Chapter 3

Single-Stage Attention based Object Detection for Autonomous Vehicles

This chapter describes the deep learning-based model developed for multi object tracking. The model is introduced in Section 3.1. The description of the proposed model is given in Section 3.2. Section 3.3 gives the result generated by the proposed model and its analysis. Section 3.4 concludes the chapter.

3.1 Introduction

An AV is enabled with the self-driving capability to travel from one place to another without human intervention. These vehicles can detect objects like traffic signs and lights, other vehicles and pedestrians from surroundings in real-time to ensure collision avoidance, safety and accurate control decisions. The vehicles can detect the object with the help of various sensors like cameras, lidars and radars etc. The camera sensors can accurately recognize the external objects based on different image features such as color, texture and spatial. It is cost-effective compared to other sensors. The camera sensors play an essential role in object detection for AVs because images are used as input for most deep-learning-based techniques. The accuracy of camera sensors is better than humans for object detection; thus, deep-learning-based object detection with a camera sensor is a crucial method in ADS.

There should be two conditions to be followed by any detection techniques for AVs: accuracy and speed. High accuracy ensures the vehicle avoids collisions and abides by the traffic rules, while faster speed helps to make decisions quickly. Deep-learning-based object detection methods, which are crucial for the ADS, can be categorized into two parts: the region proposal method and the region-free method. The Region-based convolutional neural network (RCNN) [189], Fast RCNN [21], and Faster RCNN [23] are the proposal-based methods that are described briefly. Some proposal-free approaches are You only look once (YOLO series) [37] [38] [35] and Single-shot detection (SSD) [190] that used for real-time object detection, which is detail explained below. The single-stage object detector is faster than the two-stage object detector because there is no need to require any region proposal, so the proposed work is based on a single-shot technique.

It is well known that human perception is mainly based on attention. The neural attention mechanism utilizes a neural network that can focus on a subpart of its input (or features). The attention helps drive to reduce road accidents, avoid breaking the traffic rules and find more and fast control of vehicles. Attention-based object detection is very helpful in identifying the object class and location exactly. The spatial attention mechanisms can be divided into two parts based on state-of-the-art methods. The former is global [191] [192] and the latter one is local [193] [194] [195] and the description of these two mentioned below. In CSA-SS, the channel attention module works as global attention while spatial attention is used as local attention. The CSA-SS uses a combination of attention mechanisms to enhance the features by incorporating the attention block in the backbone network. The attention module is combined with channel attention and spatial attention. The channel attention mechanism provides more grained refine features and emphasizes 'what' is a semantic part from a given input. Apart from the channel attention mechanisms, spatial attention emphasizes 'where' is meaningful information, which works as a performance booster for the attention block. The channel and spatial attention modules work sequentially, as shown in Fig- 3.1 to produce refined, deep, semantic, shallow and high-resolution features. However, the existing methods of developing attention modules for better performance are not more suitable in the case of small objects. The proposed CSA-SS generates global channel attention by using Global average pooling (GAP), global max-pooling (GMP), and local spatial attention with a different aspect of the feature map efficiently and achieved a remarkable trade-off between speed and accuracy.

In this thesis, firstly, the model explores the attention mechanism with a backbone network to provide more accuracy and the anchor-free detection method help to reduce the computational cost. However, in practice, such as automatic driving, it is necessary to perform fast and accurate multiclass object detection. The main objective of the CSA-SS method is to provide a more accurate and faster object detector for the production systems. The model is easy to train on a limited number of GPUs with small batch sizes without affecting the accuracy of the object detector. For example, anyone who uses a GPU with less memory can train and test to achieve efficient and convenient results of real-time object detectors.

3.2 Proposed Method and Model

The CSA-SS is an effective and simple attention module that uses input features generated from the convolution layer and produces more refined features for classification. The base model uses FRN instead of batch normalization, surpassing state-of-the-art results even with small batch size. The CSA-SS evaluates the effectuality of the proposed attention module through ablation studies.The CSA-SS achieved state-of-the-art results on two standard datasets (KITTI [1] and BDD [2]) with a compact model. This model features conduct end-to-end training and more accurate result for the low resolution of input images, further maintaining the speed vs. accuracy trade-off.

The main objective of an Av is to accomplish object detection accurately, which

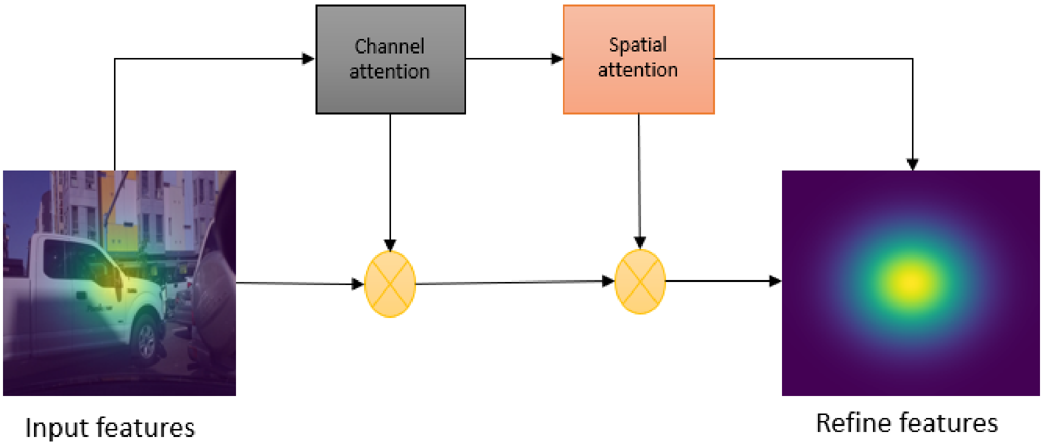

FIGURE 3.1: Configuration of the attention modules

depends on visual perception of the surrounding environment. The crucial task for high-level automation is to model the surrounding environment, also called environment perception. The onboard sensor's data have to be processed accurately to describe the surroundings, which is essential for automatic and safely navigating the car. The environment possesses both static and dynamic or moving contents.

3.2.1 Channel Spatial attention based Object Detector

The proposed method has a one-stage object detector with two attention blocks. The CSA-SS is a feed-forward-based convolutional network that generates bounding boxes and confidence scores. The confidence scores represent the instances of the object class that are present in those boxes. To detect the object from feature maps, CSA-SS used the same policy as YOLOv3. The detection kernel of size 1×1 is used to generate the prediction feature map. There are three prediction boxes, where every prediction box contains the coordinated boundary box (bx, by, bw and bh), the class score and the objectness scores. The class output (define object category) represents the score between 0 and 1 and objectness (shows the

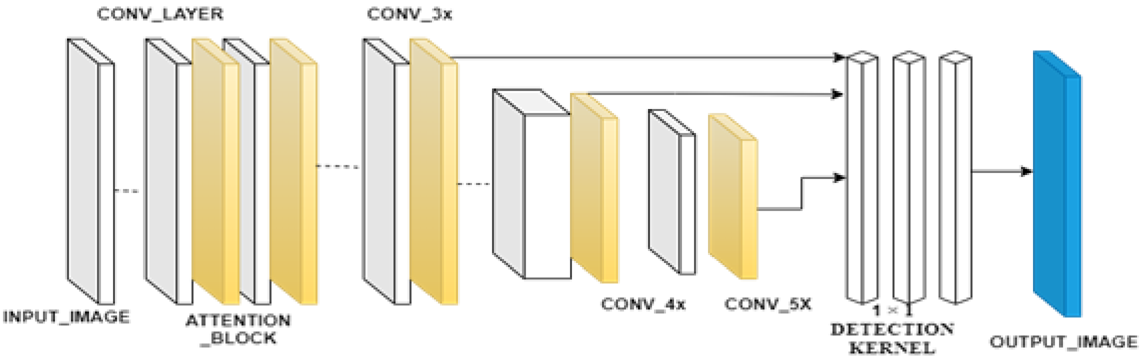

FIGURE 3.2: CSA-SS Architecture

presence of objects in the boundary box). The product of these two helps to detect the object. CSA-SS uses logistic regression and classification for the object score and class probability, respectively. Instead of the object score, the detection method focuses on deterministic coordinate values of the object, so the confidence score of the boundary box is unknown. A Non-maxima suppression (NMS) is used to select the final boundary box for an object. The working of the model is explained in the algorithm. Then, the proposed model added auxiliary structure to the network to produce detection with the following key features:

3.2.1.1 Filter Response Normalization

The base network is a variant of ResNet with filter response normalization [138] and attention blocks. The batch normalization is a cornerstone of the current high-performing deep neural network model but the BN reliances on sufficient large batch size, when trained with the small batch size, exhibits a significant degradation of performance. The filter response normalization (FRN) layer consists of a normalization and activation function that eliminates this type of shortcomings. FRN operates every batch sample on every activation map independently, removing the dependency between the batch samples responsible for better accuracy even on small batch sizes.

3.2.1.2 Attention Module

This attention block is helpful to find more exact detection results by utilizing a more fine-grained feature map. The attention block is the combination of channel attention and spatial attention. The channel attention module uses the generated feature from the Residual block as input and produces a more refined feature, while the spatial feature uses the conditional distribution of column features corresponding the spatial location that generates spatial feature to boost the performance of the backbone network. The motivation behind this approach is to provide better detection results with less computational cost. The global average pooling (GAP) is to identify the object of extent, while the global max pooling (GMP) tends to help for the position contains by the object feature map. The GMP is very suitable for the detection of small object and when the feature map scales are shrinking with the spatial dimension.

For a feature map $M\epsilon F^{W\times H\times C}$ that generated mid-layer of ResNet using as input, CSA-SS infers single dimension channel attention feature map $Ac\epsilon F^{1\times 1\times C}$ to generate feature map M' and a two-dimension spatial attention feature map $As\epsilon F^{W\times H\times 1}$ sequentially as depicted in Fig-3.1. The complete process of the attention mechanism can be calculated as:

$$M' = Ac(M)\bigotimes \text{ M} \tag{3.1}$$

$$M'' = \text{As}\bigotimes \text{M'} \tag{3.2}$$

Where $\bigotimes$ used for element-wise multiplication.

The values of channel attention are transmitted across the spatial dimension during multiplication, and vice versa. M represented the final refined feature map. Fig-3.3 illustrated the working process of the channel attention map. The description of these two attention modules is as follows.

Channel Attention Module The inter-channel relationship of features is used

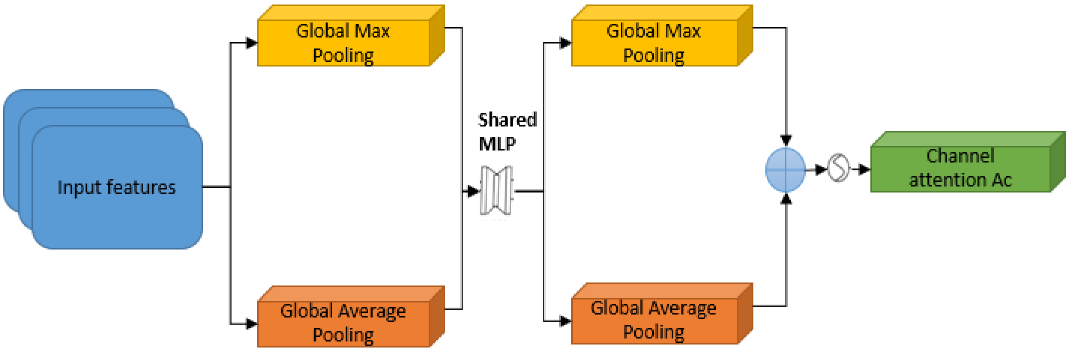

FIGURE 3.3: Channel attention sub-module

to extract semantic channel features of the attention map. Yet average pooling is commonly used to accumulate spatial information. Zout et al. suggested an average pooling to acquire knowledge about the target object effectively. Hu et al. also adopt this to compute the statistics of spatial information in the attention module. Sanghyun Woo et al. proposed that max-pooling gathered with average pooling gives a more refined feature map through channel-wise attention. Inspired by this method, the CSA-SS uses GAP and GMP to generate the channel attention feature map.

The global average pooling is more endogenous to the convolution structure through imposing correlations between object classes and feature maps. Thus the class confidence map can be easily accessed through the feature maps. Thus, another factor to using global average pooling is that there is no need for parameter optimization; thus, over-fitting is avoided at the attention layer. Global pooling layers are used to reduce the spatial dimensionality that deducts the computational overhead of the attention mechanism. GAP and MP provide two different spatial context descriptors to accumulate the feature map's spatial information. MGAP and MMP are the features generated through the GAP and MP layers, respectively. As shown in Fig-3.3. A shared network is MLP with a single hidden layer used to generate a channel attention map by using the spatial descriptor. The channel attention map is $A_C \epsilon F^{C\times1\times1}$ the activation size of the hidden layer is set to $F^{C/r\times1\times1}$. To get the output feature vector from shared MLP, apply concatenation on this. The channel attention calculated as

$$A_c(M) = \sigma(MLP(GAP(M) + MLP(MP(M))) \tag{3.3}$$

$$A_c(M) = \sigma(W_1(W_0(M_{GAP})) + W_1(W_0(M_{MP}))) \tag{3.4}$$

Where $W_0 \epsilon F^{C/r\times C}$, $W_1 \epsilon F^{C\times C/r}$ and σ is the sigmoid function. W_0 and W_1 are the MLP weights that are shared for both inputs and W_0 follow the ReLU activation function.

Spatial Attention Module For the input feature map $M \epsilon F^{C\times W\times H}$, CSA-SS aimed to generate the spatial feature $As \epsilon F_{W\times H}$. where C is the number of channels for the input image, W and H are the row and column of the input image. Let M $= m_{1,1}\ldots\ldots m_{w,h}$ where $m_{i,j}$ be a column feature belongs to a particular spatial location (i,j). Similarly, $A_s = a_{1,1}\ldots\ldots a_{m,n}$ where $a_{i,j}\epsilon F$ be the positional feature corresponding to $M_{i,j}$.

Formally the CSA-SS wants to predict the spatial features through the conditional probability p(As/M). Most conventional attention mechanisms [51] predict the attention values directly that can be considered as expected value (point value) under the formulation. The global attention mechanism predicts As from M directly from a fully connected network. However, global attention does not predict conditional distribution p(As/M) factorization. As the size of M increases, this becomes tractable because more parameters are used in the fully-connected layer. Another side of the local attention mechanism makes high independence assumption between the attention variable $a_{i,j}$. Particularly, spatial local attention assumes every attention variable $a_{i,j}$ free from other variables given for local spatial context $\delta(M_{i,j})$. From a previous studies viewpoint, the CSA-SS globally applied channel attention while the spatial attention applied locally that work can be calculated as

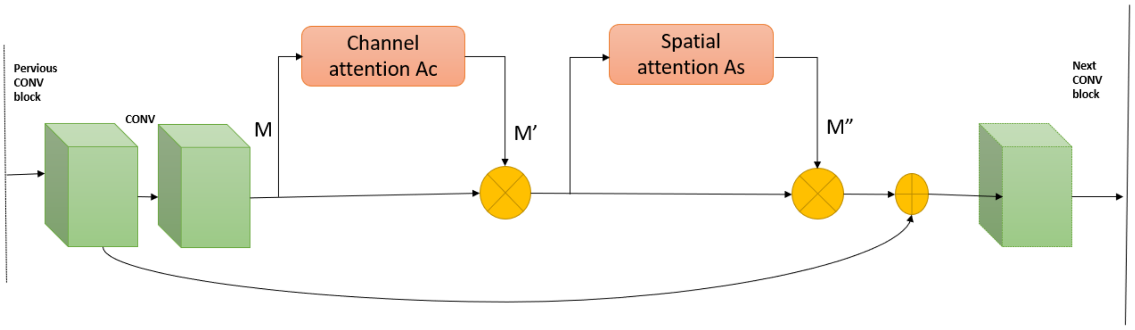

FIGURE 3.4: Association of attention module with ResNet network

$$p\left(\frac{A_s}{M}\right) = \prod_{i=1,j=1}^{i=W,j=H} p\left(a_{i,j}|\delta\left(M_{i,j}\right)\right) \tag{3.5}$$

Arrangement of sub-module with ResNet The CSA-SS combines these two attention sub-modules sequentially and the whole attention block considers as a new block that can be assembled with the backbone network easily. As illustrated in Fig-3.4 the attention block used as an input feature generated from the previous convolution block and generates a more refine feature that works as an input for the next convolutional block.

3.3 Result Analysis and Discussion

3.3.1 Experimental Setup

The test set is used for the evaluation in this experiment. The IOU threshold is 0.7 for cars and 0.5 for cyclists and pedestrians of the KITTI dataset. Apart from this the IOU threshold is 0.75 for all classes of the BDD dataset. The finalization of bounding boxes using NMS for every training set of KITTI and BDD. Object detection performance is measured based on mean Average Precision (mAP). The training is done with a randomly sampled image scale of 416 × 416. The model trained for 30k iterations with 8 batch sizes starting from 0.01 learning rate and

Algorithm 1: Algorithm for CSA-SS model

```
Input: Image Datasets for different object
Output: Identification of Classes with bounding box and a Confidence score for objects
Step 1: Start
 Step 2: Take images as inputs and passes them through the proposed model
 Step 3: Extract Feature Maps
   Step 3.1: CSA-SS extract the features with ResNet variants where each residual block contain 2 convolution block followed by FRN to obtain feature map $M \epsilon F^{W \times H \times C}$.
 Step 3.2: Apply attention block after each residual block on feature map $M \epsilon F^{W \times H \times C}$ as in Eq-3.1 & 3.2 and Fig-3.4.
 Step 3.2.1: channel attention Ac takes $M \epsilon F^{1 \times 1 \times C}$ as input and passes it through channel attention as in eq-3.3 &3.4.
 Step 3.2.2: spatial attention As takes $M \epsilon F^{W \times H \times 1}$ as input and pass it through spatial attention as in 3.5
       Step 3.2.3: concatenate these two outputs as in Fig-3.1
 Step 4: These feature maps pass through detection layers and regression layers to get the bounding boxes and class probability as in Fig-3.2
 Step 5: Selection of bounding boxes
    Step 5.1: Discard all bounding boxes having a probability less than or equal to a predefined threshold (0.5)
    Step 5.2: For the remaining boxes(NMS):
       Step 5.2.1: Pick the box with the high probability and take that as an output probability
       Step 5.2.2: Discard any other boxes which have IoU greater than the threshold with the output from the above step.
       Step 5.3: Repeat step 4.2 until all the boxes are either taken as the output prediction or discard
 Step 6: End
```

dividing by 10 at every 10k iterations. The batch size depends beyond the capacity of GPU memory due to using FRN. Thus, CSA-SS maintained its result even on small batch sizes. The proposed model has used early stopping to avoid over-fitting. The early stopping is a measure used to handle over-fitting during training, whose documents can be found in Keras library. The proposed model set the patient parameter for the early stopping to 5. The patient parameter is to be set in early stopping. If the value of the monitoring matric is minimized over the patient value, then the training will continue; otherwise, it will be halted. The KITTI dataset was generated with clear weather and daytime hour in Europe, while the BDD dataset was generated under diverse scenes like a residential area, city street and highways in North America.

These datasets have lots number of classes, and some are useful for AVs like cars, trains, trucks, pedestrians, bicycles, traffic lights, etc. Object detection performance is measured based on mean Average Precision (mAP). The training is done with a randomly sampled image scale 832×832, to reduce over-fitting, while inference is used for a single scale of 1024 pixels. The model used a few images per GPU as a mini-batch (16 on 8 GPU) and trained for 30k iterations starting from 0.01 learning rate and dividing by 10 at every 10k iterations. The batch size of the model is set to 2. The batch size depends on the capacity of GPU memory. The weight decay is 0.0001 and the momentum is set to 0.9. While the proposed technique is a single-stage detector, it provides better results than some single-stage detector models.

3.3.2 KITTI Dataset Result

The KITTI and BDD datasets have been taken for the experiment, which is common for AVs. There are 3 classes, car, cycle and pedestrian in the KITTI dataset that contain 7481 samples of images for training and 7518 sample image with a resolution 1242×375 for testing. The KITTI datasets don't have the GT for the test set so the train and validate datasets are created by randomly splitting of training dataset in half.

3.3.3 BDD Dataset Result

BDD dataset [35] is properly annotated, includes detection of road object, driveable area segmentation, instance segmentation and detection of lane markings annotations. The detection of road objects contains 10 categories person, car, traffic light, bike, traffic sign, train, truck, motor, rider and bus for 100,000 images for

Table 3.1: Comparison of performances on KITTI validation set

METHODS	**AVERAGE PRECISION**									**mAP%**	**FPS**	**INPUT SIZE**
	CAR			**PEDESTRIAN**			**CYCLIST**					
	EASY	MID	HARD	EASY	MID	HARD	EASY	MID	HARD			
MS-CNN [61]	92.54	90.49	79.23	87.46	81.34	72.49	90.13	87.59	81.11	84.71	8.13	1920×576
SINet [62]	99.11	90.59	79.77	88.09	79.22	70.3	94.41	86.61	80.68	85.42	23.98	1920×576
SSD [65]	88.37	87.84	79.15	50.33	48.87	44.97	48	52.51	51.52	67.6	28.93	512×512
RefineDet [196]	98.96	90.44	88.82	84.4	77.44	73.52	86.33	80.22	79.15	84.36	27.81	512×512
CFENet [63]	90.33	90.22	84.85	-	-	-	-	-	-	-	0.25	512×512
RFBNet [64]	87.41	88.35	83.41	65.85	61.3	57.71	74.46	72.73	69.75	73.44	39.2	512×512
YOLOV3 [37]	85.68	76.89	75.89	83.51	78.37	75.16	88.94	80.64	79.62	80.52	43.57	512×512
Gaussian YOLO V3 [66]	90.61	90.48	89.47	87.84	79.57	72.3	89.31	81.3	80.2	83.61	43.13	512×512
FCOS [68]	89.7	-	-	79.8	-	-	87	-	-	85.54	-	-
RetinaNet [67]	89.6	-	-	80.3	-	-	86.2	-	-	85.34	-	-
EFL-B [70]	89.8	-	-	82	-	-	86.6	-	-	86.1	-	-
CSA-SS Proposed	92.56	91.95	91.35	90.56	83.65	75.48	90.84	84.56	80.9	87.76	41.37	416×416

2D bounding boxes annotations. The ratio of splitting the testing, validation and training set is 2:1:7. The IoU^{TH} is set to 0.7 for evaluation on the testing set.

Table 3.2: Comparison of performances on a different backbone architectures

METHODS	**mAP%**	**FPS**	**INPUT_SIZE**
MS-CNN [61]	5.7	6	1920×576
SINet [62]	9	18.2	1920×576
SSD [65]	24.3	23.1	512×512
RefineDet [196]	17.4	22.3	512×512
CFENet [63]	19.1	21	512×512
RFBNet [64]	14.5	39	512×512
YOLOV3 [37]	26.6	42.9	512×512
GAUSSIAN YOLOV3 [66]	30.7	42.5	512×512
FCOS [68]	41.79	-	-
RetinaNet [67]	40.71	-	-
EFL-B [70]	42.68	-	-
CSA-SS Proposed	43.55	41.6	416×416

This section has the experimental details conducted on both BDD and KITTI datasets. The proposed model first compares the CSA-SS method with other existing state-of-the-art methods and the results are shown in Table-3.1 & Table-3.2. The CSA-SS outperforms all current state-of-the-art methods for weakening the trade-off between speed and accuracy of object detectors for AVs. In both Table 3.1 & Table- 3.2, the proposed CSA-SS achieves 1.66 and 1.13 with the closest to the competitors.

The CSA-SS provides better results by 1.66 mAP compared to EFL-B and the detection speed is 41.37 fps, which is enabled for real-time with 416 × 416 resolutions. The speed of CSA-SS is lesser than YOLOv3 and Gaussian YOLOv3, but the model predicts more accurate results and competes with the speed of the real-time system. The RFBNet is faster than all previous models except YOLOv3 and Gaussian YOLOv3, and the CSA-SS has improved speed by 2.17 fps. SINet has the maximum accuracy of 99.11, but the resolution of input tensors is very high compared to the CSA-SS model.

The BDD test set performance for CSA-SS and other state-of-the-art methods have represented in Table 3.3. CSA-SS produced better mAP by 1.13 with Gaussian YOLOv3 while the input size of CSA-SS is 416 × 416. The CSA-SS has a detection speed of 41.6, which is not faster than the Gaussian YOLOv3 but has the speed for a real-time system. Except for YOLOv3 and Gaussian YOLOv3, The CSA-SS is faster than the other state-of-the-methods as shown in Table 3.2. The CSA-SS performed excellently by 1.5 mAP while on lesser input size. The CSA-SS can remarkably improve the precision with little compensation in speed compared to baseline model YOLOv3 and Gaussian YOLOv3, and the CSA-SS is better than the previous techniques.

Due to the assemble attention block with the Conv layer, the computation complexity will increase, but the proposed CSA-SS model has negligible computations, as shown in the tables. However, it improves the model accuracy and beats the state-of-the-art approaches with a remarkable difference while compromising with a negligible frame rate. Furthermore, to further verify the effectiveness of the proposed CSA-SS, an ablation study is also conducted.

3.3.4 Ablation Study

To understand the effectiveness of the proposed model and evaluate the results with different settings and arrangements of the model on different datasets.

Generally, the CSA-SS model is trained with a 32 batch size but to see the effectuality of the model, it is set to 8 and the results are illustrated in Table 3.3. The performance of KITTI and BDD with different arrangements of network configuration. The best arrangements of the attention layer and FRN are in the last row of Table- 3.4. The model has analyzed that when the applied spatial attention directly on the feature map, then the performance of the model slightly improved but the channel attention gives the more prominent results as illustrated in Table 3.4.

TABLE 3.3: Comparison of performances on different backbone architectures

Methods	**mAP (KITTI)**	**mAP(BDD)**
ResNet101	77.9	39.8
ResNet101+Spatial	78.1	40.4
ResNet101+Channel	80.5	40.7
ResNet101+Channel+Spatial	82.4	41.2
ResNet101+Channel+Spatial+BN	83.8	42
ResNet101+Channel+Spatial+FRN	86.76	43.55

The total number of learnable parameters of the CSA-SS is 44.9 million and the CSA-SS achieved 90.76 top-1 Accuracy with 7.35 Flops. Table- 3.4 compares different attention-based state-of-the-art approaches concerning the number of parameters and accuracy. It can be observed from Table-3.3 that the proposed CSA-SS is highest in the list next to ResNet-101.

As shown in Fig-3.5 the comparison between different attention mechanisms corresponds with the number of parameters used by the model and top-1 accuracy of the different models on the COCO dataset. The model is trained with the COCO dataset to compare with attention-based state-of-the-art methods and get effective results. The proposed attention mechanism has shown the best performance on less number of parameters for accuracy. All selected learnable parameters of the proposed CSA-SS and compared its top-1 accuracy with other attention-based state-of-the-art approaches.

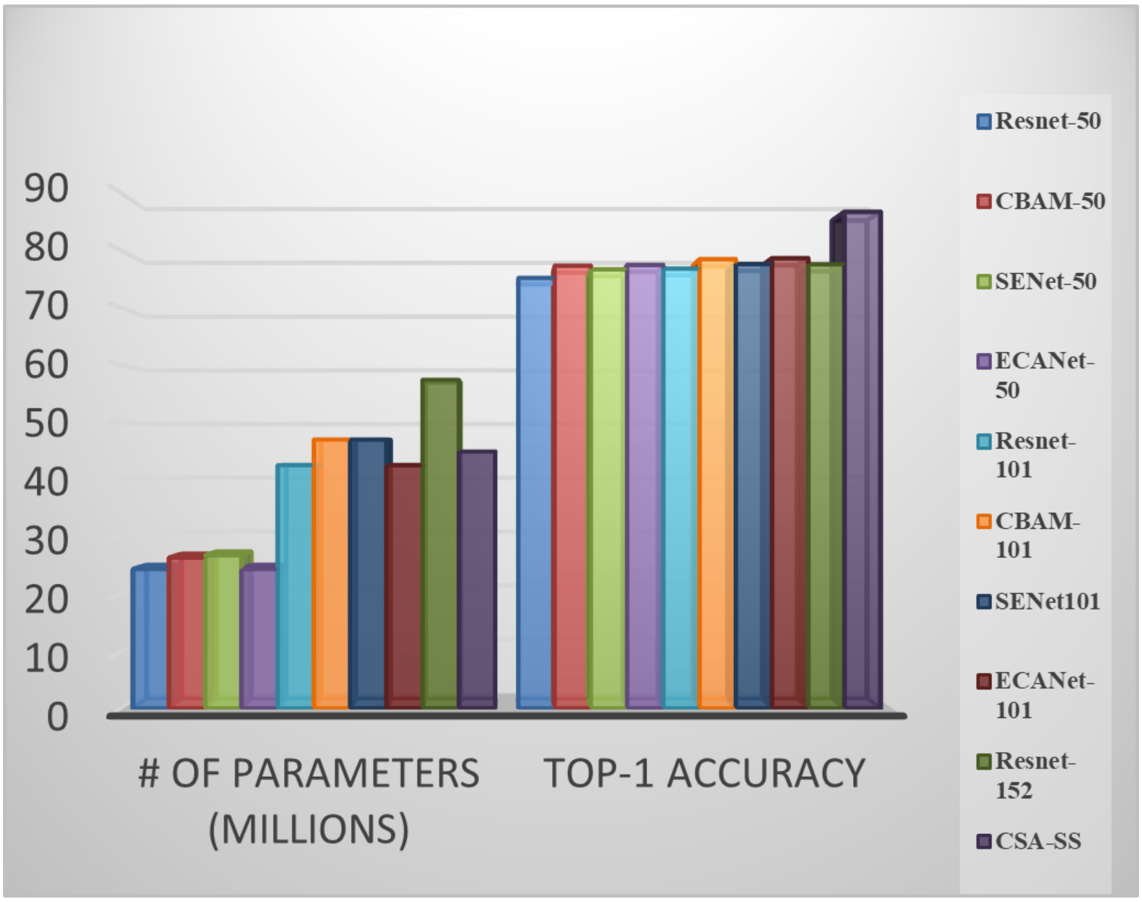

FIGURE 3.5: Comparison of different attention mechanism efficiency

TABLE 3.4: Comparison of different attention mechanism efficiency in terms of number of parameters, floating point operations per second (FLOPs) and Top-1 accuracy

Model	**# of parameters (Millions)**	**Top-1 Accuracy**	**Flops**
Resnet-50	24.37	75.2	3.86G
CBAM-50	26.37	77.34	3.87G
SENet-50	26.77	76.71	3.87G
ECANet-50	24.37	77.48	3.86G
CSA-SS	23.99	79.58	3.86G
Resnet-101	42.49	76.83	7.34G
CBAM-101	47.01	78.49	7.35G
SENet101	47.01	77.62	7.35G
ECANet-101	42.49	78.65	7.35G
CSA-SS	44.9	90.76	7.35G

3.4 Conclusion

An accurate and fast object detector is a crucial task for AVs. Various methods conducted camera-based AVs but did not maintain the trade-off between speed and accuracy. The CSA-SS has a context for the backbone network for object detection. It depicts the importance of having an attention block in a network using attention modules. This helps to get more effective and efficient features and training with fewer batch sizes without compromising the result. The experiment results on two datasets, KITTI and BDD, demonstrate that the CSA-SS framework has achieved state-of-the-art performance. An apple-to-apple comparison between the baseline model and CSA-SS has improved mAP by 1.66 and 1.13 for the KITTI and BDD, respectively. Consequently, the CSA-SS can effectively improve object detection for AVs and is expected to contribute to the general use in autonomous driving applications. A compact model can be designed to provide more accurate results with fewer parameters in the future.