classes, the output will be composed by 21 heatmaps, each one with the same dimensions (height x width) of the input image.

The authors also introduce three variants of the architecture: 32s, 16s and 8s. Each variant is related to how the output is generated. In the first case, the output is generated by directly upsampling the pixel-wise predictions using a stride of 32 (32s), so that the predictions match the dimensions of the input. In the other cases (16s and 8s), before being upsampled, the output pixel-wise predictions are combined with coarser feature maps, obtained from earlier layers in the network.

The main insight of this approach is to combine structural features (earlier layers) with semantic features (final layers) through skip connections, in order to obtain a more detailed output prediction.

Besides implementing the model based on a VGG16 core, proposed by the authors, we extended the concepts to a DenseNet [48] core. For both base models, we used the implementations available in Keras [49-50].

3.3.2 SegNet

Proposed in [8], the SegNet model consists of a convolutional encoder-decoder architecture; the encoder uses as base model the feature extractor – model without classification head - of the VGG16 architecture. Its key concept is related to the decoder and how the upsampling operation is performed.

Motivated by scene comprehension tasks and designed to be efficient both in terms of memory used and inference time, the network has a reduced number of parameters, thanks to the concept of pooling indices, introduced by the authors.

The pooling indices encapsulate the position of the terms selected during the max pooling operation in a given encoder block. Then, in the correspondent decoder block, that information is used to perform the upsampling, generating a sparse feature map. In this feature map, the non-zero values are stored in the positions indicated by their correspondent pooling indices. After that, a convolution with learnable weights is applied, finally generating a dense feature map.

The use of pooling indices is justified to retain information related to the contour of the extracted image representation, so that the model not only produces smooth segmentations for large classes, but also precisely delineates small objects.

The model was developed using the VGG16 implementation also available at [49].

3.3.3 Pix2Pix

Proposed in [19], this model consists in a framework for image-to-image translations. It is based on the Conditional Generative Adversarial Networks (cGAN), which unlike the standard GANs [14] considers the inputs as part of the loss calculation. This characteristic allows the model to be suitable to a wide variety of applications.

As semantic segmentation can be defined as the process of classifying an image at pixel level, we can naturally consider using the Pix2Pix model for semantic segmentation tasks, as it operates in a similar level of image-to-image translation.

The Pix2Pix model was implemented based on the code available at [51], with some changes to adapt it to the current application.

4 EXPERIMENTS

After creating the dataset and implementing the models, we followed to the training step. During the experiments, we studied the influence of different batch sizes in the models' performances.

4.1 Performance Metrics

4.1.1 Mean Intersection Over Union (mean IoU)

Also known as Jaccard Index, the intersection over union (IoU) metric is one of the most commonly used when evaluating semantic segmentation models [9, 10, 11, 36]. It can be defined as the area of overlap between the predicted segmentation and the ground truth, divided by the area of union between them [52] (Fig. 6). The average value of this metric, also called mean IoU, is calculated as the average of the IoU values obtained for each class considered in the problem.

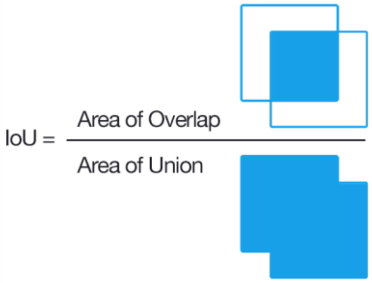

Figure 6: IoU calculation visualized [50].

4.1.2 Model efficiency

We also evaluated the models' efficiencies, based on their mean IoU value and total number of parameters, either trainable or not. The logic behind this measure works basically as follows: the fewer the number of parameters and the higher the value of mean IoU, the more efficient the model.

4.1.3 Inference Time

We finally evaluated the models based on their inference time on the test set; that is, the time taken for a given RGB-D image to be translated to its correspondent segmentation. This is an important analysis since the faster the prediction, the more time the system has to plan an act in order to recover from a potentially dangerous situation.

4.2 Environment Setup

The models were trained on an Acer Nitro 5 notebook with the configuration presented in Table 1. To configure and manage the package dependencies we used virtual environments.

Processor	Intel Core i5-8300H
Memory (RAM)	8GB
GPU	NVIDIA GeForce 1050
Operating System	Windows 10

Table 1: Environment configuration.

4.3 Dataset Setup

The distribution of the dataset into training, validation and test sets is depicted in Fig. 7. In order to increase the number of training examples, the data augmentation technique was also used to generate synthetic samples. This technique consists in applying transformations to the original data, in order to obtain a greater variety of representations. The types of transformations applied in the context of this project were: horizontal and vertical translations, rotation and horizontal flip (Fig. 8)

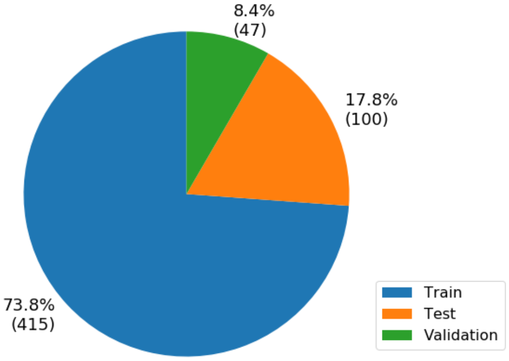

Figure 7: Number of examples per subset (train, validation and test).

4.4 Training

All models were trained from scratch, for 50 epochs and using the SGD optimizer with default parameters (learning rate = 0.01, momentum = 0.0). In order to study the hyper parameters' influence in model performance, we trained the variants with 2, 4 and 8 images per batch.

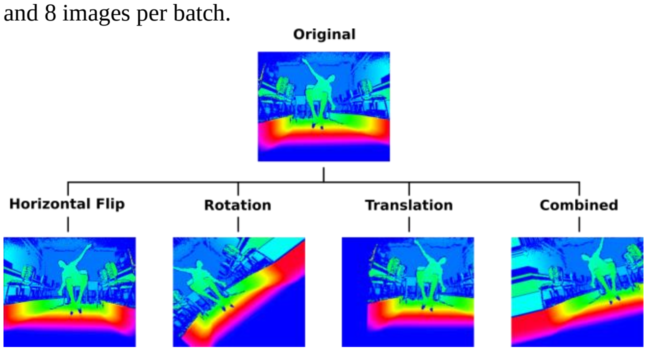

Figure 8: Examples of data augmentation.

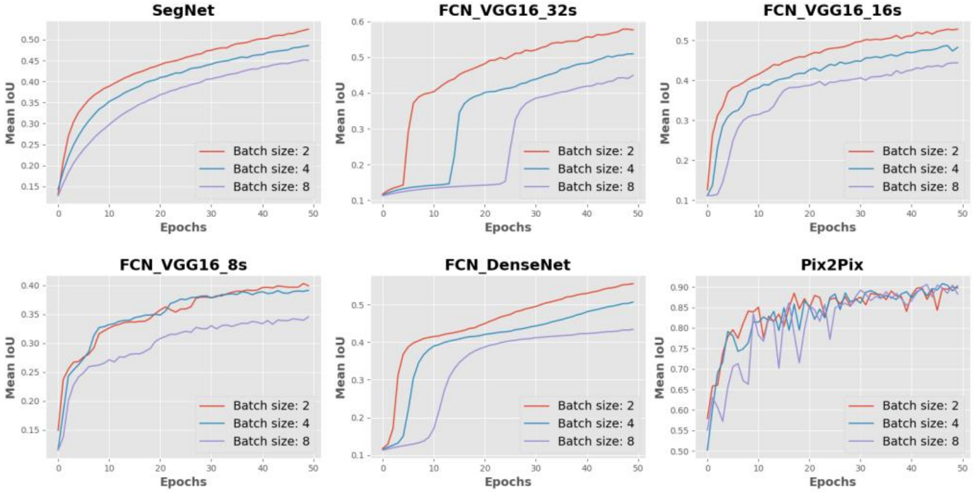

Figure 9: Evolution of neural learning, with respect to the mean IoU value, for different batch size values.

5 RESULTS

During the training phase, enough data was generated to carry out different types of analyses. First, we studied the influence of batch size in the performance; then we evaluated the models with regards to the mean Intersection Over Union (mean IoU) value, model size and inference time.

5.1 Influence of Batch Size

Fig. 9 shows that, in general, the best performances were obtained for the batch size set to 2, the smallest considered in the analyses. Although this behavior is not observed for the Pix2Pix model, since the model performed similarly for the three batch sizes tested, the value 2 can also be selected as the best one.

In the following analyses we only consider the models trained with batch size 2, as it results in the best performance for all models.

5.2 Mean IoU

Considering the configurations with the best performance according to the previous analyses, we compared the models' performances with regards to the mean IoU value obtained when evaluating them in the test set.

The Pix2Pix model outperforms all other methods, with a result about 30% higher than the second best (Table 2).

Model	Mean IoU		Inference Time (s)
	Mean	Standard Deviation	
Pix2Pix	0.90	0.073	0.183
FCN VGG16 32s	0.61	0.101	0.049
FCN VGG16 16s	0.58	0.091	0.055
SegNet	0.53	0.113	0.067
FCN DenseNet	0.52	0.082	0.054
FCN VGG16s 8s	0.42	0.062	0.059

Table 2: Mean IoU and Inference time per model.

5.3 Efficiency

The third type of analysis addresses model efficiency with respect to number of parameters. This type of evaluation is justified by the need for precise and robust models, when considering its integration in the embedded computer vision system of an autonomous mobile robot. In other words, considering the limited hardware resources of an autonomous system, we are looking for a model with good performance in terms of mean IoU value (precision) and which requires the lowest storage space.

Therefore, the higher the mean IoU value and the smaller the number of parameters, the more efficient the model. Equivalently, the lower mean IoU value and the greater the number of parameters, the less efficient the model.

Following this criteria, Fig. 10 presents Pix2Pix as the model with highest efficiency, followed by SegNet. The choice for the Pix2Pix model was due to its high mean IoU value, which compensates the greater number of parameters with respect to the SegNet model.

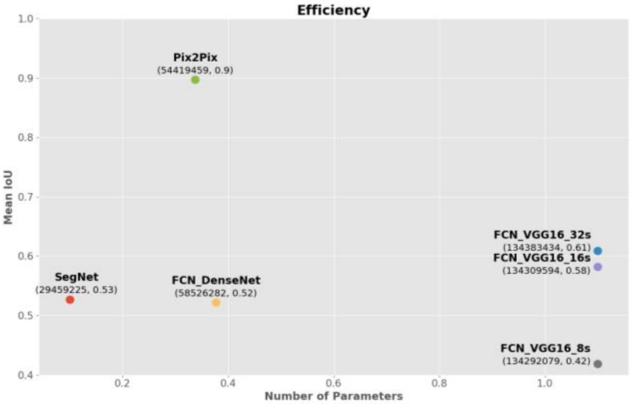

Figure 10: Efficiency of the models with respect to their mean IoU value and number of parameters.

5.4 Inference Time

Table 2 shows the performances of the models with respect to the inference time. As we can see, the best inference time was achieved by the FCN VGG16 32s model, which was approximately 26% faster than the Pix2Pix model (best mean IoU value).

5.5 Visual Inspection

The last type of analysis corresponds to a subjective assessment of the quality of the segmentations. Although being the simplest type of analysis, it provides a way to evaluate the performances in a more practical and intuitive manner. Therefore, it can be used both for an initial analysis of the models, in order to select the most suitable for the application in question, as well as for the validation of objective analysis.

Fig. 11 shows that, even though all the models achieved certain success in segmenting the safe zones, borders and unsafe zones, they struggled at segmenting the obstacles. The only model that achieved almost perfect performance was Pix2Pix, what validates our previous analyses. It smoothly labeled classes with big areas in the image, as well as delineated with precision the smallest objects' boundaries.

An interesting fact to note is that even in the initial training epochs of the Pix2Pix model ("Pix2Pix (2 ep)" in Fig. 11), the results obtained were already clearly superior in precision and quality of segmentation, when compared to the final results achieved by the other models.

6 CONCLUSION

In this paper, we have conducted a comparative study of different deep learning models for RGB-D indoor semantic image segmentation. First, we presented computer vision as an essential component of autonomous mobile robots, with utmost importance for their correct operation and as a means of safety guarantee for users and other people in the environment. We then introduced semantic segmentation as one of the most important levels of analysis provided by computer vision for scene understanding. We walked through the advances in methodology, from handcrafted feature analysis to deep learning, and data available, from 2D to RGB-D image datasets, for semantic segmentation study purposes. We then presented the main related comparative works.

Second, we provided a detailed explanation of the methodology adopted in this work. The full pipeline for data acquisition and annotation for the dataset creation, model architectures explanation and implementation details were presented.

Finally, we performed experiments comparing the models in the task of RGB-D deep semantic image segmentation. We first studied the influence of different batch sizes in performance. Then, we conducted a comparative evaluation of the performances according to both quantitative - precision (mean IoU), efficiency (mean IoU *versus* model size) and inference time – and qualitative – visual inspection – metrics. These metrics were chosen taking into account the fundamental concern with efficiency and reliability required from these models, for their correct and safe operation when incorporated in autonomous systems, preventing them to cause or to be involved in potentially fatal situations.

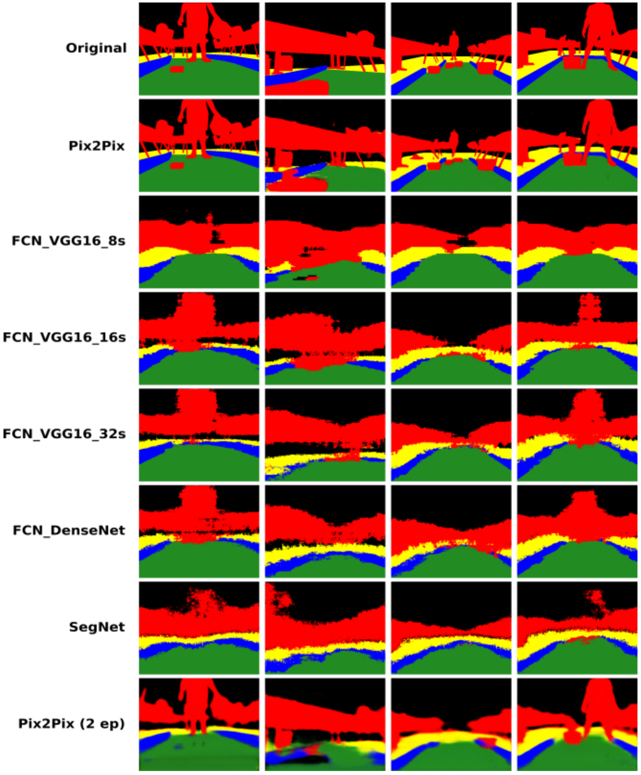

Figure 11: Visual inspection and comparison of the segmented images, evaluated in the test set. The last row corresponds to the results generated be the Pix2Pix model after two epochs of training. The other results were generated after fifty epochs.

The model with best results was the Pix2Pix, a GAN-based model. Although not providing the best inference time (Table 2), it was the one that best met the project's expectations in terms of precision (mean IoU, Table 2), efficiency (Fig. 10) and quality of segmentations (Fig. 11), outperforming the other methods by a large margin. Those characteristics configures it as the most suitable to be used as part of the vision system of an autonomous mobile robot.

A valid observation is that we used the Kinect sensor for demonstration and exploratory analysis purposes. Applying it to real autonomous robots navigation, or even more specifically, to autonomous driving systems, requires the use of more precise and, consequently, more expensive sensors, since the data quality is essential to a correct operation, improving safety for users. The LIDAR is an example of widely used 3D sensor in autonomous navigation.

As a future direction for this research, we could evaluate the influence of assigning different importance levels to different classes. For instance, the class person should be assigned more importance than the class sky.

In this work, we considered the segmentation of static images, assuming no relationship between them. However, since in a real scenario the process of segmentation is performed on an input video sequence, this work could be extended to consider the temporal correlation between the frames being processed.

Lastly, further analysis could also study the influence of pre-training the models in a separated RGB-D indoor dataset and then adapting them to the context of this work.

REFERENCES

[1] Phil McCausland. 2019. Self-driving Uber car that hit and killed woman did not recognize that pedestrians jaywalk – NBC. Retrieved November 9, 2019 from https://nbcnews.to/3ojSC1U

[2] BBC News. 2020. Tesla Autopilot crash driver 'was playing video game' – BBC. Retrieved February 26, 2020 from https://bbc.in/3qqPYcz

[3] Gary Bradski and Adrian Kaehler. 2008. Learning OpenCV: Computer Vision with the OpenCV Library (1st. ed.). O'Reilly, Cambridge, UK.

[4] Derrick Mwiti. 2019. Guide to Semantic Segmentation. Retrieved June 23, 2020 from https://bit.ly/2BQElr8

[5] Tingwu Wang. 2020. Semantic Segmentation. University of Toronto. Retrieved from https://bit.ly/3hNioZk

[6] A. Khan and R. Srisha. 2013. Image Segmentation Methods: A Comparative Study. International Journal of Soft Computing and Engineering (Jul. 2013).

[7] Jonathan Long, Evan Shelhamer, Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 7-12, 2015 Boston, MA. IEEE, 3431-3440. https://doi.org/10.1109/CVPR.2015.7298965

[8] Vijay Badrinarayanan, Alex Kendall and Roberto Cipolla. 2016. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. Retrieved from https://arxiv.org/abs/1511.00561 (v3)

[9] Çağrı Kaymak and Ayşegül Uçar. 2018. A Brief Survey and an Application of Semantic Image Segmentation for Autonomous Driving. Retrieved from https://arxiv.org/abs/1808.08413.

[10] Mennatullah Siam et al. 2020. RTSeg: Real-time Semantic Segmentation Comparative Study. Retrieved from https://arxiv.org/abs/1803.02758.

[11] M. Siam et al. 2017. Deep Semantic Segmentation for Automated Driving: Taxonomy, Roadmap and Challenges. https://arxiv.org/abs/1707.02432

[12] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. https://arxiv.org/abs/1409.1556.

[13] Olaf Ronneberger, Philipp Fischer and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. Retrieved from https://arxiv.org/abs/1505.04597

[14] Ian J. Goodfellow et al. 2014. Generative Adversarial Nets. arXiv:1406.2661. Retrieved from https://arxiv.org/abs/1406.2661

[15] Tero Karras, Samuli Laine and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. Retrieved from https://arxiv.org/abs/1812.04948

[16] Yunjey Choi et al. 2018. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. Retrieved from https://arxiv.org/abs/1711.09020

[17] Nasim Souly, Concetto Spampinato and Mubarak Shah. 2017. Semi Supervised Semantic Segmentation Using Generative Adversarial Network. 2017 IEEE International Conference on Computer Vision (ICCV), October, 22-29, 2017, Venice, Italy. IEEE, 5689-5697. https://doi.org/10.1109/ICCV.2017.606

[18] Jun-Yan Zhu et al. 2020. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Retrieved from https://arxiv.org/abs/1703.10593

[19] Phillip Isola et al. 2018. Image-to-Image Translation with Conditional Adversarial Networks. Retrieved from https://arxiv.org/abs/1611.07004 (v3)

[20] Tsung-Yi Lin et al. 2015. Microsoft COCO: Common Objects in Context. Retrieved from https://arxiv.org/abs/1405.0312

[21] Mark Everingham et al. 2010. The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision 88 (Jun. 2010), 303–338. DOI: https://doi.org/10.1007/s11263-009-0275-4

[22] Marius Cordts. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. Retrieved from https://arxiv.org/abs/1604.01685 (v2)

[23] German Ros et al. 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27 – 30, 2016, Las Vegas, NV. IEEE, 3234-3243.

[24] Microsoft. 2020. Kinect para Windows. Retrieved from https://bit.ly/3g54O23

[25] Alberto Garcia-Garcia et al. 2017. A Review on Deep Learning Techniques Applied to Semantic Segmentation. Retrieved https://arxiv.org/abs/1704.06857

[26] Shuran Song et al. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015), June 7 – 12, 2015, Boston, MA. IEEE, 567-576.

[27] Nathan Silberman et al. 2012. Indoor Segmentation and Support Inference from RGBD Images. In Computer Vision – ECCV 2012, Oct. 7 – 13, 2012, Firenze, Italy. Springer, Berlin.. https://doi.org/10.1007/978-3-642-33715-4_54

[28] Wilfrido Gómez-Flores et al. 2020. A comparative study of pre-trained convolutional neural networks for semantic segmentation of breast tumors in ultrasound. Computers in Biology and Medicine, Vol. 126. https://doi.org/10.1016/j.compbiomed.2020.104036

[29] Parham Yazdekhasty et al. 2020. Bifurcated Autoencoder for Segmentation of COVID-19 Infected Regions in CT Images. https://arxiv.org/abs/2011.00631

[30] Caio César Teodoro Mendes et al. 2016. Exploiting fully convolutional neural networks for fast road detection. IEEE International Conference on Robotics and Automation (ICRA), May 16 – 20, 2016, Stockholm, Sweden. IEEE, 3174-3179. https://doi.org/10.1109/ICRA.2016.7487486

[31] Daniela A. Ridel et al. 2015. Obstacle segmentation with low-density disparity maps. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015), Sep. 28 – Oct. 2, 2015, Hamburg, Germany.

[32] Tsung-Yi Lin et al. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312v3. Retrieved from https://arxiv.org/abs/1405.0312

[33] Roozbeh Mottaghi et al. 2014. The Role of Context for Object Detection and Semantic Segmentation in the Wild. IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH. IEEE, 891-898. DOI: https://doi.org/10.1109/CVPR.2014.119

[34] Di Feng et al. 2020. Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. arXiv:1902.07830v4. Retrieved from https://arxiv.org/abs/1902.07830

[35] Andrew G. Howard et al. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. https://arxiv.org/abs/1704.04861

[36] F. Fooladgar and S. Kasaei. 2020. A survey on indoor RGB-D semantic segmentation: from hand-crafted features to deep convolutional neural networks. Multimedia Tools and App. 79. DOI: 10.1007/s11042-019-7684-3

[37] Seichter et al. 2020. Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis. Retrieved from https://arxiv.org/abs/2011.06961

[38] J. Jang et al. 2018. RedNet: Residual Encoder-Decoder Network for indoor RGB-D Semantic Segmentation. Retrieved: https://arxiv.org/abs/1806.01054

[39] Nathan Silberman and Rob Fergus. 2011. Indoor scene segmentation using a structured light sensor. 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). Nov. 6-13, 2011, Barcelona, Spain. IEEE, 601-608. DOI: https://doi.org/10.1109/ICCVW.2011.6130298

[40] Iro Armeni et al. 2017. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. Retrieved from https://arxiv.org/abs/1702.01105 (v2)

[41] Michal Uricar et al. 2020. Yes, we GAN: Applying Adversarial Techniques for Autonomous Driving. Retrieved from https://arxiv.org/abs/1902.03442 (v2)

[42] Pauline Luc et al. 2016. Semantic Segmentation using Adversarial Networks. arXiv:1611.08408. Retrieved from https://arxiv.org/abs/1611.08408

[43] Zhongyi Han et al. 2018. Spine-GAN: Semantic Segmentation of Multiple Spinal Structures. Medical Image Analysis 50. DOI: 10.1016/j.media.2018.08.005

[44] Kentaro Wada. 2016. labelme: Image Polygonal Annotation with Python. Retrieved from https://bit.ly/3l3brWy

[45] Python. 2020. Welcome to Python.org. Retrieved from https://www.python.org/

[46] Tensorflow. 2020. Uma plataforma complete de código aberto para machine learning. Retrieved from https://www.tensorflow.org/

[47] Tensorflow Core. 2020. Keras. Retrieved from https://bit.ly/37tUdvo

[48] Gao Huang et al. 2018. Densely Connected Convolutional Networks. arXiv:1608.06993v5. Retrieved from https://arxiv.org/abs/1608.06993

[49] Tensorflow Core v2.3.0. 2020. Keras Applications: VGG16. Retrieved from https://bit.ly/345I6EM

[50] Tensorflow Core v2.3.0. 2020. Keras Applications: DenseNet121. Retrieved from https://bit.ly/2Yat2Sv

[51] Tensorflow Core. 2020. Pix2Pix. Retrieved from https://bit.ly/2I7yIYy

[52] Ekin Tiu. 2019. Metrics to Evaluate your Semantic Segmentation Model. Retrieved from https://bit.ly/3h8GKwO