Ensemble-based Sidewalks detection from street view images

Ibne Farabi Shihab
Sakib Ferdous

Team Number 15

April 28, 2022

Final Project Report for
HCI575: Computational Perception

Iowa State University
Spring 2022

ABSTRACT

Autonomous driving is the next big thing in the world. Though the demand for autonomous vehicles are increasing rapidly,the safety aspect of it is not well focused.In this work, we focused on detecting sidewalks from an image, a small part of the safety measure that need to be taken. This sidewalk identification is important for a vehicle to avoid any haphazard events like injuring pedestrians, property damage and also at the same time will make the free movements of pedestrians easy. As of the goal, we aim to identify sidewalks with higher miou score using ensemble of the state-of-the-art object detection models Using ensemble we were able to achieve 93.1%,90.3% 90.6% respectively on Cityscapes dataset, Ade20k dataset and Boston datasets.

1 Introduction

During the inception of the deep learning era, deep learning could not flourish to its expected level. Two particular aspects were instrumental behind its cause- the scarcity of data and the other is the lack of computational resources [1]. Nevertheless, due to the rampant advancement in computational resources nowadays, deep learning started to rise again, making it one of the buzzwords of the modern era. In addition, models based on deep learning started to suppress human-level accuracy in recent days. Deep Blue, Alpha Go, and Google's inception model are some examples [2, 3, 4]. These very deep learning-based approaches inspire us to develop innovative ideas, which is happening because deep learning-based approaches surmounted existing machine learning and statistical-based approaches. Autonomous driving is one of the revolutionary prospects that came to light after the revival of deep learning. Companies like Tesla have already come up with autonomous vehicles. Nowadays, it is not a matter of whether autonomous vehicles are a possibility; rather, it is high time to consider how to take this idea further. Autonomous vehicles have few aspects of dealing with. For example, the level of autonomy, additivity, Lidar to infer the surroundings, safety, etc. Among these aspects of autonomous vehicles, we are concerned with the aspect of safety, more specifically, human safety. The first step toward autonomous vehicles is that a vehicle will have a notion of safety. To achieve this particular objective, a vehicle needs to know where and distinguish between a road and sidewalks. To comply with this idea, we will be working on finding sidewalks from an image in our work. More specifically, we will focus on detecting sidewalks from the road, a small subset of the safety aspect. Having the sense of where the sidewalks are important for two reasons. One is to eliminate the risk of accidents, making it safe for pedestrians to move on the sidewalks easily. The second one is to make sure that the vehicle can function safely (avoiding accidents and property damage) even in the absence of the internet or reduced access to the internet.

To the best of our knowledge, there have been few works based on traditional techniques and deep learning for sidewalks detection. Classical computer vision-based techniques have been used where affine transformation followed by a dynamic contour model worked best [5, 6]. However, this work will not suit sidewalk detection due to 80% accuracy, where safety is a top priority. Later, people switched to deep learning-based approaches, which will be discussed in section 2. Our work

will combine the image segmentation technique followed by the object detection technique to detect where sidewalks are in an image. The related work section will discuss the detailed reasoning for why we combined both techniques. This project is intended for people interested in object detection and eager to work on sidewalks or even curb detection from an image.

The rest of the paper will be as follows where section 2 will represent work related to our project; section III will describe the experimental platform, section 4 will describe the working procedure while the result will be discussed in section 5 we will conclude with discussion followed by conclusion and future works.

2 Related Work

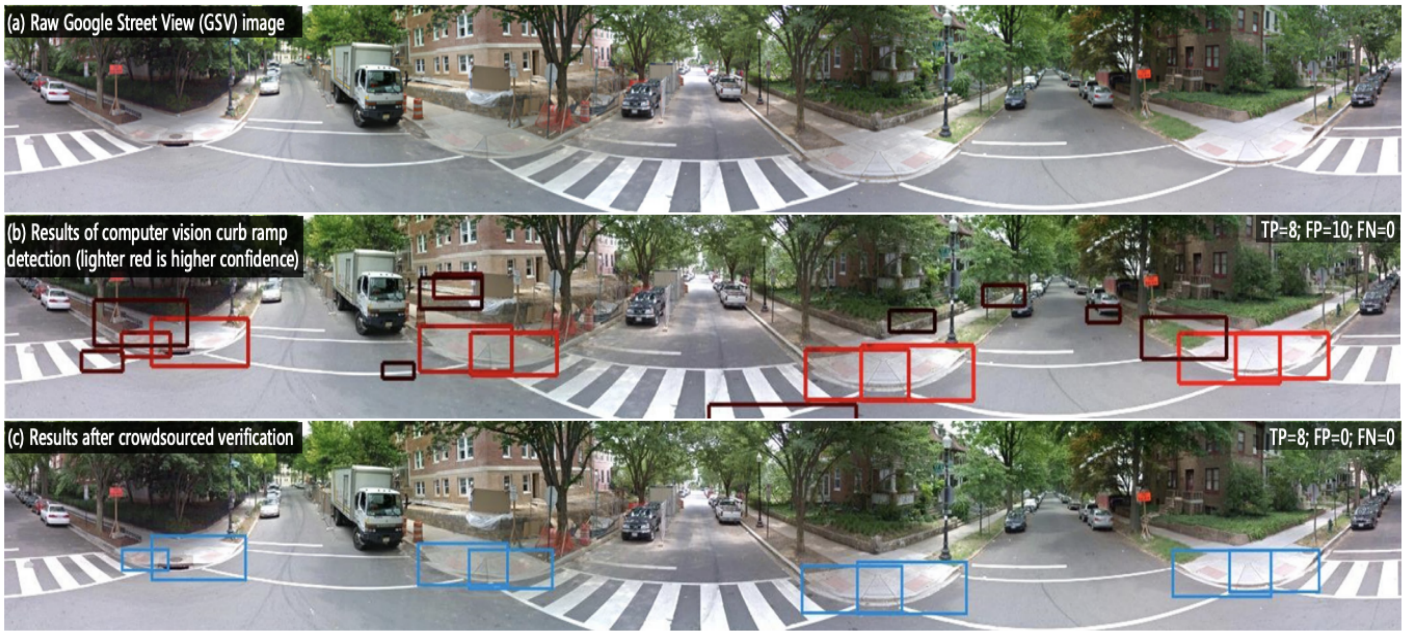

Figure 1: In this paper, we present *Tohme*, a scalable system for semi-automatically finding curb ramps in Google Streetview (GSV) panoramic imagery using computer vision, machine learning, and crowdsourcing. The images above show an actual result from our evaluation.

Figure 1: A snapshot of work from [7]

Object detection deals with obtaining the category and bounding box of an object, while semantic segmentation is the objects according to their categories. Several approaches have been taken to study sidewalk detection (not exactly) based on object detection and image segmentation. We will start with object detection. From a machine learning perspective, Support Vector Machine (SVM) and B, ag of words are applied to imagery analysis [7], where results are not promising.

However, in recent days deep learning started to suppress other approaches for which it has been the point of scholarly attention across many aerial-based object detection tasks [8, 9]. Few works have been done on sidewalk detection though they are more involved in finding the accessibility [19]. Among those two of the most notable ones is the work by Chang et al. In this work , they used a deformable part model (DPM) for detecting curbs ramp [10]. However, in most cases, DPM fails to localize objects with multiple objects [Figure 1]. Another research by Sun and Jacob addressed the challenge differently - Instead of finding what is in the image, they tried to find what is missing in the image, which is curb ramps in this case. They used a fully convolutional network integrated with the Siamese network for the task. In addition, they combined machine learning and CV techniques where they created a context map to take account of the missing curb ramps and later added it with object detection to search for the missing region [figure 2]. Like the previous work, the 27% recall score is also a concern. In all of the works, they used four datasets for bench-markings. This dataset will be kept addressed as we go along. However, the details use of the two datasets in our work will be explained in the dataset sections. To start with, Segnet [11] is the first one of a few models that are closed to our research. In this paper, they worked with encoder-decoder to do segmentation on pixel-based though their miou score [12] on sidewalk detection is around 80% on kiti dataset [13], which we did not use. A more improved version of Segnet is DeepLab [14]. This paper used" atrous convolution" and "atrous spatial pyramid pooling" for robust segmentation of images with higher control and multiple scales. Their approach also helped to perform localization more efficiently than the current state of the art at that time. Their work could take the miou [12] to 82% on cityscapes data [15], the dataset we used for our work. This dataset, along with ade20k [16], will be discussed in the next section. DeeLabv3 is a step further to improving the miou score of their previous version DeepLab [17]. In this paper, they bench-marked on PascalVoc 2012 with Segnet, and DeepLabV3, and their version show that the miou goes to around 87 % sidewalk detection on cityscapes data [15]. In addition, the work by Weld et al. gave the best accuracy of 92 %. This worked with aerial view images and used street views as supplementary images. As far as the concerns of the network, they did not propose any new model. Rather they used YOLACT [21] and PSPNet [22] for aerial and street view images, respectively. By following their process, they could achieve the mentioned accuracy. Lastly, Hierarchical multi-scale attention for semantic segmentation (HMASS) gave the state of art

result of 89.38%(miou score) for sidewalk detection [23]. In their work, they used HRNet-OCR with multi-scale attention to achieve that.

They always tried to train a single model in all of the mentioned works. It is tricky to detect where the sidewalks are as depending on the place, the structure of sidewalks is a bit different. Based on this idea, it is obvious to say a particular model cannot capture this difference based on places, and it is because we know how vulnerable deep models are to a slight change and adversarial attacks are one of the examples [27]. To overcome those shortcomings, we are proposing an ensemble-based approach. Ensemble is an approach known to work better than a single learner as we can aggregate answers from different models in one place. Our sidewalk detection project is technically semantic segmentation work- a more advanced version of object detection. It is important to notice that the ensemble does not solve the adversarial attack problem. Rather it aggregates the result based on outputs from models. Usually, the aggregation happens based on what the majority of the models are saying about the class of an image (For example, if it is a dog or cat) [25]. This ensembling aspect has been implemented in sidewalk detection. In our work, the ensemble technique will be used. We will pick the best three models from the state of the art, and a voting classifier will be used to aggregate the result [28]. The main differences between their approaches and ours are that we will be using google earth images with an ensemble of state-of-art semantic segmentation models, which is missing in the mentioned works.

3 Experimental Platform

In this section, we will give a detailed description of our data source, what platform we will use, feature selection, and models that we will use.

Data always lies at the heart of any machine learning model. For the project we are aiming at, the data from google earth seems to be a suitable match [29]. We need to check out the topography in this data as we don't need a close view of the image. However, we need the data to be annotated. Manually annotating the data is a cumbersome job. Because of this, we used the dataset of Cityscapes and Ade20k (detail in the related work section). We bench-marked our ensemble model on these two datasets. To increase our data source and add more variety, we used the data used by Seong and Jaewan [30]. This means that we will be

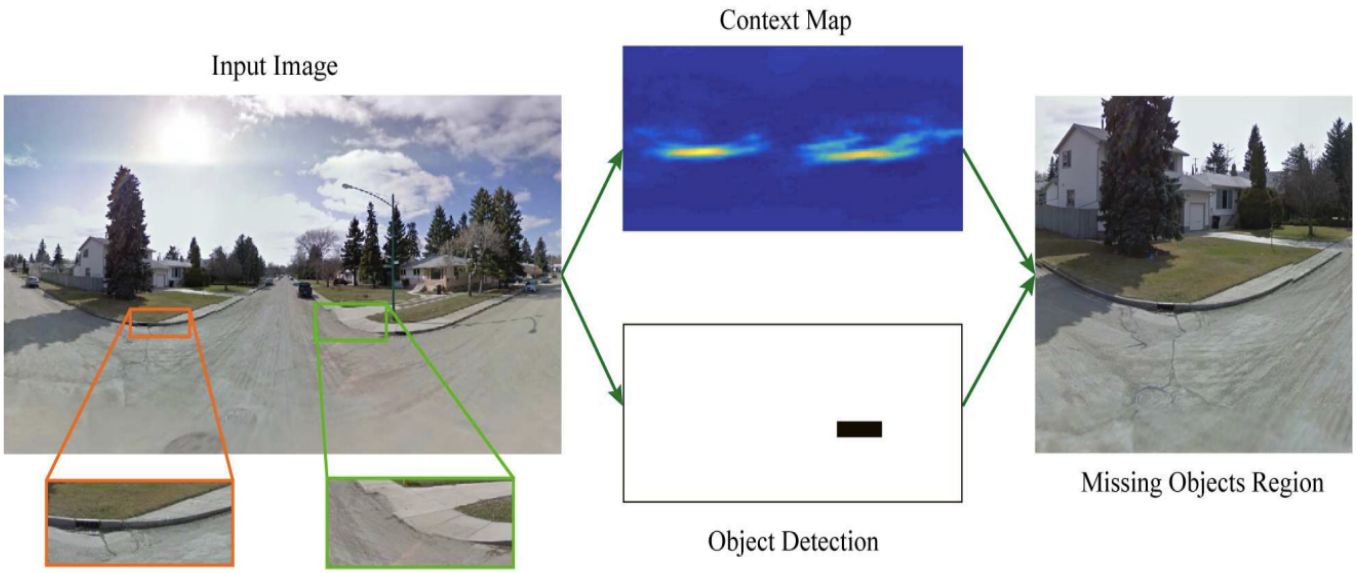

Figure 1: When curb ramps (green rectangle) are missing from a segment of sidewalks in an intersection (orange rectangle), people with mobility impairments are unable to cross the street. We propose an approach to determine where objects are missing by learning a context model so that it can be combined with object detection results.

Figure 2: A snapshot of work from [19]

bench-marking our approaches on three datasets.

3.1 Computational Resource

Training deep models is time consuming and expensive in terms of resources. to overcome this issues, we used Lamda stack server [32] build with 24gb Nvidia 3090 graphics and 12 cores. Using this server significantly reduced our experimental time and gave us scope explore different types of models.

3.2 Description and Preprocessing of Datasets

Before entering the ensemble model, the aforementioned dataset needs to get through some preprocessing steps. The Cityscapes dataset consists of images of the road from fifty different cities. It has 5000 annotated images which are enough for this project. From there, 2100 are selected. From this 2100, 20% of the data is used for testing, while the other 80% has been used for training purposes. The ADE20k dataset contains around 25000 images from which 3000 images are selected where there are sidewalks. The number of images is selected, bearing in

Dataset name	Total images	Training images	Testing images
Cityscapes	2100	1680	420
Ade20k	3000	2400	600
Boston Street	2000	1600	400

Table 1: Distribution of datasets

mind to keep a balanced comparison with our Cityscapes dataset.

Similar to Cityscapes, 80% of the images are taken for training and the rest for testing. Similarly, we extracted 2000 images from the Boston Street dataset and did a split of 80% and 20% for training and testing, respectively. One important thing to notice here is that we took different images from different datasets. Due to the exhaustive experiments, we found that this amount of images works best for our ensemble model. For a better view, the percentage and number of images from these three datasets have been shown in 1. In addition to that, an example from each dataset has been given in Figure [3,4,5]

Considering the availability of a wide variety of frameworks, the python programming language is used.

Pytorch will be used as the deep learning framework considering its applicability in the task, availability of necessary libraries, and our general ease and comfortability in using it [31].

Before diving into the training process, it is important to describe data processing and the type of training used (training from scratch or transfer learning). The transfer learning approach [33] is preferred because deep learning models are time-consuming. Another justification for doing that is the availability of fine-tuned pre-trained models. The input images need to be in the shape of that pre-trained model for using transfer learning. Thus, the shape of the input, images differed

Figure 3: An example from cityscapes dataset

depending on the type of models being used. Three different input shapes are used for the three models discussed here. We wrote a python script according to the model, and we used the transfer learning approach by training the last few layers of our three models. The initial weights are kept as it is in the pre-trained models. The input size according to the models is given in Table 2. However, it is important to notice that this input size is for the initial network (or first network, as there are different variations of them). Different sizes for input have also been used, which will be discussed in the methodology section.

4 Methodology

The most important thing is the deep learning network itself. Initially, we have a few options in mind. They are discussed briefly-

- Based on the problem specified, object detections are considered a suitable choice. For that, we considered using YOLO [34]. The idea was to draw

Figure 4: An example from ade20k dataset

bounding boxes around the sidewalks.

- We also explored the idea of using Faster-RCN [35] due to popular demand for it in deep learning and computer vision. This model is based on finding a region; in other words, this model proposes a region.
- The last one was the Single Shot multi-box Detector [36]-this model is a box detector that aligns with the end goal.

An example from YOLO and RCNN is given in Figures [7,8]. In addition, a pictorial of this architecture is shown in Figure 6. We realized that drawing the bounding box would not be of any help from these two images. There are two reasons for that. One is that vehicles need to know the exact location of the sidewalks to be on the safe side. As we were drawing bounding boxes, it did not portray the exact shape of the sidewalks. Another reason is that we plan to extend it for curb finding. Knowing the exact location of the curb is much more important than just drawing a bounding box. For the mentioned reasons, we went for a more

Figure 5: An example from boston dataset

precise option: segmentation, where we can tell where the object is rather than just vaguely drawing a bounding box. Make no mistake thinking that we are just segmenting images instead of detecting. Image segmentation is a more precise version of object detection. A pictorial view of the updated architecture is shown in Figure 9.

To start with, we did the preprocessing, which we described in the Experimental Platform section. We needed to choose a backbone network for all of these three networks [37]. This is a trial and error process to determine which backbone network works well. More detail about the backbone network will be described in the following paragraph. It is almost obvious for most deep learning-based works that the model is not built from scratch. There are a few reasons behind that. One is the amount of time we need to do it on our machine. Another reason is that we might not have the computational power needed to do those high-end training. Lastly, the most important point is that deep learning is not all about training. Rather, it is more about doing the training in the right way with perfect

Dataset name	Input size
YOLACT	550x550
HMASS	600x600
DeepLabV3	650x650

Table 2: Input size of three models

fine-tuning exhaustive hyperparameters search. Because of the mentioned reason, we went for the transfer learning as we mentioned before. In transfer learning, we take the dataset from our task and do some fine-tuning on the last few layers with the help of our training dataset. In this way, we reduced the need for a lot of data, and that is why there is not much data in our mentioned datasets given in Table 1. At the same time, we do not need many resources for computational purposes while we are getting a well-trained model from experts., The training is done on the last few layers. The model adapts to our datasets, which helps get rid of training from scratch. However, as said before, we need backbone networks. Those backbone networks will be pre-trained on our datasets(Cityscapes, Ade20k, Boston Dataset. From now on, when we say datasets, we will be pointing to these three datasets). By exhaustive trial and error, for HAMM and DeepLabV3, we found four backbone networks that work well on our datasets. To be more precise, we used Resnet-50 for our HAMM as backbone networks and Xception, Xception-JFT, and Resnet-101 for DeepLabV3. One might ask why there is only one backbone network for HAMM, and it is because we empirically found that this is the only backbone that works well for our datasets. Throughout the discussion, we did not mention anything about YOLACT because we found it hard to look for a good backbone model for it. By going through the paper on YOLACT, we found that four versions of backbone networks have been used to establish their result. Due to time constraints and the exhaustive process that will take to find the best suitable models, we used those mentioned three backbone models, and they are R-101-FPN, R-50-FPN, and D-53-FPN. As these models were not pretrained on our datasets, we took pre-trained models on our datasets from the detectron2 GitHub repository [38]. Here R stands for Resnet [39], and FPN stands for feature pyramid network [40], while D stands for darknet [41]. The numbers 50,101, and 53 stands for how many layers are there. In the case of YOLACT, the

results did vary depending on the size of the images. Empirically, the best results using YOLACT are found for image sizes of 400x400,550x550, and 700x700. In addition to that, we also tried the mentioned three backbones for each image size, and we picked the best five models, which will be discussed in the result section.

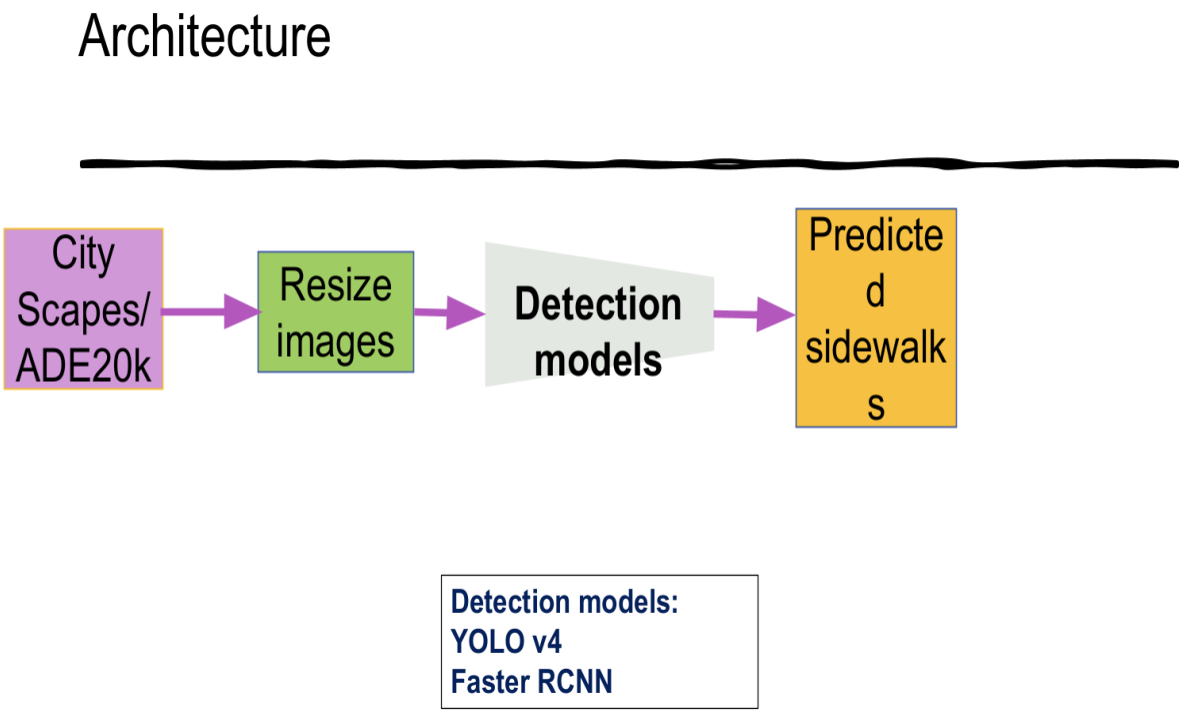

Figure 6: An example of bounding box using YOLO from cityScapes dataset

5 Results

As we used three datasets, it will be a viable option to split them into experiments as follow.

5.1 Experiment on Cityscapes

We first started our experiments with the Cityscapes dataset. Here we tried different image sizes with a few different backbone networks. However, we did not have that much success except using the Resnet-50 backbone for HAMM. In the case of DeepLabv3, we were able to find more than one backbone network that works well: Xception, Xception-JFT, and Resnet-101 [39, 42]. For YOLACT, we picked the five best models that work best for our cityscapes dataset. The process of picking these models is described in the methodology section. Following the mentioned process, we got nine models, and for those, the mIOU score has been reported in Table 3. The next step is

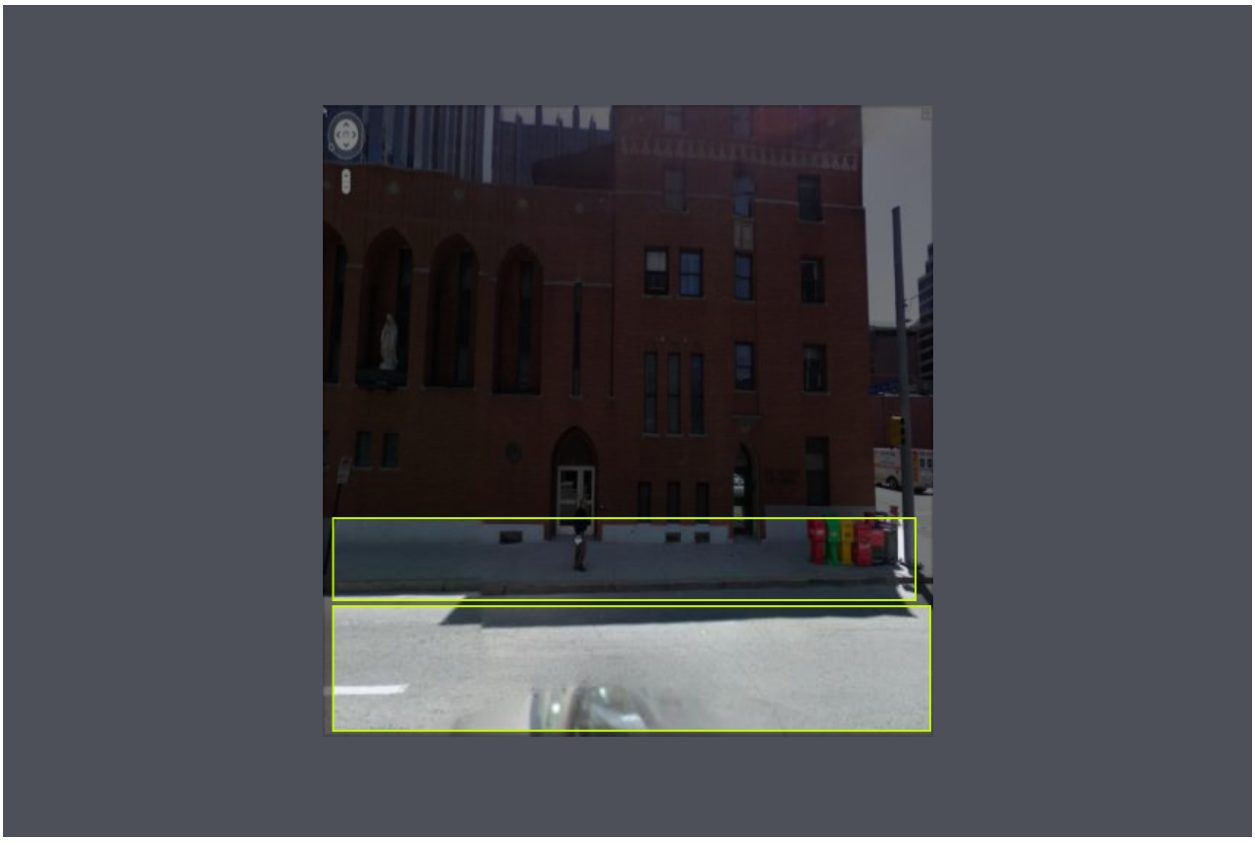

Figure 7: An example of bounding box using YOLO from cityScapes dataset

to see if we can increase the mIOU score of the models on the Cityscapes dataset. For this, we tried to do an ensemble. We picked the best models from the nine models we have based on the mIOU score. Here the higher, the better. For this, we selected YOLACT700, HAMM, and DeepLabV3 with Resnet-50. One might wonder why we left out the DeepLabV3 model even though it has more mIOU score. It is because the idea of the ensemble was to get diversity among the three models. So, taking two DeepLabV3 models would hamper the idea of an ensemble. The mIOU score of 93% makes our stand stronger. This 93.1% is the best state-of-the-art score on the Cityscapes dataset containing sidewalks. The score has been reported in Table 4. An example of segmentation after ensemble is shown in Figure 11, while Figure 10 is the original one.

5.2 Experiment on Ade20k

This subsection will discuss the model's result on the Ade20k dataset. In terms of models, we kept the same models that were for the Cityscapes dataset. In these experiments on the Ade20k dataset, YALACT550 with R-50-FPN backbone, HAMM, and DeepLabV3 with Xception-JFT backbone worked best in terms of mIOU score, which has been reported in Table 5. Here, Figure 11 is the output of the ensemble-based model, and Figure 10 is the original one. Following the same way as Cityscapes, we did an ensemble of the best models and got an mIOU score of 90.3%, a state-of-art result on Ade20k datasets for sidewalks detection. The score is shown in Table 6. Following the

Network name(input size)	Backbone	MIOU Score(%)
YOLACT400	R-101-FPN	82.2
YOLACT550	R-101-FPN	84.3
YOLACT550	R-50-FPN	86.5
YOLACT550	D-53-FPN	83.4
YOLACT700	R-101-FPN	87.2
HAMM	Resnet-50	89.2
DeepLabV3	Xception	87.8
DeepLabV3	Xception-JFT	89.0
DeepLabV3	Resnet-101	75.6

Table 3: Performance of Cityscapes dataset on different model

Network name(input size)	Backbone	MIOU Score(%)	Ensembled Score(%)
YOLACT700	R-101-FPN	87.2	
HAMM	Resnet-50	89.2	93.1
DeepLabV3	Xception-JFT	89.0	

Table 4: Performance of Cityscapes dataset on different model

Network name(input size)	Backbone	MIOU Score(%)
YOLACT400	R-101-FPN	81.3
YOLACT550	R-101-FPN	84.7
YOLACT550	R-50-FPN	85.1
YOLACT550	D-53-FPN	84.4
YOLACT700	R-101-FPN	82.5
HAMM	Resnet-50	88.1
DeepLabV3	Xception	87.8
DeepLabV3	Xception-JFT	88.1
DeepLabV3	Resnet-101	79.2

Table 5: Performance of Ade20k dataset on different model

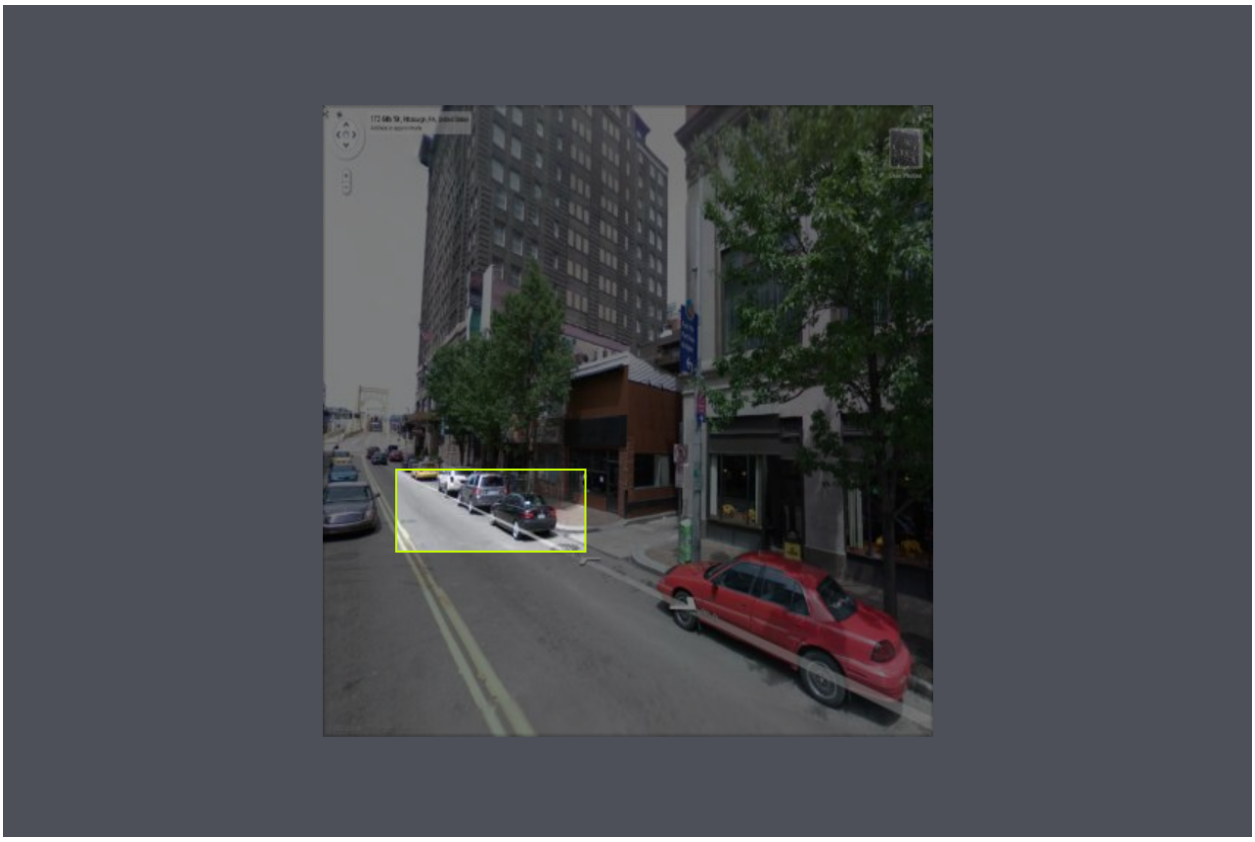

Figure 8: An example of bounding box using Faster-RCN from cityScapes dataset

Network name(input size)	Backbone	MIOU Score(%)	Ensembled Score(%)
YOLACT550	R-50-FPN	85.1	
HAMM	Resnet-50	88.1	90.3
DeepLabV3	Xception-JFT	88.1	

Table 6: Performance of Ade20k dataset on different model

same way as Cityscapes, we did an ensemble of the best models and got an mIOU score of 90.3% which is also a state-of-art result on Ade20k datasets for sidewalks detection. The score has been shown in Table 6.

5.3 Experiment on Boston dataset

Lastly, we used our nine models on Boston Dataset and reported the mIOU score for every model in Table 7. From the table, we can see that YOLACT550 with a backbone network named R-50-FPN, HAMM, and DeepLabV3 with Xception-JFT backbone did the best in terms of mIOU score. Hence, we picked these three models for the next round, which is ensemble. Following the previous two subsections, we built an ensemble of the best models using our voting classifier and got an mIOU score of 90.6%, which is a state-of-art result on Boston datasets for sidewalks detection. The score is shown in Table 8.

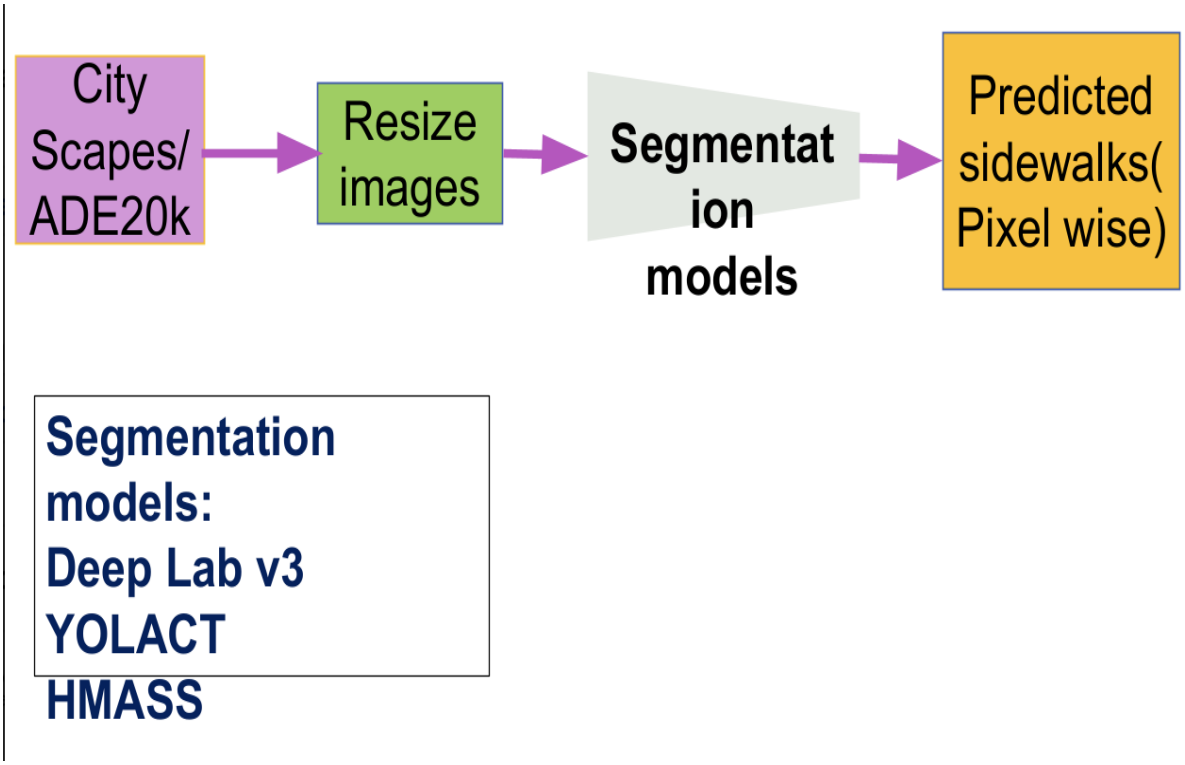

Figure 9: Pictorial view of work

This concludes the result section where we show our experimental results, and we could imitate the almost exact detection of sidewalks.

6 Discussion

This segmentation-based approach helped us to detect sidewalks as polygons instead of a rectangular box, and thus we concluded our work as we went to our goal that we had set initially.

We were able to achieve our goal of detecting sidewalks with higher accuracy though not the way we thought initially due to our lack of knowledge at that time. However, going through the literature did help us to achieve our goal. As we mentioned before, we could detect sidewalks using object detection models, which we considered a failure as they did not detect the sidewalks as a polygon. Instead, we could draw polygons using a semantic image segmentation approach. One fix that might be possible for YOLO, faster RCN is to give them the ability to draw polygon instead of just limiting them to just drawing bounding boxes. Due to the time limitation and the fact that we needed to train all of the models from scratch, we did not go in that way. Rather, we went for a different approach.

Figure 10: Pictorial view of work

7 Conclusion and Future Work

We started with the goal of ensuring the safety aspect of autonomous vehicles. Keeping the big picture in mind, we picked a small subproblem which was detecting sidewalks. Initially, due to lack of knowledge, we thought that object detection models like YOLO and faster RCN would work better as they helped detect objects from an image. However, after implementing them, we realized that it was not something we were looking for. It is because object detection models draw a bounding box around the objects. Thus, YOLO and faster RCN drew a bounding box around the sidewalks. But the problem with sidewalks from the images is that they do not have a particular shape, and hence they did not serve our purpose. The result is shown in the methodology section to justify this. Just detecting the shape of the sidewalks or roughly where it did not help us to reach our project goal. The reason is that we need to know the exact shape of the sidewalks. The project's objective was to detect sidewalks to ensure the safety aspect. Thus, after going through the literature, we learned that semantic image segmentation was something that we were looking for. There are a lot of segmentation models out there from which we need to pick our models. Later, we did some exhaustive trial and error to find out that YOLACT, HAMM, and DeepLabV3 are three models that performed well in our three datasets named Cityscapes, Ade20k, and Boson Dataset. While going through this exhaustive process, we also found that the size of images plays an important role in getting better performance. We tried to do this image resizing in all of our models. However, it turned out that image resizing works better only for YOLACT, and we tried a few models based on image size. In addition, it came to our observation that backbone networks

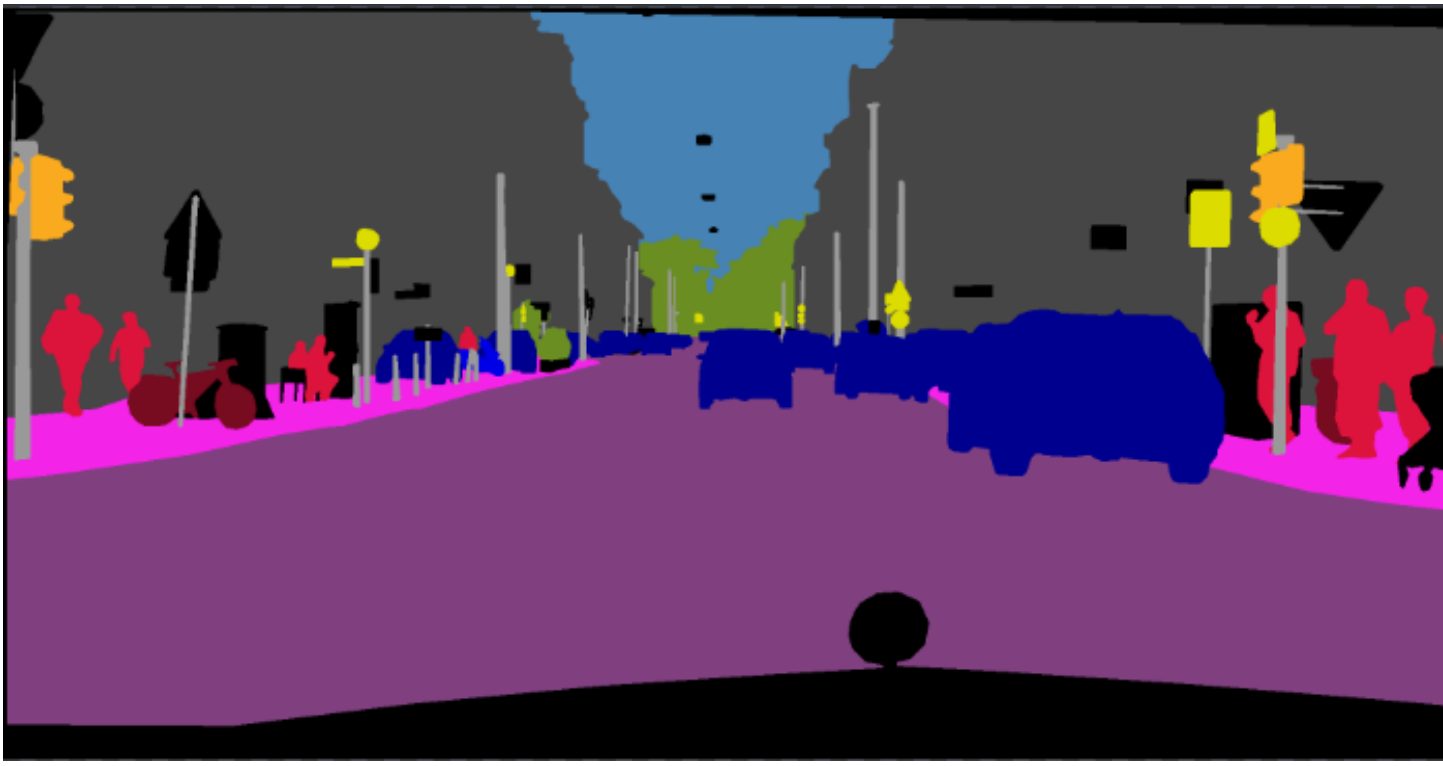

Figure 11: Pictorial view of work

also play a good role in achieving higher mIOU scores. After doing all of the mentioned processes, we picked nine better models for sidewalk detection. We trained 9 nine models for each of these three datasets and reported the mIOU score for each model for every dataset. We were fortunate enough to get pretrained backbone models for all of our nine models, and via transfer learning, we achieved a good score. In the case of training, we just trained the last few layers with our training set so that the models could adapt to our datasets. We used 2100 images for Cityscapes, 3000 images for Ade20k, and 2000 images for Boston Datasets from which we hold out 20% data for our testing purpose, and from that 80% training data, we used 10% data for validation purposes while training the models. However, these nine models could not go beyond the 90% mIOU score, and hence it turned out to be a matter of concern as we are dealing with the safety aspect. Thus we needed to come up with a better approach. We looked at state-of-the-art and found that ensemble might be a good way to improve our accuracy. Taking all the nine models for doing ensembling did not look logical and a feasible idea due to a good amount of varieties in terms of mIOU score. Also, computational resources were limited, and it would be too much in terms of computation. Thus, we decided to take one model each from YOLACT, HAMM, and DeepLabV3. To keep up with this analogy, we did not take two models from there for the Cityscapes dataset even though two models performed well. It is because of staying with the idea of having one model each from YOLACT, HAMM, and DeepLabV3. In doing so, we achieved state-of-the-art results for all three datasets by detecting sidewalks with more than 90% accuracy. In terms of future work, a few things are possible from here. All these years, the work that has been done is always based on segmentation techniques, region proposals, and so on. Nowadays, language modeling is revolutionary. There are few deep learning techniques based on language

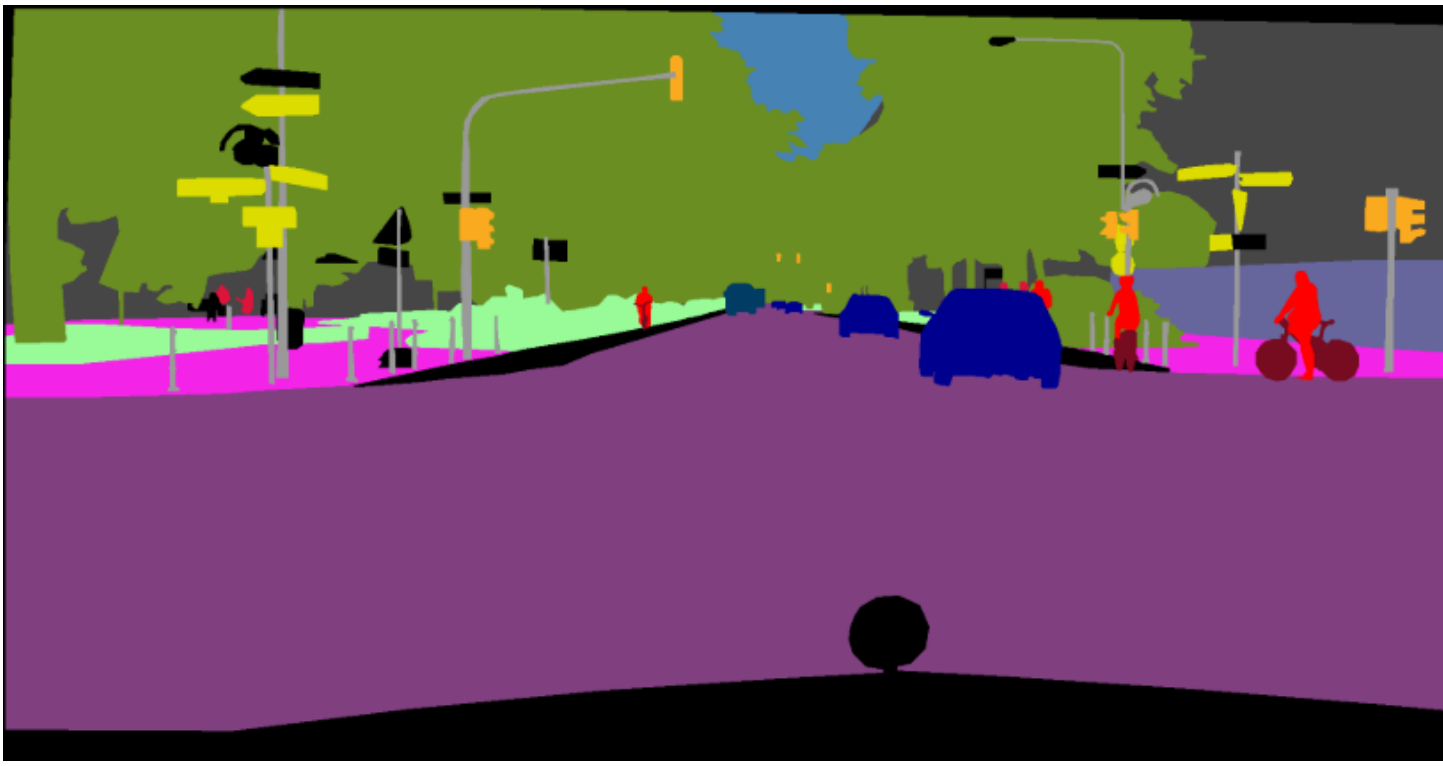

Figure 12: Pictorial view of work

modeling approaches. Machine translation, speech recognition, and text categorization have been done based on language modeling approaches [43, 44]. Recently, a paper named Pix2seq did the same [45]. They considered object detection as a language modeling task. It will be a good idea to use that model and see if it detects sidewalks more accurately. Another option can be training a new backbone network. Somehow, the backbone networks are confined within Resnet, Darknet, VGG, or attention mechanism-based [39, 41, 46, 47]. It is high time to improve them. Without any doubt, attention is the best technique we have in recent days. However, this attention needs to split over the different regions of an image as multiple objects or an object can have multiple features to be noticed. ResNet followed the same kind of approach [48]. It has shown great promises to be a top-notch backbone network. Lastly, an obvious solution is an increase in the quality of data. The data are mostly from the same place apart from Cityscapes if we look closely at the data. Object detection models need data with clear resolution though there are models that work in low lights. However, those models do not work exceptionally, and we cannot take the risk of using models with low accuracy where safety is our concern. Though the Cityscapes dataset has a wide variety of data from 50 cities, it has a major flaw in data quality. There are images with low lights that look alike roads and sidewalks, which is visible from the example image that we have shown in the experimental platform section. This kind of flaw confuses the model and leads to a low score.

Figure 13: Pictorial view of work

Network name(input size)	Backbone	MIOU Score(%)
YOLACT400	R-101-FPN	81.2
YOLACT550	R-101-FPN	84.8
YOLACT550	R-50-FPN	86.5
YOLACT550	D-53-FPN	81.7
YOLACT700	R-101-FPN	82.9
HAMM	Resnet-50	88.5
DeepLabV3	Xception	88.3
DeepLabV3	Xception-JFT	89.2
DeepLabV3	Resnet-101	80.2

Table 7: Performance of Boston dataset on different model

Network name(input size)	Backbone	MIOU Score(%)	Ensembled Score(%)
YOLACT550	R-50-FPN	86.5	
HAMM	Resnet-50	88.5	90.6
DeepLabV3	Xception-JFT	89.2	

Table 8: Performance of Cityscapes dataset on different model

REFERENCES

[1] Anyoha R. (2017). The History of Artificial Intelligence. Blog, Special edition on artificial intelligence. doi:https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/

[2] Deep Blue.(2011) https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/. Accessed Apr. 28, 2022.

[3] Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489 (2016). https://doi.org/10.1038/nature16961

[4] Szegedy, Christian, et al.(2015). Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition.

[5] Kayama, K., Yairi, I. E., & Igi, S. 2007. Detection of sidewalk border using camera on low-speed buggy. Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi-conference: Artificial Intelligence and applications. Retrieved April 28, 2022, from https://dl.acm.org/doi/abs/10.5555/1295303.1295344

[6] Lobregt S, Viergever MA. A discrete dynamic contour model. IEEE Trans Med Imaging. 1995;14(1):12-24. doi: 10.1109/42.370398.

[7] Dey V, Zhang Y and Zhong M (2010) A Review on Image Segmentation Techniques with Remote Sensing Perspective. In: Proceedings of ISPRS TC VII Symposium – 100 Years ISPRS, Vienna, Austria: 31–42.

[8] Chen X, Li G, Yang L, et al. (2020) Profiling unmanned aerial vehicle photography tourists. Current Issues in Tourism 23(14): 1705–1710.

[9] Cheng, M., et al. "Curb detection for road and sidewalk detection." IEEE Transactions on Vehicular Technology 67.11 (2018): 10330-10342.

[10] Felzenszwalb P., David M., and Deva R. "A discriminatively trained, multiscale, deformable part model." 2008 IEEE conference on computer vision and pattern recognition. Ieee, 2008.

[11] Badrinarayanan, Vijay, Ankur Handa, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling." arXiv preprint arXiv:1505.07293 (2015).

[12] Nitr, C. (2021, December 14). MIoU Calculation - CYBORG NITR. Medium. https://medium.com/@cyborg.team.nitr/miou-calculation-4875f918f4cb

[13] Geiger, Andreas, et al. "Vision meets robotics: The kitti dataset." The International Journal of Robotics Research 32.11 (2013): 1231-1237.

[14] Chen, LC., et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40.4 (2017): 834-848

[15] Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[16] Zhou, Bolei, et al. "Scene parsing through ade20k dataset." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[17] Chen, LC, et al. "Rethinking atrous convolution for semantic image segmentation." arXiv preprint arXiv:1706.05587 (2017).

[18] Vicente, Sara, et al. "Reconstructing pascal voc." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[19] Hara K.*et al.* (2014). Tohme: detecting curb ramps in google street view using crowdsourcing, computer vision, and machine learning. In Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14). Association for Computing Machinery, New York, NY, USA, 189–204. DOI:https://doi.org/10.1145/2642918.2647403

[20] Weld, Galen, et al. "Deep learning for automatically detecting sidewalk accessibility problems using streetscape imagery." The 21st International ACM SIGACCESS Conference on Computers and Accessibility. 2019.

[21] Zhao, Hengshuang, et al. "Pyramid scene parsing network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[22] Bolya, Daniel, et al. "Yolact: Real-time instance segmentation." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[23] Tao, Andrew, Karan Sapra, and Bryan Catanzaro. "Hierarchical multi-scale attention for semantic segmentation." arXiv preprint arXiv:2005.10821 (2020).

[24] Tao, Andrew, Karan Sapra, and Bryan Catanzaro. "Hierarchical multi-scale attention for semantic segmentation." arXiv preprint arXiv:2005.10821 (2020).

[25] Ganaie, M. A., and Minghui Hu. "Ensemble deep learning: A review." arXiv preprint arXiv:2104.02395 (2021).

[26] Sun J. and Jacobs D.(2017) "Seeing What is Not There: Learning Context to Determine Where Objects are Missing," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017 pp. 1234-1242. doi: 10.1109/CVPR.2017.136.

[27] Akhtar, N., and Ajmal M.. "Threat of adversarial attacks on deep learning in computer vision: A survey." Ieee Access 6 (2018): 14410-14430.

[28] Ruta, D., and Bogdan G. "Classifier selection for majority voting." Information fusion 6.1 (2005): 63-81.

[29] Earth Engine Data catalog,google developers. (n.d.). Retrieved March 11, 2022, from https://developers.google.com/earth-engine/datasets/

[30] Liu, Y., et al. ”Deep network for road damage detection.” 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020.

[31] Pytorch. PyTorch. (n.d.). Retrieved March 11, 2022, from https://pytorch.org/

[32] (GPU Cloud, Workstations, Servers, Laptops for Deep Learning — Lambda, 2022)

[33] Torrey, L., and Jude S. ”Transfer learning.” Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global, 2010. 242-264.

[34] Redmon, J. , Divvala, S. , Girshick, R. , & Farhadi, A. . (2016). You only look once: unified, real-time object detection.

[35] Ren, Shaoqing, et al. ”Faster r-cnn: Towards real-time object detection with region proposal networks.” Advances in neural information processing systems 28 (2015).

[36] Liu, Wei, et al. ”Ssd: Single shot multibox detector.” European conference on computer vision. Springer, Cham, 2016.

[37] Zhang, R., et al. ”Comparison of backbones for semantic segmentation network.” Journal of Physics: Conference Series. Vol. 1544. No. 1. IOP Publishing, 2020.

[38] Yuxin W.,Alexander K.,Francisco M.,Wan-Yen L., Ross G.,Detectron2,Retrieved March 11, 2022, from https://github.com/facebookresearch/detectron2,2019

[39] He, K., et al. ”Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[40] Lin, T., et al. ”Feature pyramid networks for object detection.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[41] Redmon J.,Ali F.. ”Yolov3: An incremental improvement.” arXiv preprint arXiv:1804.02767 (2018).

[42] Chollet, F. ”Xception: Deep learning with depthwise separable convolutions.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[43] Singh, S., et al. ”Machine translation using deep learning: An overview.” 2017 international conference on computer, communications and electronics (comptelix). IEEE, 2017.

[44] Wang, S., et al. ”An overview of unsupervised deep feature representation for text categorization.” IEEE Transactions on Computational Social Systems 6.3 (2019): 504-517.

[45] Chen, T., et al. ”Pix2seq: A language modeling framework for object detection.” arXiv preprint arXiv:2109.10852 (2021).

[46] Chen, S., et al. ”Reverse attention-based residual network for salient object detection.” IEEE Transactions on Image Processing 29 (2020): 3763-3776.

[47] Yang, F., et al. ”Multi-scale feature integrated attention-based rotation network for object detection in VHR aerial images.” Sensors 20.6 (2020): 1686.

[48] Zhang, Hang, et al. ”Resnest: Split-attention networks.” arXiv preprint arXiv:2004.08955 (2020).