HYBRIDNETS: END-TO-END PERCEPTION NETWORK

Vu Thanh Dat

Ngo Viet Hoai Bao

A thesis submitted in part fulfilment of the degree of BSc. (Hons.) in Computer Science with the supervision of Assoc. Prof. Phan Duy Hung.

Bachelor of Computer Science

Hoa Lac campus - FPT University

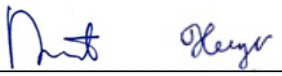
20 April 2022

This thesis is dedicated to our hard-working Assoc. Prof. Hung, whose much-needed assistance, continuous support and funding made this work reach its fullest potential as of today.

“Hope is the one thing that can help us get through the darkest of times”

DECLARATION

This thesis is the result of our own work and includes nothing, which is the outcome of work done in collaboration except where specifically indicated in the text. It has not been previously submitted, in part or whole, to any university of institution for any degree, diploma, or other qualification.

Signed: 

Date:___04/20/2022_____

Vu Thanh Dat, Ngo Viet Hoai Bao and full qualifications

FPT University

ABSTRACT

End-to-end Network has become increasingly important in multi-tasking. One prominent example of this is the growing significance of a driving perception system in autonomous driving. This thesis systematically studies an end-to-end perception network for multi-tasking and proposes several key optimizations to improve accuracy. First, the study proposes efficient segmentation head and box/class prediction networks based on weighted bidirectional feature network. Second, the study proposes automatically customized anchor for each level in the weighted bidirectional feature network. Third, the study proposes an efficient training loss function and training strategy to balance and optimize network. Based on these optimizations, we have developed an end-to-end perception network to perform multi-tasking, including traffic object detection, drivable area segmentation and lane detection simultaneously, called HybridNets, which achieves better accuracy than prior art. In particular, HybridNets achieves **77.3 mean Average Precision** on Berkeley DeepDrive Dataset, outperforms lane detection with **31.6 mean Intersection Over Union** with **12.83 million** parameters and **15.6 billion** floating-point operations. In addition, it can perform visual perception tasks in real-time and thus is a practical and accurate solution to the multi-tasking problem. Code is available at https://github.com/datvuthanh/HybridNets.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

Abbreviations	Meaning
BDD100K	Berkeley DeepDrive Dataset
BFLOPs	Billion Floating-point Operations
BiFPN	Weighted Bi-directional Feature Pyramid Network
ERNet	Efficient Residual Neural Network
FCN	Fully Convolutional Networks
FPN	Feature Pyramid Network
HSV	Hue, Saturation, Value
IoU	Intersection over Union
mAP50	Mean Average Precision at IoU threshold 0.5
mIoU	Mean Intersection over Union
PSPNet	Pyramid Scene Parsing Network
R-CNN	Region-based Convolutional Neural Network
RPN	Region Proposal Network
SCNN	Spatial Convolutional Neural Network
SSD	Single Shot MultiBox Detector
SSN	Strong-Structural Convolution Neural Network
YOLO	You Only Look Once
YOLOP	You Only Look Once for Panoptic Driving Perception

1 INTRODUCTION

1.1 Background

Recent advances in embedded systems' computational power and neural networks' performance have made autonomous driving an active field in computer vision. Ideally, to create a vehicle capable of driving itself is to feed it with every bit of information available in its immediate surroundings. However, unlike conventional thinking, lidar and radar are not required to create an accurate perception field for intelligent vehicles. From time to time, it has been shown that such vehicles can make relatively good driving decisions with just the assistance of a single camera attached to the front. There is a general consensus that the three most critical tasks in guiding intelligent vehicles are: traffic object detection, drivable area segmentation, and lane line segmentation.

Each one of these tasks has got its state-of-the-art networks, including but not limited to SSD [14], YOLO [21] for object detection; U-Net [23], SegNet [1], ERNet [10] for semantic segmentation; LaneNet [29] and SCNN [19] for lane line detection. Still, passing an image through three different networks creates unreasonable latency. Many researchers (MultiNet [28], DLT-Net [20], YOLOP [30]) have thought about combining the networks into a simple encoder-decoder architecture, where the backbone and neck generate context for three different heads to process. The architecture can be improved even further with proper selection of the feature extractor and fusing lane line with drivable area into one segmentation head. This experiment achieves the highest recall of 92.8% and segmentation IoU of 70.8%, outperforming existing multi-task networks on the challenging BDD100K dataset [31], as shown qualitatively in Figure 1.

Improvements are made upon the excellent multi-scale feature fusion BiFPN in EfficientDet [26], together with an EfficientNet [27] backbone pre-trained on ImageNet with its balanced trade-off between accuracy and computational overhead. A BiFPN decoder is constructed to utilize existing multi-scale features into the newly designed segmentation head. For an input resolution of 640x384, the entire network comes in at 15.6 BFLOPs on 12.83M parameters, comparable to the latest multi-task network YOLOP at 18.6 BFLOPs on 7.9M parameters. A multi-stage learning strategy is employed to help with the convergence of multiple loss functions [5].

To finetune even further, we also tinker with anchor box generation in this study [22]. Because anchor boxes theoretically cannot be generalized well for every dataset, nevertheless having a significant impact on the performance of one-stage detectors, we empirically choose the best possible aspect ratios and scales for the driving dataset BDD100K, where objects vary from large upfront trucks to tiny further cars.

To sum it up, the main contributions of this research are:

1. HybridNets, an end-to-end perception network, achieving outstanding results in real-time on the BDD100K dataset.
2. Automatically customized anchor for each level in the weighted bidirectional feature network, on any dataset.
3. An efficient training loss function and training strategy to balance and optimize multi-task networks.

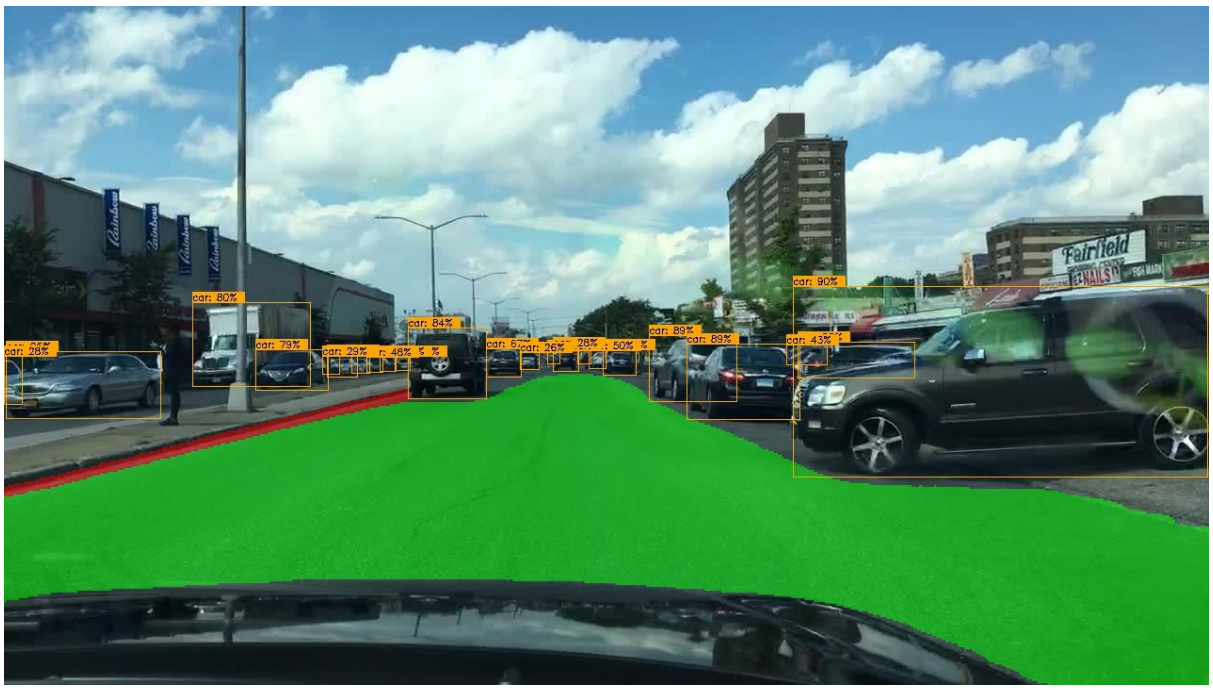


Figure 1: Results for inference of HybridNets. Our proposed network performs three tasks, including traffic object detection, drivable area segmentation and lane line detection. The green areas indicate the drivable area, the blue lines are the lane lines, and the orange boxes are the traffic objects.

1.2 Related works

This section will review some of the best networks in each respective task, then conclude with the latest multi-task networks to emphasize the strength of this unified architecture.

1.2.1 Traffic object detection

Current developments in improving detectors' performance have nearly split the area into two distinct branches: region-based and one-stage detectors. While region-based methods are more accurate, one-stage detectors gained more attraction due to their efficiency in embedded systems with limited hardware constraints. When FPN came about, it initially supported RPNs by providing a top-down pathway to construct higher resolution layers from a semantic-rich layer [11]. Then BiFPN officially showed the performance boost of bidirectional feature fusion to one-stage detectors. They can now take in multiple scales of the feature map in just one pass, alleviating the apparent weakness of YOLOs and the like.

1.2.2 Drivable area segmentation

Semantic segmentation has also made remarkable steps with deep-learning instead of the old-fashioned segmentation algorithms. FCN [25] sparked the flame with the first fully convolutional segmentation network. From then on, researchers have found various ways to improve the performance, such as encoder-decoder architecture with U-Net [23], the pyramid pooling module of PSPNet [32], or even semisupervised learning based on generative adversarial networks [6]. SSN [18] incorporated conditional random field units in the post-processing stage to increase segmentation performance. Many data augmentation techniques have been tested throughout to enhance the learning generalization of road detection networks [17]. Image analysis is still being explored in segmenting road scenes [9].

1.2.3 Lane line detection

Traditional lane line detections algorithms have been in wide use until recently, a notable algorithm being Hough transform [33]. Then LaneNet [29] proposed individual lane lines as instances to be segmented. SCNN [19] preferred slice-by-slice convolutions over deep layer-by-layer convolutions, emphasizing objects with heavy spatial relationships but barely noticeable appearances, such as poles, traffic lights, or lane lines. ENet-SAD [8] created self attention distillation, a technique allowing models to self-learn. It works by using attention maps generated in earlier training points as a form of supervision for later, surpassing SCNN by a large margin.

1.2.4 Multi-task network

Many published papers attempted to combine perception tasks into a unified network. Mask R-CNN [7] inherited RPN from Faster R-CNN while adding a third output branch for object mask, enabling the parallelization of object detection and instance segmentation. BlitzNet [4] also showed that object detection and semantic segmentation could benefit from each other. LSNet [13] came with a novel loss function named cross-IoU to add pose estimation into the output. MultiNet put forward the encoder-decoder structure, allowing DLT-Net to design special shared tensors between decoder heads for mutual information streams. Not long after, YOLOP became the first real-time state-of-the-art on the BDD100K dataset on three perception tasks: vehicle detection, drivable area, and lane line segmentation. However, the two similar segmentation heads left room for the obvious optimization task of reducing them to a single better-performing one. As hardware constraint is also of utmost importance to the application of any real-time decision-making network, model scaling must be taken into consideration.

2 METHODOLOGY

2.1 Network architecture

Based on these challenges, this research has proposed an end-to-end network architecture that can multi-task named HybridNets. As shown in Figure 2, our one-stage network includes one sharing encoder and two separated decoders to solve distinct tasks. The resolution of each feature map level P_i represents a feature level with resolution of $1/2^i$ of the input images. For instance, if input resolution is 640x384, the P_2 represents feature level 2 $(640/2^2, 384/2^2) = (160,96)$, while P_7 represents feature level 7 with resolution $(5,3)$.

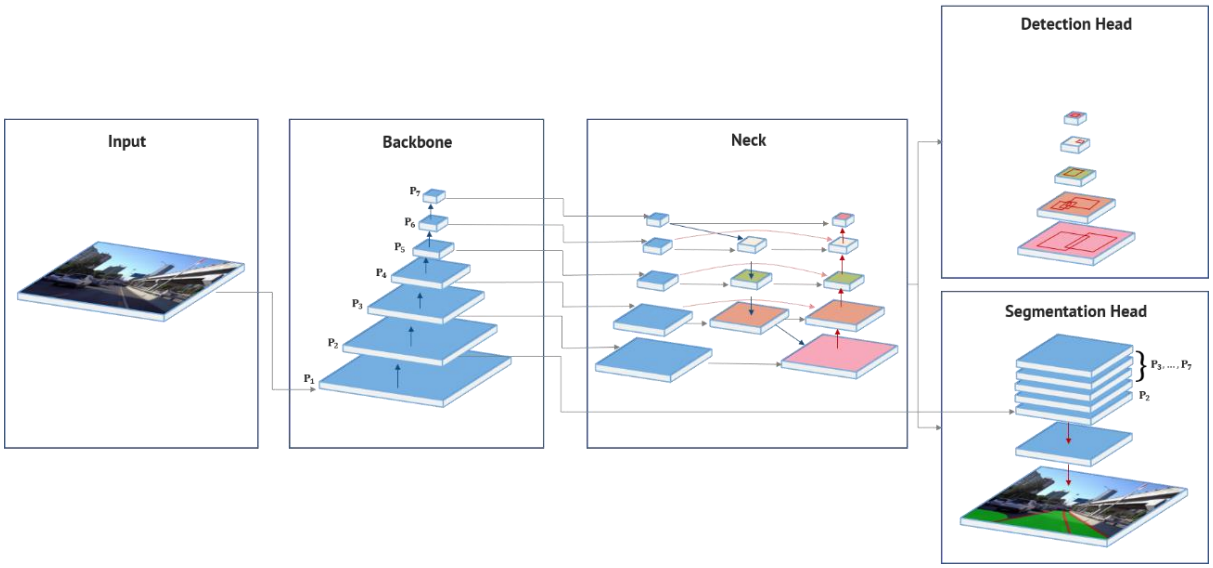


Figure 2: HybridNets Architecture. It consists of one encoder: backbone network and neck network; and two decoders: Detection Head and Segmentation Head. The backbone network generated 5 feature maps from P_1 to P_5. By down-sampling the feature map P_5, we obtain two feature maps P_6 and P_7.

2.2 Encoder

The feature extracting, serving as a backbone, is an essential part of the model that can help a variety of networks achieve excellent performance in various tasks. Many modern network architectures currently reuse networks that have good accuracy in the ImageNet dataset to extract features. Recently, EfficientNet showed high accuracy and efficient performance over existing CNNs, reducing FLOPs by orders of magnitude. We choose EfficientNet-B3 as the backbone, which solves the problem of network optimization by finding depth, width, and

resolution parameters based on neural architecture search to design a stable network. Therefore, our backbone can reduce the computational cost of the network and obtain several vital features.

The feature maps from the backbone network are fed to the neck network pipeline. Multi-scale feature representation is the main challenge; FPN recently proposed a feature extractor design to generate multi-scale feature maps to obtain better information. However, the limitation of FPN is that information feature is inherited by a one-way flow. Therefore, our neck network uses a BiFPN module based on EfficientDet. BiFPN fuses feature at a different resolution based on cross-scale connection for each node by each bidirectional (top-down and bottom-up) path and adds weight for each feature to learn the importance of each level. We adopt the method to fuse features in our work.

2.3 Decoder

Each grid of the multi-scale fusion feature maps from the Neck network will be assigned nine prior anchors with different aspect ratios. Similar to YOLOv4 [2], this study uses kmeans clustering [16] to determine anchor boxes. In addition, we choose 9 clusters and 3 different scales for each grid cell. In order to various feature map levels, this study uses scale constant to create bounding box priors that covers all regions from small to large. Thus, this proposed network can work well on complex dataset. The detection head will predict the offset of bounding boxes and the probability of each class as well as the confidence of the prediction boxes. This is described as

$$\begin{aligned} b_x &= \sigma(r_x) + c_x \\ b_y &= \sigma(r_y) + c_y \\ b_w &= c_w e^{r_w} \\ b_h &= c_h e^{r_h} \end{aligned} \tag{1}$$

Where r_x, r_y, r_w, r_h is the center, width and height of each bounding box, respectively from network prediction. Each anchor box has a center c_x, c_y, width c_w and height c_h.

Segmentation head has 3 classes for output, which are background, drivable area and lane line. This study keeps 5 feature levels $\{P_3, \dots, P_7\}$ from Neck network to segmentation branch. First, this study up-samples each level to have the same output feature map with size $(\frac{W}{4}, \frac{H}{4}, 64)$. Second, feeding P_2 level to convolution layer to have the same feature map

channels with other levels. Then, we combine them to obtain a better feature fusion by summing all levels. Finally, we restore the output feature to the size $(W, H, 3)$, representing the probability of each belonging pixel class. This research scales feature maps to the size of P_2 level, because P_2 level is a strongly semantic feature map. Additionally, we feed P_2 feature map from backbone network which represents low-level feature into the final feature fusion that helps network improve output precision, as shown in Figure 3.

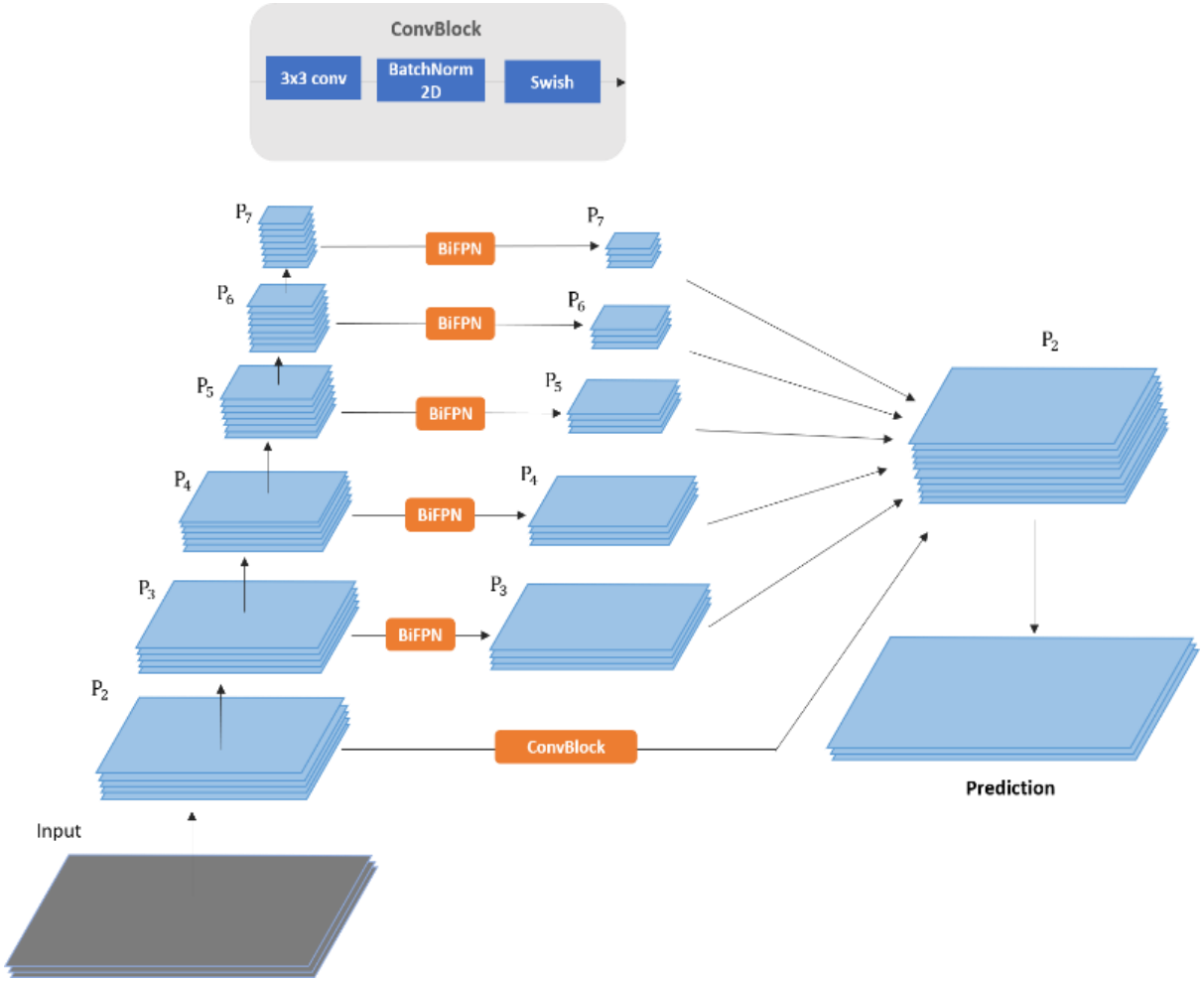


Figure 3: The Segmentation branch of HybridNets architecture.

2.4 Loss Function and Training

This study uses multi-task loss to train end-to-end network. Equation 2 expresses the total loss function by summing of two parts.

$$\mathcal{L}_{all} = \alpha \mathcal{L}_{det} + \beta \mathcal{L}_{seg} \tag{2}$$

Where α, β are tuning parameters to balance the total loss, $\mathcal{L}_{det}$ is the loss for object detection task and $\mathcal{L}_{seg}$ is the loss for segmentation task, the formulation can be written as follow

$$\mathcal{L}_{det} = \alpha_1 \mathcal{L}_{class} + \alpha_2 \mathcal{L}_{obj} + \alpha_3 \mathcal{L}_{box} \tag{3}$$

$\mathcal{L}_{class}$ and $\mathcal{L}_{obj}$ are focal loss [12], which is implemented for classifying class and the confidence of objects, respectively. The focal loss reduces the slope of loss function and focuses on misclassified examples. $\mathcal{L}_{box}$ is computed by smooth L1 loss, which takes absolutely between the predicted box and ground truth box, can be expressed as

$$\begin{aligned} \text{smooth}_{L1}(\text{x}) &= \begin{cases} \delta_1 x^2 & if \quad x < \delta_2 \\ x - \delta_1 \end{cases} \\ \text{x} &= \delta b_p \cdot \left| b_x - \hat{b}_x \right| + \left| b_y - \hat{b}_y \right| + \left| b_w - \hat{b}_w \right| + \left| b_h - \hat{b}_h \right| \end{aligned} \tag{4}$$

Where $\hat{b}$ is the prediction of bounding box and b is the ground truth, and b_p is determined a positive label has been assigned to a grid cell. In this study, we force size some anchor boxes to the regression network can learn smoothly, b_p can be written as

$$b_p = \begin{cases} 1, & if \quad IoU(c_i, b_j) \ \geq \ 0.5 \quad i = 1,...,\sum_{k=1}^{5} n_k m_k; j = 1,...,p \\ 0, & otherwise \end{cases} \tag{5}$$

Where c_i is the anchor box i^{th}, the total of anchor boxes is combining of each feature map level with n_k, m_k is the resolution of feature map, and p is the total of ground truth bounding boxes of each input image. Next $\mathcal{L}_{seg}$ is multiclass hybrid loss that is utilized for multi-class segmentation of background, drivable area and lane line. Small object segmentation is a challenge in semantic segmentation caused by imbalanced data distribution. Therefore, this study combines $\mathcal{L}_{Tversky}$ Tversky loss [24] and $\mathcal{L}_{Focal}$ Focal loss [12] to predict the class to which a pixel belongs. $\mathcal{L}_{Tversky}$ performs well at class-imbalanced problems and optimizes the maximization of score, whereas $\mathcal{L}_{Focal}$ aims to minimize the classification error between pixels and focuses on hard labels.

$$
\begin{aligned}
TP_p(c) &= \sum_{n=1}^{N} p_n(c) g_n(c) \\
FN_p(c) &= \sum_{n=1}^{N} (1 - p_n(c)) g_n(c) \\
FP_p(c) &= \sum_{n=1}^{N} p_n(c)(1 - g_n(c)) \\
\mathcal{L}_{seg} &= \mathcal{L}_{Tversky} + \lambda \mathcal{L}_{Focal} \\
\mathcal{L}_{Tversky} &= C - \sum_{c=0}^{C-1} \frac{TP_p(c)}{TP_p(c) + \varphi FN_p(c) + (1-\varphi) FP_p(c)} \\
\mathcal{L}_{Focal} &= -\lambda \frac{1}{N} \sum_{c=0}^{C-1} \sum_{n=1}^{N} g_n(c)(1 - p_n(c))^{\gamma} \log(p_n(c))
\end{aligned}
\tag{6}
$$

Where $TP_p(c), FN_p(c), FP_p(c)$ are true positives, false negatives and false positives for class N, $p_n(c)$ is the predicted probability for pixel n belonging to class c, $g_n(c)$ is the ground truth for pixel n being in class c. C is the number of classes and N is the total number of pixels in the input image.

During training, this study does several experiments to finetune a lot of hyperparameters and suitable architecture networks. Training from an end-to-end approach will cost computation and training time. In addition, several optimization algorithms have also been experimented. Therefore, to compress the training time and optimize hyperparameters, we construct a training strategy in order to train the model step by step and quickly transform the experiments. Algorithm 1 illustrates the strategy of our training method.

Algorithm 1: HybridNets training stage. First, we only train Encoder and Detection Head as object detection task. Second, we freeze the Encoder, Detection head and unfreeze parameters from Segmentation Head. Finally, the final network is trained jointly for all tasks.

Input: Target end-to-end network $\mathcal{F}$ with parameter group

$\Theta = \{\theta_{enc}, \theta_{det}, \theta_{seg}\}$;

Training dataset $\mathcal{T}$;

Threshold for convergence $\gamma = \{\gamma_1, \gamma_2, \gamma_3\}$;

Total loss function $\mathcal{L}$;

Pivot strategy $\mathcal{P} = \{\{\theta_{enc}, \theta_{det}\}, \{\theta_{seg}\}, \{\theta_{enc}, \theta_{det}, \theta_{seg}\}\}$

Output: Proposed network: $\mathcal{F}(\mathcal{X}, \Theta)$

procedure Train $(\mathcal{F}, \mathcal{T})$
    for i = 0 to length$(\mathcal{P})$ - 1
        $\Theta \leftarrow \Theta \cap \mathcal{P}[i]$ // Freeze parameters
        repeat
            Sample a mini-batch $(\mathbf{x}_m, \mathbf{y}_m)$ from training dataset $\mathcal{T}$
            $\ell \leftarrow \mathcal{L}_{all}(\mathcal{F}(\mathbf{x}_m; \Theta), \mathbf{y}_m)$
            $\Theta \leftarrow \arg\min_\theta \ell$
        until $\ell < \gamma[i]$
        if i < length$(\mathcal{P})$ - 1 **then**
            $\Theta \leftarrow \Theta \cup \mathcal{P}[i+1]$
        endif
    end for
end procedure
Train $(\mathcal{F}, \mathcal{T})$
return Proposed network $\mathcal{F}(\mathcal{X}, \Theta)$

3 EXPERIMENTATION AND EVALUATION

3.1 Experiment settings

The BDD100K dataset is used in training and validating the model. Since the test labels of 20K images are unavailable, we opt to evaluate on the validation set of 10K images. The dataset for three tasks is prepared according to existing multi-task networks trained on BDD100K to aid in comparison. Of all the ten classes in object detection, only {car, truck, bus, train} is selected and merged into a single class {vehicle} since DLT-Net and MultiNet can only detect vehicles. Two segmentation classes {direct, alternative} are also merged into {drivable}. We follow the practice of calculating two lane line annotations into a central one, dilating the annotations in training set to 8 pixels while keeping validation set intact [8]. Images are resized from 1280x720 to 640x384 due to three main reasons, in order of importance: respecting the original aspect ratio, maintaining a good trade-off between performance and accuracy, and making sure the dimensions are divisible by 128 for BiFPN. Basic augmentation techniques such as rotating, scaling, translating, horizontal flipping, and HSV shifting are used. Mosaic augmentation, first introduced in YOLOv4 with great results [2], is utilized while training detection head specifically.

We jump-start the model by using EfficientNet-B3 weights pre-trained on ImageNet. The custom anchor box settings found automatically have scales of $(2^0, 2^{0.7}, 2^{1.32})$ and ratios of $[(0.62,1.58),(1.0,1.0),(1.58,0.62)]$. The chosen optimizer is AdamW [15] with $\gamma = 1e^{-3}, \beta_1 = 0.9, \beta_2 = 0.999, \xi = 1e^{-8}, \lambda = 1e^{-2}$. When the model stucks around for 3 epochs, learning rate is decreased tenfold. For object detection, the model uses smooth L1 loss with $\delta_1 = 4.5, \delta_2 = 1/9$ for regression and focal loss with $\alpha = 0.25, \gamma = 2.0$ for classification. When matching anchor boxes to annotations, the model uses an IoU threshold of 0.5 for annotations larger than 100 pixels in area but only 0.25 for those smaller. We emphasize regression 4 times more than classification because one-class classification is easy to converge. For drivable area and lane segmentation, the model uses a combination of Tversky loss with $\alpha = 0.7, \beta = 0.3$ and Focal loss with $\alpha = 0.25, \gamma = 2.0$. We train with a batch-size of 16 on a RTX 3090 for 200 epochs.

3.2 Evaluation metrics

On traffic object detection task, this proposed method uses mAP50. mAP50 is computed by the average of the Average Precision calculated for all the classes at single IoU threshold 0.5. Average Precision is the area under the precision-recall curve. This study only evaluates one class, focusing on how good the proposed method can find all the positives. This study sets the lowest confidence and all bounding boxes is computed by mAP50. On semantic segmentation task, IoU metric is used to evaluate drivable area and lane line segmentation. To be more specific, this study presents mIoU as average of IoU for each class and IoU metric for single class.

3.3 Cost computation performance

Table 1 compares HybridNets with other multi-task networks. Although our HybridNets has more extensive parameters (12.83M) than YOLOP (7.9M), the number of computations of HybridNets is lower than the compared networks. By adopting depth-wise separable convolutions [3], the computations are significantly reduced to **15.6 BFLOPs**. In addition, we have also compared the inference latency on V100 GPU FP16. Specifically, our V100 latency is the time processing of the model, not including preprocessing and NMS postprocessing. Compared to previous multi-networks, HybridNets are up to **1.4x** faster on GPU. Therefore, HybridNets can run in real-time on standard devices and embedded devices.

Model	Params	FLOPs	Latency (ms) V100
YOLOP	7.9M	18.6B	52
HybridNets	12.83M	15.6B	37

Table 1: Cost computation result for various multi-networks. Params and FLOPs denote the number of parameters and the number of computations. Latency is for inference with batch size 1.

3.4 Multi-task performance

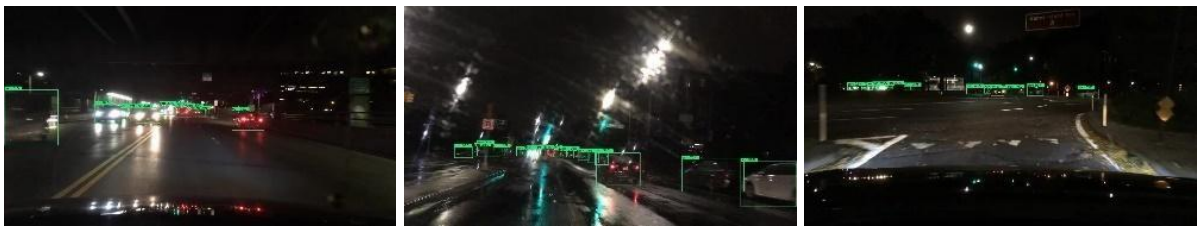
The second experiment presents results on three tasks, including traffic object detection, drivable area segmentation, and lane line segmentation. We present the vehicle detection results and compare them to six models on the BDD100K dataset.

Model	Recall (%)	mAP50 (%)
MultiNet	81.3	60.2
DLT-Net	89.4	68.4
Faster R-CNN	77.2	55.6
YOLOv5s	86.8	77.2
YOLOP	89.2	76.5
HybridNets	92.8	77.3

Table 2: The comparison result on traffic object detection task. The experiment settings include confidence threshold of 0.001 and NMS threshold of 0.6. This research mainly focuses on obtaining highest Recall IoU.

As listed in Table 2, HybridNets outperforms performance to previous networks on the BDD100K dataset. Our model achieves **3.6%** better recall and achieves the best mAP50 at **77.3%**. HybridNets can detect incredibly small objects ranging from 3 pixels to 10 pixels with input size (640,384,3) thanks to our automatically customized anchor aspect ratio and scale. Figure 4 illustrates the visualization of traffic object detection.

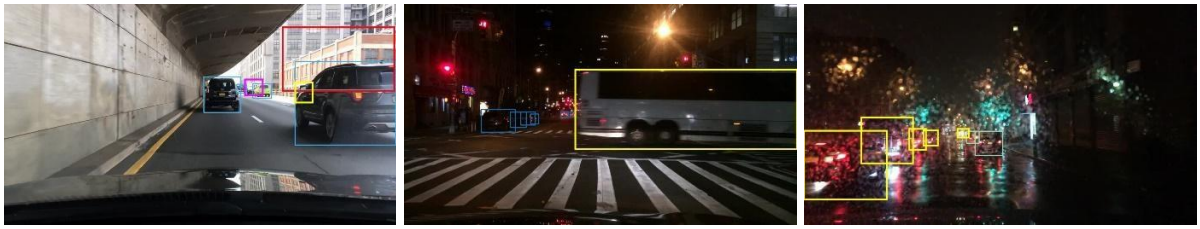
(a) Day-time result



(b) Night-time result

Figure 4: Visualization of the traffic object detection results of HybridNets. Fig. 4. (a) shows results in day-time series with different weather conditions such as clear, heat stroke and heat-wave. Fig. 4. (b) shows results in night-time series with different weathers such as cool and flurries.

As shown in Figure 5, our proposed architecture makes further improvement compared to YOLOP. Specifically, HybridNets detects small objects and large objects in traffic object detection task, whereas YOLOP has high False Negative score and detects wrong objects. In addition, HybridNets works well in various complex weather conditions and the bounding boxes are more accurate.



(a) YOLOP

(b) HybridNets

Figure 5: Comparison between YOLOP and HybridNets on traffic object detection. The first row shows issues of YOLOP and the second row shows the result of HybridNets. The red bounding boxes are the false positive, the yellow bounding boxes are the false negative and the purple bounding boxes are not accurate.

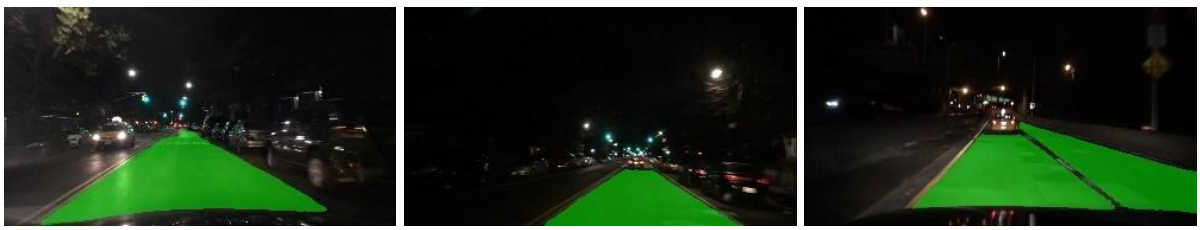
Next we evaluate the drivable area segmentation task. IoU metric is used to evaluate the segmentation performance of various networks.

Model	Drivable mIoU (%)
MultiNet	71.6
DLT-Net	71.3
PSPNet	89.6
YOLOP	91.5
HybridNets	**90.5**

Table 3: Performance comparison on drivable area segmentation task.

Table 3 shows the Drivable IoU of five networks. Our HybridNets achieves 90.5 % mIoU, pale in comparison to YOLOP (91.5%). We built a decoder network for multi-classes, whereas YOLOP constructed two decoders for specific tasks. Therefore, our HybridNets is more flexible and optimistic than theirs. Figure 6 visualizes the semantic segmentation output of drivable areas in various conditions. As shown in Figure 7, the comparison between HybridNets and YOLOP on Drivable Segmentation task, HybridNets is more accurate than YOLOP. To be more specific, YOLOP focuses on evaluating the pixel to which class it belongs, while needing to consider the intersection of bounding boxes. Therefore, the YOLOP model does not work well in harsh conditions such as night or areas with a lot of noise. Based on our Neck Network using BiFPN architecture, the information of different receptive fields is combined from various feature map levels with weighted parameters. Thus, HybridNets can improve the performance of drivable area segmentation task.

(a) Day-time result



(b) Night-time result

Figure 6: Visualization of the drivable area segmentation results of HybridNets. Fig. 6. (a) shows semantic segmentation results in day-time with various views. Fig. 6. (b) shows results in night-time series with various brightness views.

(a) YOLOP

(b) HybridNets

Figure 7: Comparison between YOLOP and HybridNets on drivable area segmentation. The first row shows the issue of mismatched pixels of YOLOP and the second row shows the result of HybridNets. The red regions are false positive and the yellow regions are false negative.

Finally, lane detection is one of the main challenges in autonomous driving. The evaluation metrics we use for lane detection are accuracy and IoU. As shown in Table 4, our HybridNets outperforms all previous models with accuracy **85.4** % and IoU **31.6** %. The proposed method works well in various complex weather conditions as shown in Figure 8. As shown in Figure 9, the lane detection results from YOLOP have mismatched pixels and less accuracy, whereas HybridNets works well on lane detection task. The lane lines from HybridNets are continuous and have high accuracy with less sparse supervisory. However, lane line has low IoU because of our approach in preprocessing the training dataset, making lane line annotation easier to learn with the drawback of suboptimal results. Thus, this study has added another metric of accuracy to evaluate in a more objective and fair manner.

Model	Accuracy (%)	Lane Line IoU (%)
Enet	34.12	14.64
SCNN	35.79	15.84
Enet-SAD	36.56	16.02
YOLOP	70.50	26.2
HybridNets	**85.4**	**31.6**

Table 4: Performance comparison on lane detection task.

(a) Day-time result

(b) Night-time result

Figure 8: Visualization of the lane detection results of HybridNets. Fig. 8. (a) shows results in day-time series with various weather conditions. Fig. 8. (b) shows results in night-time series with various brightness views.

(a) YOLOP

(b) HybridNets

Figure 9: Comparison between YOLOP and HybridNets on lane detection. The first row shows the issue of mismatched of YOLOP and the second row shows the result of HybridNets. The green regions are false positive and the yellow regions are false negative.

Figure 10 shows the results of HybridNets. The red lines are the lane lines, the green areas are the drviable area, and the orange bounding boxes are traffic objects. Our HybridNets has great performance in most scenarios. Based on the context structures, the drivable area provides information for the model to help train the model to converge faster. Moreover, each task provides context structure for other tasks. Therefore, our HybridNets can detect vehicles object easily, which challenges many other prior detection models. Therefore, the model can more easily predict traffic objects, which challenges many prior models. In general, our HybridNets works well in most complex scenarios such as severe reflective scenes and extreme weather conditions. However, our model is unable to adapt to the crossroads, the lane lines detection is broken and the drivable area is misjudged to be on the other side of the road in some cases.

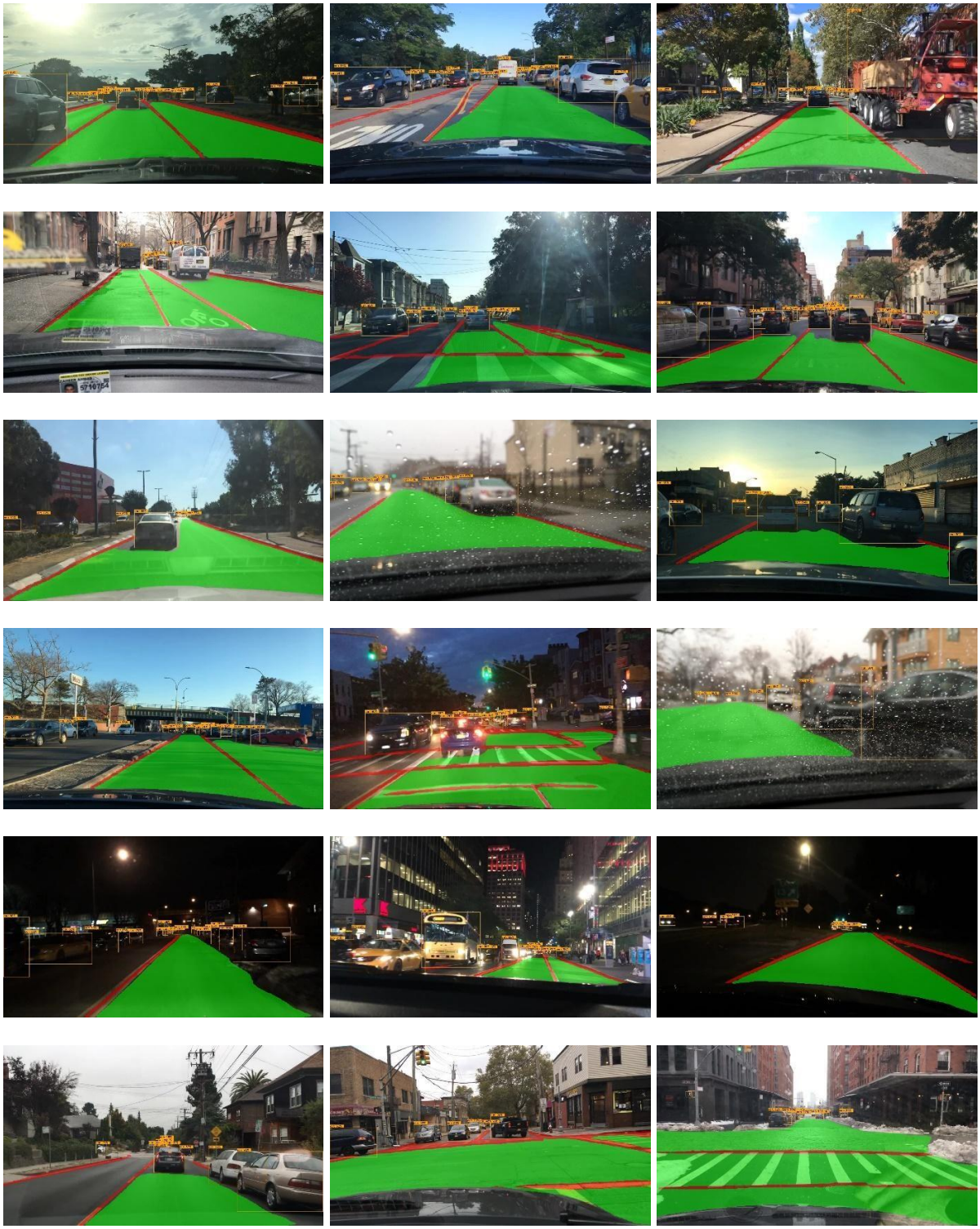


Figure 10: Multi-task results using HybridNets. The red lines are the lane lines, the green areas are the drivable area, and the orange bounding boxes are traffic objects.

4 CONCLUSION AND PERSPECTIVE

In this research, we systematically study network architecture design choices for multi-tasking, propose an efficient end-to-end perception network, customize automatic aspect ratios for each level in the weighted bidirectional feature network, and build efficient training loss function and training strategy to improve accuracy and performance. Based on these optimizations, we develop a new end-to-end multi-network, named HybridNets, which achieves better accuracy and efficiency than prior art across a broad spectrum of resource constraints. Most importantly, our network HybridNets achieves state-of-the-art accuracy with fewer FLOPS than previous multi-network models.

In future works, we would like to propose a robust network, which can perform many tasks related to perception and improve parameters and FLOPs of network. To be more specific, our work will focus on processing problems in autonomous driving such as building a decoder network that can detect 3-D object detection with only one input and classify several objects. We will try to ameliorate lane lines performance as well as context of structures in drivable area segmentation.

REFERENCES

1. V. Badrinarayanan, A. Kendall and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” IEEE Trans. Pattern Anal. Mach. Intell. 39, 12, 2481–2495, 2017, https://doi.org/10.1109/TPAMI.2016.2644615.
2. A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” arXiv:2004.10934, 2020.
3. F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1800–1807, 2017, https://doi.org/10.1109/CVPR.2017.195.
4. N. Dvornik, K. Shmelkov, J. Mairal and C. Schmid, "BlitzNet: A Real-Time Deep Network for Scene Understanding," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4174-4182, https://doi.org/10.1109/ICCV.2017.447.
5. V.A. Golovko, A.A. Kroshchanka, E.V. Mikhno, “Deep Neural Networks: Selected Aspects of Learning and Application,” Pattern Recognit. Image Anal. 31, 1, 132–143, 2021, https://doi.org/10.1134/S1054661821010090.
6. X. Han, J. Lu, C. Zhao, S. You and H. Li, "Semisupervised and Weakly Supervised Road Detection Based on Generative Adversarial Networks," IEEE Signal Processing Letters, vol. 25, no. 4, pp. 551-555, April 2018, https://doi.org/10.1109/LSP.2018.2809685.
7. K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 386-397, 1 Feb. 2020, https://doi.org/10.1109/TPAMI.2018.2844175.
8. Y. Hou, Z. Ma, C. Liu and C. C. Loy, "Learning Lightweight Lane Detection CNNs by Self Attention Distillation," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1013-1021, https://doi.org/10.1109/ICCV.2019.00110.
9. K.I. Kiy, “A New Method of Global Image Analysis and Its Application in Understanding Road Scenes,” Pattern Recognit. Image Anal. 28, 3, 483–495, 2018, https://doi.org/10.1134/S1054661818030100.
10. B. Li, J. Zang, J. Cao, “Efficient Residual Neural Network for Semantic Segmentation,” Pattern Recognit. Image Anal. 31, 2, 212–220, 2021. https://doi.org/10.1134/S1054661821020103.
11. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936-944, https://doi.org/10.1109/CVPR.2017.106.
12. T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318-327, 2020, https://doi.org/10.1109/TPAMI.2018.2858826.
13. B. Liu, H. Chen, Z. Wang, “LSNet: Extremely Light-Weight Siamese Network For Change Detection in Remote Sensing Image,” arXiv:2201.09156, 2022.

14. W. Liu et al., “SSD: Single Shot MultiBox Detector”. In: Leibe, B. et al. (eds.) Computer Vision – ECCV 2016. pp. 21–37 Springer International Publishing, Cham, 2016, https://doi.org/10.1007/978-3-319-46448-0_2.
15. I. Loshchilov, F. Hutter, “Decoupled Weight Decay Regularization,” arXiv:1711.05101, 2019.
16. J.B. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observations,” in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, University of California Press, Berkeley, 281-297, 1967.
17. J. Muñoz-Bulnes, C. Fernandez, I. Parra, D. Fernández-Llorca and M. A. Sotelo, "Deep fully convolutional networks with random data augmentation for enhanced generalization in road detection," in Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 366-371, https://doi.org/10.1109/ITSC.2017.8317901.
18. Y. Ouyang, “Strong-Structural Convolution Neural Network for Semantic Segmentation,” Pattern Recognit. Image Anal. 29, 4, 716–729, 2019, https://doi.org/10.1134/S1054661819040126.
19. X. Pan, J. Shi, P. Luo, X. Wang, X. Tang, “Spatial As Deep: Spatial CNN for Traffic Scene Understanding,” arXiv:1712.06080, 2017.
20. Y. Qian, J. M. Dolan and M. Yang, "DLT-Net: Joint Detection of Drivable Areas, Lane Lines, and Traffic Objects," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 11, pp. 4670-4679, Nov. 2020, https://doi.org/10.1109/TITS.2019.2943777.
21. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, https://doi.org/10.1109/CVPR.2016.91.
22. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517-6525, https://doi.org/10.1109/CVPR.2017.690.
23. O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28.
24. S.S.M. Salehi, D. Erdogmus, A. Gholipour, “Tversky Loss Function for Image Segmentation Using 3D Fully Convolutional Deep Networks,” In: Wang Q., Shi Y., Suk HI., Suzuki K. (eds) Machine Learning in Medical Imaging. MLMI 2017. Lecture Notes in Computer Science, vol 10541. Springer, Cham. https://doi.org/10.1007/978-3-319-67389-9_44.
25. E. Shelhamer, J. Long and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640-651, 1 April 2017, https://doi.org/10.1109/TPAMI.2016.2572683.
26. M. Tan, R. Pang and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10778-10787, https://doi.org/10.1109/CVPR42600.2020.01079.

27. M. Tan, Q.V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” arXiv:1905.11946, 2020.
28. M. Teichmann, M. Weber, M. Zöllner, R. Cipolla and R. Urtasun, "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving," in Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2018, pp. 1013-1020, https://doi.org/10.1109/IVS.2018.8500504.
29. Z. Wang, W. Ren, Q. Qiu, “LaneNet: Real-Time Lane Detection Networks for Autonomous Driving,” arXiv:1807.01726, 2018.
30. D. Wu, M. Liao, W. Zhang, X. Wang, “YOLOP: You Only Look Once for Panoptic Driving Perception,” arXiv:2108.11250, 2022.
31. F. Yu et al., "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2633-2642, https://doi.org/10.1109/CVPR42600.2020.00271.
32. 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 (CVPR), 2017, pp. 6230-6239, https://doi.org/10.1109/CVPR.2017.660.
33. F. Zheng, S. Luo, K. Song, C-W. Yan, M-C. Wang, “Improved Lane Line Detection Algorithm Based on Hough Transform,” Pattern Recognit. Image Anal. 28, 254– 260, 2018, https://doi.org/10.1134/S1054661818020049.