

**SEMANTIC SEGMENTATION USING CONDITIONAL GAN WITH
PERCEPTUAL LOSS**

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted by:

Gaurav Sohaliya (2K19/ISY/08)

Under the supervision of

Prof. Kapil Sharma



DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi – 110042

JULY, 2021


DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE’S DECLARATION

I, Gaurav Sohaliya (2K19/ISY/08) student of M.Tech. (Information Systems), hereby declare that the Thesis/Dissertation titled: “**Semantic Segmentation using Conditional Gan with Perceptual Loss**”, which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

July 22, 2021



Gaurav Sohaliya (2K19/ISY/08)

INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Thesis/Dissertation titled “**Semantic Segmentation using Conditional Gan with Perceptual Loss**” which is submitted by Gaurav Sohaliya (2K19/ISY/08), Department of Information Technology, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is a record of the project work carried out by the student with my guidance. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

PROF. KAPIL SHARMA

July 22, 2021

SUPERVISOR

ACKNOWLEDGEMENT

I am very thankful to **Prof. Kapil Sharma** (Department of Information Technology) and all the faculty members of the Department of Information Technology of DTU. They all provided us with immense support and guidance for the project. I would also like to express my gratitude to the university for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions. I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Gaurav Sohaliya

2K19/ISY/08

ABSTRACT

Image-to-semantic labels classification is a very challenging task in image processing. Convolutional neural networks (CNN) have managed to achieve the state-of-the-art quality of the segmented image in semantic segmentation tasks. Still, the classification capability of such algorithms is not satisfactory to segment images that contain complex object boundaries and minimal regions. Recently, the Generative Adversarial Networks (GAN) were introduced, which can solve the overfitting of the generator network using the adversarial loss. In this paper, a GAN-based segmentation model is proposed, in which the Conditional Generative Adversarial Networks (CGAN) model is used as base architecture. Perceptual loss is introduced in this composite model to solve the identification and classification of visually small elements in images. A pre-trained deep convolution neural network is adopted to generate improved segmentation masks to calculate Perceptual loss. The usage of Perceptual loss has demonstrated the high quality of the output labels. The evaluation of the proposed model on the cityscapes dataset has shown the effectiveness of GAN-based architecture in semantic segmentation of multiclass images. The proposed model achieved 83.3% accuracy on the test dataset, which is superior to most semantic segmentation state-of-the-art methods.

CONTENTS

CANDIDATE'S DECLARATION	I
CERTIFICATE	II
ACKNOWLEDGEMENT	III
ABSTRACT	IV
LIST OF FIGURES	VI
LIST OF TABLES	VII
CHAPTER 1 : INTRODUCTION	1
1.1 EVALUATION OF IMAGE SEGMENTATION.....	3
1.2 DIFFERENT APPROACHES OF SEMANTIC SEGMENTATION.....	4
1.3 ORGANIZATION OF THE THESIS.....	6
CHAPTER 2 : LITERATURE REVIEW	7
2.1 NAIVE APPROACH.....	8
2.1.1 <i>Fully Convolutional Network (FCN)</i>	9
2.1.2 <i>Pyramid Scene Parsing Network (PSPNet)</i>	14
2.1.3 <i>DeepLab</i>	15
2.1.4 <i>U-net Architecture</i>	15
2.2 WEAKLY AND SEMI-SUPERVISED LEARNING.....	17
2.3 GENERATIVE-BASED APPROACHES.....	21
2.3.1 <i>Semantic segmentation using Adversarial Networks</i>	25
2.3.2 <i>SegGAN</i>	26
CHAPTER 3 : DATASET	28
3.1 MAPILLARY VISTAS.....	28
3.2 APOLLOSCAPE.....	30
3.3 KITTI.....	30
3.4 BDD 100K.....	31
3.5 CITYSCAPES DATASET.....	32
CHAPTER 4 : METHODOLOGY	37
4.1 DEEP LEARNING.....	37
4.2 OBJECT DETECTION USING DEEP LEARNING.....	42
4.3 REVIEW OF GENERATIVE ADVERSARIAL NETWORKS (GAN).....	43
4.4 PERCEPTUAL LOSS.....	45
4.5 NETWORK ARCHITECTURE.....	47
4.6 OPTIMIZATION.....	53
CHAPTER 5 : EXPERIMENTAL RESULTS	55
5.1 EXPERIMENTAL SETUP.....	55
5.2 EVALUATION METRICS.....	58
5.2.1 <i>Accuracy</i>	59
5.2.2 <i>Memory Footprint</i>	60
5.2.3 <i>Execution Time</i>	60
5.3 RESULTS AND ANALYSIS.....	61
RELATED PUBLICATIONS	72
REFERENCES	72

LIST OF FIGURES

<i>Fig. 1 : Development of scene understanding or object recognition</i>	<i>3</i>
<i>Fig. 2 : Existing semantic segmentation approaches classification.....</i>	<i>7</i>
<i>Fig. 3 : Architecture of FCN</i>	<i>9</i>
<i>Fig. 4 : PSPN Architecture.....</i>	<i>14</i>
<i>Fig. 5 : U-net Architecture.....</i>	<i>16</i>
<i>Fig. 6 : with and without adversarial training results.....</i>	<i>26</i>
<i>Fig. 7 : Number of annotated pixels per class in Mapillary Vistas dataset.....</i>	<i>29</i>
<i>Fig. 8 : Number of annotated pixels per class in Apolloscape Dataset.....</i>	<i>30</i>
<i>Fig. 9 : Number of annotated pixels per class in Kitti dataset.....</i>	<i>31</i>
<i>Fig. 10 : Number of annotated pixels per class in BDD 100K dataset.....</i>	<i>32</i>
<i>Fig. 11 : Images from Cityscapes Dataset.....</i>	<i>34</i>
<i>Fig. 12 : Number of annotated pixels per class.....</i>	<i>35</i>
<i>Fig. 13 : Sample Images from the challenging urban scene understanding datasets..</i>	<i>36</i>
<i>Fig. 14 : (a) ground truth (b) output of U-net.....</i>	<i>45</i>
<i>Fig. 15 : feature map of (a) ground truth (b) U-net prediction (c) Ground truth (d) U-net prediction.....</i>	<i>46</i>
<i>Fig. 16 : The proposed architecture.....</i>	<i>47</i>
<i>Fig. 17 : Generator network for the proposed method.....</i>	<i>49</i>
<i>Fig. 18 : The architecture of the Discriminator.....</i>	<i>52</i>
<i>Fig. 19 : Class wise accuracy of Visually larger region of different methods.....</i>	<i>64</i>
<i>Fig. 20 : Class wise accuracy of Visually smaller region of different methods</i>	<i>65</i>
<i>Fig. 21 : Training and Validation Accuracy without Perceptual Loss.....</i>	<i>67</i>
<i>Fig. 22 : Training and Validation Accuracy with Perceptual Loss.....</i>	<i>67</i>
<i>Fig. 23 : Output sample generated by proposed model at 100 epochs.....</i>	<i>68</i>
<i>Fig. 24 : Output sample generated by proposed model at 200 epochs.....</i>	<i>69</i>
<i>Fig. 25 : Output sample generated by proposed model (a) original (b) ground truth (c) output.....</i>	<i>70</i>

LIST OF TABLES

<i>TABLE I - Dataset Classification</i>	<i>33</i>
<i>Table II - Segmentation performance on the cityscapes dataset.....</i>	<i>62</i>
<i>Table III - Segmentation performance on the cityscapes dataset on class bases.....</i>	<i>62</i>
<i>Table IV - Segmentation performance on the pascal voc dataset.....</i>	<i>66</i>

CHAPTER 1 : INTRODUCTION

Object detection, image classification, and image segmentation are very classic problems in computer vision and still challenging tasks in the current time, which may be used to still 2D pictures, video, and even 3D or volumetric data. Semantic segmentation is one of the high-level processes that leads to full scene understanding in the grand scheme of things. The importance of scene understanding as a fundamental computer vision topic is highlighted by the fact that an increasing number of applications rely on inferring knowledge from pictures. Identifying whether a particular object is present in an image or not and classifying the image according to it is object detection and image classification, respectively. But the semantic image segmentation phenomenon is very complex. It's the technique of assigning a semantic name to each pixel in a picture (for example, vehicle or people). This task is one of the most challenging since natural environments include many classes, some of which are visually identical to one another.

The term “semantic segmentation” was coined in the 1970s [1]. This phrase was similar to picture segmentation at the time, but it emphasized the importance of isolated sections that were “semantically relevant”. “Object segmentation and recognition” [2] are two-class segmentation problems created in the 1990s to distinguish essential things from the background. Because it's difficult to entirely separate foreground and background items, a two-class picture segmentation problem called sliding window object detection was created to partition objects using bounding boxes. At the time,

this notion was comparable to image segmentation, but it emphasized on the importance of different "semantically significant" parts. Object segmentation and recognition is a two-class image segmentation problem invented in the 1990s to distinguish essential items from the background.

We are interested in finding all the things present in a picture with exact regional boundaries and categorizing each pixel present in an image into a set of classes [3] like bus, car, sky, building, tree, road, etc., assigning a label to it. This is very challenging task in image processing because of multiple angles and viewpoints of different objects and illumination, texture of visual scenes, and the high variation in appearance. Because separating foreground and background items is challenging, a two-class picture segmentation problem known as sliding window object detection was created to partition objects using bounding boxes.

Excellent two-class fragmentation techniques like limited parametric min-cutting are effective in recognising objects in circumstances. Two-class image segmentation, on the other hand, is unable to determine what these split items are. As a result, object detection (or identification) was eventually enhanced to semantic segmentation or multi-class image labeling in the contemporary meaning, to establish both the location and the nature of the objects in the picture.

In order to achieve high-quality semantic segmentation, two primary difficulties must be addressed. These include how efficient representation of characteristics may be built to distinguish between different classes and how contextual information can be used to ensure consistency between pixel labels.

Semantic segmentation can be used in automated cars [4], virtual reality, human-computer interaction [5], sky monitoring cameras, delivery drones, etc.

1.1 Evaluation of Image Segmentation

Semantic segmentation has grown in importance as it relates to different elements of computer vision. This evaluation will not be able to cover all of the information available on the subject. We won't go into detail on classic image segmentation, object segmentation, or object identification because there are already several great assessments of research findings.

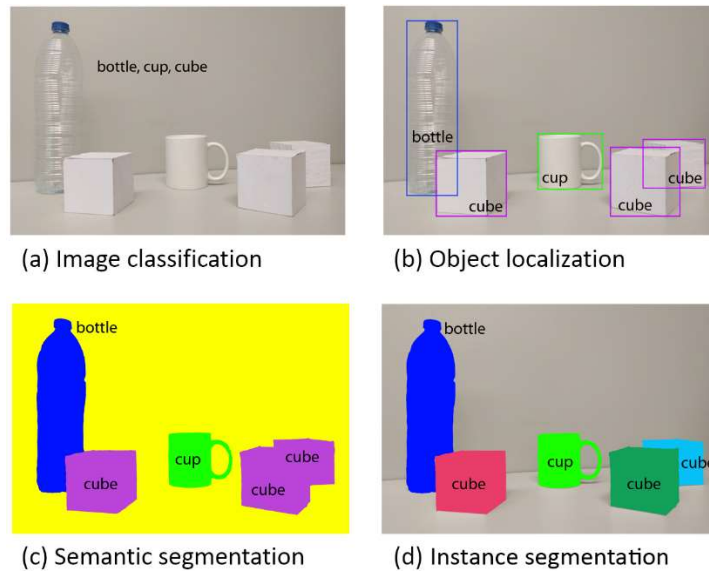


Fig. 1 : Development of scene understanding or object recognition

- **Image classification:** The image might be categorized, which entails generating a prediction for the whole input, such as identifying which object in an image is the object or even constructing a ranked list if there are several, it is referred as image classification.

- **Localization or detection:** is not simply the classes but also supplementary information about their spatial position, such the centres or bounding boxes. is the next stage in the picture categorization.
- **Semantic segmentation:** is the obvious next step after object localization; its goal is to provide dense predictions for each pixel of image, labelling each pixel with the class of the item or region it surrounds.
- **Instance segmentation:** is the next stage following semantic segmentation, where it assigns different labels for various instances of the same class. The above-mentioned development is depicted in Figure 1.

1.2 Different approaches of semantic segmentation

Deep learning can solve these tasks using deep convolution architecture, and some of the problems are also resolved successfully. Several classic computer vision and machine learning approaches have already been used to tackle this challenge. Convolution neural network architectures (CNN) [6, 7] pushed the accuracy and efficiency of image segmentation to a new height by surpassing other methods with a wide margin. Despite their widespread usage, the deep learning revolution has turned the tables, and deep architectures, most notably CNNs, are now being utilised to tackle a wide range of computer vision problems, including semantic segmentation, surpassing traditional approaches in terms of accuracy and speed.

CNNs can solve the problem of semantic segmentation. Still, it requires many modifications because originally, CNN is developed for the image classification task. Because of that, CNN suffers from various problems such as the low resolution of generated output, unable to predict object boundaries accurately. In comparison to other well-established disciplines like computer vision and machine learning, deep learning is still in its infancy.

Using the capabilities of CNN and overcoming the problem, several deep architectures and research works have been proposed, such as Fully Convolution Network [8], Pyramid Scene Parsing Network [9], DeepLab [10], U-Net [11], etc. These techniques can solve image segmentation problems successfully at a certain level, but some issues like accurate boundary extraction and area prediction remain to solve. As a result, there are few unifying works and state-of-the-art assessments available. Due to the huge volume of new material being generated, keeping up with the field's ever-changing state is difficult, and keeping up with its evolution pace is a time-consuming endeavor. This makes it difficult to stay on top of semantic segmentation research and correctly evaluate concepts, eliminate ineffective techniques, and validate results.

A convolution neural network tries to learn a parametric translation function in a fully supervised environment. Significant manual efforts are required to compose accurate loss between ground truth pixels and predicted pixels. Ian J. Goodfellow proposed generative adversarial networks (GANs) [12] to address this problem. GAN tries to identify the relation between an input dataset and an output dataset. GANs can be used in several applications, such as object transfiguration, photo enhancement, style transfer, season, or daylight transfer.

There are certain limitations of GANs, such as the training of GAN is sensitive and unstable, it cannot control the data being generated by the generator. To tackle these problems, Mehdi Mirza proposed Conditional Generative Adversarial Nets (CGAN) [13], in which condition has been imposed over discriminator to produce the required output. We use CGANs as basic architecture with perceptual loss [14] for high quality and accurate regional segmentation in our proposed architecture.

1.3 Organization of the Thesis

This dissertation demonstrates how recent advances in the field of semantic segmentation have led to the development of a novel architecture based on generative adversarial networks that has achieved the best segmentation accuracy of any GAN-based approach described to date. In Chapter 2, we briefly examined the linked works, including their introduction, what they accomplished, and their limits. The tools and platform we utilised, the libraries and datasets we used, and the general structure of the tracking effort are all explained in Chapter 3. It provides a comprehensive description of the step-by-step algorithms. Finally, Chapter 4 presents the experimental results we were able to accomplish using our suggested technique, as well as a comparison of its accuracy metrics with prior methods used in the same area. Chapter 5 summarises our ground breaking work, as well as the findings, problems we encountered while putting the framework in place, and additional works that may be used to enhance our framework.

CHAPTER 2 : LITERATURE REVIEW

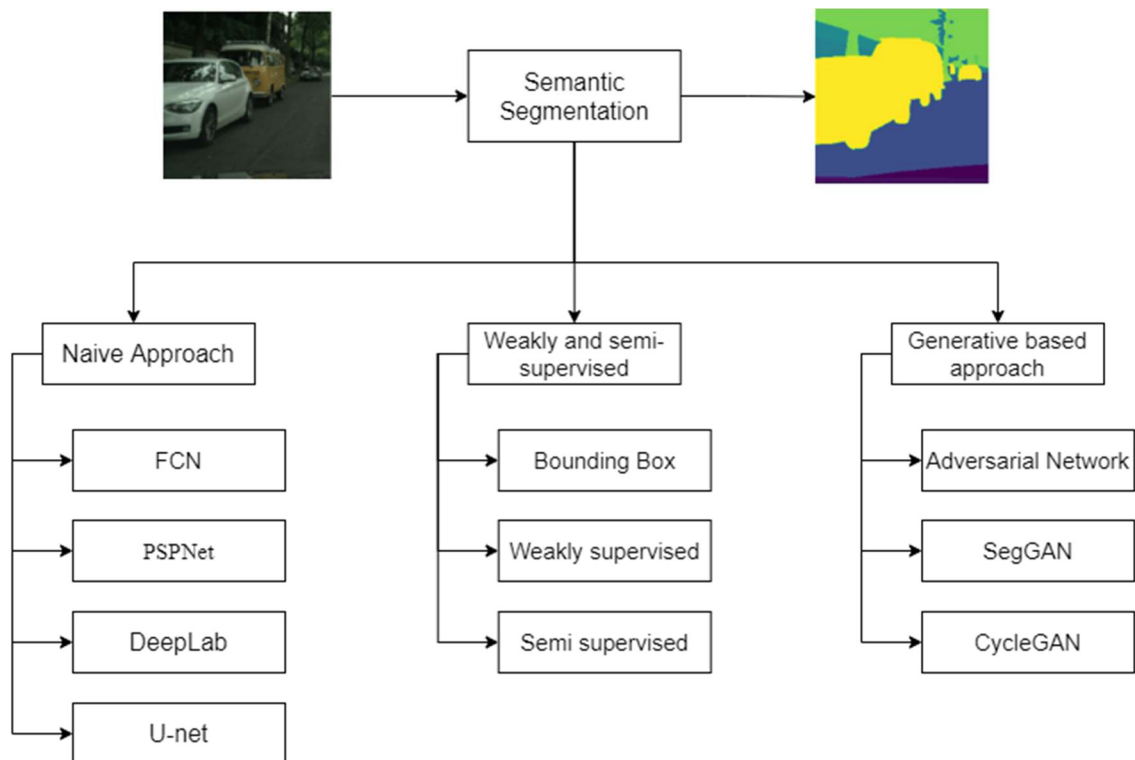


Fig. 2 : Existing semantic segmentation approaches classification

CNNs In this chapter, as shown in Fig. 2 shows, we will see methods to solve semantic segmentation and its sub approaches. Then after, we will explore available dataset for urban scene semantic segmentation.

2.1 Naive approach

The region/pixel level categorization is followed by feature extractors. CNNs are commonly used to display patches of fixed size centred on each pixel in the context of pixel categorization. On the other hand, it is inadequate to mark each pixel with a tiny region surrounding it. When making a local decision, it is essential to look at the wider context. The size of picture patches is a simple technique to do this.

However, this technique would significantly rise CNN parameters, which would result in a significant increase in processing complexity. Multi-scale approaches were created to deal with a wide range of circumstances while keeping computational efficiency. Farabet et al. [15] used CNN on a multi-scale pyramid of data to analyze patches of images. The feature maps produced at different sizes were combined using a size matching approach. As a result, each pixel is linked to a feature vector comprised of multiple patches with progressively larger but equal-sized visual fields.

Coupric et al. [16] used a multi-scale architecture to extract features from the depth information of RGB-D sensors. In the same way, different scales were taken into account. Although feeding picture patches to a traditional CNN to obtain dense predictions is conceptually simple, it is computationally inefficient. Because patches centred at neighbouring pixels are considerably overlapped, there are a lot of duplicate convolutions. One method to approach this challenge is to use the notion of region proposals to undertake region-wise categorization.

After warping, this approach has the benefit of immediately putting the rectangle region suggestions into a CNN for classification. This system can also recognise and segment items at the same time. Rectangular region

suggestions, on the other hand, include not just the object class, but a few others as well. As a result, many bottomup area suggestions must be examined while deciding on the label for a single pixel. Mostajabi et al. [17] used a CNN as a feature extractor and upsampled the resulting feature maps to produce pixel-level features.

Then, by pooling pixel-level data, superpixel characteristics were derived. In order to acquire multi-scale data, the authors also recommended that the characteristics of intermediate layers be concatenated. Upsampling these multi-scale feature maps to picture resolution would take a lot of memory because each scale can include hundreds of feature maps. A recursive network of context propagation was used to enhance each superpixel with contextual information, which comprised an upgrade and an upgraded context phase.

2.1.1 Fully Convolutional Network (FCN)

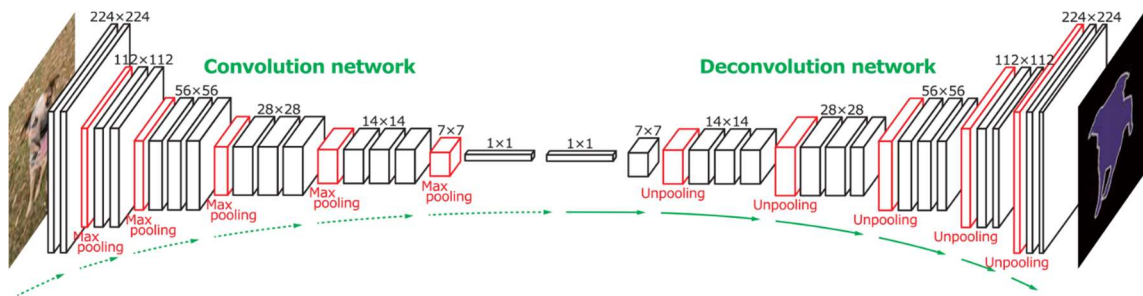


Fig. 3 : Architecture of FCN

FCN [8] is the first successful architecture (Fig. 3) in terms of accuracy gain in image segmentation, which uses a CNN to reconstruct image pixels into a set of categories. The last dense layer of convolutional neural networks is replaced by transposed convolution layer in the proposed FCN architecture.

Hence the predicted pixels can have one-to-one conformity in terms of spatial positions with the input image. FCN first uses the consecutive convolution layers to capture image features. Then it transforms the dimensions of the intermediate layer feature block to the dimensions of the input image by using the transposed convolution layer. The model's output will have the exact dimensions as the input image. The final output channel contains the class prediction of the pixel of the corresponding spatial dimension.

The dense layer in CNN has many trainable parameters, and it requires a lot of time to train; this layer is eliminated in FCN. Hence FCN takes less time for training in comparison to CNN. It does not have any dense layer and contains only convolutional layers, because of which it can accept variable size input images.

The majority of recent advances in semantic segmentation have employed FCN. The main idea behind FCN is to substitute a fully connected layer in a conventional CNN with a 1×1 convolution layer to produce low-resolution predictions, as shown in Fig. 3. Shift-and-stitch [18] is a simple method for sewing predictions generated from numerous shifted copies of the input into pixel dense forecasts. Because the CNN's output moves in lockstep with the input, this approach works. The shift-and-stitch method must evaluate s^2 shifted copies of the input picture when assessing output score maps with input strides. A more efficient technique is to upsample over the coarse predictions. We called it UP FCN to keep things easy. [8, 19] for example, utilised a deconvolution layer to up-sample low-resolution predictions. End-to-end training is possible with the UP FCN-based methodology, making it a preferred method for scene labelling.

In end-to-end training, the feature representation and pixel-wise classifiers are both learned at the same time. Based on the structure of FCN,

methods for improving semantic segmentation may be divided into three categories. It's possible that upsampling low-resolution forecasts to picture resolution (typically 1/32 of the original input) will result in a significant loss. To extract the feature maps, the SegNet [20] employed a hierarchy of decoders. The authors used dimensionality reduction on the feature maps to reduce the model's size, however this reduced the model's accuracy.

Some down-sampling procedures may be discarded before upsampling to get finer-resolution forecasts. Unfortunately, this may result in a reduction in the receptive field widths of the final layer, resulting in the loss of contextual information and critical global for semantic segmentation. Dilated convolution is a similar approach described in [21]. The fundamental concept is to dilate the convolution filters before the deleted down sampling stages. The atrous convolution increases the number of zeros in the initial convolution mask, whereas the dilated convolution accomplishes the same. The main benefit of the atrocious FCN model is that it can give forecasts without the requirement for additional parameters.

However, because the model's wide receptive field fails to capture low-level visual information, adding skip connections to intermediary layer features is another approach for producing finer-resolution outputs. For example, before fusing the multi-resolution scores to generate the final forecast. Feature maps should be up-sampled rather than score maps. All of these approaches used a summing procedure to aggregate the multi-resolution score or feature maps after upsampling.

Making the most of a poor circumstance includes the following strategies: Despite the fact that many deep CNNs' receptive fields are theoretically close to or even greater than the total input, Zhou et al. [22] discovered that the true receptive field is significantly smaller than the

theoretical one, indicating that many segmentation networks do not adequately use higher-level contexts. Zhao et al. achieved this by splitting the FCN receptive field into various sizes (multiple parallel pooling). Context was gathered at both low and high levels by combining the features of various receptive fields. [10] suggested an atrous spatial pyramid pooling approach that is similar but more efficient.

Lin et al. recently reported a superpixel-based receptive field that produced several receptive fields by using varied superpixel sizes. Mean-field inference was used by Chen et al. to create a dense CRF with sharp limits over up-sampled predictions. [23] calculated paired pixel affinities utilising semantic limits given by a trained CNN instead of low-level pairwise potentials defined on colour contrast. Lin et al. [24] used a non-associative potential based on CNN.

Zheng et al. [25] were able to train CNN and CRF simultaneously by translating several rounds of CRF inference to a recurrent neural network by combining the capabilities of the CNN and CRF in one unified framework (RNN). Arnab et al. [26] shown that the CRF with higher order potentials may also be incorporated into the FCN, resulting in substantial improvements over the CRF-RNN. These CRF embedded FCNs are substantially more complex than standard FCNs, notwithstanding their effectiveness [23]. [27] addressed the problem by casting the paired potential to a contextual classifier over unary scores, which reproduced CRF inference as a convolutional network. Because approximation inference methods are common in discrete CRF-based approaches, [28] employed Gaussian CRF to pursue precise inference. While [28] and [25] both used the same notion of unrolling fixed inference steps to train a deep network.

Several scene labelling techniques based on this information used RNN in conjunction with CNN to capture long-range spatial relationships. RNN works on the premise of memorising data created in the current prediction and then passing it on to the next prediction. As a result, previous data from this repeating process may influence each projection. The idea of temporal sequence is broadened to encompass spatial sequence when it comes to scene labelling.

By seeing the image horizontally and vertically in both directions, Visin et al. [29] created four 1D spatial sequences. The pros and cons of each approach are their own. Expanding the receptive field requires a simpler model and is easier to implement than using CRF/RNN. A difficult-to-understand learning process, on the other hand, implicitly and thoroughly represents the structural relationships between pixels. Using CRF/RNN, on the other hand, requires a significant amount of time and effort to train the model, despite the fact that it allows for more precise structural correlations.

Keeping the notion of hard pixels in mind when learning: A class-imbalance problem plagues most semantic segmentation datasets, resulting in performance variations in identifying pixels of various classes. Furthermore, not all pixels in a single picture are easily identifiable. All of these findings point to the need to discriminate between "hard" and "easy" pixels. This is done by eliminating pixel samples that have been "correctly recognised," which is assessed by assessing if the accurate label's predicted probability exceeds a certain threshold. Because most early projections for most pixels have low confidence levels in contrast to reality, pixels are regarded the same at the beginning of training. As the training proceeds, easy pixels become increasingly visible and disregarded. In deep layers, the Deep Layer Cascade just overlooks basic pixels rather than totally discarding them.

Current approaches have mostly concentrated on enhancing two core FCN architectures: atrous UP-FCN and skip connected UP-FCN, as shown in the graph. These advancements include the use of more powerful CNN models, the integration of several CRF or higher-level contexts, and the use of cascaded methods. The FCN may be used as a general model for constructing semantic segmentations, as illustrated in previous presentations. When using finer-resolution feature maps, memory constraints might arise during training, which is a current and common problem with FCN models. As a result, the finer-resolution used is typically 1/8 of the input resolution or less. This indicates that further study will be done in the future to increase output resolution.

2.1.2 Pyramid Scene Parsing Network (PSPNet)

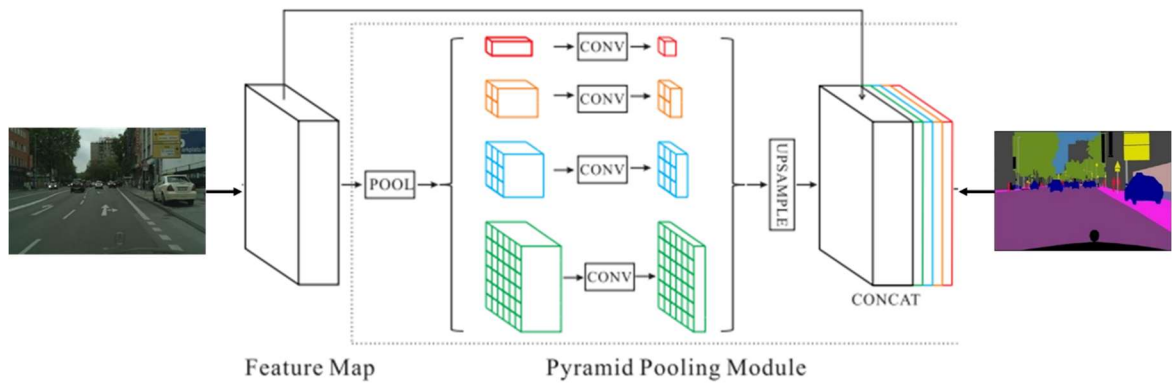


Fig. 4 : PSPN Architecture

PSPNet [9] proposed the scene parsing concept, which refers to parse an image semantically into different regions that belong to a set of classes such as sky, road, and car. PSPNets employ pyramid pooling to use the capabilities of the global context of the image. PSPNet uses FCN in its core, where it has series of convolution layers, and then pyramid pooling adds a new layer between the convolution layer and the fully connected layers. It helps to map

variable size input to constant output size. PSPNet provides a very efficient architecture for interpreting sophisticated scenes and pixel-level prediction ability in the image. It includes predicting the shape, size, and object position, similar to the human visual approach. PSPnet enormously enhances the ability of image description and provides an obvious direction for the pixel-level prediction.

2.1.3 DeepLab

When applying DCNN on images for segmentation, there are certain limitations, and DeepLab [10] tries to overcome these limitations. The first solution is dilated convolution, which solves the problem of reduced feature resolution. Due to multiple pooling and downsampling layers in DCNN reduce spatial resolution significantly from the resultant feature map. Conditional Random field has been employed to increase localization accuracy by considering the neighbor's context while predicting the label. After the first DeepLab model, various revised models have been proposed, including DeepLabV2 [30], DeepLabV3 [31], and DeepLabV3+ [32]. The main idea is to combine the capabilities of DCNN and conditional random fields.

2.1.4 U-net Architecture

U-net [11] was proposed for the segmentation of Biomedical Images. The architecture contains two paths. It is divided into two parts, same as FCN; the first part is the contraction path employed to extract the context and details of the image. It is also addressed as encoder architecture and just a stack of convolution layers without dense layers. The decoder part is the symmetric expanding path which is a stack of deconvolutions layers.

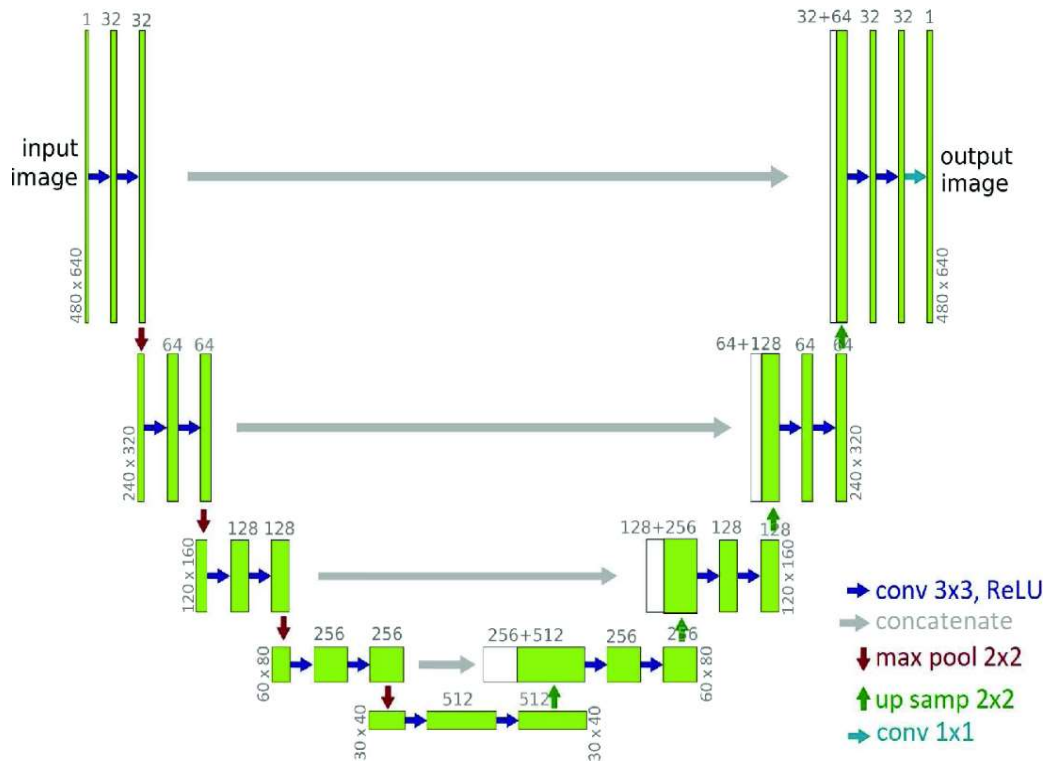


Fig. 5 : U-net Architecture

The problem in encoder-decoder architecture compresses the input linearly; hence, it creates a bottleneck in which all feature information cannot be passed to a decoder, leading to low resolution and inaccurate output. Here the U-net differs from other architecture. U-Net is a kind of architecture that is in U shape. It has two parts that are the left and right sides as shown in Figure 5. The left side, which is called the contracting path, is the usual convolution neural network. Two convolutional blocks are applied.

Each block consists of two convolutional layers, each of which is followed by a ReLU (Rectified Linear Unit) activation function and a max-pooling layer, allowing for down-sampling. After every down-sampling, the number of channels of features reduces. The right side, which is called the expansive path, upsamples the components. By this step, the numbers of feature channels doubles. This step combines with a concatenation layer added along with feature maps from the respective contracting path and two convolution layers,

each of which again followed by a ReLU (Rectified Linear Unit) activation function. After the last deconvolution block, a convolution layer of filter size equal to the number of classes is applied and a segmented image will be generated.

U-net allows the network to propagate context information from the lower layer to the higher layers. The higher resolution feature maps from the encoder's successive layers are concatenated with upsampled features of the decoder network. It helps to learn better representations with subsequent deconvolution operations. Upsampling or deconvolution is a sparse operation, so to increase the resolution and represent better localization, it needs prior knowledge from successive layers. This yields U-shaped architecture, and hence u-net architecture has achieved good accuracy in semantic segmentation [33] [34].

2.2 Weakly and Semi-Supervised Learning

The initial domain adaptation approaches for semantic segmentation were created via adopting domain adaptation methods for classification in the classification area, similar to how previous semantic segmentation research began using image classification techniques. However, approaches focusing only on the semantic segmentation problem began to emerge soon after, taking into consideration the unique characteristics of the spatial components as well as the costly (pixel-wise) labour. Simultaneously, methods with weak or incomplete supervision, which are the subject of this section, were widely used prior to unsupervised domain adaptation.

As previously stated, training a deep learning model for semantic segmentation necessitates a large quantity of data with pixel-level semantic labels, which is difficult, inconvenient, and time-consuming to get. Other

computer vision applications, such as image classification and object recognition, are less impacted by this issue since image-level tags or bounding boxes are much simpler to come by, and large annotated datasets are easily accessible. Many studies suggest that a model be trained using just weakly labelled data or a combination of many weakly labelled samples and a few samples with the more expensive pixel-level semantic map in the segmentation problem.

As indicated in [35], A weakly monitored semantic segmentation is one method to address the problem as a multiple-instance learning problem. A approach for calculating the probability of unnoticed pixel labels was added to the foundation, which was founded on the standard feature-based Semantic Texton Forest (STF) approach. The STF's structure was then enhanced using a unique approach in which a geometric context estimate job is used as a regularizer in a multi-task learning framework.

[36] also proposes a deeper network for semantic segmentation, with the application of an EM algorithm, in a semicircular and poorly monitored situation. The method alternates between pixel-level annotation estimate (limited to weak annotations) and network segmentation tweaking. [37] is presented as a model for the inclusion of limited CNNs (CCNNs) in training. Linear limitations are introduced to the area where labels from the image level tags exist and are expected to distribute, and a unique loss function has been developed to improve the set of constraints.

[38] proposed a weakly-supervised semantic segmentation paradigm that ranged from elementary to sophisticated. Simple and difficult photographs are differentiated in the article: the former contain a single object of a single category in the foreground and a clean backdrop, whilst the latter may have many things of numerous categories and a crowded background.

First, using salient item identification approaches, semantic maps are constructed from weakly annotated basic pictures, and then three separate networks are trained successively from these to enable sophisticated image segmentation.

[19] proposes a semi-supervised method that consists of three key elements: a segmentation network, a classification network and some bridge layers that connect the two networks. The suggested training is separated by first training of the classification network with poorly noted information, followed at the same time by training of bridge layers and segmentation network with strong examples. After feeding the input picture into the classification network, the bridge layers extract a class-specific activation map from the classification network intermediate layer, which is then fed into the segmentation network. This technique allows for training with a limited number of semantically annotated samples since the number of parameters in the segmentation network may be reduced. The classification network's collection of key labels and spatial information, which is then improved by the bridge layers, makes the task of the segmentation network much easier in practise.

Using just bounding-box annotated input, [33] developed an iterative technique for training a segmentation network. For each image, region proposal methods are utilised to create a huge number of possible segmentation. The candidate mask that overlaps the ground truth bounding box as much as feasible with the right label is chosen using an overlapping objective function. One candidate mask is chosen for each bounding box at each iterative phase, and the resultant semantic labels are utilised to train the segmentation network. The segmentation network's outputs are then employed in a feedback loop to enhance candidate label selection for the following phase. Both the selected candidate labels and the segmentation network outputs improve with each iteration.

Starting with [39], where the discriminator network is changed to do semantic segmentation, generative adversarial networks have proved to be effective in this field. The discriminator assigns a label from one of the semantic classes or a false label to each pixel in the input picture. Fake (manufactured) data, unlabeled data for regularisation, and labelled data with pixel-level semantic mappings are used to train the discriminator. In a weakly-supervised scenario, another method is to use conditional Generative Adversarial Networks (GANs) with weak image-level annotation at both the generator and discriminator inputs in a weakly-supervised scenario.

Using a traditional Seeded Region Growing (SRG) technique, the areas are then extended to neighbouring pixels with comparable characteristics, resulting in precise labelling of the pixel level that are utilised to train a segmentation network. The SRG technology detects the resemblance between the seed and the surrounding pixels using the output of the segmentation network. Consequently, each repeat improves the divisional network and dynamic labels created by SRG. [38] provides a similar approach for gradually localising and enlarging object areas using a unique harsh erasing procedure. The suggested technique is divided into four phases, with two tasks and two domains:

1. In order to establish a common representation for the domains, one task network is trained on data from both domains to resolve the first task;
2. The second network is solely trained in the first field to deal with the second problem;
3. On the first domain a third network is trained to map depths acceptable for the first job to features that are appropriate for the second task ;

4. Finally, in the first area the nth network is trained in mapping the first task deep features to fit the second task function;

This approach moves picture segmentation tasks from a generated to a real area by using feature maps from both areas. Because depth cameras and 3D scanners make it simpler to collect depth maps, they may be inferior to semantic maps as annotations.

2.3 Generative-Based Approaches

Unsupervised image-to-image translation (I2I) aims to create a function that can translate pictures between domains using just unpaired training data from the domains in question as supervision. The objective is to extract unique features from a collection of photos and transfer them to a different data set. The I2I job, in a more formal description, tries to find a common distribution of pictures from several domains. Since the problem is unsuccessful, and the marginal ones can infer an unlimited number of joint distributions, suitable restrictions have to be imposed to find satisfactory results.

The discovery of the target set's conditional distribution with regard to the source set, the statistical gap between source and target pixel-level statistics in principle should be able to cross and eliminate an initial covariate shift that caused the performance of the classifier to decrease. The objective is to keep the source semantic information while transferring visual characteristics from the target domain to the source domain. Many research have suggested an input-level adaptation approach based on a generative module that changes pictures between source and destination domains based on this concept. Regardless of the approaches used, the objective of all of these articles is to achieve visual domain invariance by decreasing cross-domain differences in picture arrangement and organisation. This allows for the use of

source annotations while learning a segmentation network on data from a translated source domain.

In cross-domain picture translations, the semanticized predictor is obliged to keep semanticized information, a degree of semantic discord between the original image and its translated counterpart, minimised by the optimisation of the translation network. Because the prediction maps are fundamentally incorrect, specifically in the target set when annotations are lacking, the generative module's learning of picture projections may be hampered by the erroneous semantic information provided.

To achieve input-level domain adaptation, a large amount of research has relied on the effective CycleGAN [40] unsupervised I2I framework. [40] present a method for conditional image translation in both the target-to-source and source-to-target directions across two domain sets based on a pair of generative adversarial models. The necessity for cycle consistency links the two antagonistic modules even tighter, further separating the cross-domain projections. This criteria is necessary to maintain the input scene's structural geometrical characteristics, but it does not ensure semantic coherence during translation. In reality, the mapping functions may entirely ruin the semantic categorization of incoming data, although maintaining geometrical coherence.

Li et al. [41] use the CycleGAN-based adaptation technique to build a bidirectional learning framework. The I2I and segmentation modules are alternatively trained in an optimization approach in which both modules get positive data from the other. The segmentation network is enhanced by original supervision and target-like translated source pictures, while the predictor assists the generative network in maintaining semantic consistency. I2I semantic accuracy and quality both improve with time with this closed-loop structure.

Li et al. [42] proposes to use soft gradient sensitive losses in order to support the cyclically consistent I2I framework in the maintenance of semantic substances inside the cross-domain projection with an emphasis on semantic borders in cross-domain projection. The aim of this technique is that the semantic uniform area borders should be apparent regardless of how low-level visual features change between domains, regardless of the distribution utilised to create the image. As a result, a gradient-based edge detector should be able to recognise edge mappings that are consistent in both the original and modified pictures. They also develop a semantic-aware discriminator structure, which is based on the concept that semantically distinct sections of an image should be modified differently. The discriminator can then assess the semantic similarity of the original and translated samples.

By training generative networks with a continuous variable representing the domain, Gong et al. [43] modified the CycleGAN model to produce a continuous flow of domains spanning from source to destination domains. The objective of collecting intermediate domains that span the two starting domains is to make the adaptation process easier by gradually explaining the domain shift that affects input data distributions. They also assert that using target-like training data from a variety of target-like domain distributions increases the generalisation of the segmentation network.

To minimise the processing cost for the CycleGAN bi-directional structure, several researches are using a light-weight adaptation module based on a generative adversarial frame to slip the backward-to-target projection branch. The link to a related effort, which is done concurrently with semantic segmentation, ensures translation consistency. Choi et al. [44] instead improve the fundamental GAN framework's generator by adding feature normalisation modules at various stages to provide style information to source

representations while maintaining source content. A semantic consistency loss from a pre-trained segmentation network enhances image translation coherence in the absence of the cycle-consistency effect, resulting in a regularising impact.

Style-transfer techniques are examined as a separate adaptation category in order to achieve an invariance in image levels throughout the source and target domains. The concept that each image is separated in two components is based on those methods: content and style. The content offers a low-level domain-specific texture, while the style transmits domain-invariant, high-level structural features. As such, the ability to integrate style characteristics with semantically maintained source material from target data should enable the generation of goal-delivered training information while retaining the original source annotations.

Yang et al [46] are creating a type of objective monitoring from source translated data using image-to-image target-to-source translation instead of a common source-to-target translation to decrease the distortion towards a source domain. The source-like target pictures are then utilised in the supervised training of the predictor by utilising pseudo labelling. The training of the segmentation network in the source domain permits complete use of the original source annotations in the pixel-to-target adaptation situation, while also reducing the potential for semantical alterations. To match feature representations across domains, they additionally develop a label-driven reconstruction network. Rather of utilising feature-based reconstruction approaches, semantic maps obtained from the segmentation result are used to generate generative replication of input pictures. They intend to do this by directing the category-wise alignment of segmentation network embeddings, penalising reconstructions that stray from the objective, and ensuring semantic consistency in network predictions.

In order to characterise the remainder of a representation between source and destination maps, Hong et al. [45] employ a conditionally generated function, optimised in an adverse environment. They avoid depending on a single domain-invariant latent space assumption, which may or may not be satisfied due to the highly organised nature of semantic segmentation. The generator is called upon to create high-level characteristic maps with target distribution from low source characteristics maps and noise samples using a disc that measures the statistical distance between original and replicated target representations. The original source and domain modified representations are given a dense classifier to calculate the loss of cross-entropy.

2.3.1 Semantic segmentation using Adversarial Networks

In all the previous models, the loss function is majorly based on pixel-to-pixel loss. As discussed previously in [10], a Conditional Markov random field (CRFs) is an effective way to emphasize spatial integrity in output labeled images. CRF Model has a limitation of higher-order consistency. To overcome this limitation, in 2016, facebook researchers proposed to use an Adversarial Network [47]. Adversarial loss of GAN architecture helps to enforce higher-order consistency forms, which is impossible with pixel-to-pixel loss.

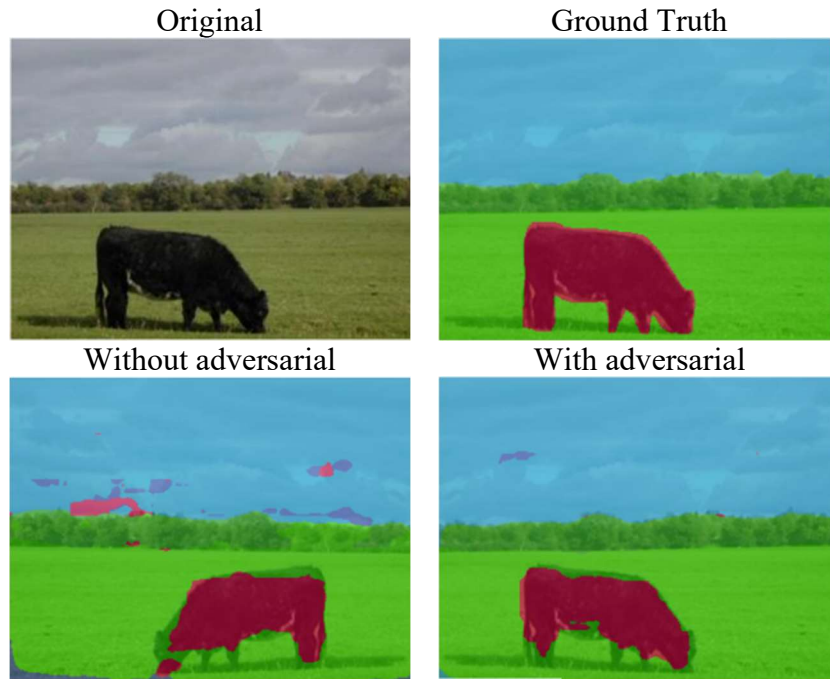


Fig. 6 : with and without adversarial training results

The results are shown in Fig. 6 prove that adversarial training helps to get spatial consistency and brings smoothness and sharpness of the large object or class, e.g., sky and grass.

2.3.2 SegGAN

SegGAN [48] has been proposed to utilize the capabilities of adversarial loss. It uses the GAN as a kind of loss to segmentation a given image and achieve effective results. In SegGAN architecture, there are mainly three components:

- Pretrained DeepLab model – it is used for semantic segmentation.
- Generator G – it takes predicated masks from the DeepLab model and tries to generate an original image synthetically.
- Discriminator D – it distinguishes synthetically generated images from a generator and real images from a dataset.

Here, generator G follows a simple encoder-decoder architecture with four convolution and four deconvolution layers. The discriminator uses four convolution layers with ReLU as the activation method. In this paper, another significant thing is the loss function; they integrated three losses overall:

- Segmentation Loss – it is a loss between the predicated segmentation mask map from Segmentation Network and the actual mask.
- Content Loss – this is pixel-wise MSE loss between ground truth image and reconstructed image by Generator G.
- Adversarial Loss – generator G and discriminator D try to improve themselves by fooling each other.

The result obtained by SegGAN is superior in comparison to DeepLab and GAN-based architecture, and the research has successfully demonstrated GAN and adversarial loss capabilities.

CHAPTER 3 : DATASET

3.1 Mapillary Vistas

The Mapillary Vistas [49] is a large-scale street-level photo dataset with 25000 high-resolution images divided into 66 item categories and additional, instance-specific labels for 37 classes. In a rich and fine-grained way of annotation, individual elements are demarcated using polygons. Mapillary Vistas is five times the size of the total number of excellent comments for Cityscapes, and it contains images from all over the world, captured in a range of weather, season, and daytime conditions. Photographers of various ability levels create images using a range of imaging equipment. Our dataset has been created and assembled in this manner to encompass diversity, richness of detail, and geographic scope. Fig. 7 shows number of pixels per class in Mapillary Vistas dataset and total classes.

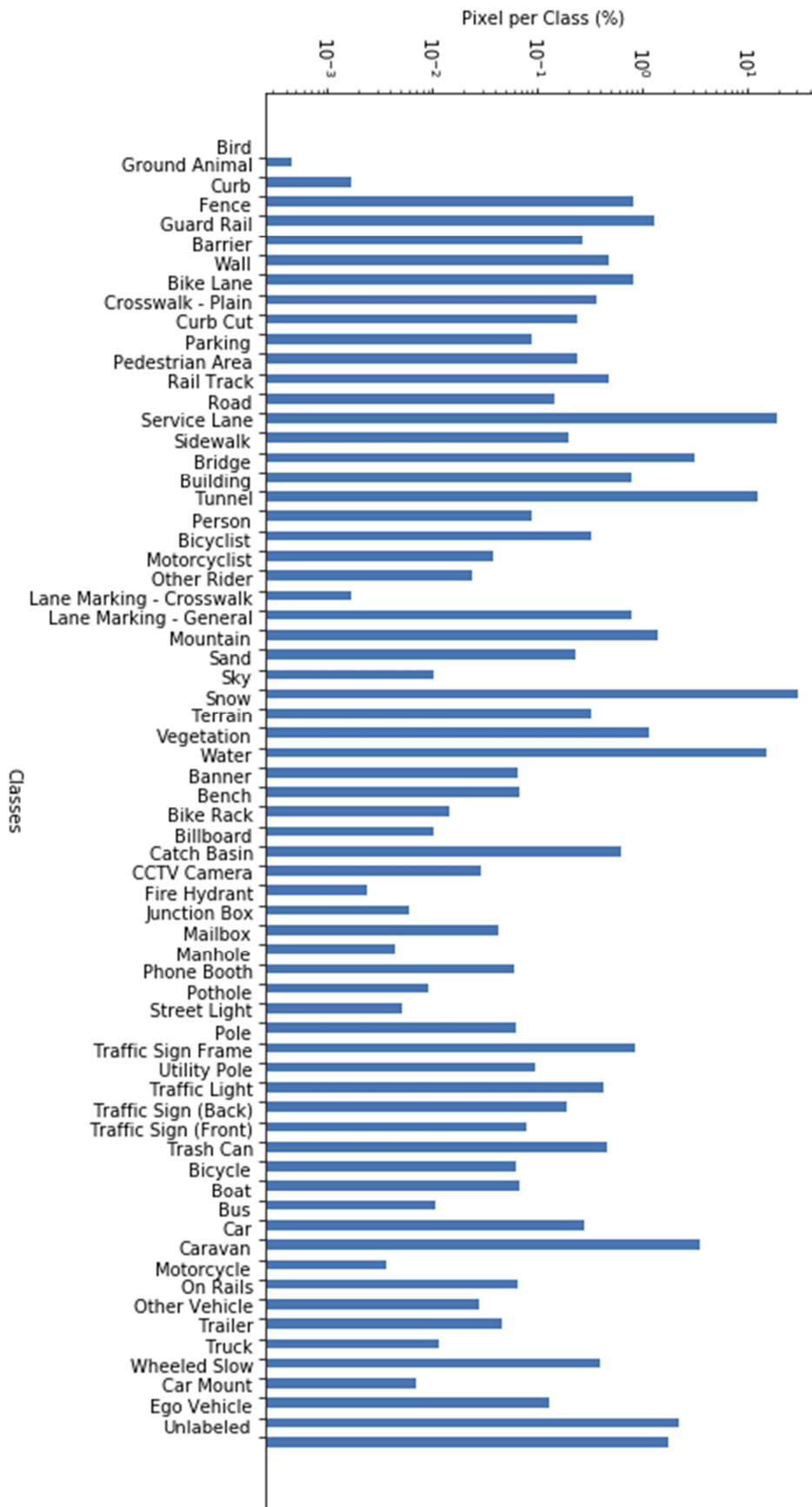


Fig. 7 : Number of annotated pixels per class in Mapillary Vistas dataset

3.2 ApolloScape

The Apollo Scape [50] dataset includes 147k pictures with pixel-level annotations. The static background's posture information and depth maps are also included. All photos were taken with a Riegl VMX-1HA camera system with a resolution of 3384 x 2710 pixels. The class definitions are identical to those in the Cityscapes dataset, except they included a new tricycle class that encompasses all types of three-wheeled vehicles due to the prevalence of the tricycle in East Asian nations. Fig. 8 shows number of pixels per class in Apollo Scape dataset and total classes.

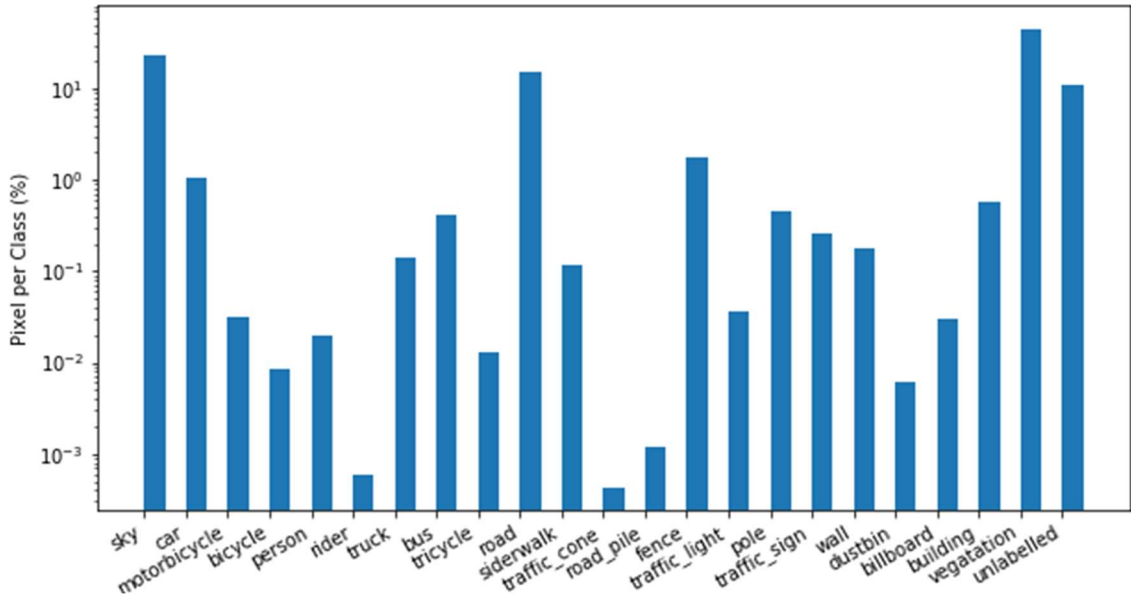


Fig. 8 : Number of annotated pixels per class in ApolloScape Dataset

3.3 Kitti

The KITTI [51] semantic segmentation dataset contains 200 training and 200 test semantically annotated pictures. In addition to semantic segmentation, the KITTI collection includes datasets for optical flow, depth assessment, lane/road detection, scene flow, object tracking, 2D and 3D object recognition, and object tracking. This dataset was collected while driving about Karlsruhe, in rural regions, and on highways. Per picture, up to 15

automobiles and 30 people can be seen. Fig. 9 shows number of pixels per class in Kitti dataset and total classes.

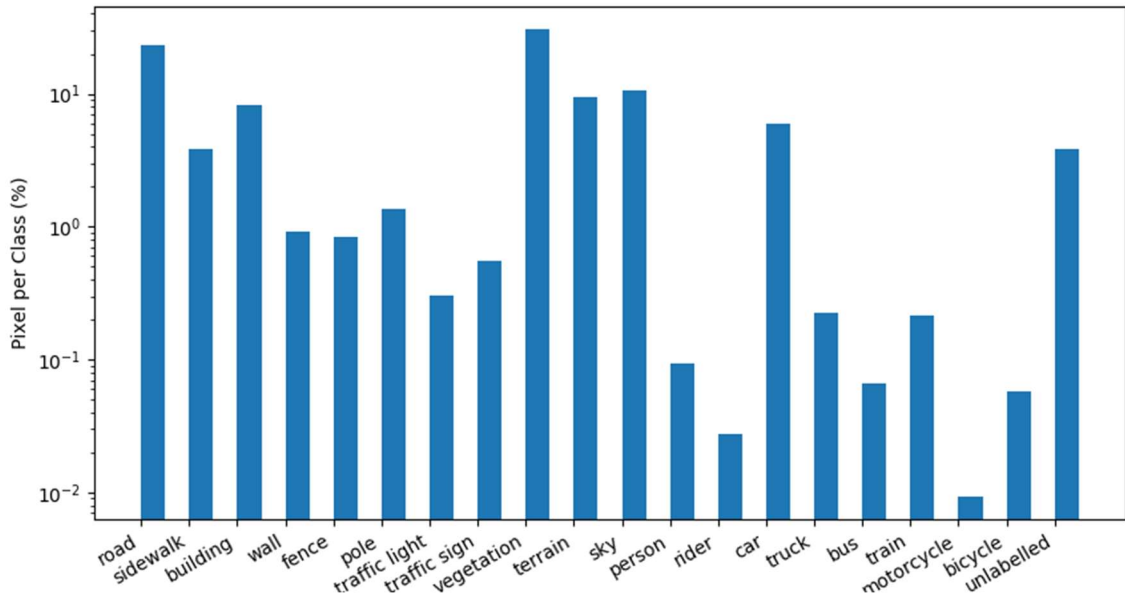


Fig. 9 : Number of annotated pixels per class in Kitti dataset

3.4 BDD 100K

The BDD100K [52] dataset is 800 times larger than ApolloScape and the largest freely available self-driving dataset. Geographic, environmental, and meteorological variety are all included in the dataset, which is beneficial for training models that are less likely to be shocked by unexpected situations. The majority of the data in this collection comes from various parts of the United States. It created a benchmark for heterogeneous multitask learning and investigated how to solve the tasks collaboratively. Fig. 10 shows number of pixels per class in BDD100K dataset and total classes.

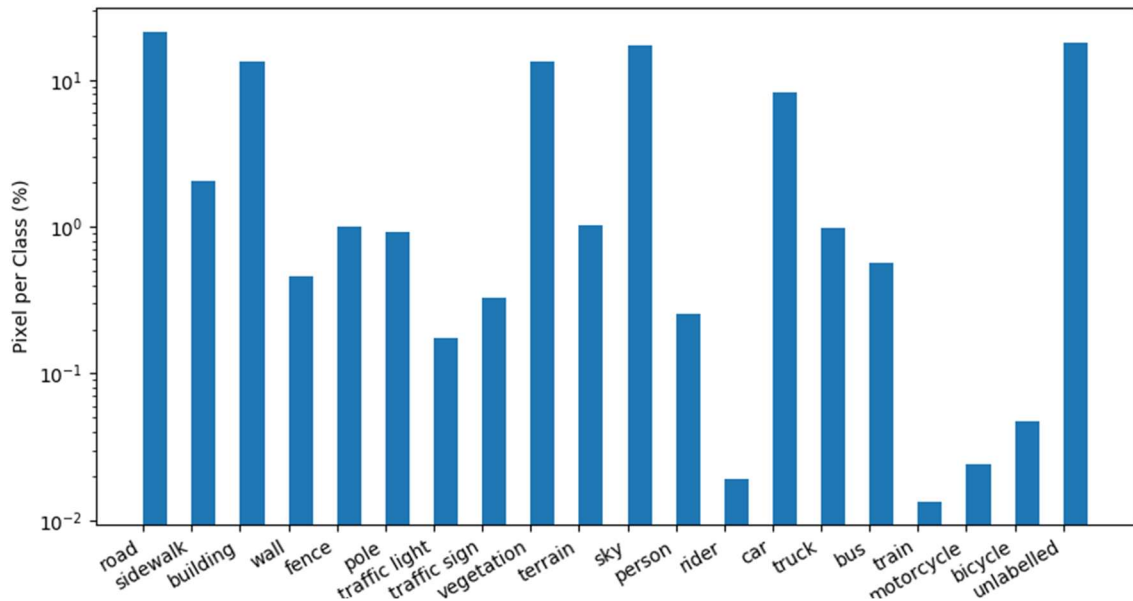


Fig. 10 : Number of annotated pixels per class in BDD 100K dataset

3.5 Cityscapes Dataset

We used the Cityscapes [53] dataset for this research. It is a huge dataset that focuses on semantic segmentation, through which we can understand urban street scenes. Hundreds of thousands of frames were captured from a moving car in 50 locations over the course of many months throughout the spring, summer, and fall seasons, mostly in Germany but also in neighboring countries. In Cityscapes dataset, they purposefully avoided recording in inclement weather, such as torrential rain or snow, because such situations necessitate specific approaches and datasets.

Images were captured at a frame rate of 17 Hz using an automotive-grade 22 cm baseline stereo camera with 1/3 in CMOS 2 MP and rolling shutters. The sensors were installed beneath the windscreen and provide pictures with a 16-bit linear colour depth and a strong dynamic range. Each pair of 16-bit stereo pictures was then delayed and tweaked after that. The images produced are less accurate, but they are more aesthetically attractive and simpler to annotate. By hand-picking 5000 photographs from 27 cities for

detailed pixel-level annotation, the goal was to produce a high level of variation in the foreground, backdrop, and overall scene layout.

A single image was chosen for coarse annotation every 20 seconds or every 20 metres of driving distance in the remaining 23 cities, totaling 20 000 photos. The dataset consists of around 20000 coarse annotated ones, and 5000 fine annotated images. Dataset contains daytime images captured from 50 different cities. There are mainly 34 classes like objects, humans, flat surfaces, constructions, sky, vehicles, nature, void, etc. The class contribution in images is hugely unbalanced in terms of region area, and it also includes a diverse scene layout and a large number of objects. It will help us to check the robustness of the model. It provides three types of annotation, e.g., instance-wise, semantic, and dense pixel annotations. Since this dataset contains high-resolution images, we converted it into 256*256 resolution so that class information assigned to each pixel is not loose.

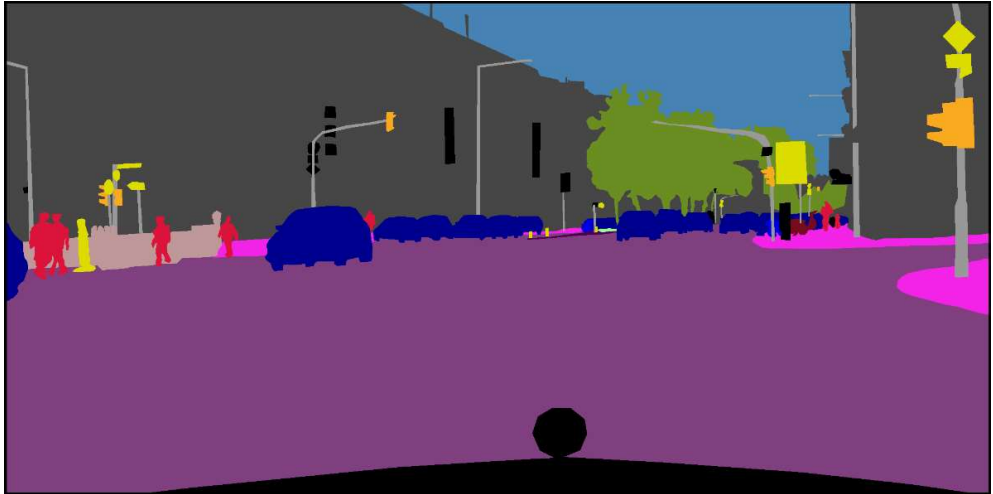
TABLE I - Dataset Classification

Group	Classes
flat	Rail track, parking, sidewalk, road
human	Rider, person,
vehicle	Trailer, caravan, bicycle, motorcycle, on rails, bus, truck, car
construction	Tunnel, bridge, guard rail, fence, wall, building
object	Traffic light, traffic sign, pole group, pole
nature	Terrain, vegetation,
sky	sky
void	Static, ground, dynamic

Real Image



Fine annotated colored Image



Fine annotated Image



Fig. 11 : Images from Cityscapes Dataset

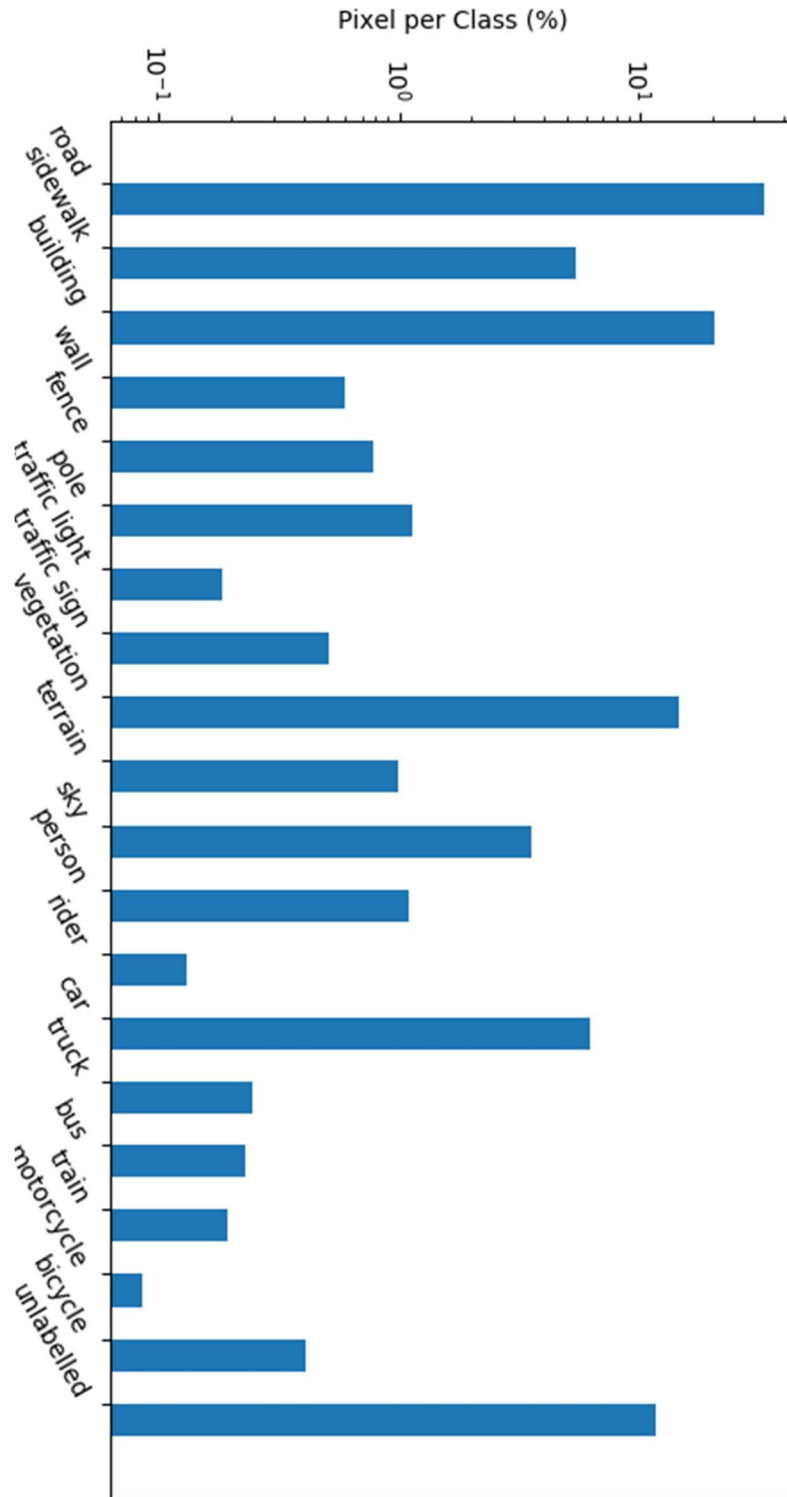


Fig. 12 : Number of annotated pixels per class

Fig. 12 shows number of pixels per class in Kitti dataset and total classes. We used a fine annotated dataset with 2975 training, 500 validation, and 1525 test images for our experiment.

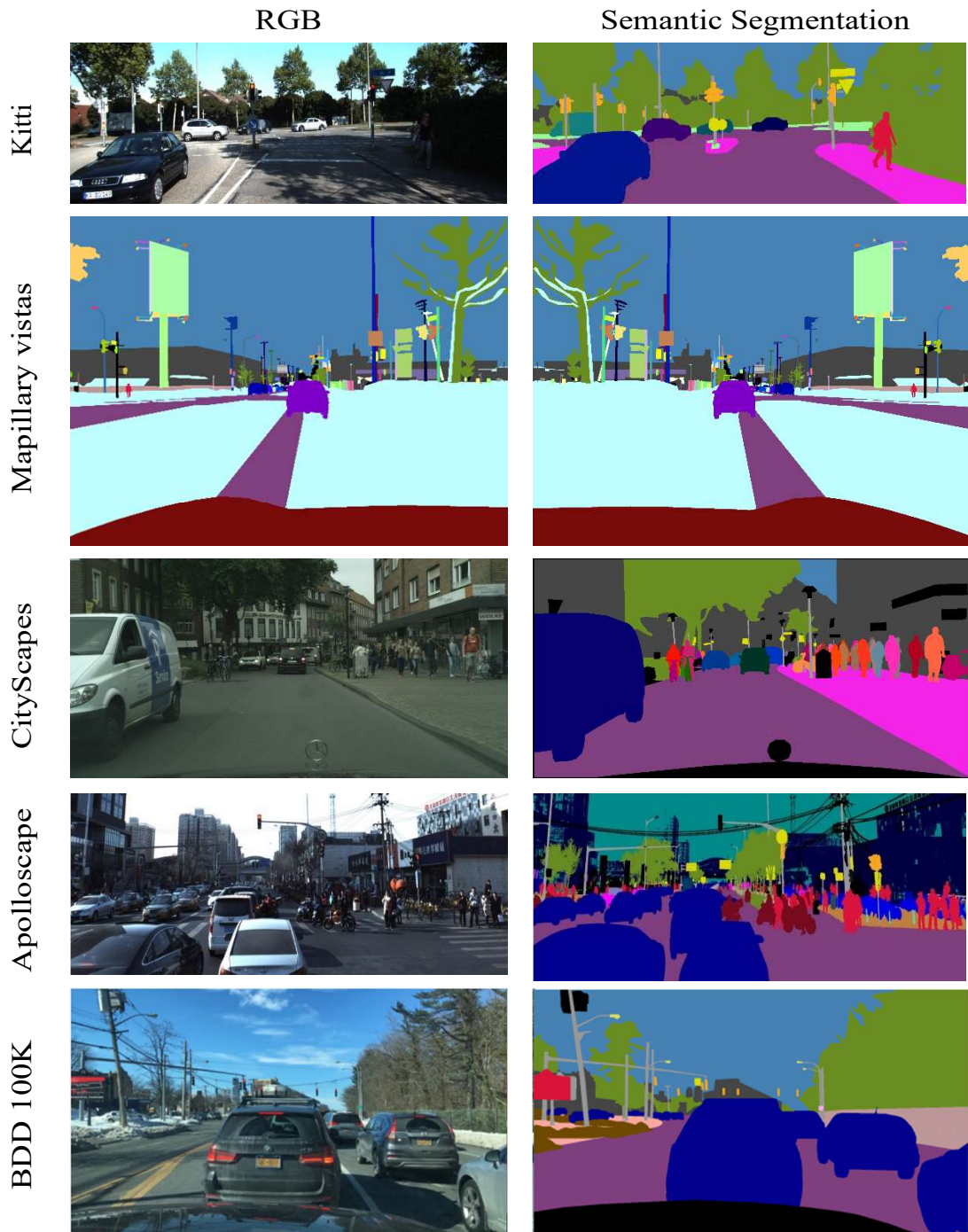


Fig. 13 : Sample Images from the challenging urban scene understanding datasets

CHAPTER 4: METHODOLOGY

4.1 Deep Learning

Web search, internal social network filtering, and offers on 'e-commerce' websites, as well as consumer products like cameras and phones, all employ machine learning technology. The machine learning system recognises item images, converts speech to text, adapts to new items, articles, or commodities, as well as user requests, and selects relevant search results. Deep learning is a sort of technology that's becoming more common in these systems.

Deep learning is a subfield of artificial intelligence. Deep learning is primarily concerned with computations that are enlivened by the structure and capability of fictitious neural networks that are driven by the human mind. Deep learning is used without lifting a finger to forecast the unexpected. Deep learning is a specialist with an incredibly sophisticated set of abilities that allows them to get much higher outcomes from similar data sets. It is sometimes referred to as a subset of AI. It is simply based on the NI (Natural Intelligence) mechanics of the biological neuron structure. It has a befuddling variety of skills in light of the strategies it employs for preparation, such as learning in deep learning relies on "learning information representations" rather than "task-explicit computations," as is the case with other systems.

Traditional machine learning approaches are restricted in their ability to analyse raw natural data. Engineers and field skills have been necessary for decades to create feature extractors that transform input (such as pixel values in pictures) into internal representations in pattern building systems or machine learning systems. The learning subsystem (typically the classifier) is able to recognise and categorise patterns in the input. The technique of representation training allows a machine to examine raw data and locate the representatives required for detection or distribution.

A model of representation with several layers of representation is an in-depth research. It's made up of basic but unusual combinations. Each model evolves from a depiction of one degree of knowledge to a higher and higher level of comprehension. Very tough jobs may be accomplished with the right connections, such as transitions. Higher-level representatives focus on tactics that are essential to diversity and avoid incorrect exchanges when it comes to the division of work.

A picture, for example, is a pattern with pixel values, and the characteristics learned in the initial representation are commonly replaced in the image by or without specific borders and regions. Regardless of any little modifications in the working edges, the second layer generally discovers a motor by visualising the arrangement of the edges. The third layer can transform motifs into bigger textures that fit the normal material's shell, while the outer layer captures the material as these parts are linked. The major point of the research is that the structure's layers were not designed by human engineers, but rather learnt from data using a generic learning approach.

Deep learning is a collection of basic models that are all (or almost all) learnt and that frequently calculate non-linear input-output grids. Each model in the group alters the format, enhancing the representation's variety and

ambiguity. Using many non-layered layers (e.g., depths of 5 to 20) allows the system to access information that is not sensitive to mass and does not vary, such as backdrop, motion, illumination, and surroundings, while also utilising the material's extremely complicated properties.

The backpropagation method for computing the slope of an Function of objective with regard to the stack weight of a multilayer module is just a practical implementation of the chain rule for derivatives. Working backwards from the slope of a module's output, the derivative (or slope) of a target may be calculated regarding the module's input. We may multiply the gradient by recursively applying the back-division equation to all modules. From the top (where the network creates predictions) to the bottom (where the network generates predictions), you may multiply the gradient. It's simple to calculate the gradients in proportion to the weight of every component after these gradients have been established.

Neural networks are used to learn how to transform a fixed-sized input (such a picture) into a fixed-size output in many deep-learning applications (e.g., probability for each of several categories). A series of units calculates the weighted amount of their incoming layer and transmits the outcome to a nonlinear function to advance from one to the next layer.

The rectified linear unit (ReLU), which is nothing more than a half-wave rectifier with $f(z) = \max(z, 0)$, has become the most often used nonlinear function. In recent decades, smooth nonlinearities such as $\tanh(z)$ or $\frac{1}{(1 + e^{-z})}$ have been used in neural networks, but ReLU learns far faster on multilayer networks, allowing the construction of a deeply monitored network without prior supervised training. Units that are not at the input or output level

are known as hidden units. Hidden layers are nonlinear input distortions that allow the categories to be removed from the final layer in a linear manner.

Back propagation and neural networks were mostly disregarded by the machine learning community. It was likewise disregarded by the computer communities' visual and speech-recognition communities. Learning meaningful multi-step extracts with minimal prior information was considered to be impossible. A basic slope computation, in example, was considered to be locked in weak local minima - heavy settings where even minor modifications would not lower the mean error.

A bad local minimum is usually an issue in many networks in practice. Despite the original conditions, the system usually constantly produces high-quality results. The local minimum is not a major concern, according to recent theoretical and theoretical conclusions. The scene, on the other hand, is encased in a tangle of numerous saddle points, each with a gradient of zero and a top curve in most dimensions and curves in the rest. The study appears to suggest that the saddle points are many, with just a few curve orientations at the bottom, but nearly all of them have the same amount of objective movement. As a result, it doesn't matter where of these saddle points the algorithm is gathered in.

Then there was a surge in interest in deep forward learning. Uncontrolled learning techniques were developed by the researchers, allowing them to build layers on detectors without requiring tagged data. In order to simulate the performance of function detectors in the layer below, objective learning of each layer of function detectors has to be able to reconstruct. By pre-training the detector layers of multiple increasingly more complicated functions, the original network's bulk may be reduced to a manageable level. Afterwards it was possible to add the last layer of output

units to the top layer of the system and configure the entire deep system with conventional backpropagation. This was notably useful for detecting handwritten numerals or identifying pedestrians when the quantity of data indicated was restricted.

The introduction of powerful GPUs that were easy to programme and permitted researchers to train networks 10 or 20 times faster helped to make speech recognition the first important use of this pre-training technique. This approach was used in 2009 to convert a collection of probabilities of distinct speech fragments that may be demonstrated by a frame at the window's centre from a short time window of coefficients derived from sound waves. It set new records in conventional voice recognition benchmarking tests with a short vocabulary, and it was soon improved to hit new highs in huge vocabulary tasks.

Many big voice teams have been using the Deep Web version since 2009, and it had been installed on Android phones by 2012. When the number of tagged instances is limited or when there are numerous examples in the transfer settings for some "source" activities, unsupervised pre-training helps to minimise over-fitting, but few Generalization benefits are considerably better when applied for specific "target" tasks. It turns out that once deep learning is stabilised, only small data sets require a pre-training phase.

However, unlike networks with full connection between adjacent layers, there existed a form of deep advanced network that was considerably easier to train and generalised far better. The convolutional neural network was responsible for this (ConvNet). It has had a number of practical triumphs at a period when neural networks were underutilised, and it has recently gained widespread acceptance in the computer vision field.

4.2 Object Detection using Deep Learning

The volume of image data on the Internet has increased dramatically as a result of the advent of mobile internet and the popularity of various social media platforms, yet humans are unable to properly process such enormous amounts of image data. As a consequence, these data processing activities are expected to be carried out automatically with the help of a computer to handle large-scale visual difficulties. Complete image interpretation and exact identification of the picture's target object become increasingly important as understanding of image processing technology improves.

People are concerned not only with basic image classification, but also with accurately getting the semantic category of an item and its location in the image, therefore object detection technology has received a lot of attention. Object detection technology aims to recognise target items, define semantic categories for these things, and indicate the exact position of the target object in the picture using image processing and pattern recognition theories and methodologies.

Using computer technology to automatically recognise items in a real-world application is a difficult challenge. Background complexity, noise disruption, occlusion, low-resolution, size and attitude changes, and other variables will all have a significant impact on object recognition ability. The traditional object detection technique relied on a hand-crafted feature that was not resistant to changes in light and lacked generalisation capabilities. In the PASCAL VOC competition from 2010 to 2012, it was noted that progress in object detection was sluggish, with limited advances made by building ensemble systems and using slight modifications of conventional approaches.

As a result, a number of approaches for improving object detection performance have been presented. As a successful model of deep learning, the convolutional neural network (CNN) [54] has the ability of hierarchical learning features, and research indicates that the feature extracted by CNN has a higher discriminating and generalisation capacity than hand-crafted features.

4.3 Review of Generative Adversarial Networks (GAN)

Adversarial learning has been presented as Generative Adversarial Networks (GANs) to achieve a generative goal [12]. (i.e., creating false pictures that seem like genuine ones). In order to solve the generative problem, the unknown probability distribution must be established. The advent of adversarial learning in the generative context was groundbreaking because it removed the need for explicit modelling of the fundamental target distribution and the requirement to train the model with a specific aim.

A generator must learn to generate data on the same statistical distribution of training samples in the adversarial method. It does it by using a discriminator to determine if the incoming data is from the original set or was made up. Simultaneously, the generator is tweaked to trick the discriminator by generating outputs that appear to be identical to the originals. Finally, the generated data must match the training set's statistics.

In the form of a learnable discriminator, the GAN model may learn a structured loss in order to regulate the generative network optimization process. As a consequence, the objective function may be conceived of as automatically adapting to the present situation, without the need for complex losses to be explicitly designed. As a result, the adversarial learning system

may be simply modified to suit a wide range of activities needing various types of application-specific objectives.

GANs [12] are powerful generative models. GAN is working on Game Theory, where it consists of a generator and a discriminator. The generator generates a new image, and the discriminator discriminates whether that newly generated image is actual (as real as training data) or fake. Suppose the discriminator assesses the generated image as a fake image. In that case, it is a loss for the generator network, and if it fails to discriminate as a fake image and classify as an actual image, then it is a loss for the discriminator network. So, it's like a game between these two networks, both are trying to fool each other, and they evaluate themselves by facing loss, using backpropagation, and adjusting weights. The optimization function of GANs is constructed as follows:

$$\min_G \max_D \mathbb{E}_{x \sim \mathbb{P}_r} [\log D(x)] + \mathbb{E}_{z \sim \mathbb{P}_g} \left[\log \left(1 - D(G(z)) \right) \right] \quad (1)$$

where $x \sim \mathbb{P}_r$ are images from the input dataset, and $z \sim \mathbb{P}_g$ are random noise units, $G(z)$ are the generated images and $D(x)$ is the probability of x being actual.

GAN cannot constrain the data being produced by the generator, and to address this issue, Mirza et al. [13] extended GANs to a CGAN (conditional generative adversarial networks). To achieve specific results from GAN networks, certain auxiliary conditions or information x emphasized on the generator and the discriminator, e.g., class labels. The objective function of CGANs is constructed as follows:

$$\min_G \max_D \mathbb{E}_{x \sim \mathbb{P}_r} [\log D(x, y)] + \mathbb{E}_{z \sim \mathbb{P}_g} \left[\log \left(1 - D(x, G(z)) \right) \right] \quad (2)$$

4.4 Perceptual Loss

In CNN, the feature extraction is done hierarchically. So, the response of the lower layer (layers that are nearest to the input layer) of CNN is much similar to the input image because feature extraction at lower layers is at a lower tendency, so it just reproduces much similar output with input. Higher layers give high-level content (e.g., objects in the scene, the relative positions of objects, boundaries of regions), so we prefer to use a higher-level feature map for perceptual loss [55]. The primary purpose of perceptual loss is to leverage the knowledge of a pre-trained network that can identify more minor details and boundaries of regions of image very effectively. U-net and other segmentation algorithms are struggling in the identification and classification of small object and their areas. We use the content loss to bring more information, accurate boundaries, and smooth resultant segmented image to overcome this limitation.

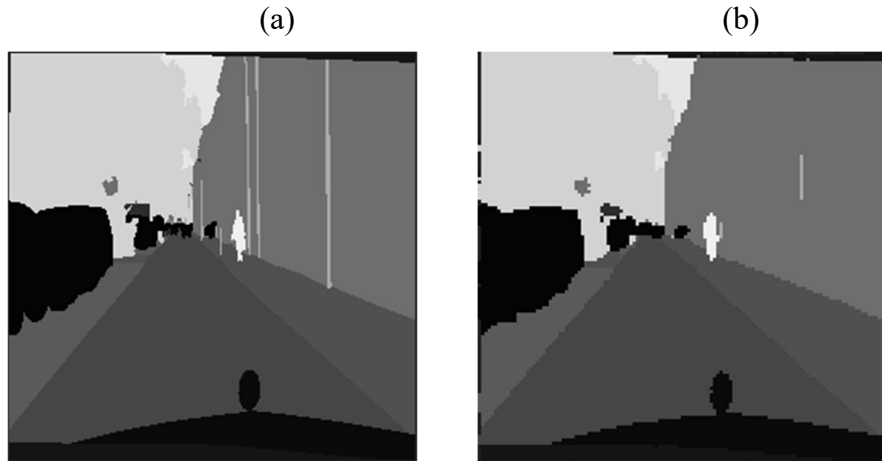


Fig. 14 : (a) ground truth (b) output of U-net.

Fig. 14(a) is the ground truth segmented image from the dataset, whereas Fig. 14(b) is an output of U-net. Some details like street lights and stands are not visible, and boundaries are also not smooth in the predicated image compared to the ground truth image. If the original image (not segmented) is passed to pre-trained CNN, it can identify minor details where

U-net is failed to classify it. Fig. 15 shows the comparison of the higher-level feature map of the actual image and predicted image.

The difference between the higher-level feature map of the original image and the predicted image can be employed as a perceptual loss to gain a sharp and detailed result.

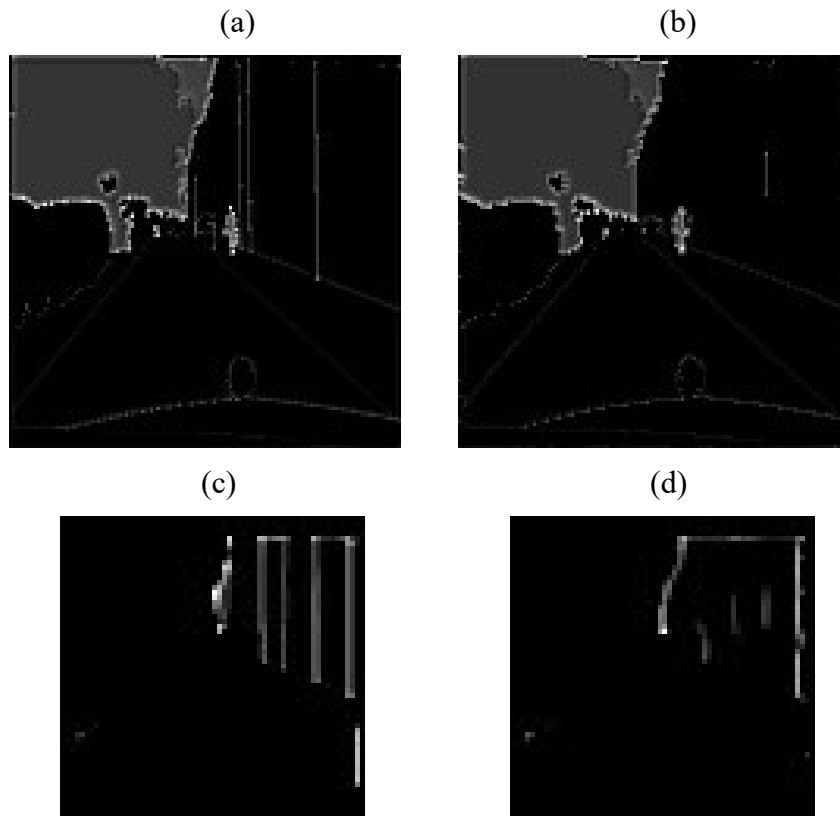


Fig. 15 : feature map of (a) ground truth (b) U-net prediction (c) Ground truth (d) U-net prediction.

So, let \vec{o} and \vec{g} be the original and generated image, respectively, whereas O^l and G^l are their respective feature representation in layer l . C , H , and W represent the number of filters in feature map, height, and width of the image, respectively. We then define the squared-error loss (L2 Norms) between the two feature map representations. The perceptual loss for a layer l can be formulated as:

$$\mathcal{L}_{feat}(\vec{o}, \vec{g}, l) = \frac{1}{C*H*W} \sum_{i=0}^C (G_i^l - O_i^l)^2 \quad (3)$$

The above Equation is for one-layer content representation only; we can take multiple layers, with each layer has a different contribution factor.

4.5 Network Architecture

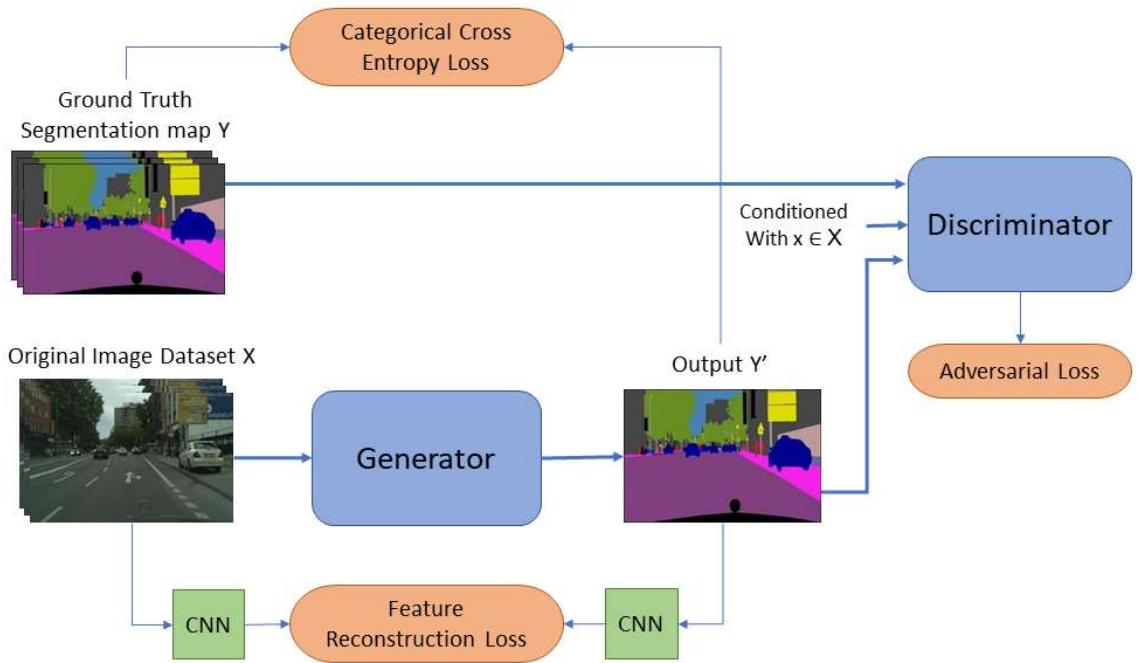


Fig. 16 : The proposed architecture

The Proposed method (Fig. 16) is a different technique in which the conditional GAN is combined with the pre-trained VGG19 [54] network for calculating the perceptual loss. This hybrid architecture consists of a generator model, discriminator, and VGG nets.

U-net has shown excellent performance in image segmentation, and hence we are incorporated U-net with slight modification as generator architecture (Fig. 5). U-net is built on the encoder-decoder model, including

skip connection, where skip connection helps u-net in shuffling low-level details directly between the encoder input to decoder output. The encoder aims to map the input data space into output data space, whereas the decoder forces encoder to do a meaningful reconstruction of output from a given input space.

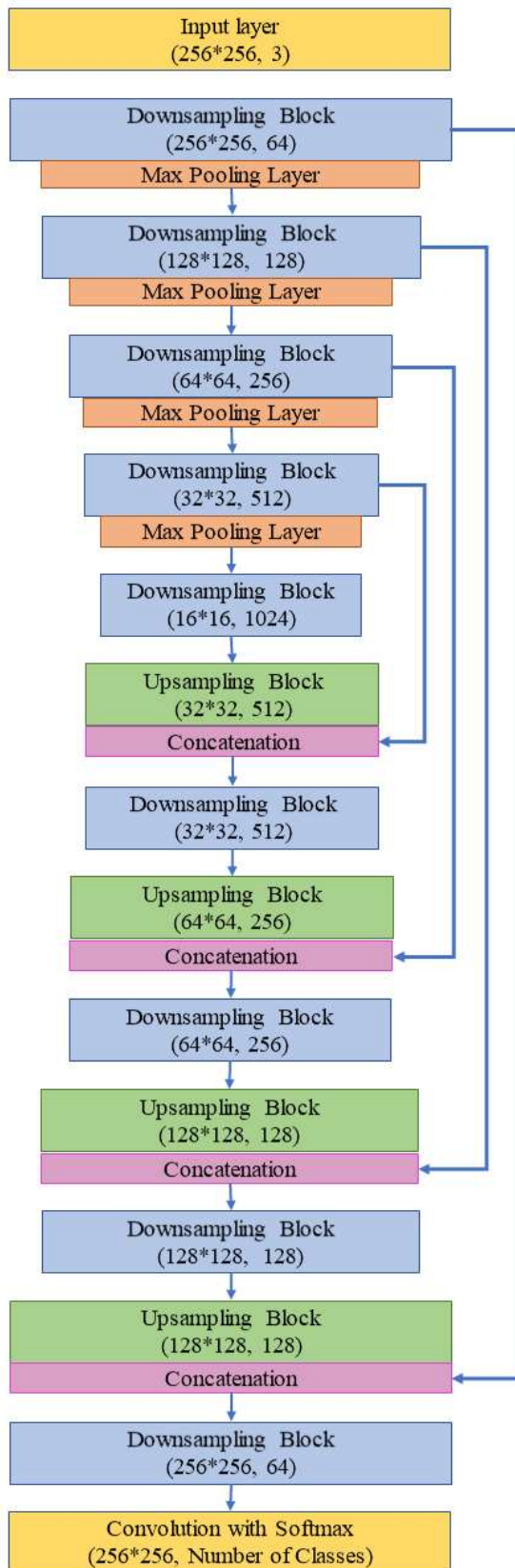


Fig. 17 : Generator network for the proposed method

The encoder part of a generator (fig. 17) consists of 5 downsampling blocks, whereas the decoder part consists of 4 composite blocks of an upsampling block, concatenation layer, and downsampling block. ReLU activation function is used for both the convolution and deconvolution processes. Two convolution layers are followed by a batch normalising layer in each downsampling block. The transposed convolution layer is followed by a batch normalising layer in the upsampling block. The output layer of the generator has softmax as an activation function because it predicts each class's probability for each pixel. In contrast, the discriminator accepts two images, generated image and the conditioned image (original image), with three channels. Generated image has N number of channels, and discriminator requires only three channels. Hence, we added argmax and stack layer between generator and discriminator.

The PatchGAN is used as a discriminator, which is proposed in [56], aims to categorize every patch into real or fake of an image. Discriminator runs over the entire image patch by patch and does convolution over each patch. The mean of all the values will generate the final output of the discriminator. PatchGAN projects the image as same as CRF (Markov random field), and hence it acts as a form of texture.

When it comes to picture creation issues, it's well known that L2 loss – as well as L1. Although these losses don't help with high-frequency clarity, they do a good job of capturing low frequencies in most cases. When this is the case, we don't need a whole new system to assure accuracy at low frequency for problems. L1 is adequate. The use of an L1 term to compel low-frequency accuracy by restricting the GAN discriminator to only represent high-frequency structure is motivated by this (Eqn. 4). We just need to concentrate on the structure of small picture patches to portray high-frequency signals. As a result, the PatchGAN, a discriminator architecture that penalises structure purely on the basis of patch size.

This discriminator attempts to determine if each of an image's $N \times N$ patches is authentic or not. We use a complex approach to apply this discriminator to the image, averaging all answers to get D's final result. PatchGAN is a smaller version of PatchGAN that has less parameters, runs quicker, and can be used on any size picture. This type of discriminator treats the picture as a Markov random field, assuming that pixels separated by greater than a patch diameter are independent.

Both the conditioned actual sample from the target domain, $D(x, y|x)$, and the conditioned fake sample created by G, $D(x, G(z|x))$, are given to the discriminator network. D examines the data distribution to check if it was generated or derived from the target domain's data. The chance that the sample came from the training distribution is represented by discriminator D's output. $G(z|x)$ and actual y will be inputs to the discriminator D, which will output whether the picture is real or fake; each of D's inputs has a $1/2$ probability of being true and $1/2$ probability of being false (acts as a binary classifier for the produced data, training to recognise the synthetic images as accurately as possible).

PatchGAN is a D that has been patched together. To decrease gradient sparsity, we utilised Leaky ReLU activation instead of ReLU activation on all D's levels and conducted Batch Normalization on all layers except the input. Instead of pooling operations, we used strided convolutions and zero padding to reduce information loss at the borders. D's architecture is depicted in further depth in Figure 18.

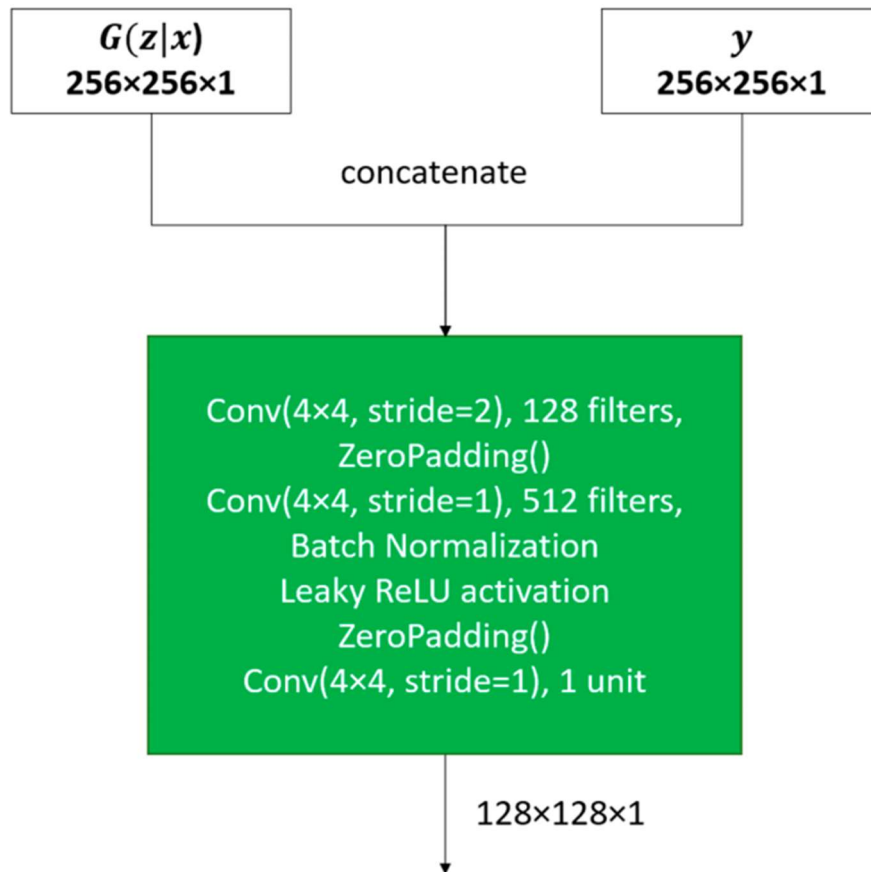


Fig. 18 : The architecture of the Discriminator

To compute perceptual loss, we employed the VGG19 pre-trained architecture. It's an extremely deep CNN that was trained on a big dataset and proven to be more accurate at extracting tiny characteristics than other pre-trained networks.

D classifies bigger picture patches (128×128 , instead of 32×32) to determine if the input is real or false by operating convolutionally over the 256×256 patch. The discriminator takes four 128×128 patches as input and classifies them according to the distribution they came from (real or fake). To get a final forecast for the input tile, the decision scores are summed.

These enhancements reduced the overall number of parameters from 2,770,433 to 1,071,105, a 61.3 percent reduction, when compared to the original PatchGAN [56] network utilised in Pix2pix.

4.6 Optimization

The objective function \mathcal{L} of our method consists of three vital components: Adversarial Loss \mathcal{L}_{GAN} , Categorical cross-entropy \mathcal{L}_{CCE} , and perceptual loss \mathcal{L}_C . In detail, the final loss function for our segmentation architecture, \mathcal{L}_{SGAN} , will be:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{CCE} + \lambda_2 \mathcal{L}_{FEAT} + \lambda_3 \mathcal{L}_{GAN} \quad (4)$$

Where λ_1 , λ_2 , and λ_3 are weighting hyperparameters of each loss.

Here we adopted categorical cross-entropy to calculate the distance between ground truth segmentation image and generated segmentation image because each pixel belongs to a particular class. Therefore, the discriminator task remains as it is, but the generator has to be near the pre-segmented image and fool the discriminator. So categorical cross-entropy is:

$$\mathcal{L}_{CCE}(y, G(x)) = \sum_i^N \sum_c^C G(x_{ic}) \log(y_{ic}) \quad (5)$$

Here x is from the original image, and y is the ground truth segmented image. $G(x_{ic})$ represents the probability of class c for generated pixel i and y_{ic} represents the probability of class c for ground truth pixel i .

Generated segmentation images will not be sharp, and the edges of objects will not be smooth and accurate without perceptual loss. It guides the generator to follow the content and object edges of an image x . The block5_conv2 layer of VGG19 is used to calculate perceptual loss because this layer contains the higher-level feature of an image which generally represents the shape and edges of objects. So, perceptual loss will be:

$$\mathcal{L}_{FEAT}(x, G(x)) = \frac{1}{C*H*W} \sum [\|x^l - G(x)^l\|_2] \quad (6)$$

Here l is the layer of VGG19.

The adversarial loss is used to improve and enhance the constructed result by generator, which can be defined as:

$$\mathcal{L}_{GAN}(G, D) = \min_G \max_D \mathbb{E}_{x,y} [\log D(x, y)] + \mathbb{E}_{x,y} [\log (1 - D(x, G(x)))] \quad (7)$$

CHAPTER 5: EXPERIMENTAL RESULTS

5.1 Experimental Setup

The first convolution layer contains 64 filters in both the generator and discriminator. We used upsampling factor and downsampling factor as 2 for every convolution and deconvolution block present in the generator and the discriminator with a kernel size of 3×3 . Both the networks use an Adam optimizer with a 0.0002 learning rate. Moreover, this type of architecture uses one as the batch size; we use the same and train for 200 epochs. λ_1 , λ_2 , and λ_3 are adversarial loss, perceptual loss, and categorical cross-entropy loss, respectively where $\lambda_1 = 1, \lambda_2 = 100$ and $\lambda_3 = 100$. Our networks were implemented with a Keras v2.0 with Tensorflow v1.15 framework

- **Tools Used**

In this part, we go through all of the software prerequisites that were required to complete our project, including the programming language(s), imported libraries, and so on.

1) Programming Platform: Python 3.6: Python is a popular programming language. It was created in 1991 by Guido van Rossum, a programmer, and has since been substantially expanded and utilised for numerous large-scale projects.

It's also an interpretive language. An interpreted language is a high-level language that is run and executed on the fly by an interpreter (a software that translates a high-level language to machine code and then executes it); the programme is processed one step at a time. It incorporates high-level programming, which is ideal for novices, and because of its simple syntax and wide range of import libraries, a programmer may focus on what to accomplish rather than how to do it.

2) Libraries used: The framework requires a number of libraries and open-source packages, including Tensorflow with Keras as the backend, and additional libraries essential for machine learning applications, such as scipy, numpy and pickle.

a) TensorFlow: The rise of computer science has resulted in an increase in the amount of data available. As a result, deep learning began to outperform all other machine learning techniques. To make the most of this potential, Google decided to utilise neural networks to improve its services:

- Gmail
- Photo
- Google search engine

TensorFlow was created to allow academics and developers to collaborate on AI projects. It provided a lot of people the opportunity to utilise it in its evolved and scaled version. In 2015, the first public version was published, followed by the first stable version in 2017. It is now open source thanks to the Apache Open Source licence. We are free to use, alter, and share it.

TensorFlow architecture has three parts for working:

- Pre-processing the data
- Build the model
- Train and estimate the model

TensorFlow was coined because it uses multi-dimensional arrays known as Tensors as inputs. We create a graph of functions, which is a type of flowchart that shows what we want to do with that input. We get data from one side, execute several operations on it, and receive an output from the other. This is why it's referred to as TensorFlow as input flows.

- b) Keras:** Although the deep neural networks are growing to become more and more popular, many frameworks are so complex that they have become a barrier to use it. Many high-level APIs have been proposed which are simple and better for developing neural networks, which look same but truly are very different on examining.

Keras is one of the most widely used high-level neural network APIs. It works with a variety of neural network engines on the backend. Keras offers a user-friendly interface that is modular, extendable, and simple to use with Python. Standalone modules like as optimisers, activation functions, cost functions, neural network layers, and so on can be mixed and matched to create new models. Classes and functions, for example, can be simply introduced as modules.

- c) Mat-plot Lib:** It provides excellent Python visualisations for 2D graphs. Matplotlib utilises the SciPy stack for broader use and is built on NumPy arrays for multiplatform display. John Hunter gave it to the audience in 2002. The main advantage of visualisation is that it allows us to visually perceive enormous amounts of data in easy-to-understand graphs and other formats. It includes graphs such as line, bar, scatter, histogram, and others.

- d) Numpy:** NumPy is the most important Python module for scientific computing. It's a Python library that includes a multidimensional array object, derived objects (such as masked arrays and matrices), and a variety of routines for performing fast array operations, such as random simulation, basic statistical operations, basic linear algebra, discrete Fourier transforms, I/O, selecting, sorting, shape manipulation, logical, mathematical and more.
- e) Pickle:** Pickle is a module that allows you to alter or modulate object structures in a way that is Python friendly and easy to deal with. With the aid of the pickle package, any type of Python object may be pickled and then written and saved on disc.

5.2 Evaluation Metrics

A segmentation system's performance must be extensively assessed in order for it to be helpful and make a genuine contribution to the area. The assessment has to be carried out by well-defined metrics that allow for comparisons fairly with current approaches. To establish a system's validity and usefulness, several factors such as accuracy, memory footprint, and execution time must be examined. Depending on the system's context or purpose, certain metrics may be more essential than others; for example, in a real-time application, accuracy could be compromised up to an execution speed point. All possible measurements for a given technique must be reported to preserve scientific rigor.

5.2.1 Accuracy

There have been a number of assessment criteria suggested, and they are commonly used to examine the accuracy of any semantic segmentation technique. These measurements are generally pixel accuracy and IoU fluctuations. For evaluation, how pixel labelling algorithms function in this work is the most often used semantical segmentation metrics. We stressed the following notation features in the sake of clarity: P_{ij} is the number of class i pixels that are deduced from class j as long as $k+1$ classes are available in total. In other words, p_{ii} represents the number of genuine positive, whereas p_{ij} and p_{ji} represent false positive and false negative.

- **Pixel Accuracy (PA):** It's the simplest straightforward metric, involving simply the calculation of a ratio between the number of correctly classified pixels and the total number of pixels.
- **Mean Pixel Accuracy (MPA):** a somewhat improved PA in which the correct-pixel ratio is determined per-class before being averaged across all classes
- **Mean Intersection over Union (MIoU):** This is the industry standard for segmentation. It computes the ratio of two sets, in this instance the ground truth and the segmentation we expect. The number of true positives (intersection) divided by the total number of true positives, false negatives, and false positives may be expressed as the sum of true positives, false negatives, and false positives (union). This IoU is computed each class before being averaged.

- **Frequency Weighted Intersection over Union (FWIoU):**
It's an improvement over the raw MIoU, which weights each class's significance based on how frequently it appears.

Because of its representativeness and simplicity, the MIoU is the most commonly used statistic among the measures listed above. The majority of challenges and researchers use this metric to communicate their findings.

5.2.2 Memory Footprint

Another key element in the development of segmentation algorithms is memory consumption. Although less restrictive than runtime is (typically higher memory capacity) the problem may still be limited. In some situations, RAM is not as large as in a High-Performance Server, such as on-board CPUs for robotic platforms. Another key element in the development of segmentation algorithms is memory consumption. It nevertheless may be a limiting issue even if memory capacity can typically be increased. In some situations, RAM is not as large as in a High-Performance Server, such as on-board CPUs for robotic platforms.

5.2.3 Execution Time

Because the great majority of systems must adhere to severe time constraints for the inference pass, speed, or runtime, is an extremely important measure. Knowing how long it takes to train a system can be beneficial in some situations, but because it is an offline operation, it is seldom important unless the system is extremely sluggish. In any event, giving precise times for the approaches is meaningless because they are so reliant on hardware and backend implementation that some comparisons become irrelevant.

For the purpose of repeatability and to assist colleagues, timings should be accompanied with a detailed description of the hardware on which the

experiment was conducted. This can help others to decide whether the Strategy is suitable for the application and perform fair comparisons in the same settings in order to discover which methods are the fastest. If done appropriately. But times should be coupled with a comprehensive explanation for the sake of repeatability and to assist fellow researchers of the hardware on which the system was operated, and the benchmark situations. This can allow others, when done correctly, to evaluate if the strategy is beneficial for the application or not and to do fairer comparisons under the same conditions in order to discover which approaches are quickest.

5.3 Results and Analysis

The Mean Intersection over Union (mIoU) score [8] is used to evaluate the architecture performance, which is the standard accuracy metric for semantic segmentation tasks.

Using U-net architecture in CGAN with perceptual loss, the proposed Segmentation GAN model has achieved a higher mIoU score than the original U-net. The obtained mIoU score is similar in such classes with larger region areas, e.g., road, sky, and buildings, compared to U-net as shown in Table III. But the proposed architecture managed to improve mIoU score of minor objects such as humans, traffic lights, signs, etc., because of perceptual loss adoption as shown in Table III. Hence overall mIoU score has improved.

The network managed to achieve an 83.30%, 86.90% and 97.5% mIoU score on the testing set, the validation set and the training set, respectively. It shows that more present classes in the dataset are classified accurately. The comparison table of segmentation methods performance is provided in Table II.

Table II - Segmentation performance on the cityscapes dataset

Method	IoU Score%
FCN-8s [8]	65.30
PSPNet [9]	81.20
DeepLabv2-CRF [30]	70.40
DeepLabv3 [31]	81.30
DeepLabv3+ [32]	82.10
U-net [11]	69.10
Proposed method	83.30

Table III - Segmentation performance on the cityscapes dataset on class bases

Class	Ours	FCN	PSPNET	DeepLabv2	DeepLabv3	DeepLabv3+	U-net
Bicycle	81.8198	66.7635	77.5388	68.8479	78.2985	78.8796	66.2731
Motorcycle	77.7488	51.569	70.8048	57.6633	72.0899	73.8367	53.1812
Train	87.4274	46.5414	83.6401	57.4574	85.0893	83.9089	59.4483
Bus	94.9137	48.5751	91.5102	67.4976	90.3914	90.9143	64.4488
Truck	81.728	35.2722	77.6988	56.5019	75.086	78.0207	47.3863
Car	96.8228	92.628	96.2174	93.7134	96.3178	96.4068	92.887
Rider	85.8129	51.4129	71.9132	59.8495	73.3587	73.2603	57.8774
Person	90.0094	77.1373	86.8317	79.8312	87.6126	87.953	77.8652
Sky	96.2946	93.8604	95.2983	94.192	95.8643	95.8471	94.2097
Terrain	75.1504	69.2969	72.2031	69.4396	72.3212	73.0359	69.4297
Vegetation	94.57	91.4171	93.6399	91.8508	93.7959	93.9698	91.8746
Traffic sign	92.851	65.0173	80.474	67.2847	81.3333	82.1568	65.3923
Traffic light	90.0552	60.0832	76.1225	57.8685	77.0897	78.163	61.1365
Poll	80.2949	47.4143	67.6716	49.5789	70.0424	71.3946	54.1891

Fence	76.1869	44.2369	63.6752	47.3634	63.2389	63.7375	47.0695
Wall	79.1813	34.9328	58.3855	48.7736	55.1757	59.4754	41.7583
Building	95.0358	89.2114	93.4688	90.35	93.5295	93.9102	90.2109
Sidewalk	88.7003	78.4065	86.9233	81.3219	86.1916	87.0411	81.1635
Road	98.8659	97.406	98.6814	97.8649	98.5931	98.6939	97.7055

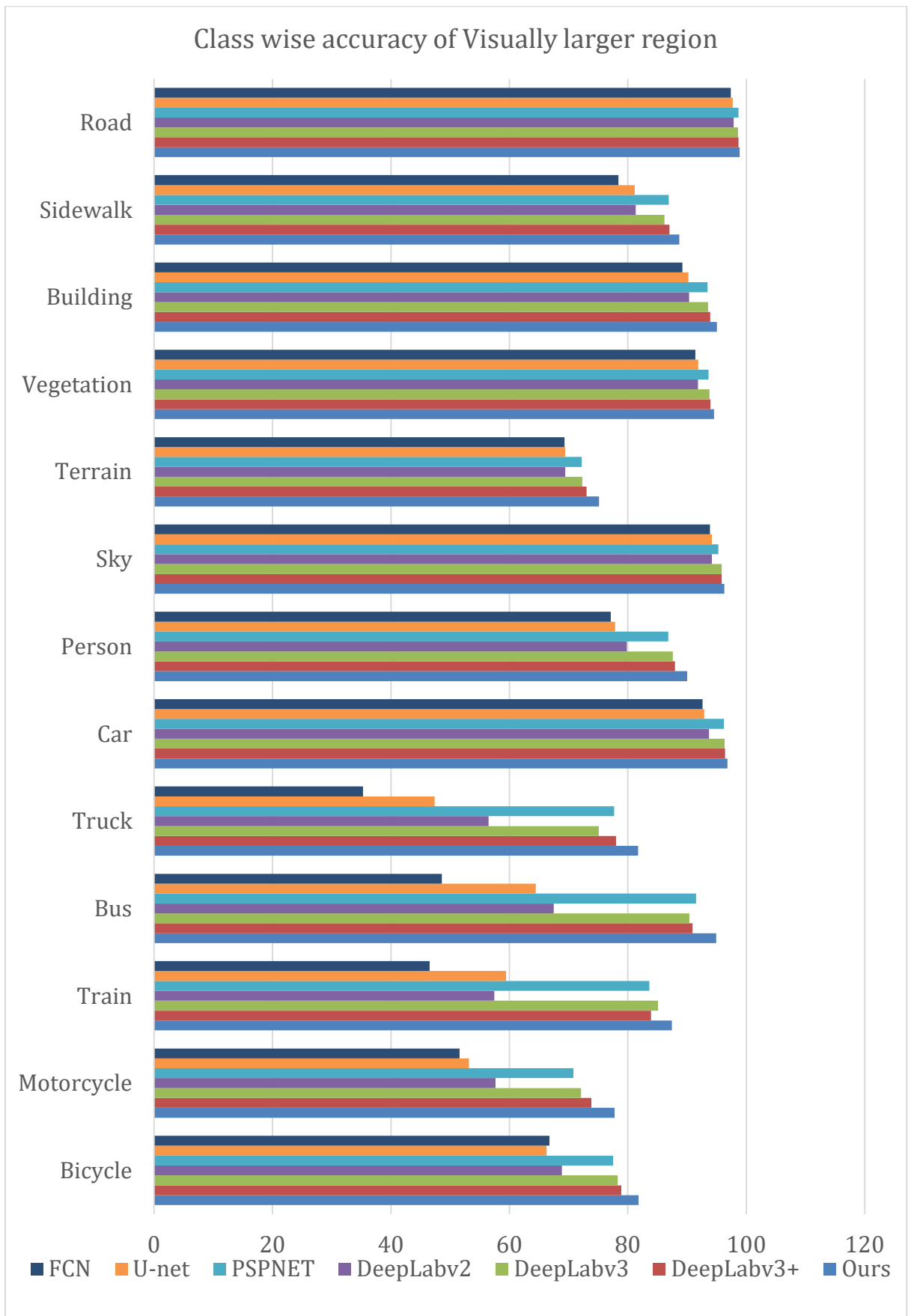


Fig. 19 : Class wise accuracy of Visually larger region of different methods

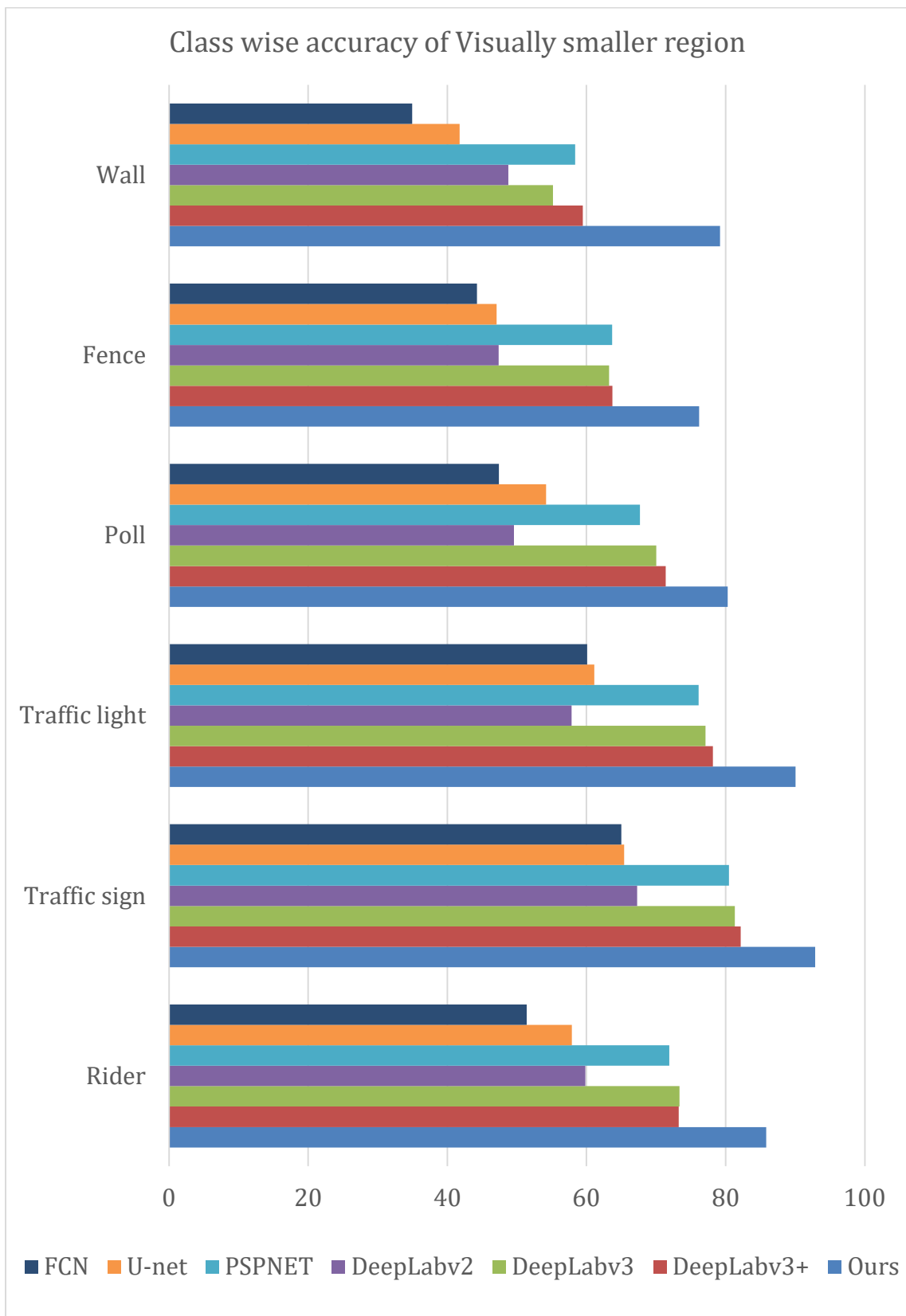


Fig. 20 : Class wise accuracy of Visually smaller region of different methods

IoU Score of all the models mentioned in Table IV. are available in cityscapes dataset site [53] except U-net. We trained U-net separately in same hardware and software environmental setup. To compare the proposed method with other GAN-based segmentation models, we used the PASCAL VOC 2012 dataset [57] for the experiment. Our method achieved a better result than other GAN-based segmentation models because this dataset includes a smaller number of categories, and objects presented in the dataset are also visually very large compared to the cityscapes dataset. The comparison table for GAN based segmentation model is provided in Table IV.

Table IV - Segmentation performance on the pascal voc dataset

GAN based method	IoU Score%
INRIA et al. 2016 [47]	54.30
SegGAN [48]	69.90
Proposed method	82.10

Fig. 21 and Fig. 22 are training and validation loss history chart with perceptual loss and without perceptual loss respectively. A model without perceptual loss able to achieve 94% training and 82% validation accuracy whereas it able to achieve 97.5% training and 87.9% validation accuracy with perceptual loss in 200 epochs. Fig. 23, Fig. 24 are results captured at 100 and 200 epochs and Fig. 25 is results after complete training and saving weights of model and loading it again using pickle [58] [59].

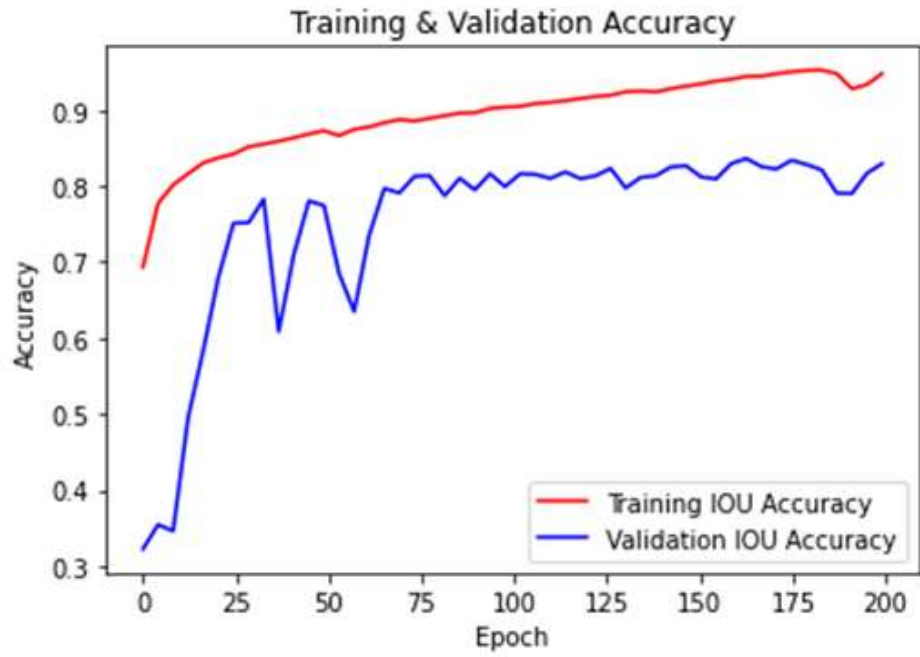


Fig. 21 : Training and Validation Accuracy without Perceptual Loss

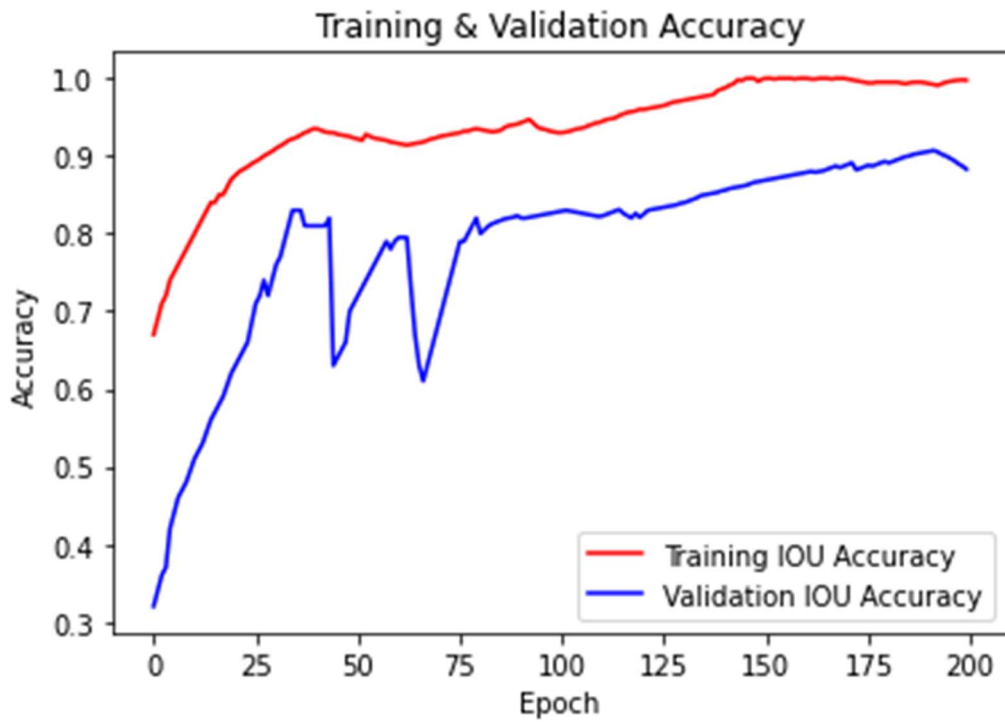


Fig. 22 : Training and Validation Accuracy with Perceptual Loss

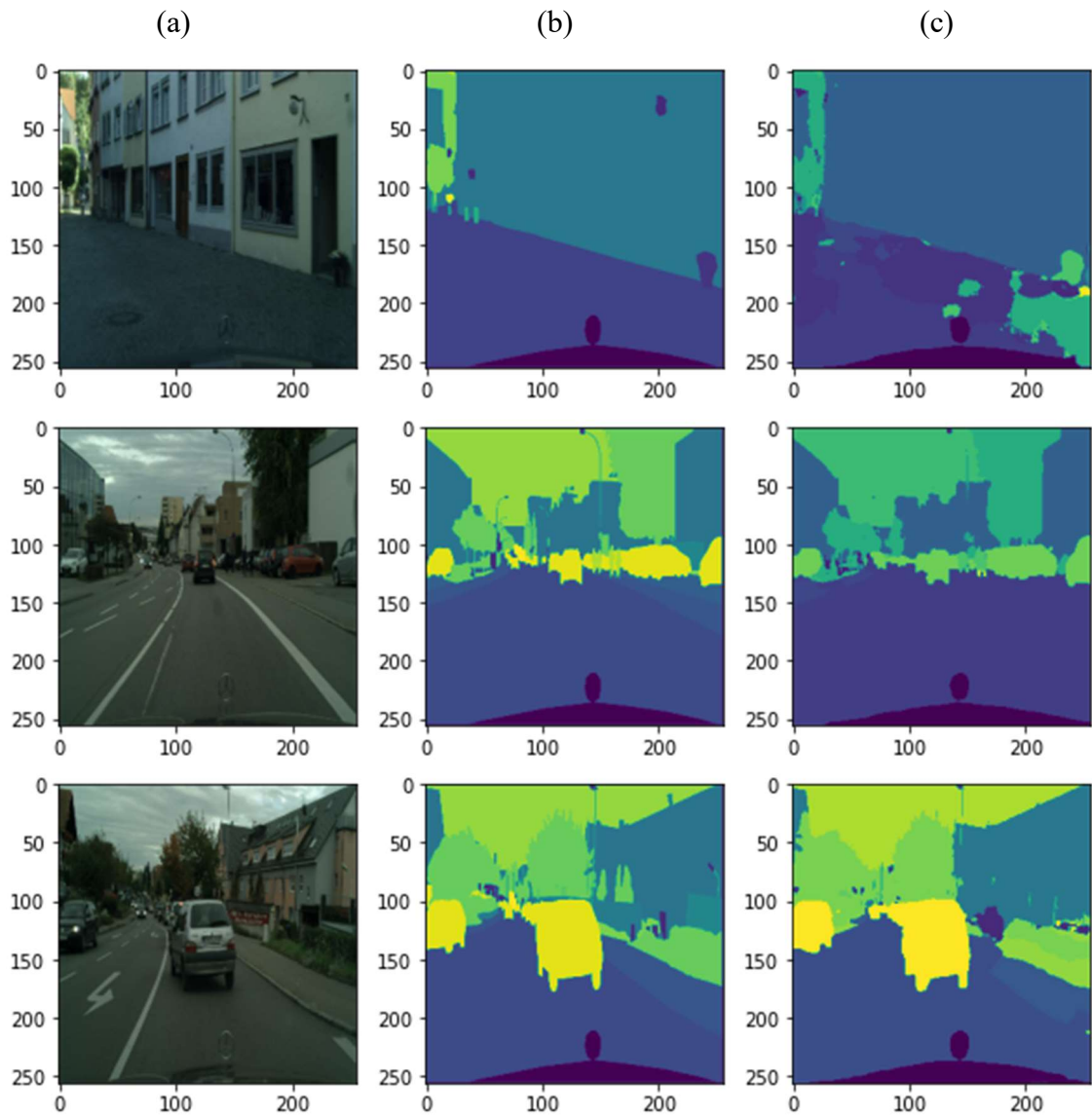


Fig. 23 : Output sample generated by proposed model at 100 epochs

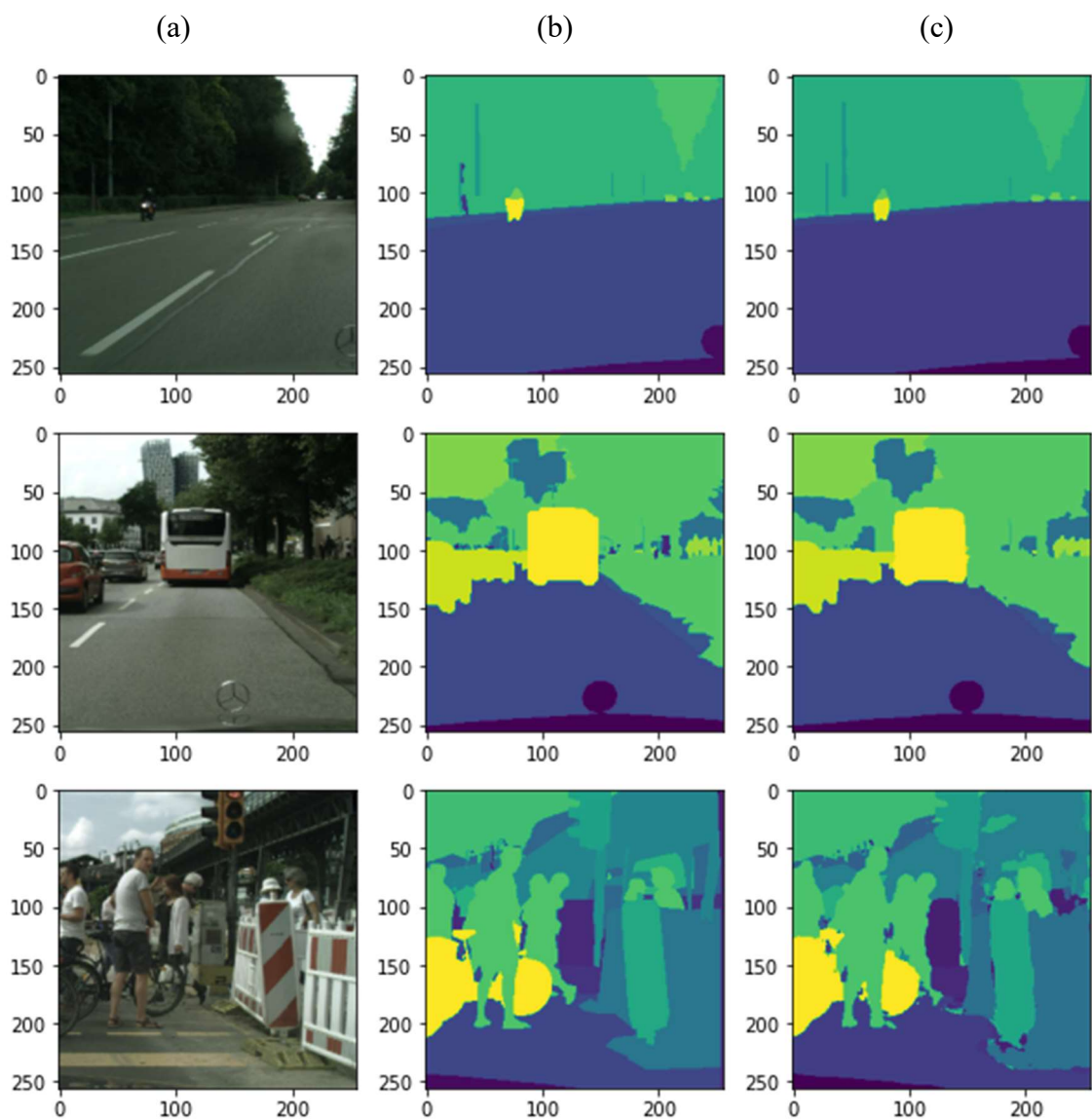


Fig. 24 : Output sample generated by proposed model at 200 epochs

Here are final results generated by the proposed architecture in cityscapes dataset.

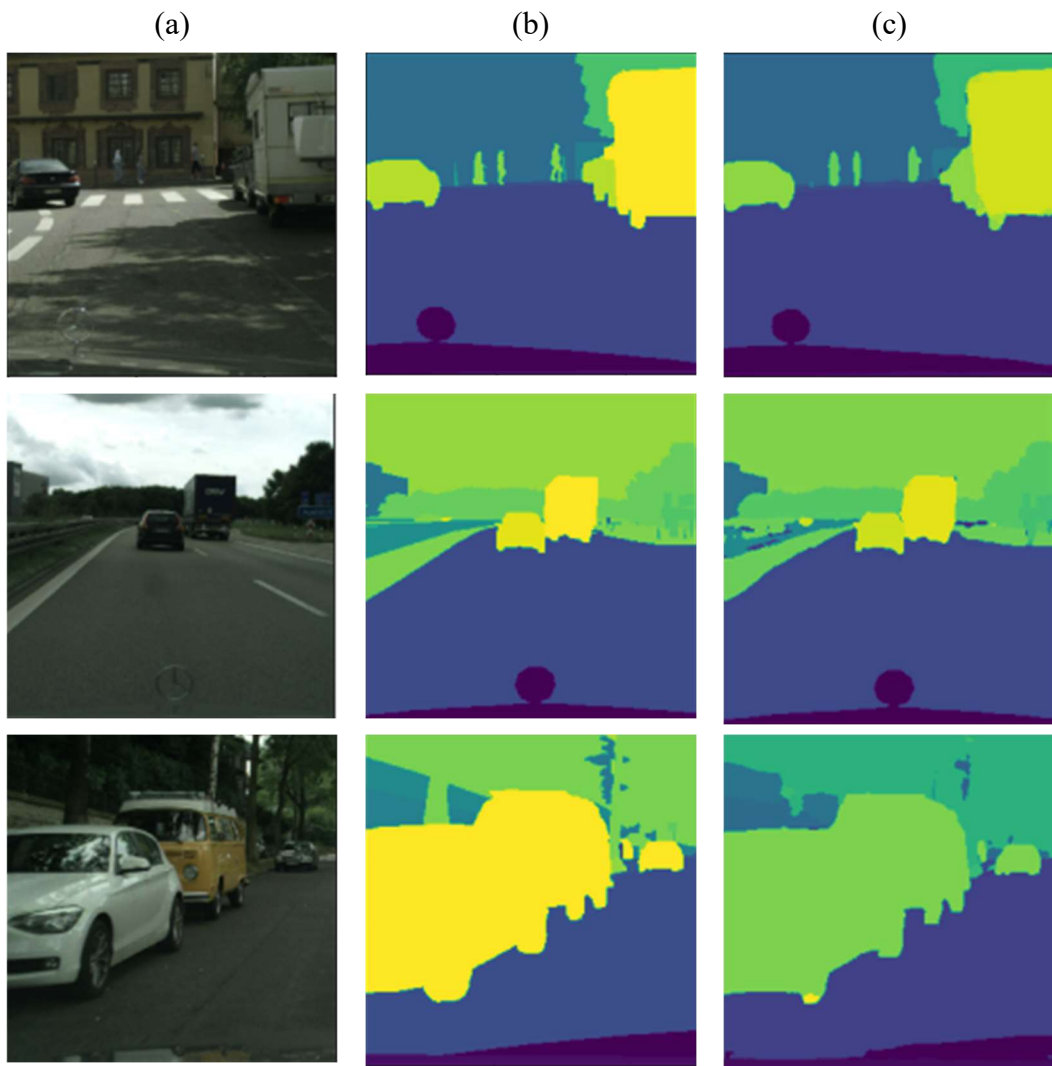


Fig. 25 : Output sample generated by proposed model (a) original (b) ground truth (c) output

CHAPTER 5 : CONCLUSION

In this thesis, we proposed a GAN-based semantic segmentation method with the aim to identify more minor region details of an image and improve the segmentation model (generator) itself by competing with the discriminator. This aim has been fulfilled by employing perceptual loss and adversarial loss. The several experiment results on the cityscapes dataset prove that the proposed method has significantly enhanced the quality and performance of the existing conventional and GAN-based semantic segmentation models. The proposed method has achieved highest accuracy among all GAN-based methods in semantic segmentation. In this method, we only adopted single layer for calculate perceptual loss, but we also can use multiple layers for feature extraction. We haven't tested this method on datasets which are taken during multiple environment conditions such as daylight, night, dusk, dawn, haze, fog, snow, and rain. It might lead to different result.

RELATED PUBLICATIONS

- [58] G. Sohaliya and K. Sharma, “Semantic Segmentation using Generative Adversarial Networks with a Feature Reconstruction Loss,” in *IEEE Asian Conference on Innovation in Technology*, 2021.
- [59] G. Sohaliya and K. Sharma, “An Evolution of Style Transfer from Artistic to Photorealistic : A Review,” in *IEEE Asian Conference on Innovation in Technology*, 2021.

REFERENCES

- [1] Y. Ohta, T. Kanade and T. Sakai, “An Analysis System for Scenes Containing objects with Substructures,” in *Proceedings of 4th International Joint Conference on Pattern Recognition (IJCPR '78)*, 1978.
- [2] S. Edelman and T. Poggio, “Integrating visual cues for object segmentation and recognition,” *Optics News*, vol. 15, p. 8, 1989.
- [3] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez and J. G. Rodríguez, “A Review on Deep Learning Techniques Applied to Semantic Segmentation,” *CoRR*, vol. abs/1704.06857, 2017.
- [4] A. Ess, T. Mueller, H. Grabner and L. Van Gool, “Segmentation-Based Urban Traffic Scene Understanding,” in *BMVC*, 2009.
- [5] M. Oberweger, P. Wohlhart and V. Lepetit, “Hands Deep in Deep Learning for Hand Pose Estimation,” *CoRR*, vol. abs/1502.06807, 2015.
- [6] Y. Yoon, H.-G. Jeon, D. Yoo, J.-Y. Lee and I. So Kweon, “Learning a deep convolutional network for light-field image super-resolution,” in *Proceedings of the IEEE international conference on computer vision workshops*, 2015.

- [7] D. Ciresan, A. Giusti, L. Gambardella and J. Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” *Advances in neural information processing systems*, vol. 25, p. 2843–2851, 2012.
- [8] J. Long, E. Shelhamer and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” *CoRR*, vol. abs/1411.4038, 2014.
- [9] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” *arXiv preprint arXiv:1412.7062*, 2014.
- [11] O. Ronneberger, P. Fischer and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, 2015.
- [12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative adversarial networks,” *arXiv preprint arXiv:1406.2661*, 2014.
- [13] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [14] J. Johnson, A. Alahi and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European conference on computer vision*, 2016.
- [15] C. Farabet, C. Couprie, L. Najman and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, p. 1915–1929, 2012.
- [16] C. Couprie, C. Farabet, L. Najman and Y. LeCun, “Indoor semantic segmentation using depth information,” *arXiv preprint arXiv:1301.3572*, 2013.
- [17] M. Mostajabi, P. Yadollahpour and G. Shakhnarovich, “Feedforward semantic segmentation with zoom-out features,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.
- [18] R. C. P. H. O. Pinherio and H. Pedro, “Recurrent convolutional neural networks for scene parsing,” in *International Conference of Machine Learning*, 2014.

- [19] S. Hong, H. Noh and B. Han, “Decoupled deep neural network for semi-supervised semantic segmentation,” *arXiv preprint arXiv:1506.04924*, 2015.
- [20] V. Badrinarayanan, A. Kendall and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, p. 2481–2495, 2017.
- [21] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.
- [22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, “Object detectors emerge in deep scene cnns,” *arXiv preprint arXiv:1412.6856*, 2014.
- [23] G. Bertasius, L. Torresani, S. X. Yu and J. Shi, “Convolutional random walk networks for semantic image segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [24] G. Lin, C. Shen, A. Van Den Hengel and I. Reid, “Efficient piecewise training of deep structured models for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [25] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang and P. H. S. Torr, “Conditional random fields as recurrent neural networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015.
- [26] A. Arnab, S. Jayasumana, S. Zheng and P. H. S. Torr, “Higher order conditional random fields in deep neural networks,” in *European Conference on Computer Vision*, 2016.
- [27] Z. Liu, X. Li, P. Luo, C. C. Loy and X. Tang, “Semantic Image Segmentation via Deep Parsing Network,” *CoRR*, vol. abs/1509.02634, 2015.
- [28] R. Vemulapalli, O. Tuzel, M.-Y. Liu and R. Chellappa, “Gaussian Conditional Random Field Network for Semantic Segmentation,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [29] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci and A. Courville, “Reseg: A recurrent neural network-based model for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016.

- [30] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834-848, 2018.
- [31] L.-C. Chen, G. Papandreou, F. Schroff and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation,” *CoRR*, vol. abs/1706.05587, 2017.
- [32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation,” *CoRR*, vol. abs/1802.02611, 2018.
- [33] S. Suri, A. Gupta and K. Sharma, “Comparative Analysis of Ranking Algorithms Used On Web,” *Annals of Emerging Technologies in Computing (AETiC)*, vol. 4, 2020.
- [34] A. K. Tripathi, K. Sharma, M. Bala, A. Kumar, V. G. Menon and A. K. Bashir, “A Parallel Military-Dog-Based Algorithm for Clustering Big Data in Cognitive Industrial Internet of Things,” *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 2134-2142, 2021.
- [35] D. Pathak, E. Shelhamer, J. Long and T. Darrell, “Fully convolutional multi-class multiple instance learning,” *arXiv preprint arXiv:1412.7144*, 2014.
- [36] G. Papandreou, L.-C. Chen, K. P. Murphy and A. L. Yuille, “Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation,” in *Proceedings of the IEEE international conference on computer vision*, 2015.
- [37] D. Pathak, P. Krahenbuhl and T. Darrell, “Constrained convolutional neural networks for weakly supervised segmentation,” in *Proceedings of the IEEE international conference on computer vision*, 2015.
- [38] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao and S. Yan, “Stc: A simple to complex framework for weakly-supervised semantic segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, p. 2314–2320, 2016.

- [39] N. Souly, C. Spampinato and M. Shah, “Semi and weakly supervised semantic segmentation using generative adversarial network,” *arXiv preprint arXiv:1703.09695*, 2017.
- [40] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017.
- [41] Y. Li, L. Yuan and N. Vasconcelos, “Bidirectional learning for domain adaptation of semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [42] P. Li, X. Liang, D. Jia and E. P. Xing, “Semantic-aware grad-gan for virtual-to-real urban scene adaption,” *arXiv preprint arXiv:1801.01726*, 2018.
- [43] R. Gong, W. Li, Y. Chen and L. V. Gool, “Dlow: Domain flow for adaptation and generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [44] J. Choi, T. Kim and C. Kim, “Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.
- [45] W. Hong, Z. Wang, M. Yang and J. Yuan, “Conditional generative adversarial network for structured domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [46] J. Yang, W. An, S. Wang, X. Zhu, C. Yan and J. Huang, “Label-driven reconstruction for domain adaptation in semantic segmentation,” in *European Conference on Computer Vision*, 2020.
- [47] P. Luc, C. Couprie, S. Chintala and J. Verbeek, “Semantic Segmentation using Adversarial Networks,” in *NIPS Workshop on Adversarial Training*, Barcelona, 2016.
- [48] X. Zhang, X. Zhu, X. Zhang, N. Zhang, P. Li and L. Wang, “SegGAN: Semantic Segmentation with Generative Adversarial Network,” in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, 2018.

- [49] G. Neuhold, T. Ollmann, S. Rota Bulo and P. Kotschieder, “The mapillary vistas dataset for semantic understanding of street scenes,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [50] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin and R. Yang, “The apolloscape dataset for autonomous driving,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.
- [51] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, p. 1231–1237, 2013.
- [52] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan and T. Darrell, “Bdd100k: A diverse driving dataset for heterogeneous multitask learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.
- [53] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth and B. Schiele, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” *CoRR*, vol. abs/1604.01685, 2016.
- [54] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [55] L. A. Gatys, A. S. Ecker and M. Bethge, “A Neural Algorithm of Artistic Style,” *CoRR*, vol. abs/1508.06576, 2015.
- [56] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” *CoRR*, vol. abs/1611.07004, 2016.
- [57] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, p. 303–338, 2010.
- [58] Goldesel. [Online]. Available: https://github.com/Goldesel23/Temporal-Ensembling-for-Semi-Supervised-Learning/blob/master/pi_model.py.