ALMA MATER STUDIORUM · UNIVERSITY OF BOLOGNA

SCHOOL OF ENGINEERING AND ARCHITECTURE
Second Cycle Degree in Automation Engineering

Lane Marking Segmentation in the Autonomous Driving Scenario with Deep Convolution Neural Networks

Supervisor:

Prof. Luigi Di Stefano

Advisor:

Dott. Luca De Luigi

Presented by:

Fikrat Gasimov

Graduation Session iii
Academic year 2017/18

Abstract

Lane Marking Segmentation is research domain on which many techniques have been proposed so far, as well as, this thesis explains how to tackle facing problems in the Autonomous Driving, on the deep learning scenarios. Previously, several approachs, such deeplab[4], and its extensions have deeply been experienced with multiple datasets, such ImageNet[11], CityScape[10] and so on. Accordingly, they all represent competitive outcomes, in terms of lass Accuracy, Mean Intersection Over Union. However, this thesis represents flatly different approach on Lane Marking Semantic Segmentation, aiming at improving seen results, and ending up with compeletly new techniques. In this respect, DeepLabV3 Plus [7], extending DeepLabV3[5], is trained on ApolloScape dataset. This large-scale dataset contains a diverse set of stereo video cropped sequences, recorded in street scenes from different cities, with high quality pixel-level annotations of 110 000+ frames. Xception network [8], being extension of InceptionV3 network[8], is considered on such a particular research, together with DeepLabV3 plus.

Proposed solution has variety of advantages in the task of Semantic Segmentation, in terms of providing, new techniques such as encoder-decoder network, Spatial Pyramid Pooling with Parallel Atrous Convolution layers[7], Depthwise Separable Convolution as well as Multi-Grid Method which all are broadly discussed in this thesis. Regardless of several state-of-art methods, there are possible challenges, needed to be explicitly analyzed, and taken good measures, to propose superior achievements. Confronting difficulties in Semantic Segmentation, concerns to obtained results, related to mIoU and Class Accuracy of 38 classes, which are, in turn, caused by Cross Entropy Loss for unbalanced dataset and Random Scale Crop function which operates on randomly scaling ground-truth images, resulting disappearance reasonable information on images, on the other hand, having 38 classes on images, bring challenge to network to classify and semantically label, great variety of road signs on the images. Accordingly, there are experimented methods suggested for those ongoing issues, for example, replacement of Cross Entropy loss, with Weighted-Cross-Entropy Loss, Random Scale with Standard Random Crop, moreover, deployment of Center Crop technique and training with two classes, namely, ”Lane Marking” and ”Non Lane Marking”, all in all have dramatically improved previous outcomes, in particular, with advent of Weighted Cross Entropy Loss.

Acknowledgements

I would like to acknowledge and thanks following important people who supported me not only during the course of master thesis, but also thorough my master degree at the University of Bologna.

I would like to acknowledge my gratitude and render my warmest thanks to my supervisor, Prof. Luigi Di Stefano who made this work possible. His friendly guidance and expert advice have been invaluable throughout all stages of the work.

I would also wish to thanks to my tutor, Dott. Luca De Luigi for extended discussions and valuable suggestions which have contributed greatly to the improvement of the thesis.

Finally, thanks are due to my family and close friends, for their continuous support, deep patience. You have not only, all encouraged and believed in me, but also helped me to focus on what has been a hugely rewarding and enriching the process.

This thesis has been written during my stay in the Computer Vision Lab of University of Bologna.

Contents

List of Tables

List of Figures

.

Chapter 1

Introduction

Semantic Segmentation is a natural step in the progression from coarse to fine inference. The origin could be located at classification, which consists of making a prediction for a whole input. The next step is localization, detection, which provides not only the classes but also additional information regarding the spatial location of those classes.Finally, semantic segmentation achieves fine-grained image classification by making dense predictions inferring labels for every pixel, so that each pixel is labeled with the class of its enclosing object region. Semantic Segmentation is widely used in different purposes such as medical, robot vision and understanding, as well as, autonomous driving (Fig 1.2)deep learning tasks. In this respect, these classes could be pedestrians, vehicles, buildings, vegetation, sky, void and so on. As it is clear from the concept that DeepLabV3 Plus comes up with an particular layer, called Convolution Layer, in this respect, convolution layers have been broadly investigated.Convolutional Neural Network (CNN) architecture has three main parts. A convolutional layer that extracts features from a source image. Convolution helps with blurring, sharpening, edge detection, noise reduction, or other operations that can help the machine to learn specific characteristics of an image. Pooling layer that reduces the image dimensionality without losing important features or patterns. Fully connected layer also known as the dense layer, in which the results of the convolutional layers are fed through one or more neural layers to generate a prediction.Then, Lane Marking Semantic Segmentation with Deep Convolution Neural Network, is one of most fascinating, in the same time, challenging research, because Lane Marking is specific

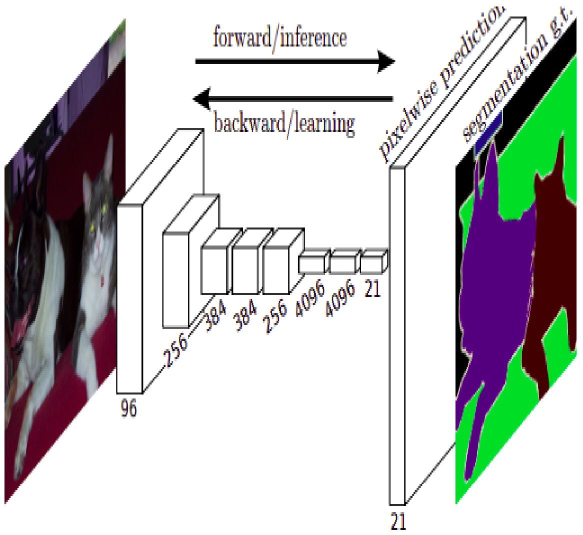


Figure 1.1: Fully Convolution Network Deployment

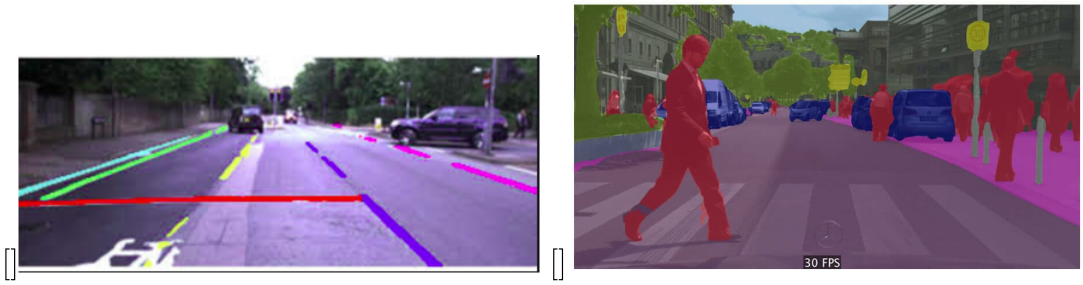

[] []

Figure 1.2: Lane Marking Segmentation(left) and Semantic Segmentation(right)

type of Semantic Segmentation, thus, our target is to develop Lane Marking Semantic Segmentation, by Deep Convolution Network. Lane Marking Segmentation principle is expressed by the left part of Fig 1.2
Moreover, some semantic networks comprises Fully Convolution networks(Fig 1.1), and they do not have Fully Connected Layers, but only convolution layers. It is evident that ConvNet has dominance on image classification as well as image segmentation.It is well-controlled on the internal task, with obtained outcome. According to these traits, ConvNet has great progression in the task of object detection and local correspondence.

Semantic Segmentation broadly used convnet, labelling invidual pixel with class of the object. Apart from this, Fully Convolution Network illustrates great performance in semantic segmentation, by training, pixel by pixel, end-to-end, without extra tools or methods.

1.1 Aims and Objectives

In order to contribute to this a project, the reasonable measures we follow are:

1. Main goal is refine semantic segmentation achievements, seen in the former experimental results, via ApolloScape Dataset, Xception Network, DeepLabV3 Plus.

2. As second part of primary target is to take advantage of multiple techniques, so as to achieve high class accuracy and intersection over union.

1.2 Description of the work

The overview of complete master thesis document:

In **Chapter 1** We introduce main ideas behind semantic and lane marking segmentation

In **Chapter 2** We comprehensively discuss DeepLab and its successors

In **Chapter 3** DeepLabV3 Plus and Xception module have both been broadly probed.

In **Chapter 4** Ultimately, we represent all considered measures to deal with facing difficulties, moreover their dramatic improvements on the Lane Marking Segmentation Task

In **Chapter 5** Finally, we mentioned several bibliographies cited in the thesis.

Chapter 2

Study on Semantic Segmentation and DeepLab

.

2.1 Apolloscape Dataset Description

We proposed to take advantage of ApolloScape Dataset for the task of Lane Marking Segmentation with Deep Convolution Neural Network. ApolloScape Dataset consists of 15 thousands of cropped images derived from real-world highway.Apolloscape Dataset is splitted into training and test set. Moreover, Training Set consist 12 thousands images, whereas validations set comprises 3 thousands images, however,format of existing ground-truth images is not suitable for fitting into network. In this respect, we tackle with this issue, by preceding on building new ground-truth images which will be further fitted into original network. Following the suggested way, we come up with Data PreProcessing Technique that is broadly described in the following section.

2.2 Data Preprocessing

It is a data mining technique that transforms raw data into an understandable format. Raw data(real world data) is always incomplete and that data cannot be sent through a model or network. That would cause certain errors. That is why we need to preprocess data before sending through a model. Following procedure opens up accurate data on which all steps are implemented:

1. Extraction of RGB color set from row data of Apolloscape dataset.
2. Consider labels encoded, according to road signs on ApolloScape Dataset.
3. Preparation of Ground-Truth values within the range of [0 255]
4. Replacement of RGB unique colors with Ground-Truth values, in order to provide Ground-Truth Images
5. Analyse ApolloScape Dataset, so as to understand multiple labels on highway lines of images

Table 2.1 displayes ID-Labels and conversion of RGB Colors, into Ground-Truth:

RGB Colors	Ground-Truth	ID-Labels
0 0 0	0	Void
70 130 180	1	Dividing
220 20 60	2	Middle Parallel line
128 0 128	3	Right line Parking
255 0 0	4	Border Line
0 0 60	5	Continuous Line
0 60 100	6	Right Turn
0 0 142	7	Guiding
119 11 32	8	Left Dash-line
244 35 232	9	Cycle line
0 0 160	10	Thick-Guide Line
153 153 153	11	Stopping
220 220 0	12	Convergence Line
250 170 30	13	Safety line
102 102 156	14	Left-Right Dash Separated Line
128 0 0	15	Chevron
238 232 170	17	Turn-Left Line
190 153 153	18	Zebra-Crossing
0 0 230	19	Double-Turn
128 78 160	21	Continuous Line
150 100 100	22	Circle Turn
255 165 0	23	Yield Sign without Words
180 165 180	24	Turn-Left
107 142 35	25	Three Merged Lines
201 255 229	26	Double Turn
0 191 255	27	Straight Line
51 255 51	28	Right Arrow
250 128 114	29	Pedestrian stand Line
127 255 0	30	Turn Right
255 128 0	31	Reduction
0 255 255	32	Attention
178 132 190	33	No Parking
128 128 64	34	Turn Allowance Line
102 0 204	35	Parking
0 153 153	36	Curve Sign
255 255 255	37	Ignored

Table 2.1: Clear Representation of Road Labels, with both, RGB and Ground-Truth Values

2.3 DeepLab

Primarily, DeepLab [4] is thoroughly investigated, according to its former [5] and state-of-art techniques [7].DeepLab introduces useful attributes in semantic segmentation task, such as Atrous Convolution, Fully Connected Conditional Random Field, as well as Deep Convolution Nets on which it did several refinements through the years.

1. DeepLab precedes with particular convolution, together with upsampled filter, in other words, Atrous convolution, with which DeepLab gains well-defined performance in dense prediction tasks, by guiding resolution on feature responses, computed in Deep Convolution Neural Networks. Moreover, Atrous convolution further represent another advantage of expanding field of view of filters, so as to put together larger contexts, without exhibiting any modification on amount of parameters it has.

2. On the other hand, there is Atrous Spatial Pyramid Pooling, segments object solidly in various rates. ASPP from [4], leans back an idea of existence of atrous convolution with filters, aiming at absorbing image context and objects on multiple rates [5].

3. Existence of max-pooling and downsampling acquire invariance from the [4], in Deep Convolution Neural Network, but has bad effect on localization accuracy, then good measure is to incorporate responses with Fully Conditional Random Field, introduced by the [7]

Possible challenges from [4], surrounds semantic segmentation task, are reduction in feature resolution, existence of objects at multiple scale, as well as ruined localization accuracy, caused by combination of max-pooling and downsampling.

2.3.1 Atrous Convolution

In ths chapter, we will more figure out ongoing corresponding challenges and their proper solutions, such that integral part is to improve reduced feature resolution which is caused by repeated combination of max pooling and downsampling at consecutive Deep Convo-

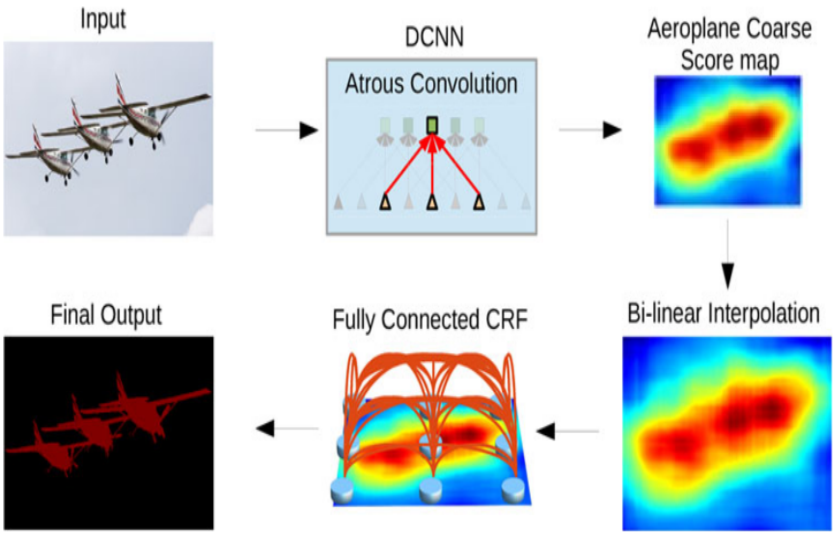


Figure 2.1: Sequence Illustration of Atrous Convolution, mapping through Binilear Interpolation, which expands feature map to original image size, then fit into Fully Connected CRF

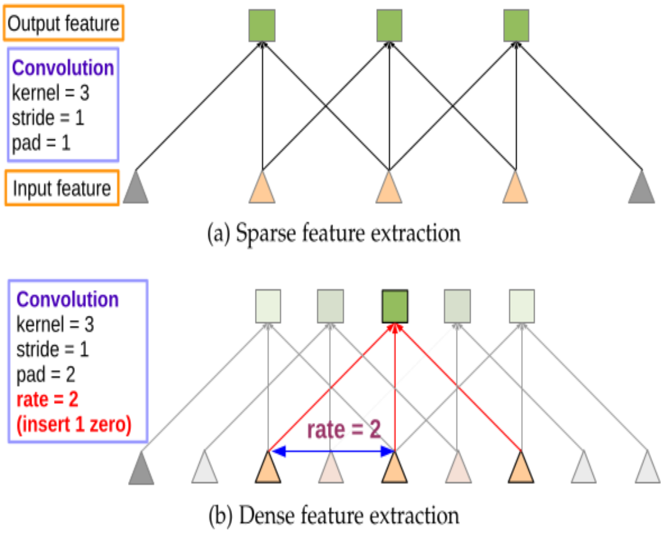
lution Neural Layers, this performs significantly reduction of resolution on feature map, when DCNN is employed in Fully Convolutional Fashion [18]

Diminished feature resolution is recovered, with an advent of experimented technique of extracting downsampling layers, amongst last max pooling layers, furthermore, appending upsample filters in successive convolution layers. Then, feature map is measure at high sampling rate.

Both, Atrous Convolution and Bilinear Interpolation of feature responses ,offer dense computation of feature maps, resulting full resolution feature map [4].

It has to be noted that Deep Convolution Neural Network can be trained in both, image classification and image segmentation, however, with fully connected layers as well fully connected convolution layers, respectively. From Fig 2.1 in [4], it is discussed that atrous convolution enhance feature resolution, then is pursued by bilinear interpolation, with aim of upsampling score map in Fig 2.1, to obtain original image resolution. As it is depicted, Fully Connected CRF is added to elaborate segmentation results. By considering one dimensional signal first, then output y[i] of atrous convolution of d input signal x[i] with a filter w[k] of length K is defined as shown in the 2.1 equation:

$$y[i] = \sum_{n=1}^{K=1} x[i + r \cdot k]w[k] \tag{2.1}$$



(a) Sparse feature extraction

(b) Dense feature extraction

Figure 2.2: Atrous and Standard Convolution (a) Standard convolution with rate r =1 . (b) Atrous convolution with rate r=2 applied on a high resolution input feature map

As it is given in equation 2.1, parameter r opens up with stride with which input signal is sampled. It should be noticed that standard convolution will be transformed into atrous with stride 2.

Clear exposition and dramatic enhancement on feature map, are introduced by both ideas on the Fig 2.3, such as standard and atrous convolution. Primarily, giving input image, standard convolution is applied with sparse feature extraction. In first case, downsampling operation is implemented with stride two, followed by convolution layers, ultimately, layers upsampled with filers, leading to implicit and low resolution input feature map. Conversely, superior method, Atrous convolution is proposed, with the stride r = 2, kernel size = 7, and stride = 1, so as to capture dense feature, at high resolution input.

It is broadly seen that Atrous Convolution is exceedingly powerful in high resolution of feature extraction. Going more detailed with it, experimentally, Atrous deploys small kernels, so as to provide fast computation and keep number of parameters unchanged. As well, Atrous Convolution beginning with stride r, generating r-1 zeros in the sequenced filter values, and broaden kernel size of k filter, to the k + (k - 1)(r - 1), without increasing number of parameters.

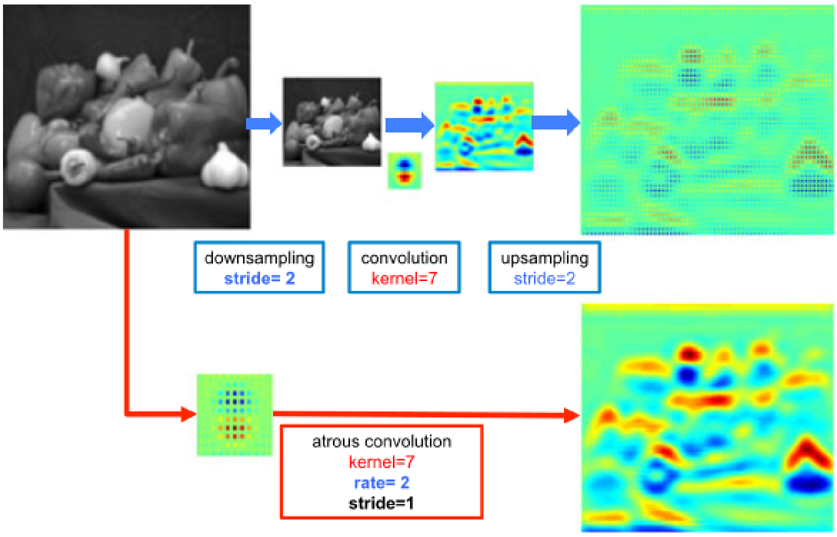


Figure 2.3: Various Feature Extraction Method with two kind of:Standard and Atrous convolution

2.3.2 Atrous Spatial Pyramid Pooling

Second facing challenge is existence of object at multiple scales, therefore, this problem had been figured out, applying rescaled version of DCNN on the same image, then further aggregate feature map . This particular approach, although increase performance, nonetheless, appears with cost of computing features responses at all DCNN layers for multi scale version of image. In this case, with aim of alleviating this issue, on the behalf of motivating Spatial Pyramid Pooling, that playes main role, and in turn provide computationally efficient scheme of resampling given feature layer at multiple rates, prior to convolution layer. With these incoming features, this appears to be probed on original image with multiple layers that has alluring effective field of views, thus capturing objects as well as useful image context at multiple scales. On the other hand, Deep Convolution Neural Network illustrates great representation by constituting small and large object scale. In principal, Atrous Spatial Pyramid Pooling is based on the concept from [13], which discusses achievement of R-CNN spatial pyramid pooling on scale of object regions on which it produces explicit, accurate outcome, once it re-generates convolution features, obtained at each single scale. In turn, this will then come up with an idea of using multiple parallel atrous convolution layers, with different sampling rates [5], thus after achieving features, at every sampling rate, they will be in turn, processed on various

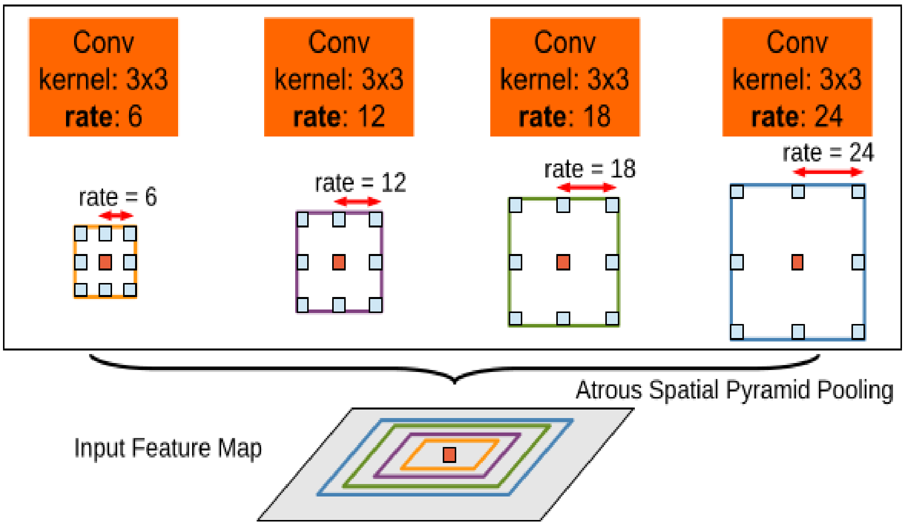


Figure 2.4: Atrous Spatial Pyramid Pooling

branches, ultimately, they will be put together to acquire result. Reference descriptions are outlined in the Fig 2.4

2.3.3 Fully Connected Conditional Random Field for Accurate Boundary Recovery

Reduction on localization accuracy and its experimental solution are thoroughly examined in [4].Fig 2.5 clearly explains score map which is quite smooth with short-range CRF, simultaneously, short-range CRF can be harmful in recovering detailed structure, then another alternative method, contrast sensitive potentials has been tested, with which are achieved improved localization, however, this also has lack of constructing thin-structures. Fig 2.5 express that regardless of DCNN score map predicts presence and solid position of object, yet is not good at discovering object border.

As challenges appears from short-range CRF, accordingly, assessment of value is taken into account, with which, system will incorporate with Fully Conditional Random Field, forming energy function:

$$E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \tag{2.2}$$

where x is the label assignment for pixels, unary potential $\theta_i(x_i) = \log P(x_i)$ where $P(x_i)$is

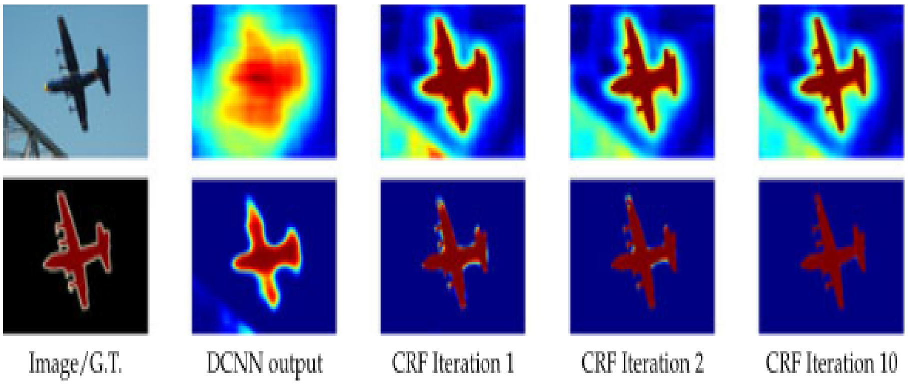


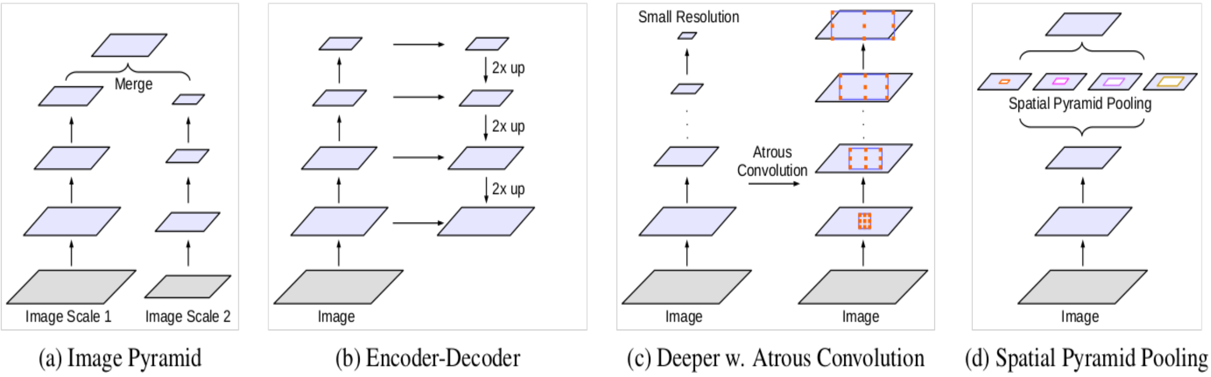
Figure 2.5: Score Map

the label assignment probability at pixel i as computed by a DCNN. As a consequence, Fully Connected Conditional Random Field is appeared to be remarkable successful for fine-grained localization accuracy, achieving accurate semantic segmentation results, as well as recovering object boundaries. as largely presented in the [6], [16].

Chapter 3

DeeplabV3

DeepLabV3 [5], is dramatically improved version of previous DeepLab [4] version, as well as dealing with the same confronting difficulties, such as existence of objects at multiple scales, reduced feature resolution caused by previuosly discussed synthesis of downsampling and maxpooling, accordingly solution is derived from taking downsampling from the amongst last few max pooling layers, preferably upsampling with corresponding filter kernels. This is preceded on the behalf of Atrous Convolution or Dilated Convolution capturing dense feature map, obviously experienced in [3], [4]. DeepLabV3 [5], supplies atrous convolution will gain control on resolution of features responses with DCNN. Existence of objects at multiple scale should be comprehensively considered, as in the case of earlier versions, in this respect, from [5], there are various possible measurements to figure out such a ongoing issues. Fig 3.1 displays multiple methods of Fully Convolution Network proposed by DeepLabV3 [5], in order to capture objects multi-scale. Each Fully



(a) Image Pyramid (b) Encoder-Decoder (c) Deeper w. Atrous Convolution (d) Spatial Pyramid Pooling

Figure 3.1: Several Experimental methods of FCN to obtain objects in multi-scale

Convolution Network is deeply investigated and sequenced, in inters of its performance.

1. Image Pyramid presented by [20] fits same weight through all scale inputs. This technique of FCN is aiming at deriving long-range contextual information from small scale inputs, whereas, achieving thin and small object details via large scale inputs, as explicitly shown in (a) of Fig 3.1 [6]

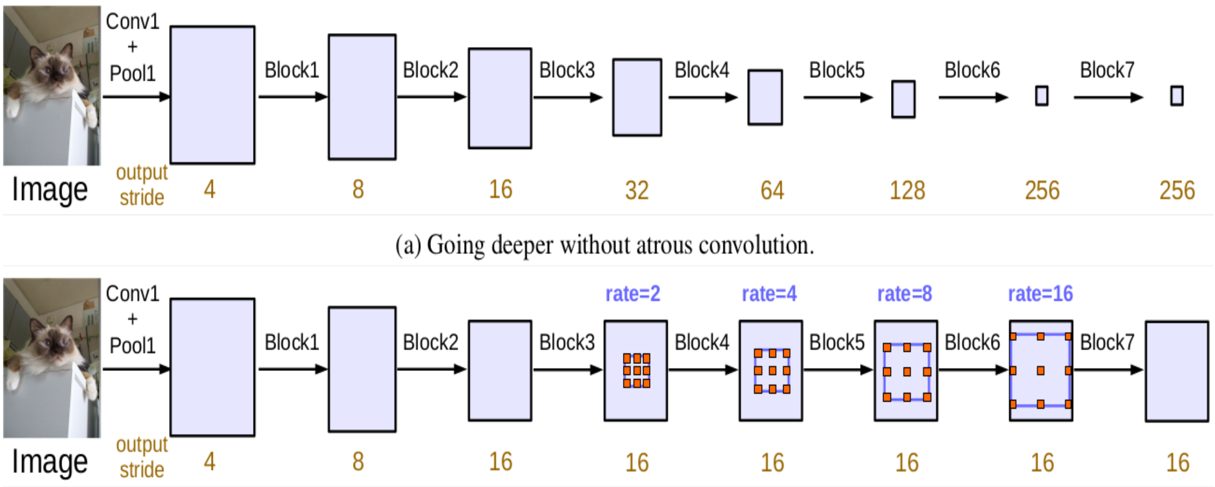
2. Encoder-Decoder appears with [1], [7] gains high popularity in recent version of DeepLab, generating fine-gained experimental results, (b) part of Fig 3.1 clearly explaines how encoder-decoder operates in FCN, such that encoder's target is to steadily reduce spatial dimension of feature map, in order to accomplish large-range information its output, as a consequent, it is produced into decoder which is left part of graphics, performing reverse operation of which encoder does, thus gradually regain lost object details and spatial dimension.

3. CRF and Deep CNN This method is conceptually divided into two parts: such as first, Deep CNN deploy many convolution layers, to encoder long-range information, as the second aim is feed DCNN into CRF [15], to apply incorpation of them, to build cascade [17] form of convolution layers.

4. Spatial Pyramid Pooling with Cascade Layers ASPP supplies outstanding work on DeepLab versions, came up with [5], [4], [7] such as DeepLabV2 [4] primarily talks about ASPP with various multiple atrous convolution, with different rates, to acquire deep scale contextual data. Then DeepLabV3 [5] explores further benefits of ASSP and ending up with recent development of corporation between atrous convolution in cascade form which feeds into feature map , together with ASPP.

3.0.1 Going Deeper in Atrous Convolution

Fig 3.2 impart deep knowledge regarding the Standard and Atrous Convolution operations. It is noticeable from the Fig 3.2 that Standard Convolution holds rate = 1, version of Atrous Convonvolution. Essentially, Atrous Convolution provides larger rates, measured by output stride which is computed as ratio of input image spatial resolution to



(a) Going deeper without atrous convolution.

(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

Figure 3.2: Cascade system with and wihout Atrous Convolution

output image spatial resolution. Atrous Convolution continuously modifies atrous rates, to have a control how densely compute feature responses. Main principal of expanding rates that Atrous Convolution follows, to allows object to be encoded in multiple scale.Coming back to the cascade form of Atrous Convolution, Fig 3.2 presents ResNet Blocks, as well as consecutive striding on those blocks and obtained outcomes with atrous and without atrous convolution. In principal, Atrous Convolution assist to achieve long contextual information, however, undoubtedly sequenced striding can deteriorate detailed information, then output stride determining striding is applied, which holds 16 value in this case.Fig 3.2 has two part : (a) and (b) differs from each other, due to value variation of output stride, according to atrous convolution case, refered to blocks, starting from 4 on which loss of information is seen, till carried on more 4 blocks.In (a) shows absence of atrous convolution, instead presence of standard convolution that leads to high value of output stride and loss of object details in cascade form, whereas (b) outlines well-defined form, mentioning Atrous convolution with output stride = 16, in the consecutive blocks.

3.0.2 Multi-Grid Method

In order to elaborate atrous convolution providing with various grid size, Multi-Grid Method is offered, allowing to impose hierarchical grid cells.Advent of Multi-Grid method suggests assumption of ultimalte atrous for each convolution layer will be calculated by

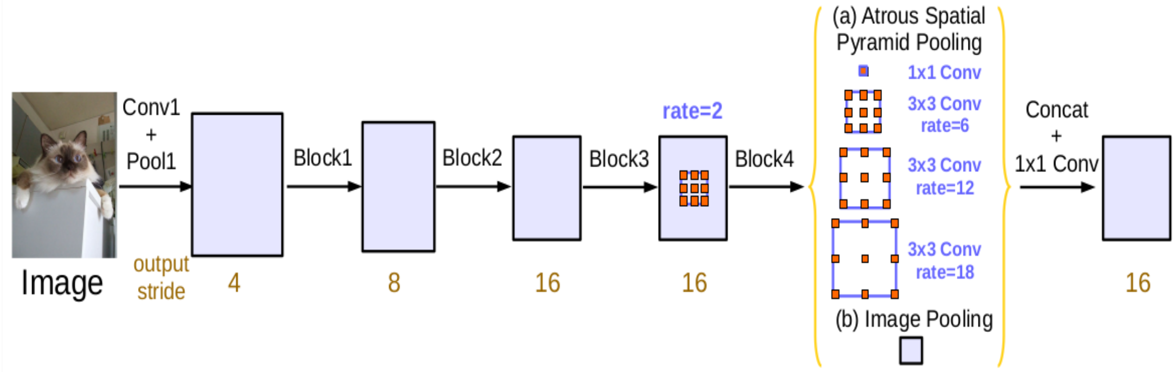


Figure 3.3: ASPP and Global Average Pooling

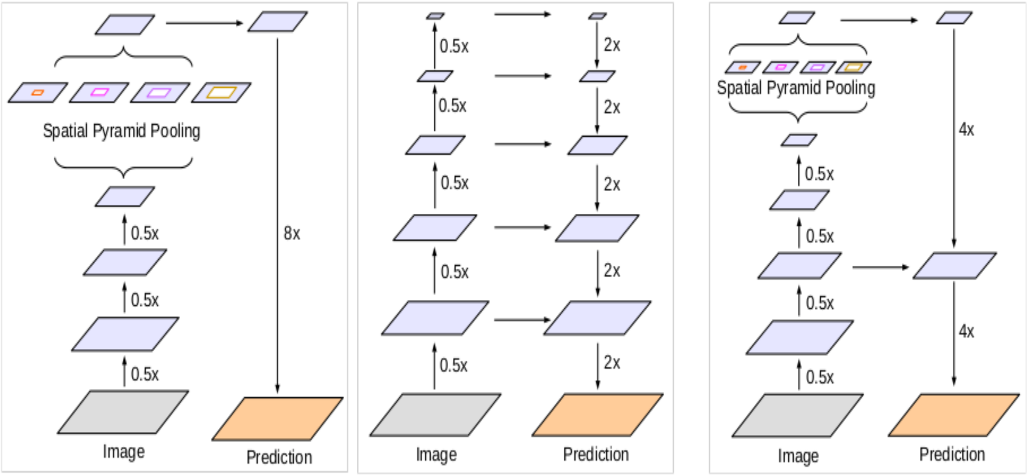
the multiplication between unit rate and atrous rate. Unit rate is formulated as (r_1, r_2, r_3). [2], [21]

3.0.3 Atrous Spatial Pyramid Pooling with Global Average Pooling

Throughout the sections, all discussed about ASPP, [4] together with its achievements, [12] however, it is paramount to note that ASPP with atrous convolution with multiple rates not only absorb long scale information, from the other hand, it filter weights turn out to be smaller, as rates increase. This is illustrated by the Fig 3.3 deeply.As it is seen, 3×3 convolutions are applied to the image, accordingly, is fed into 1×1, because central filter weight is integral point, but also leading to not to caputure whole part of image. In this respect, Global Average Pooling appears on last feature map, image level features(b) are appended to 1×1 convolution with batch normalization, ultimately, features are adapted to the desire dimension. Fig 3.3 also is splitted into (a) and (b) part of process in which (a) delivers sequential procedure of concatenated in the end.

3.1 DeepLabV3 Plus

Conceptually, DeepLabV3 Plus [7] is extension form of DeepLabV3 [5], then providing state-of-art techniques, not only combining all improved techniques of former version, but also providing additional refined version of Atrous Spatial Pyramid Pooling and Encoder-



(a) Spatial Pyramid Pooling (b) Encoder-Decoder (c) Encoder-Decoder with Atrous Conv

Figure 3.4: Concatenation of Spatial Pyramid Pooling and Encoder-Decoder with Atrous Convolution

Decoder network [7], on which Encoder will steadily reduce feature map resolution, so as to capture large contextual information, which will in turn be, gradually recovered by decoder, to absorb object boundaries. In addition, Xception [9] network which is extreme version of InceptionV3 module [22], comes up with depthwise separable convolution, further applied to ASPP and encoder-decoder module in DeepLabV3 [5].

3.1.1 Encoder-Decoder part of Network with Atrous Convolution

Previous version of DeepLabV3 plus, basically concentrates to extract large contextual information,, with multiple pooling operation and atrous convolution,Then, DeepLabV3 Plus introduces new technique such Encoder-Decoder network, on which encoder steadily reduce resolution of feature map, to inherit larger contextual information, in turn, decoder in contrast, operates gradually recovering that spatial dimension of feature map, to reconstruct object boundaries. Mainly, encoder-decoder corporate with large-scale contextual information, with which semantic information is encoded, simultaneously, atrous convolution control feature resolution, then output of encoder passes into decoder, as shown by the Fig 3.4.

Fig 3.5 present comprehensive idea of how encoder-decoder incorporates with

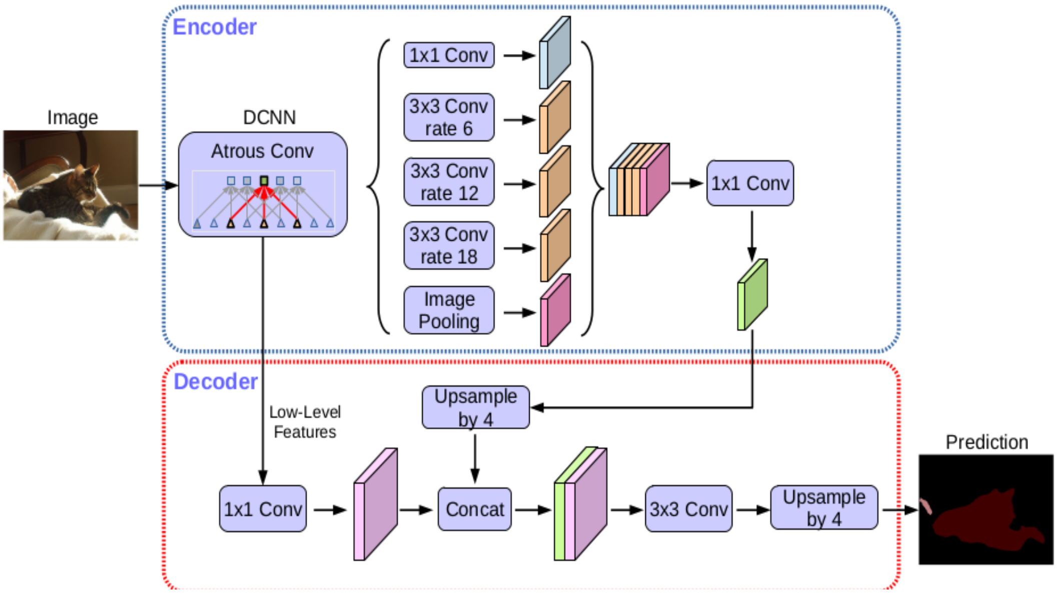


Figure 3.5: Operational Procedure of Encoder-Decoder Network

atrous convolution. Principally, leaning back suggested encoder network, with atrous convolution and image level features, from [5], output of encoder are primarily upsampled by factor of 4, then concatenated with image level features, which is in turn, applied to 1×1 convolution, because image channels can be large, causing training to be complicated. Recursively, 3 × 3 is fed into coming output, and upsampled again with factor of 4.

3.1.2 Modified Aligned Xception

Although, Xception module [9] has gain well-defined performance, however, it is further extended with not only by the name of Modified Align Xception, but also there are new additionals such as all max pooling operations, replaced by depthwise separable convolution [9] with striding, this will exceedingly improve ongoing performance, by achieving feature maps via atrous separable convolution. In addition, batch normalization [14] and Relu Activation are appended, amongst depthwise separable convolutions.However, entry flow remained unchanged, due to not loss efficiency and rapid computation.

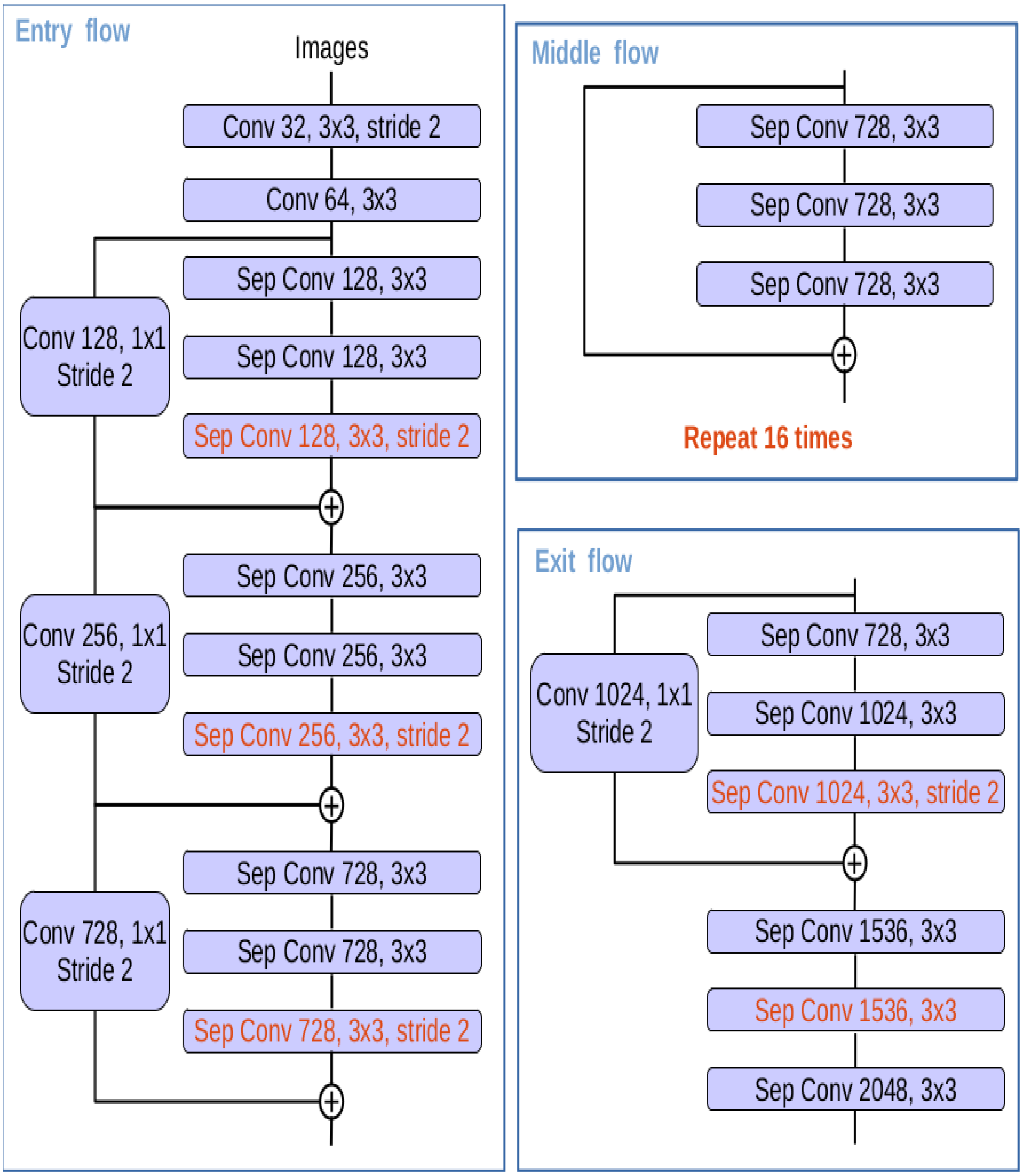


Figure 3.6: Modified Align Xception

3.2 Xception

It is already seen about Depthwise Separable Convolution [9] which in turn, plays integral role in Xception module [9]. Xception module is extreme Inception, providing exceedingly great performance and slightly better approach for different dataset. In Xception chapter, There are multiple differences between Xception and InceptionV3 [22], in terms of architecture construction. Another justification should be forgotten that Xception turns out flatly different from Inception module, because of not increase capacity, but instead efficiently usage of model parameters.

3.2.1 Inception

Conceptually, Inception module captures the same idea that Convolution follows, extracting large semantic information with less parameters.However, there are possible variance between them. It is the fact that Convolutions target is to learn filters with both, spatial and channel dimensions, thus if considered just single kernel convolution, then it will consider spatial and cross channel correlations. Whereas, in this case, Inception module try to make process easier by putting all them together sequentially. Such consecutive ordering should independently observe both cross-channel and spatial correlations. What Inception does is represented by the Fig 3.7. From this figure, it is clear that Inception concentrates primarily on cross-channel correlations via 1×1 convolution. Input is fed into multiple small branches,utilizing not only 3×3, but also average pooling layer, then further put together correlations via 3×3.

Apart from this, there is simplified version of Inception module, holds only 3×3 convolution, without average pooling, as Fig 3.8 displays.

This Inception module can be formed as large 1×1 convolution is mapped into segments of output channels, as a consequent, pursued by spatial convolutions.Fig 3.9 also express reference hypothesis. From this concept, some doubts appears such as how efficiently and separately map cross and spatial channel correlation? In this respect, Xception, [9] extreme Inception is proposed, taking advantage of 1×1, to map cross channel correlation. In contrast to previous module, spatial correlations are independently

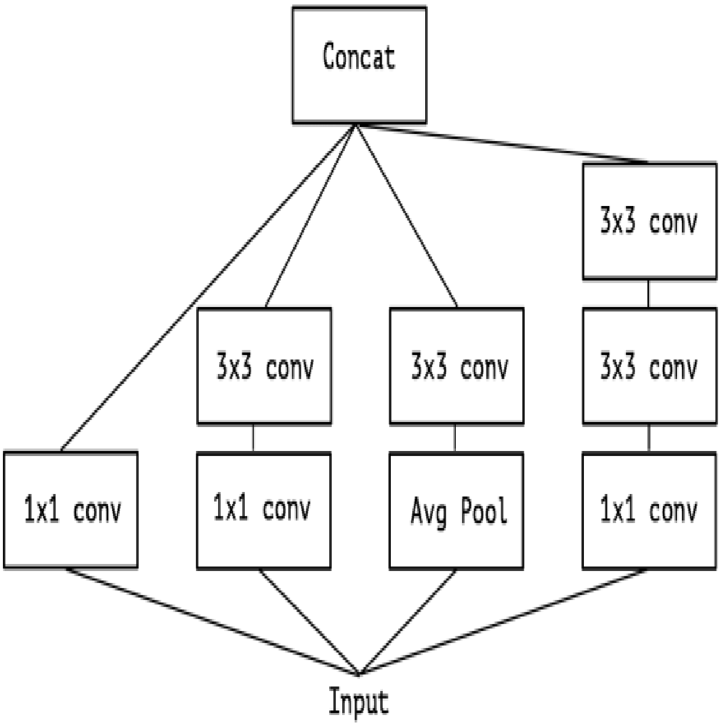


Figure 3.7: InceptionV3

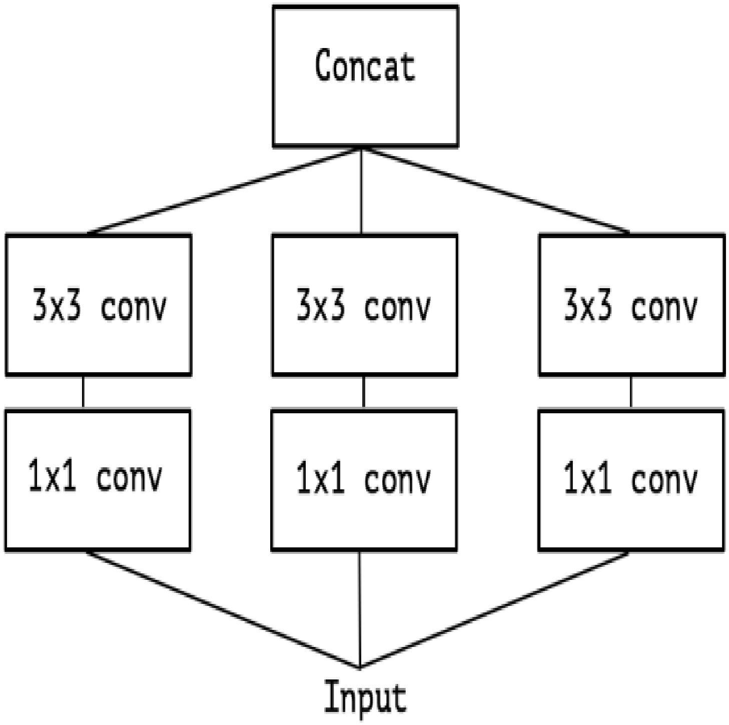


Figure 3.8: Simplified Inception Module

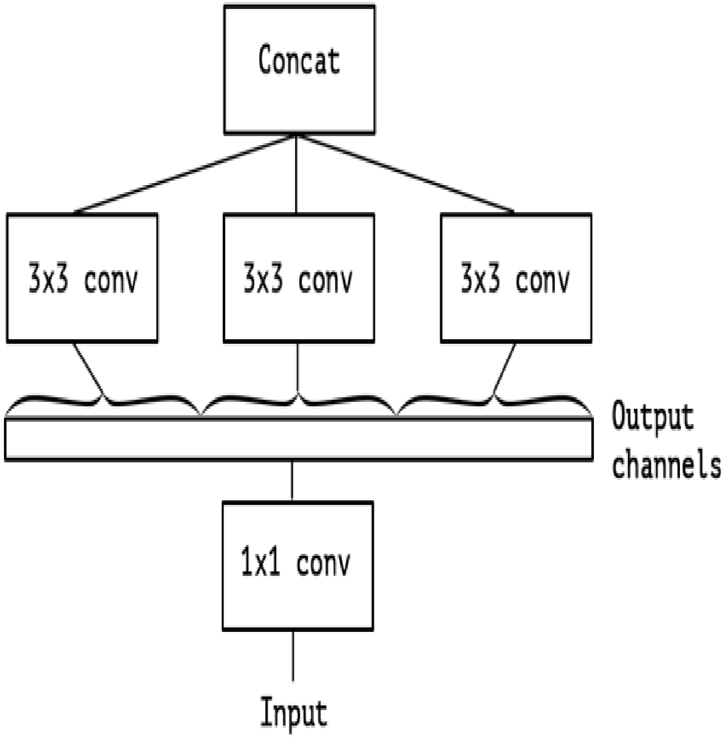


Figure 3.9: Large Convolution form of Simplified Inception

mapped for individual outcome derived from cross correlation, these are all depicted in the Fig 3.10

Regardless of some factors providing the similar assumptions between Extreme Inception and Depthwise Separable Convolution, they hold some different concepts. Depthwise Separable Convolution is explained as the following: every channel of input undergoes independently spatial convolution, consecutively, followed by pointwise convolution, holding 1 × 1 rate, projects output channels by depthwise convolution, generating new channel space. Depthwise Separable Convolution first focus on spatial convolution, then followed by 1×1, wherease, Inception module consider 1×1 convolution at first. Secondly, There are Relu non-linearity applied to Inception module, however, Depthwise Separable Convolution has no non-linearity.

Fig 3.11 introduce depthwise separable convolution union where depthwise and pointwise are explicitly formed. Moreover, Atrous benefits combination of Depthwise and Pointwise Convolution, [9] respectively.

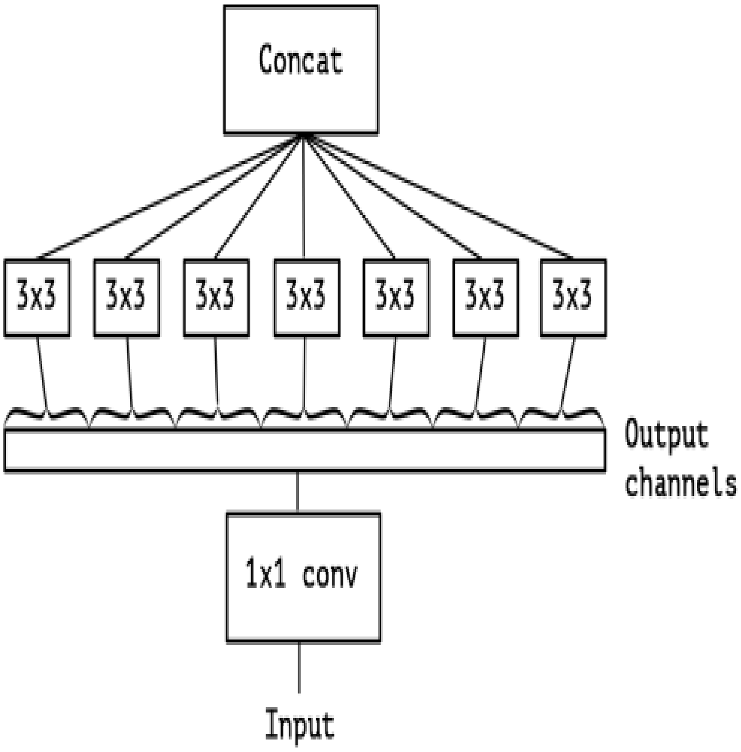


Figure 3.10: Extreme Inception Architecture(Xception)

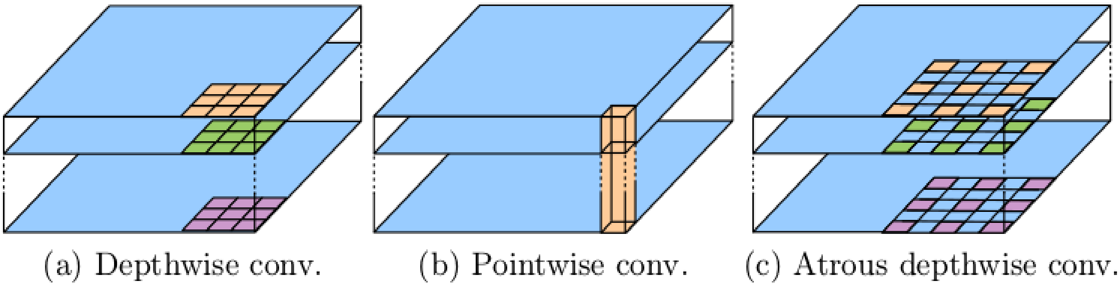

(a) Depthwise conv. (b) Pointwise conv. (c) Atrous depthwise conv.

Figure 3.11: Depthwise Separable Convolution

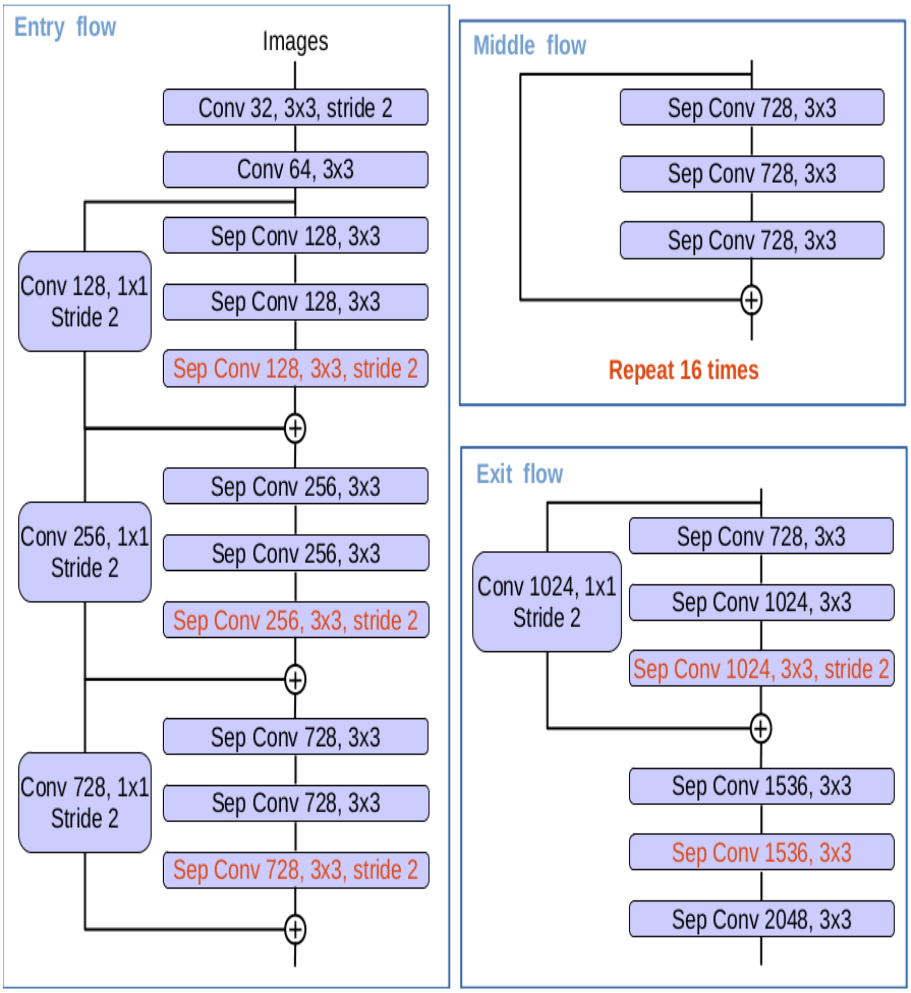


Figure 3.12: Xception Architecture

3.2.2 Xception Architecture

Xception module comes up with an idea of convolution neural network architecture leans back depthwise separable convolution, thus spatial and cross channel correlations formed on the feature map of convolution network, are separately conducted. Fig 3.12 opens up regarding the fact that Xception [9] captures 36 layers and is combined with stack of depthwise separable convolution with residual interaction. All have strong bond amongst linear stack of residual connections, but not first and last modules obeys the rule.Fig 3.13 also demonstrates modifications on each individual flow of Xception.

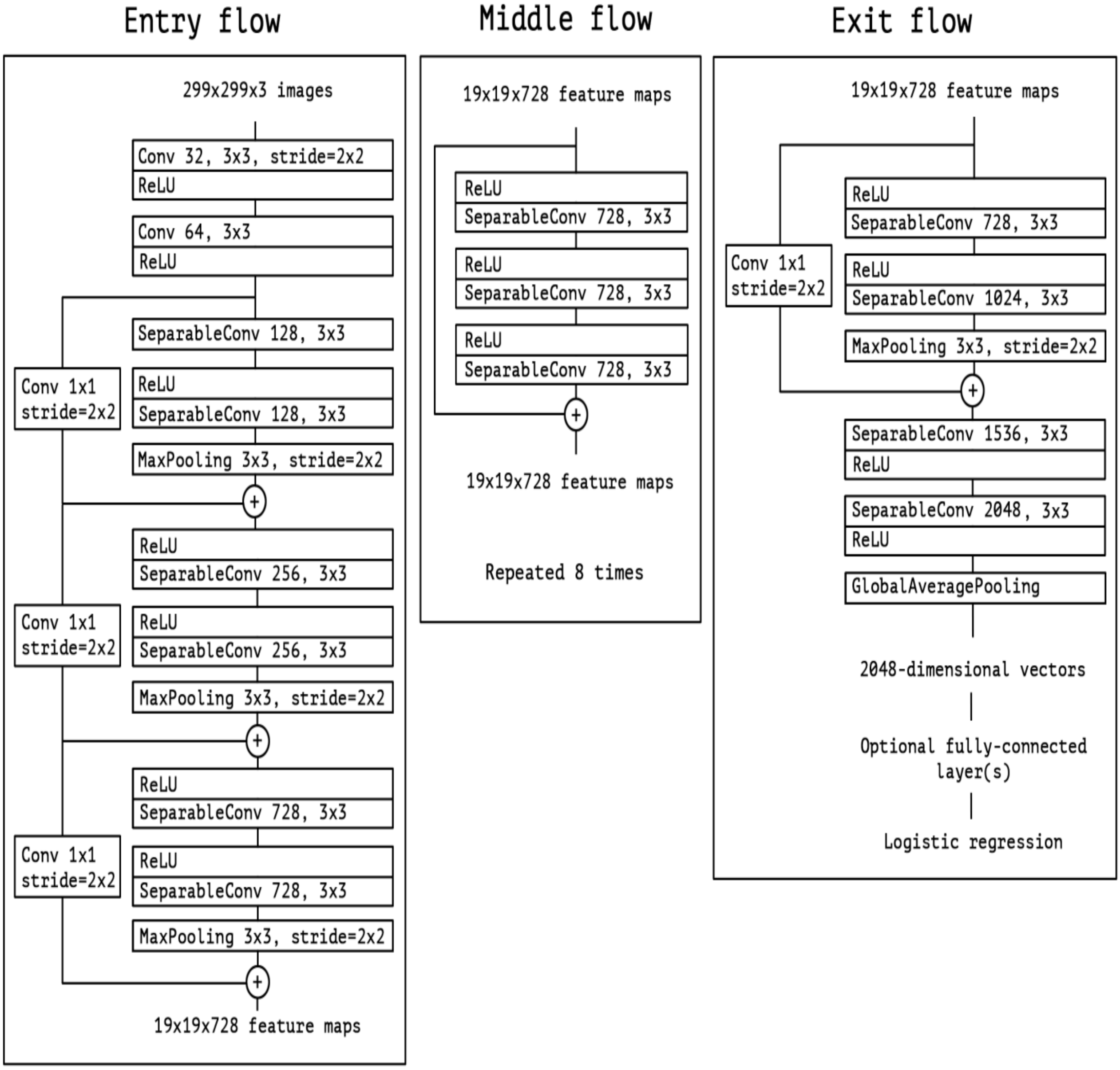


Figure 3.13: Xception Network with each flow, entry, middle, and exit

Chapter 4

Experimental Results with DeeplabV3 Plus

Throughout the all former version of DeepLabV3 Plus, we inherit deep practical and hypothetical knowledge over backbone, called Modified Aliged Xception that we considered as the optimal module with its reference advantages which is extension of Inception module, as well as Depthwise Separable Convolution together with Atrous Convolution, ultimately, DeepLabV3 Plus [9], which gather all state-of-art methods together, is represented in Lane Marking Semantic Segmentation task.In addition, we will discuss all these about, in the following sections, in more details, in terms of practical point of view.

1. Tranining Apolloscape Dataset with 38 classes, together with corresponding classes mentioned, as well as loss function called Stardard Cross - Entropy Loss

2. Tranining Apolloscape Dataset with 38 classes, together with reference class names and Weighted-Cross-Entropy Loss

3. Training with just two classes such as Lane Marking and Non-Lane Marking

4. Replacement of Random Scale Crop function, with Random Centre Crop function task.

4.1 Experimental Analysis

Primarily, we established custom-dataset for Apolloscape dataset which introduces 5000 images in total, consecutively, they are divided into train and validation set, 3980 and 1020, respectively. In addition, We make huge attempt to improve localization over boundary of existed Lane Marks on the road, as well as to reduce loss functions associated with corresponding labels of Roads. Atrous Spatial Pyramid Pooling has been proposed with parallel atrous convoluton layers, conceptually proposing inplanes = 2048, moreover output stride = 16, and 8 based on concept of Multi-Grid Union provide dilation that varies two times than each other such [1, 6, 12, 18] and [2, 12, 24, 36], respectively. [2], [21]

In decoder part of network, by considering module of 'Xception', we tested low-level-inplanes with 128, furthermore, fitting layers sequentially such as Convolution, Batc-normalization, Relu Activation and ultimately, we implement sequentual layers in at the last layer of Convolution, such as Conv2d, Batchnorm, Relu as well as Drop out regularization term. We further testes Modified Align Xception,comprising twenty blocks as well as output stride with 8 and 16 values, in order to make values of block slightly different from one to another, as represented in the Table 4.1:

Parametrization	output-stride = 8	output-stride = 16
entry-block3-stride	1	2
middle-block-dilation	2	1
exit-block-dilations	(2, 4)	(1, 2)

Table 4.1: Modified-Aligned-Xception Strides Values

Applying to Apolloscape dataset, as well as first chapter that describes data-preprocessing step in depths, comprising 38 various classes and corresponding colors. Those well-formatted ground-truth images are fitted into network.

4.1.1 Standard Cross-Entropy Loss on Semantic Segmentation

While training network with various 38 classes, there are possible challenges that has great impact on training set, caused by background and 37 reasonable classes such that they

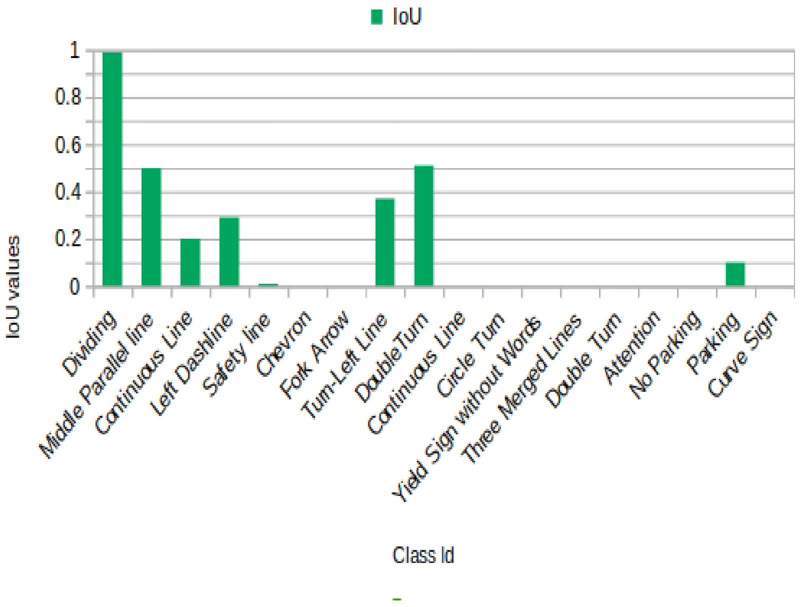


Figure 4.1: Intersection of Over Union for each class

are update with corresponding weights, instead background pixels exceeds that surrounds whole part of images, exceed reasonable pixels which leads to lack of accuracy defined for each class, global accuracy as well as IoU for consecutive classes. By visualizing results of class accuracy and IoU while training with Standard Cross-Entropy Loss, we represent graphical representation of each class accuracy and corresponding IoU, as illustrated on the Fig 4.1: As we have seen from table of 4.1, reasonable classes suffers from leakage of weighted loss on which network has great difficulties to manage proper accuracy for each class, in this case, Intersection Over union which comes up with notation of ratio between number of number of clearly classified pixels by total number of pixels over the image. Accordingly, we depicted Intersection Over Union and Individual Class accuracy, with Fig 4.1 and Fig 4.2, accordingly.

Similarly, Global Accuracy, as seen by the Fig 4.3, represents undesirable effect over the all classes. In addition, Mean Intersection over Union obtained via such a particular technique, is considered as well in the Fig 4.4. Ultimately, we provide individual Class Accuracy determined for each classes, through Standard Cross-Entropy Loss in the Fig 4.5.

As a consequence, we want to represent input - RGB images, and Ground-Truth images, on training set, and together with predicted image on validation set, as outlined

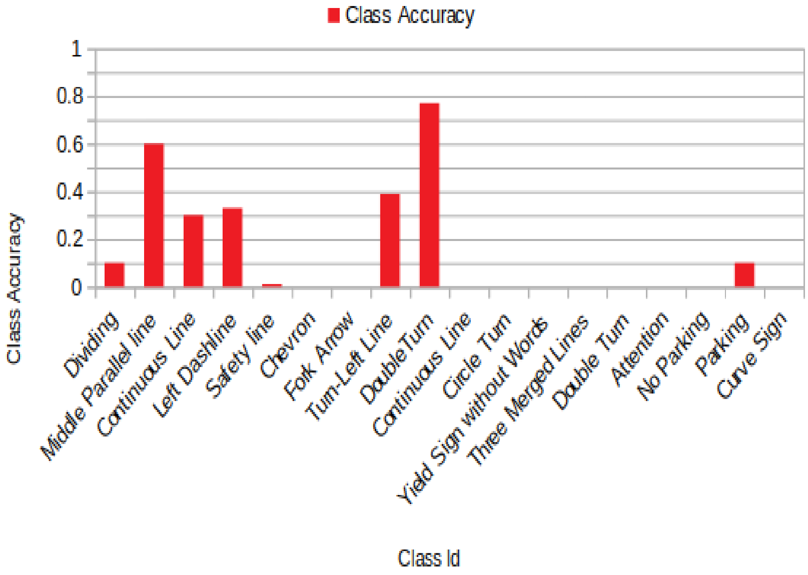


Figure 4.2: Each class Accuracy

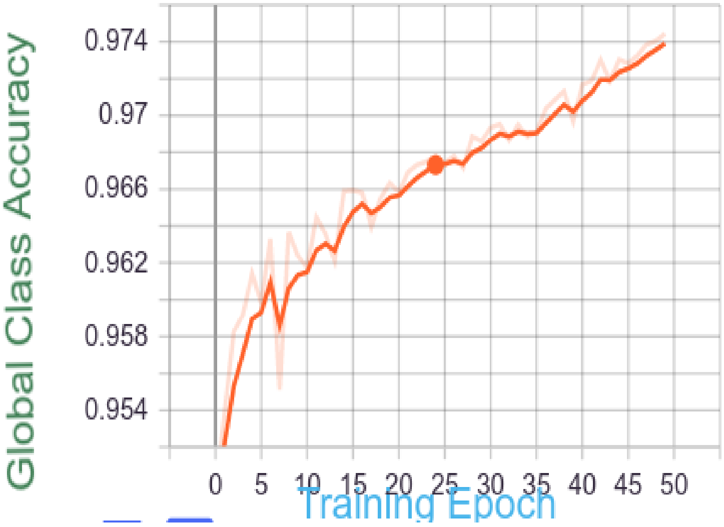


Figure 4.3: Representation of Global Accuracy with Standard-Cross-Entropy Loss

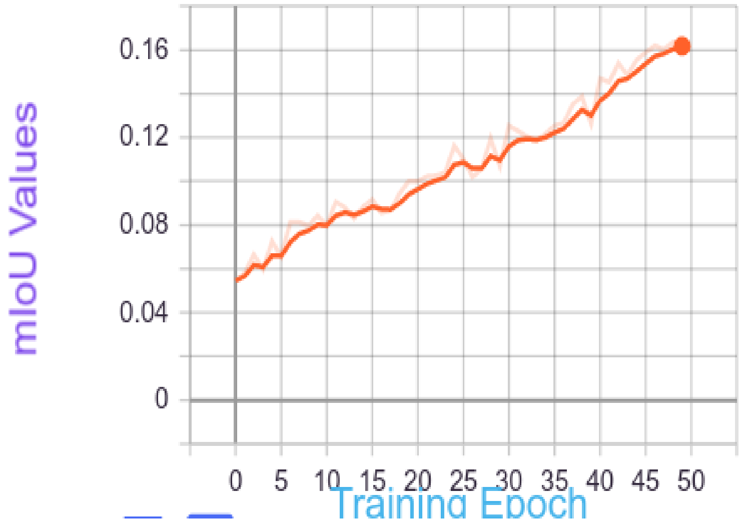


Figure 4.4: mIoU gained corresponding individual class accuracy

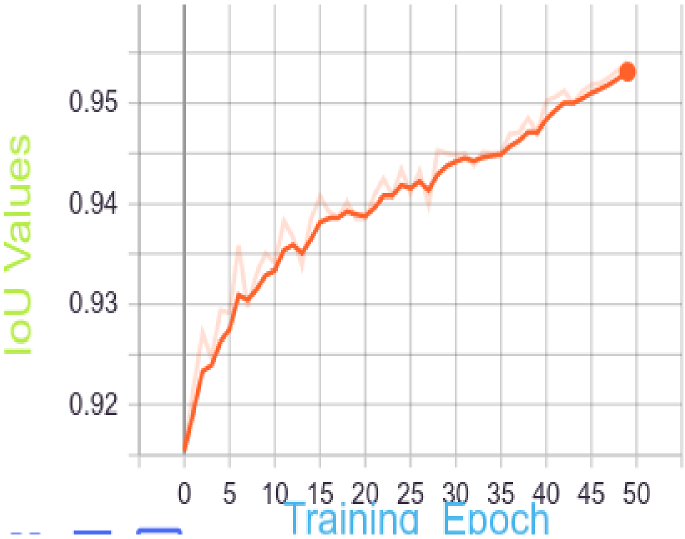


Figure 4.5: Representation of Individual Class Accuracy

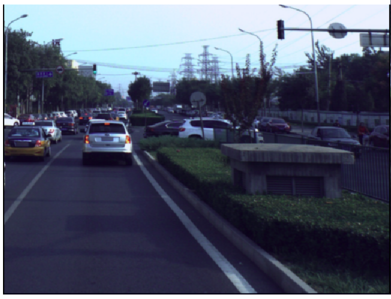


Figure 4.6: RGB(a)

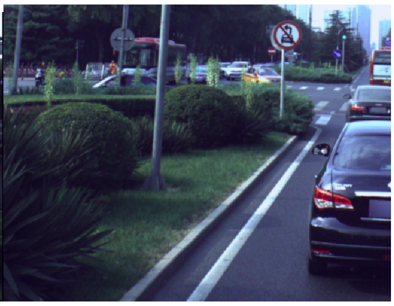


Figure 4.7: RGB(b)

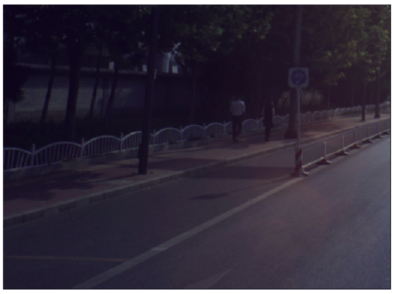


Figure 4.8: RGB(c)

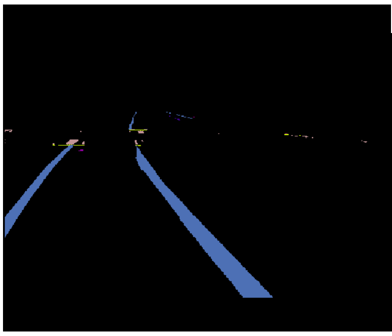


Figure 4.9: GT(a)

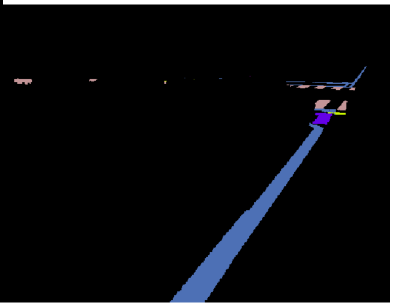


Figure 4.10: GT(b)

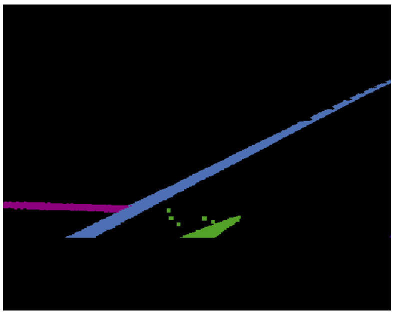


Figure 4.11: GT(c)

amongst The Figures, from 4.6 to 4.14.

Network predicted image as seen on the Fig 4.12. thorough the 50 epochs, as following the Fig 4.13 and Fig 4.14, It is important to note that We also used Focal loss [23] which is an improvement to the standard cross-entropy criterion, it is also illustrated with an Eq. 4.1. This is done by changing its shape such that the loss assigned to well-classified examples is down-weighted. Ultimately, this ensures that there is no class imbalance. In this loss function, the cross-entropy loss is scaled with the scaling factors decaying at zero as the confidence in the correct classes increases. The scaling factor automatically down weights the contribution of easy examples at training time and focuses on the hard ones.

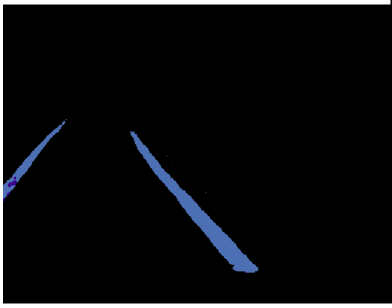


Figure 4.12: Prediction(a)

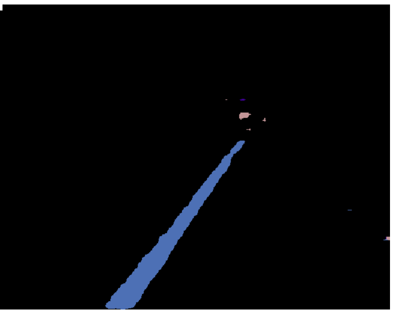


Figure 4.13: Prediction(b)

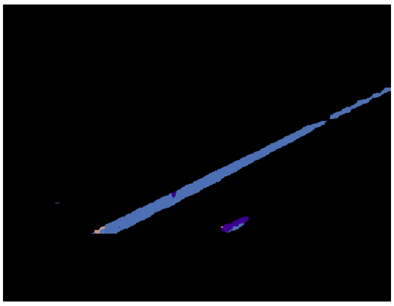


Figure 4.14: Prediction(c)

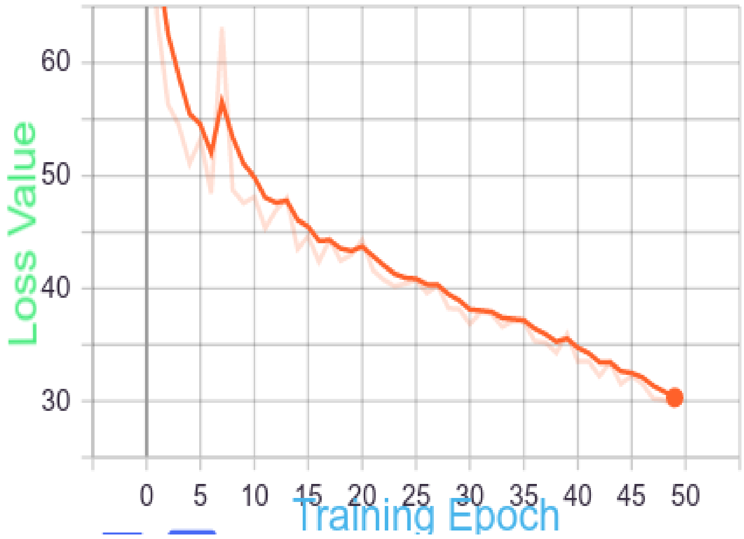


Figure 4.15: Loss value during training

$$FL(p_t) = -(1-p_t)^{\gamma}\log(p_t) \tag{4.1}$$

The Eq. 4.2 is the Cross Entropy Loss for binary classification. $y \in 1$ which is the ground-truth class and $p \in [0,1]$ which is the model's estimated probability. It is straightforward to extend it to multi-class case. For notation convenience, pt is defined and CE is rewritten as the following Eq. of 4.3:

$$CE(p,y) = \begin{cases} -\log(p), & if\ y=1 \\ -\log(1-p), & otherwise \end{cases} \tag{4.2}$$

$$p_t = \begin{cases} p, & if\ y=1 \\ 1-p, & otherwise \end{cases} \tag{4.3}$$

$$CE(p,y) = CE(p)t) = -\log(p_t) \tag{4.4}$$

As a consequent, table 4.2 clearly dedicates valid obtained Intersection over Union and Class Accuracy values for classes, and dash lines allure to existence of missing classes on test set. meanwhile, it should not be forgotten to introduce Global accuracy, Mean Intersection Over Union as well as Mean of Class Accuracy achieved during validation of entire dataset, on the table 4.3. Whole training is conducted withing just 50 epochs, seen from the Fig 4.15.

Classes	IoU	Class Accuracy
Dividing	0.99	0.10
Middle Parallel line	0.50	0.60
Continuous Line	0.20	0.30
Left Dashline	0.29	0.33
Safety line	0.01	0.01
Chevron	-	-
Fork Arrow	-	-
Turn-Left Line	0.37	0.39
DoubleTurn	0.51	0.77
Continuous Line	0	0
Circle Turn	-	-
Yield Sign without Words	0	0
Three Merged Line	-	-
Double Turn	-	-
Attention	-	-
No Parking	-	-
Parking	0.10	0.10
Curve Sign	-	-

Table 4.2: Standard-Cross-Entropy Loss with Correctly Classified pixels(Classes with NaN values, are not mentioned in the table)

Global Accuracy	mIoU	Mean of Class Accuracy
0.98	0.17	0.20

Table 4.3: Standard-Cross-Entropy Loss with Correctly Classified pixels

4.1.2 Weighted Cross Entropy Loss on Lane Marking Semantic Segmentation

Previous evaluation on training and validation set, are not desirable, due to vast amount of background pixels appearing more on the whole part of image, rather than reasonable pixels. We leveraged focal loss and Standard Cross-Entropy-Loss which was unseccessful on the multi-label tasks of semantic segmentation. In this respect, we eliminate ongoing challenge by implementation of Weighted-Cross-Entropy-With-Logits is the weighted variant of Sigmoid-Cross-Entropy-With-Logits. Basically, weighted-cross-entropy-with-logits

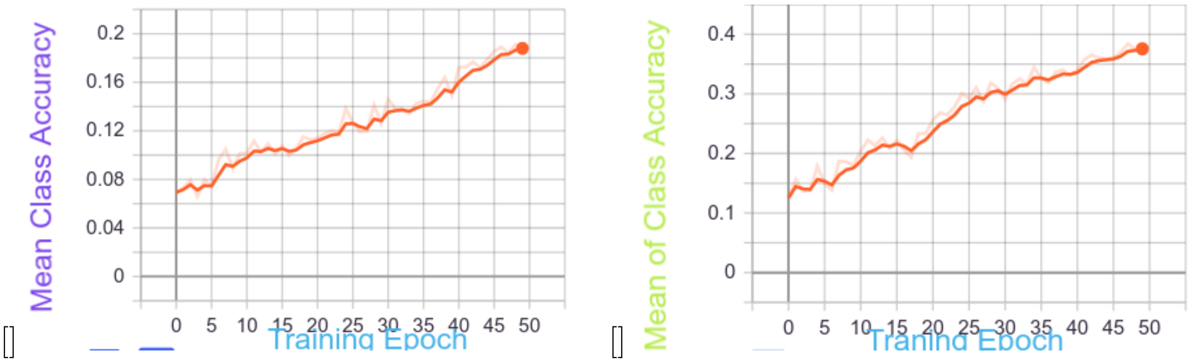


Figure 4.16: Standard (left) and Weighted Cross Entropy(right) Mean of Class Accuracy Values

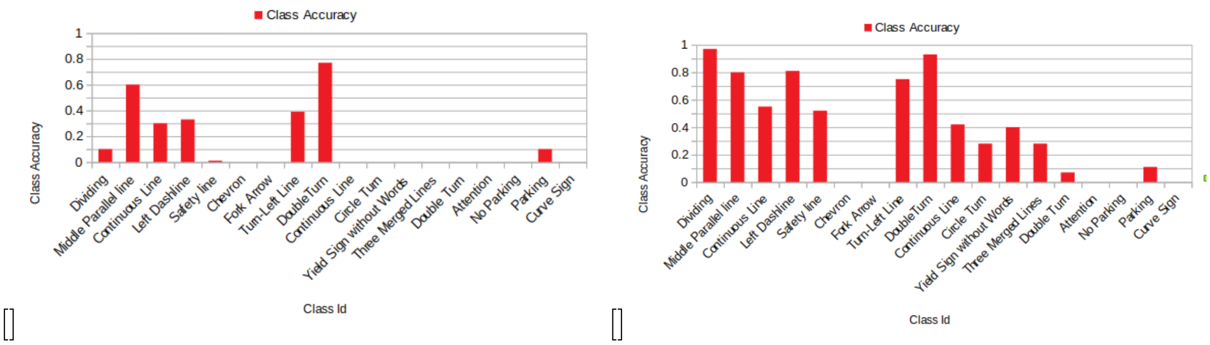


Figure 4.17: Binary Class(left) and Weighted Cross Entropy(right) Class Accuracy Values

weighting needs to happen within the computation of the loss. This is what weighted-cross-entropy-with-logits does, by weighting one term of the cross-entropy over the other. In Other Word, it weights reasonable class pixels, compared to background pixel on the imbalance dataset. Proposed method dramatically improved class accuracy over the reasonable pixels, on the cost of reduction, performed on the focus of background pixels as illustrated by obtained result during the validation set, these all accomplished outcomes are introduced by the comparison with two different loss functions, Standard and Weighted Cross Entropy Loss, on behalf of Mean of Class Accuracy,with Fig 4.16.

In order to observe drastic improvement related to Class Accuracy, conducted by two different methods, tensorboard graphic can be seen as the bar chart, in the Fig 4.17.

On the other hand, this success can not hold for the Global-Accuracy of Classes as well as Mean Intersection of Over the Union, thus, regardless of some dramatic fluctuations, seen on the achieved Global-Accuracy and mIoU, Global-Accuracy does not indicate any modification, yet mIoU show slight improvement thorough the validation

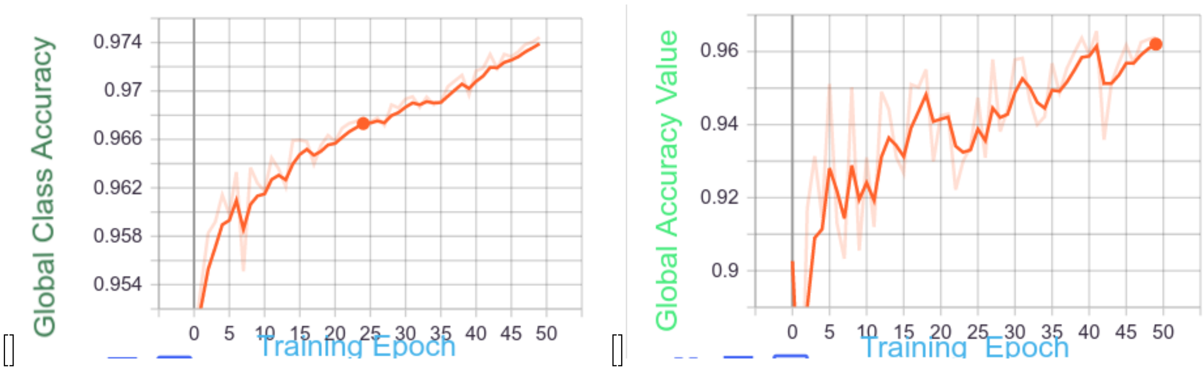


Figure 4.18: Standard (left) and Weighted Cross Entropy(right) Global Class Accuracy

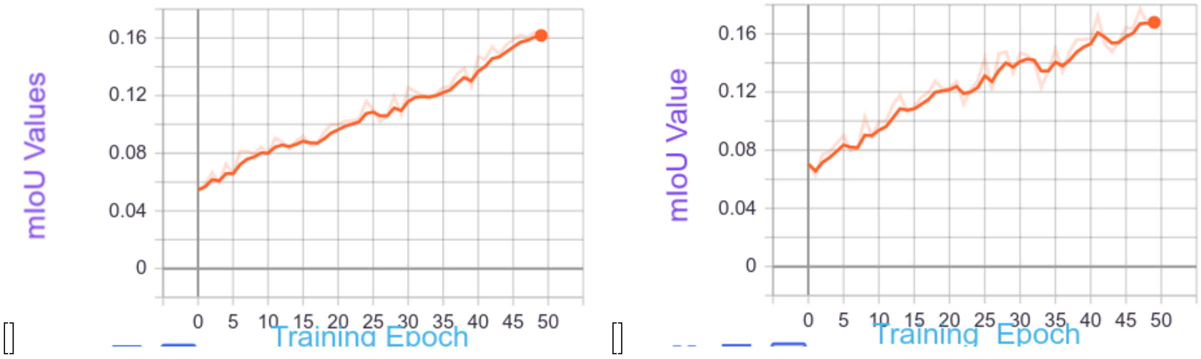


Figure 4.19: Standard (left) and Weighted Cross Entropy(right) Mean Intersection over Union

set, respectively, with advent of Weighted-Cross-Entropy loss method. Fig 4.18 denotes constant experimental results for both methods, in Global Accuracy. Following Fig 4.19 shows mIoU for Standard and Weighted Cross Entropy Loss.

Fundamentally, a weight vector $w \in R_k$ is determined with elements $w_k > 0$ defined over the range of class labels $k \in 1, 2, ..., K$ .Then, class-weighted cross-entropy [19] which plays integral role in unbalanced dataset, is mathematically introduced, by the Eq. 4.5

$$L_n(W) = -w_{c_n} \log y_{c_n}(x_n, W) \tag{4.5}$$

However, Intersection of Union obtained in both methods, appears to be proven slight improvement or reduction in some classes, seen by bar Fig 4.20.

we visualized achieved results on tensorboard together with corresponding RGB, Ground-Truth as well as Predicted Images, as the following cases, but loss value modification is also presented in the Fig 4.21. There are two examples here to depict, from Fig 4.22 to 4.30 with corresponding RGB, Ground-Truth and Predicted Images are denoted

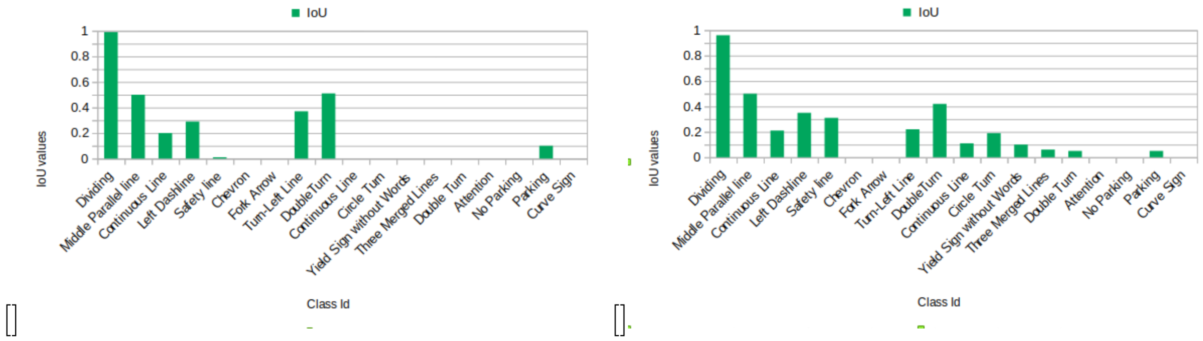


Figure 4.20: Standard (left) and Weighted Cross Entropy(right) IoU Values

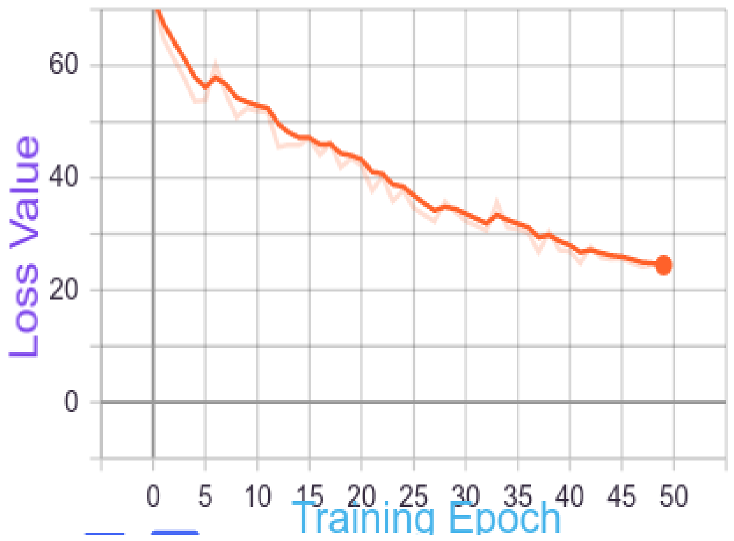


Figure 4.21: Loss value during training

as first example, then second example is detailed by RGB representative Figures 4.31, 4.32, 4.33, with their analogous achieved images by the Fig 4.37, 4.38 and 4.39.

Table 4.4 perform correctly classified pixels around the predicted image, trained with Weighted-Cross Entropy Loss

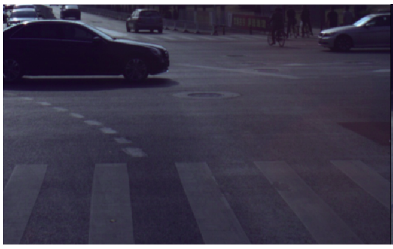


Figure 4.22: RGB(a)

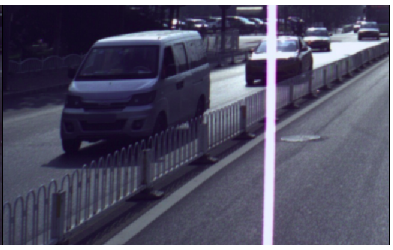


Figure 4.23: RGB(b)

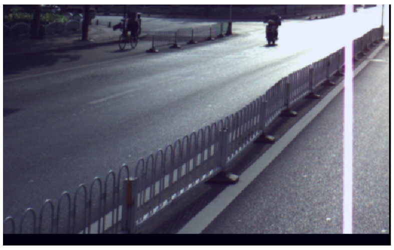


Figure 4.24: RGB(c)

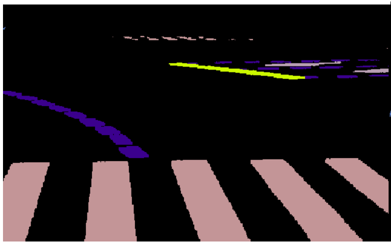


Figure 4.25: GT(a)

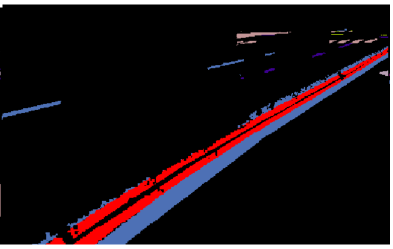


Figure 4.26: GT(b)

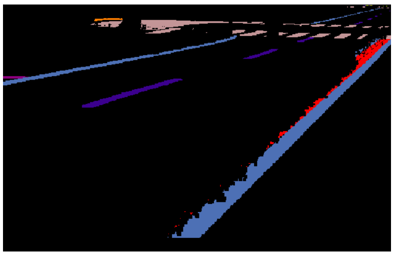


Figure 4.27: GT(c)

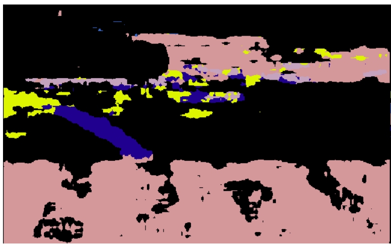


Figure 4.28: RGB(a)

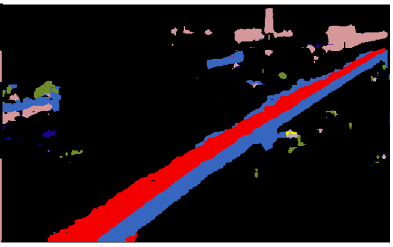


Figure 4.29: RGB(b)

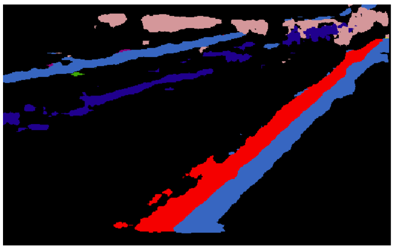


Figure 4.30: RGB(c)

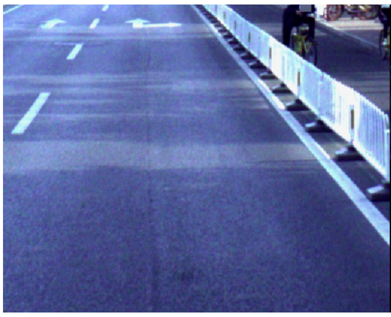


Figure 4.31: RGB(a)

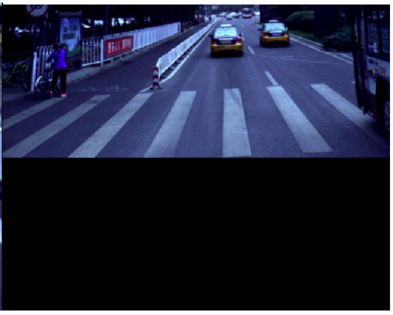


Figure 4.32: RGB(b)

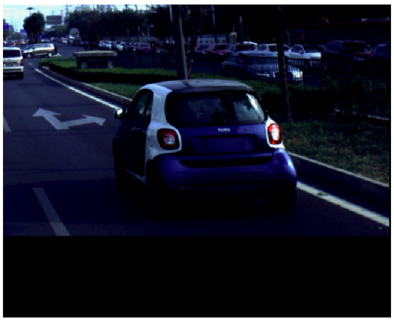


Figure 4.33: RGB(c)

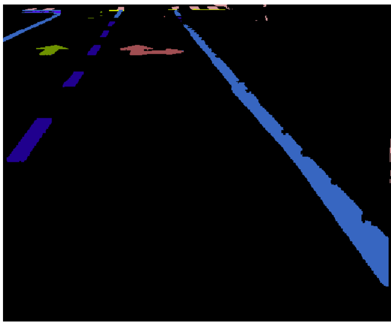


Figure 4.34: Prediction(a)

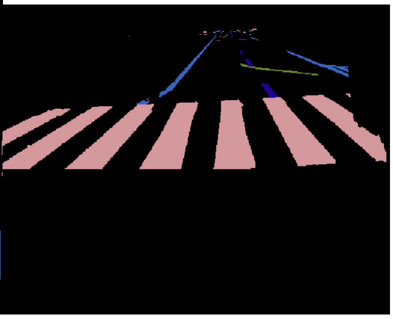


Figure 4.35: Prediction(b)

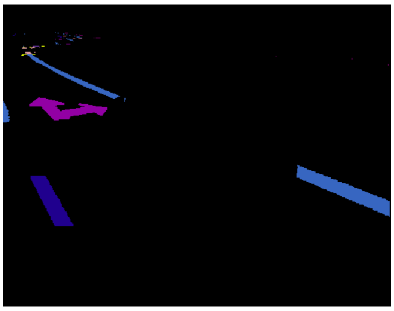


Figure 4.36: Prediction(c)

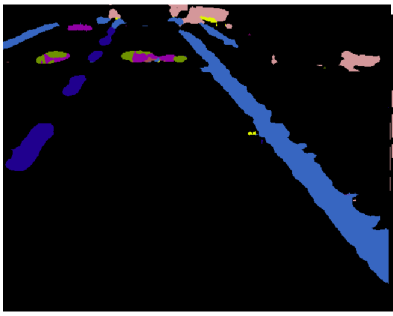


Figure 4.37: Prediction(a)

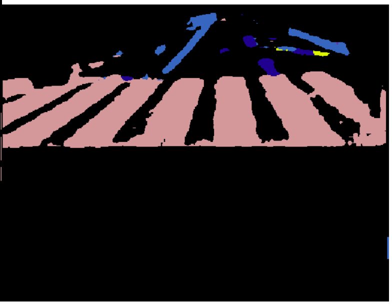


Figure 4.38: Prediction(b)

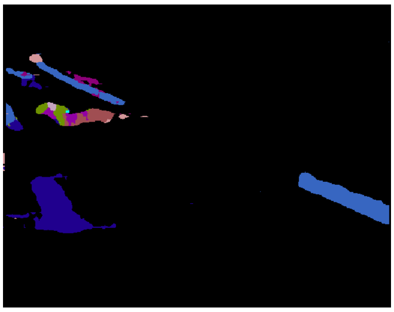


Figure 4.39: Prediction(c)

Class Id	IoU	Class Accuracy
Dividing	0.96	0.97
Middle Parallel line	0.50	0.80
Continuous Line	0.21	0.55
Left Dashline	0.35	0.81
Safety line	0.31	0.52
Chevron	0	0
Fork Arrow	-	-
Turn-Left Line	0.22	0.75
DoubleTurn	0.42	0.93
Continuous Line	0.11	0.42
Circle Turn	0.19	0.28
Yield Sign without Words	0.10	0.40
Three Merged Lines	0.06	0.28
Double Turn	0.05	0.07
Attention	-	-
No Parking	-	-
Parking	0.05	0.11
Curve Sign	-	-

Table 4.4: Weighted-Cross-Entropy Loss with Correctly Classified pixels. (Classes with NaN values, are not mentioned in the table)

4.1.3 Composition of 38 classes into Binary Classes

As we have encountered with various challenges, regardless of some dramatic refinements, by deploying new techniques such as Weighted Cross Entropy Loss[19], then we decided to merge 38 classes into 2 classes, denoted ”Lane-Marking” and ”Non-Lane Marking”. With this idea, solely, accuracy defined for each class is exponentially increased. It is explicitly seen that this is unique growth that is not experienced in the previous techniques. What we comprehensively experienced is that accuracy of each class is exceedingly high for

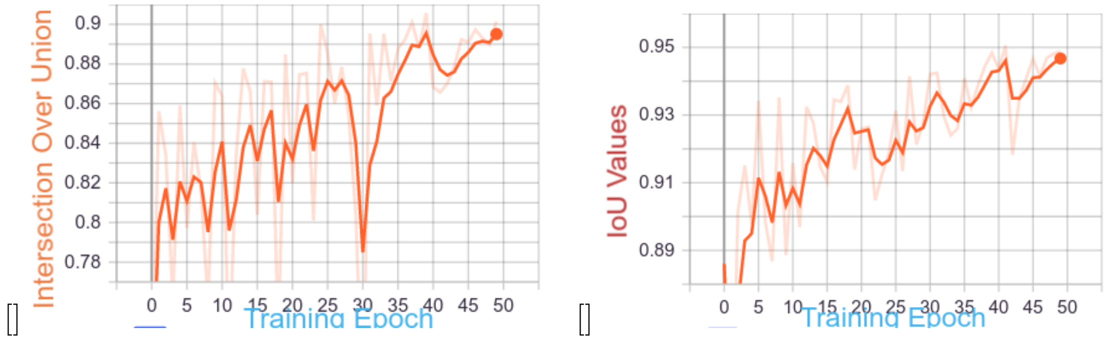


Figure 4.40: Binary Class(left) and Weighted Cross Entropy(right) Intersection Over Union

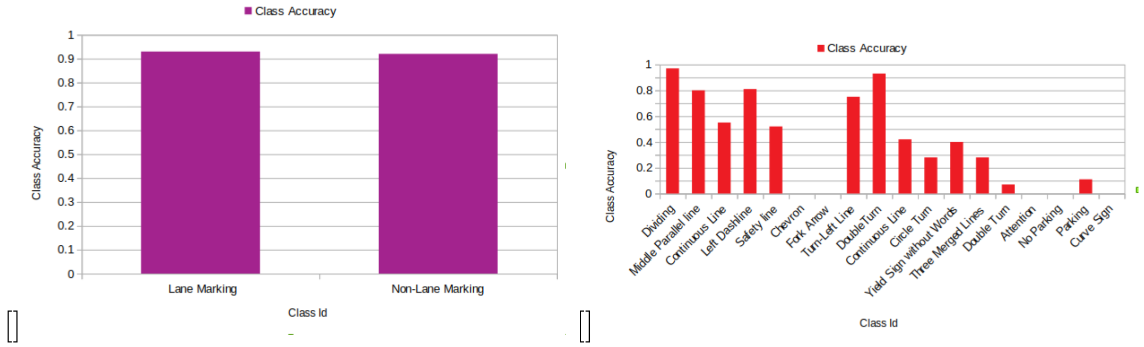


Figure 4.41: Binary Class(left) and Weighted Cross Entropy(right) Class Accuracy

lane marking task, whereas, Intersection of Union illustrates low performance in precision seen with Fig 4.40. Class Accuracy Values experienced via training with two classes, are more or less close to results, seen with Weighted-Cross Entropy Loss, also explicitly demonstrated by the Fig 4.41

Another justification is noticeable that Mean of Class Accuracy in Binary classification method, holds large values, in comparison with Mean of Class Accuracy obtained with Weighted Cross Entropy Loss, on the Fig 4.42.There achievements further provide total loss validating the entire dataset on Fig 4.43.

On the other hand, experimental analysis on images are produced from Fig 4.44 to 4.52. In addition, Intersection over Union and Class accuracy both with Binary class training, are obtained as mentioned in the table 4.5. Despite the fact that, results are high, however, qualitative results are not satisfactory for depicted by the Fig 4.40 All in All, network appears to be confused and finds misleading training, with a proposed method of composition of 38 classes into 2 classes.

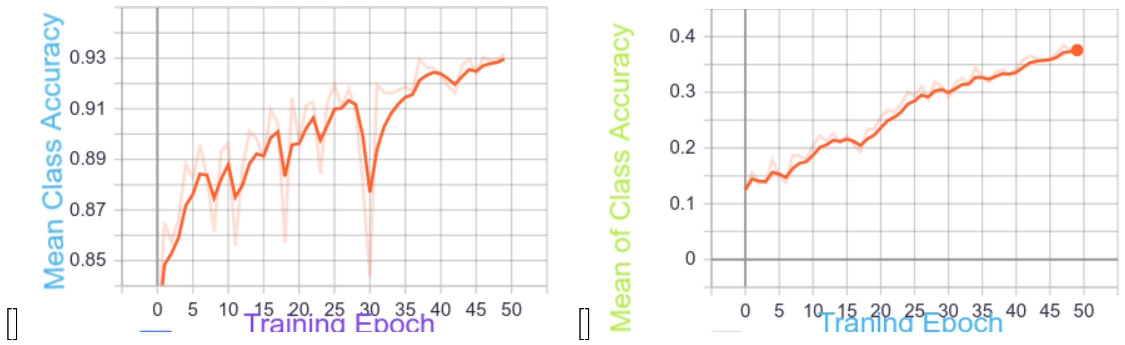


Figure 4.42: Binary Class(left) and Weighted Cross Entropy(right) Mean of Class Accuracy

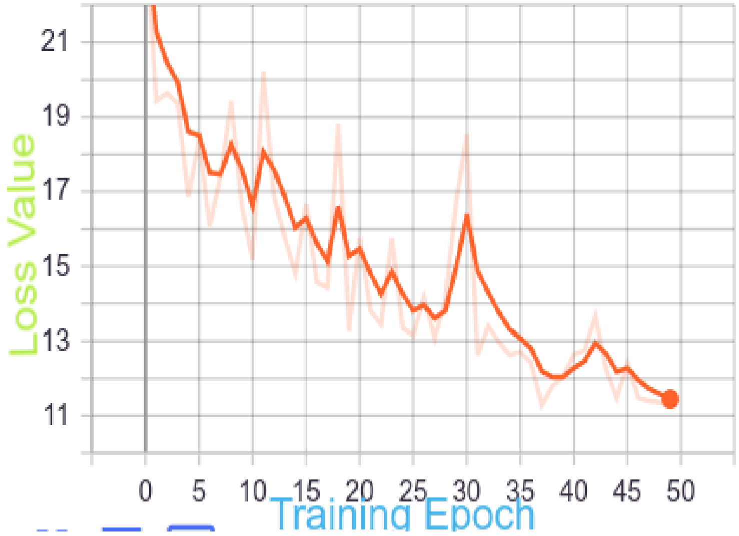


Figure 4.43: Loss value during training

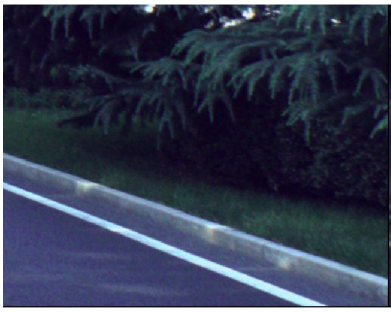


Figure 4.44: RGB(a)

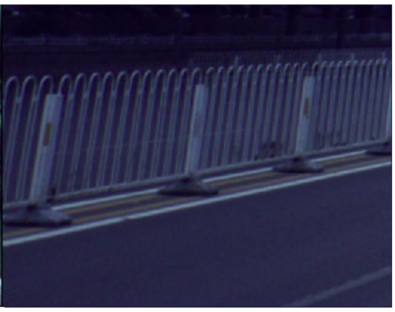


Figure 4.45: RGB(b)

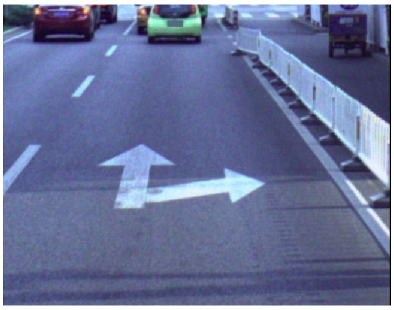


Figure 4.46: RGB(c)

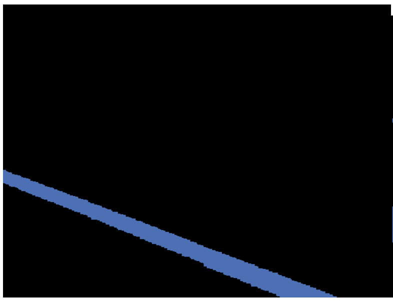


Figure 4.47: GT(a)

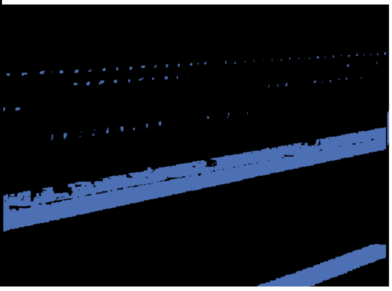


Figure 4.48: GT(b)

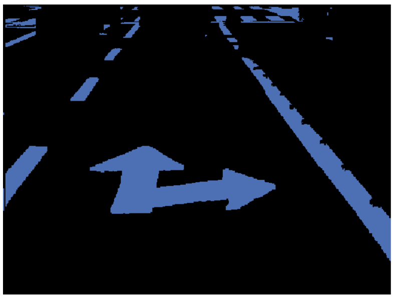


Figure 4.49: GT(c)

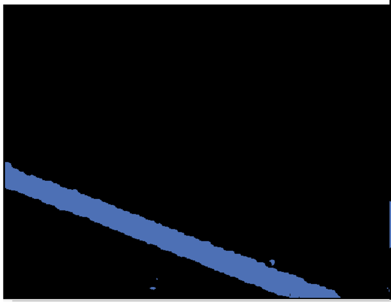


Figure 4.50: Prediction(a)

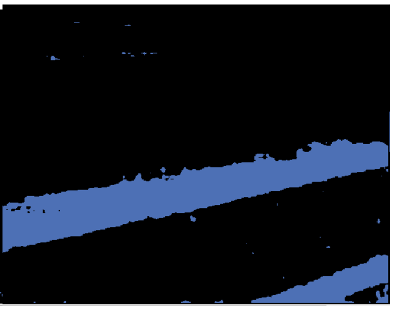


Figure 4.51: Prediction(b)

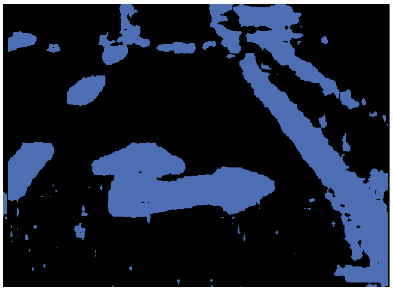


Figure 4.52: Prediction(c)

Class Id	IoU	Class Accuracy
Lane Marking	0.93	0.93
Non-Lane Marking	0.93	0.92

Table 4.5: Binary Class experimental results

4.1.4 Integration of Centered Crops method with Weighted Cross

As it is seen on the previous chapters, we make huge attempts to refine obtained results via various kind of techniques, however, they solely opens up regarding modification on loss function and composition of classes into two classes. In this chapter, we will discuss about Random Rescale crop which crops input images, random, in other word, with different scales. With this idea, it is mostly likely to loose reasonable part of image which instead could effectively be trained and test, but Ramdom Scale function unable to train accurately, as a consequence, to obtain fine-tuned semantic segmentation. In this respect, we eliminate to randomly scale input images, instead, propose more effective function Fixed Crop in which we we want network to train only reasonable part of images, therefore, defining left and right offset, as well as new height and width, will be crucial set of Fixed Crop function, conceptually will be appended to transform function of custom dataset. Ultimately, we create new built-in function of Pytorch, called Random Crop. In addition, it is interesting to note that metrices with gathered results are similar to the considered fundamental approach of Weighted Cross Entropy Loss, because we integrate technique of utilizing Center Crop method, with Weighted Cross Entropy loss, once Weighted-CE generates optimal performance, amongst the all methods considered so far, in metrices at the time of validating dataset, as well as, we enlarge total epoch, aiming at training

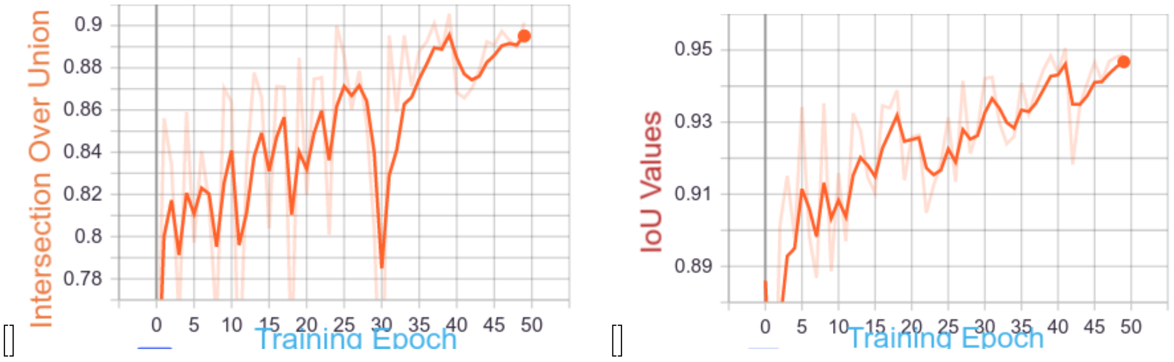


Figure 4.53: Binary Class(left) and Weighted Cross Entropy(right) Intersection Over Union

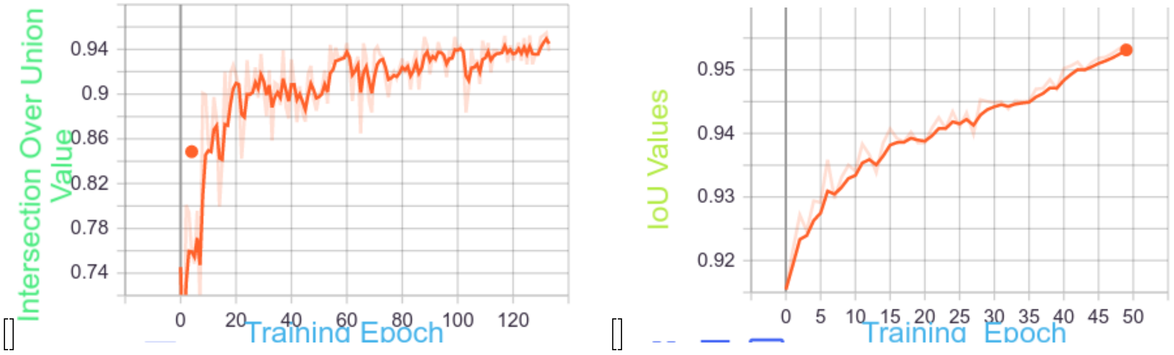


Figure 4.54: Center Crop method (left) and Weighted Cross Entropy(right) Intersection Over Union

network longer, through 120 epochs (Fig 4.61), to allow model to converge, in contrast to weighted cross entropy loss, which was 2 times smaller in iterations. It is paramount to notify different Intersection Over Union accomplished with former methods such as Binary Class, Weighted Cross Entropy Loss and Center Crop method, as demonstrated between Fig 4.53 and Fig 5.54, respectively.

As graphical representations illustrate clear idea that lowest performances on class accuracy, concern to Binary Class method by the Fig 4.55, in contrast, Weighted Cross Entropy loss (right) drastically refines class accuracy through the majority of classes, similarly, Center Crop method trained with Weighted Cross Entropy Loss delivers almost identical value for class accuracy by the Fig 4.56(left). Ultimately, Standard Cross Entropy Loss appears to be unsuccessful in the presence of unbalance dataset, regardless of dramatic growth in some classes (Fig 4.56(right)). These all can be seen by bar chart as well, as the following Figures.

Above Intersection over Union and Class Accuracy come up with interesting

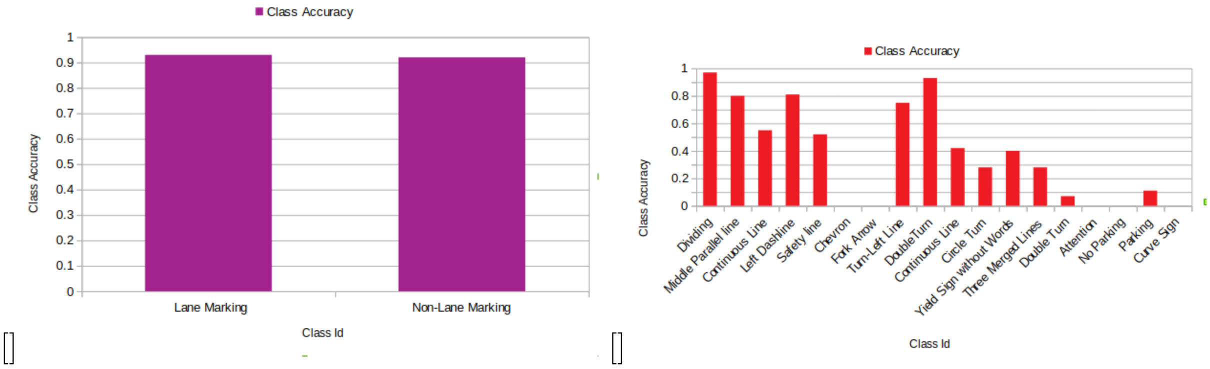


Figure 4.55: Binary Class(left) and Weighted Cross Entropy(right) Class Accuracy

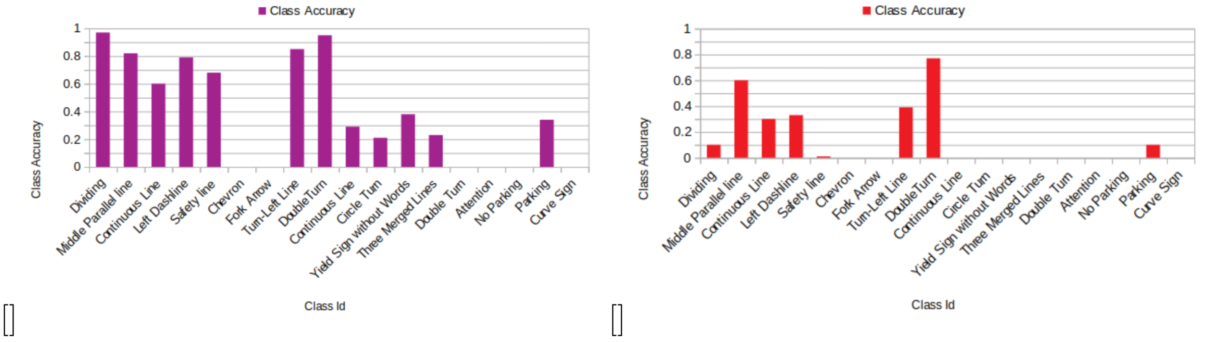


Figure 4.56: Center Crop method(left) and Standard Cross Entropy Loss(right) Class Accuracy

results when observed via bar chart, various considered method, from the Fig 4.57 and Fig 4.58, accordingly.

On the other hand, Global accuracy and Mean of Class Accuracy should not be forgotten to mention, since they have also slight modification in representation, in the Fig 4.59, and 4.60. Training and Validating network in this method as already mentioned above, will take 120 iterations (Fig 4.75).

Center Crop method solely concentrates on the reasonable part of images, which

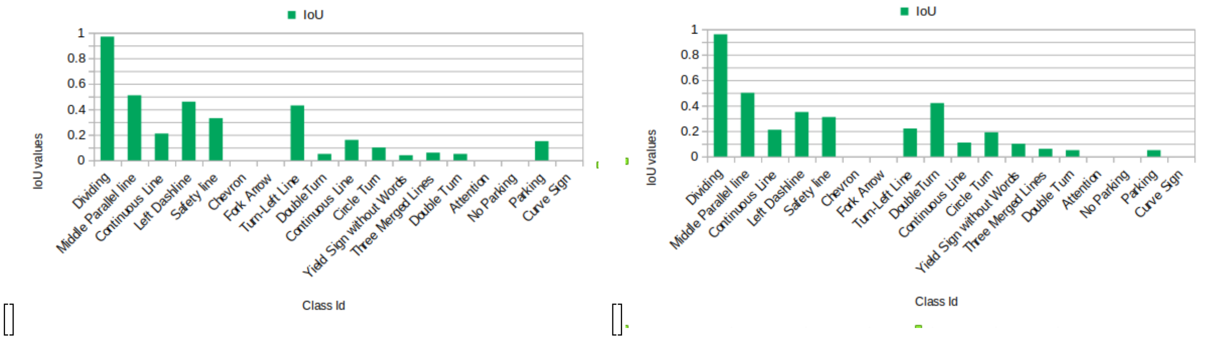


Figure 4.57: Binary Class(left) and Weighted Cross Entropy(right) IoU Values

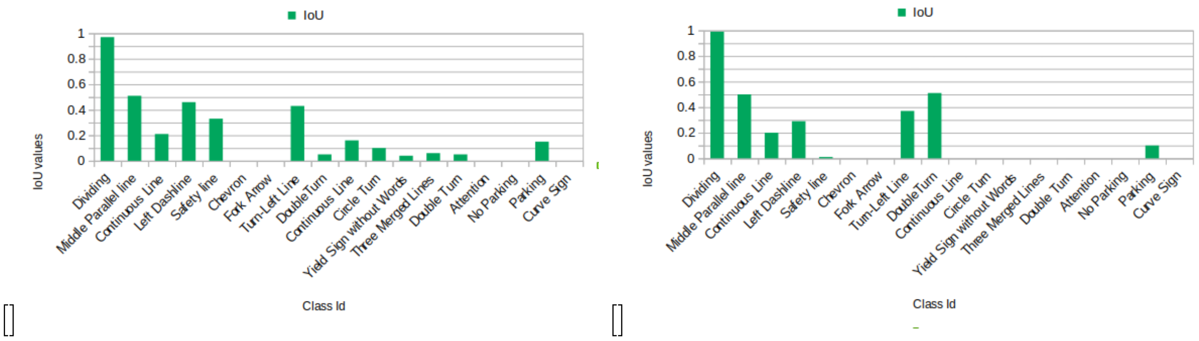


Figure 4.58: Center Crop method(left) and Standard Cross Entropy Loss(right) IoU values

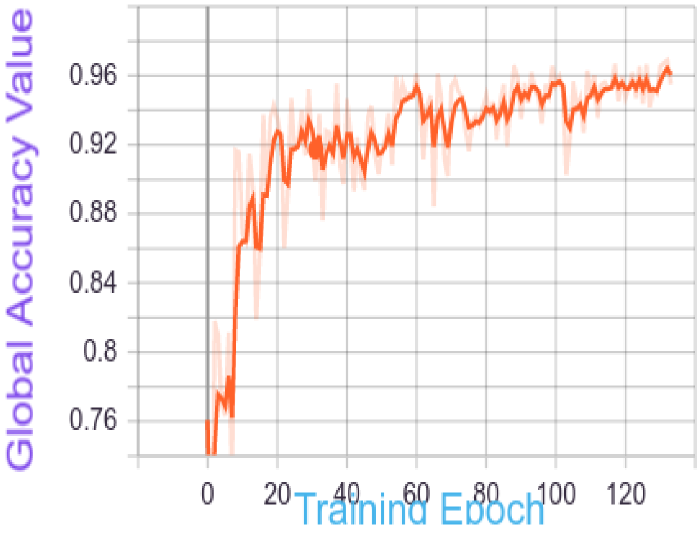


Figure 4.59: Global Accuracy

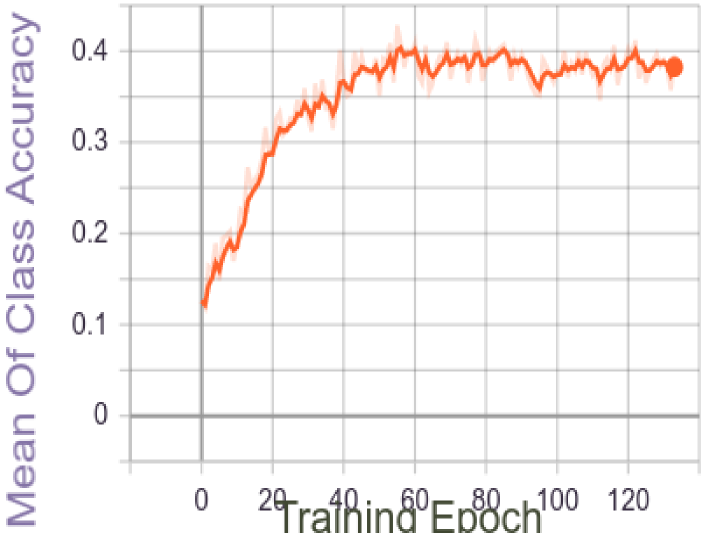


Figure 4.60: Mean of Class Accuracy

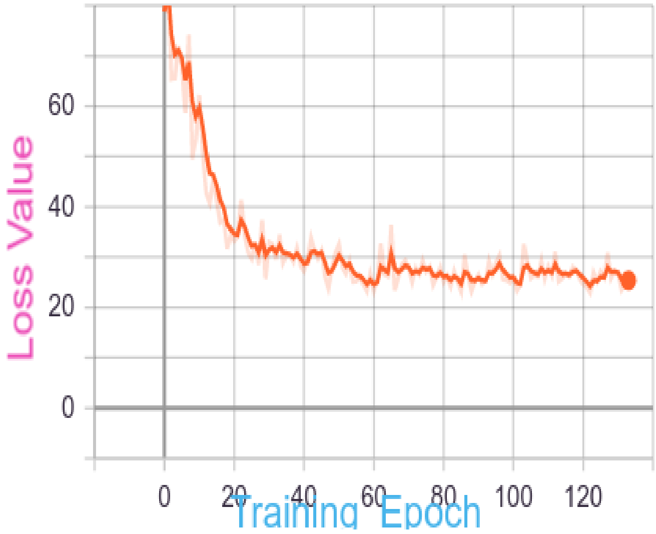


Figure 4.61: Total Loss Epoch

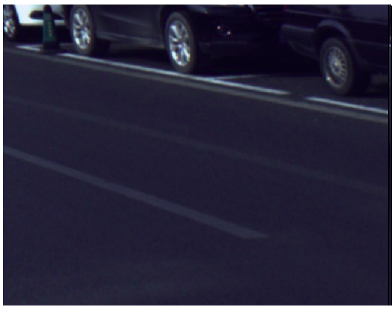


Figure 4.62: RGB(a)

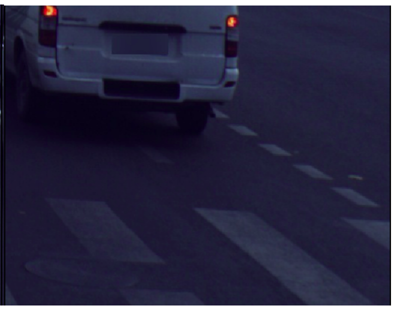


Figure 4.63: RGB(b)

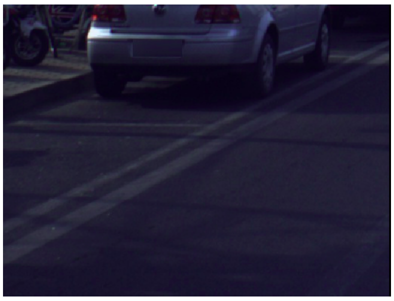


Figure 4.64: RGB(c)

are all in road signs in this case, then entire discussion relies on the achieve predicted images by generating input images to network, to train, as it is obvious starting from Fig 4.62 to Fig 4.70. Predicted images are almost identical to the ground-truth images.

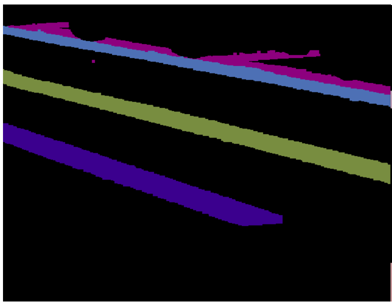


Figure 4.65: GT(a)

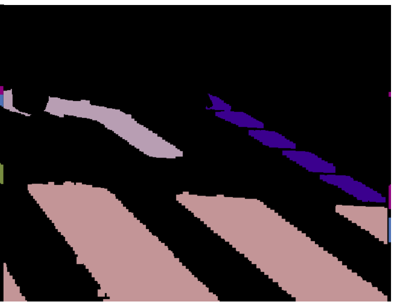


Figure 4.66: GT(b)

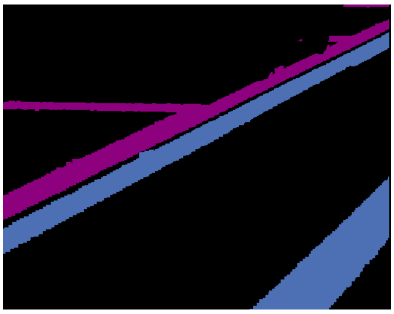


Figure 4.67: GT(c)

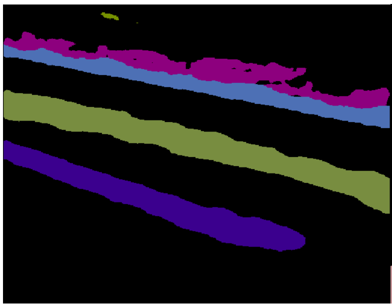


Figure 4.68: Prediction(a)

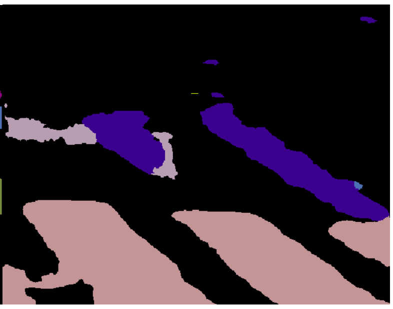


Figure 4.69: Prediction(b)

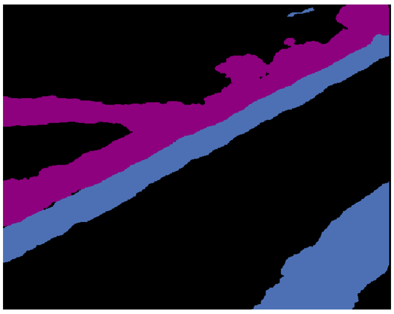


Figure 4.70: Prediction(c)

Classes	IoU	Class Accuracy
Dividing	0.97	0.97
Middle Parallel line	0.51	0.82
Continuous Line	0.21	0.60
Left Dashline	0.46	0.79
Safety line	0.33	0.68
Chevron	-	-
Fork Arrow	-	-
Turn-Left Line	0.36	0.85
DoubleTurn	0.43	0.95
Continuous Line	0.05	0.29
Circle Turn	0.16	0.21
Yield Sign without Words	0.10	0.38
Three Merged Lines	0.04	0.23
Double Turn	0	0
Attention	-	-
No Parking	-	-
Parking	0.15	0.34
Curve Sign	-	-

Table 4.6: Center Crop and Weighted Cross Entropy Loss with Correctly Classified pixels(Classes with NaN values, are not mentioned in the table)

4.1.5 Conclusion

We clearly represented all proposed techniques on the task of Lane Marking Semantic Segmentation by Deep Convolution Neural Network. Regardless of each method huge experimental attempts to assist network to produce clear and accurate prediction with 38 labels, only utilization of Weighted Cross Entropy Loss in lane marking semantic segmentation has great impact on providing high class accuracy. We all take into account methods with their details continuously modified metrices graphically, thus, finally it would be reasonable to introduce mean of class accuracy, global accuracy, as well as IoU by table 4.7

Methods	mIoU	Mean of Class Accuracy	Global Accuracy
Standard Cross Entropy Loss	0.17	0.20	0.98
Weighter Cross Entropy Loss	0.18	0.38	0.96
Composition of 38 classes into Binary	0.61	0.93	0.93
Center Crop and Weighted Cross Entropy	0.19	0.40	0.97

Table 4.7: Mean value of Class Accuracy, IoU and Global Accuracy on different methods

Bibliography

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image seg-

mentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[9] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1458–1465. IEEE, 2005.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[15] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[16] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2016.

[17] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE international conference on computer vision*, pages 1377–1385, 2015.

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[19] Sankaran Panchapagesan, Ming Sun, Aparna Khare, Spyros Matsoukas, Arindam Mandal, Björn Hoffmeister, and Shiv Vitaladevuni. Multi-task learning and weighted cross-entropy for dnn-based keyword spotting. In *Interspeech*, volume 9, pages 760–764, 2016.

[20] Pedro Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *International conference on machine learning*, pages 82–90, 2014.

[21] Demetri Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):129–139, 1986.

[22] Xiaoling Xia, Cui Xu, and Bing Nan. Inception-v3 for flower classification. In *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pages 783–787, 2017.

[23] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *The European Conference on Computer Vision (ECCV)*, September 2018.