International Journal of Computing and Digital Systems
2025, VOL. 17, NO. 1, 1–11

http://dx.doi.org/10.12785/ijcds/1570983253

Multi-Sensor Data Fusion using FPN-ResNET

Vinodh S[1] and Ramakanth P[2]

[1,2]*Department of Computer Science, R V College of Engineering, Bengaluru, India*

Received 12 March 2024, Revised 15 June 2024, Accepted 3 October 2024

Abstract: Multi-sensor data fusion is ubiquitous; therefore, the associated research is significant. There are several instances in the day-to-day activities where data fusion can be observed. The present generation autonomous driving system requires a thorough understanding followed by a voluminous dataset for training the model. The earlier efforts have utilized the point-level fusion technique, thereby supplementing the LiDAR point cloud with the camera features. In experimental data, imagery and proximity sensors are paramount for the model's performance. The sole purpose of preserving the semantic density of the imagery is compromised in point-level technique, rendering the technique ineffective. The present work attempts to enhance the conventional point-level fusion techniques by allocating prime importance to semantic density without increasing the computational time. This is facilitated by performance optimization, which identifies the hindrances and enhances the transformation of the view through bird's-eye-view pooling. ResNET-FPN is introduced to down-sample the images without affecting the semantic density; the latency is shortened by $\approx 68\%$. On the other hand, EKF is used to fuse the sensor data and evaluate the noise-covariance by compensating for the quadratic effects of the data. The proposed model is compared with the existing models based on their performance in each background class. The IoU of the existing models is compared with the proposed model, and it is observed to outperform the BEVFusion model by $\approx 3.1\%$. The detection precision is found to be 0.9684, and the detection recall is 0.9436, while the mAP is evaluated to be 74.3%, which is $\approx 5.6\%$ better than BEVFusion.

Keywords: Feature Pyramid Network(FPN), LiDAR, Multi-sensor data, Residual-Network(ResNET), Sensor fusion

1. Introduction

A wide variety of sensors are used in autonomous driving systems, rendering them complex. Various sensors offer supplementary signals to enhance the overall data collection. The inevitable usage of sensors with different modalities demands the usage of the Multi-Sensor Data Fusion(MSDF) for accurate object detection. Camera data is rich in semantic information in the perspective space; Light Detection and Ranging(LiDAR) based sensors supply spatial information in a three-dimensional space, and radar estimates the velocity at any particular instant. The present work considers the fusion of three-dimensional LiDAR sensor data with two-dimensional camera data. Mapping the semantic information from the camera with the spatial information from the LiDAR for accurate object detection forms a crucial aspect for autonomous driving[1].

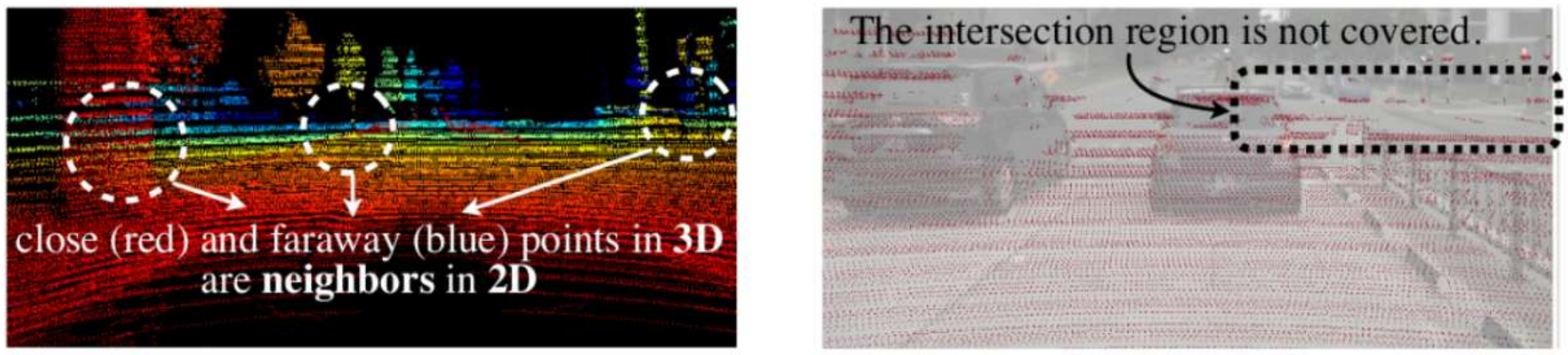
There have been several efforts to develop reliable three-dimensional object detection systems for autonomous driving. Regarding information depth, laser-based sensors excel, but cameras can capture semantic data down to a deeper level. Therefore, a fusion of camera and LiDAR-based sensors complement each other, permitting the development of a formidable three-dimensional detection system for a safe and exceptional autonomous driving experience.

However, MSDF has associated challenges due to the difference in modalities generated by the data of each sensor. To achieve a multi-modal and multi-task fusion, there is a need for a unified representation of the data from different sensors. In earlier efforts, two-dimensional perception has been achieved by projecting spatial LiDAR data onto semantic camera data. The method is less successful in detecting objects in three-dimensional space because of the geometric distortion that occurs when the LiDAR data is projected onto the camera(Figure.1a).

Some of the recent efforts in sensor fusion aim at enhancing the LiDAR point cloud data with CNN features[2], semantic labels[3], [4] and two-dimensional image-based virtual points[5]. Though there is a commendable detection performance on large-scale benchmarks, the point-level-based fusion is less impressive on tasks of a semantic nature such as BEV-Segmentation[6], [7], [8], [9], which can be attributed to the semantically-lossy behavior of the projection of camera to LiDAR(Figure.1b). Further, the differences in density are more pronounced for sparser LiDAR data.

The present work proposes a fusion of multi-modal features by maintaining geometric structure and semantic density, which is expected to enhance 3D perception. The Fea-

E-mail address: vinodh2302@gmail.com, ramakanthkp@rvce.edu.in



(a) *Geometric-lossy* (b) *Semantic-lossy*

Figure 1. Projection losses[1]

ture Pyramid Network(FPN) encoder, with Residual Network(ResNET) as the backbone, is used for image classification and object detection due to the inherent characteristic of deep convolutional layers permitting faster visual data processing. ResNET is more accurate in real-time applications, which justifies its usage in the present study. Since the view transformation consumes nearly 80% of the model's runtime, an Extended Kalman Filter(EKF) kernel is used for pre-computation and interval reduction, ensuring process speed-up by $\approx 65\%$. Furthermore, due to the non-linearity in the measured data, EKF is used to temporally estimate the state's mean and covariance on-the-run, iteratively. Lastly, the encoder measurements are prone to uncertainties in ego-vehicle localization due to the accumulation of errors, which is significant when the measurements are made from larger distances[10]. Therefore, EKF is used to fuse the sensor data by statistically minimizing the error during the estimation of the ego vehicle state vector.

The present approach also disproves the conventional wisdom that point-level fusion provides the best multi-sensor fusion solution. The present model is simple in construction and operation, rendering it more robust and reliable. Therefore, the work paves the way for future sensor-fusion developments by building upon the platform reported in the document.

2. Related Work

Over a decade, immense efforts have been put forth to develop a reliable and robust method for the fusion of sensors with different modalities. However, there is a great scope for developing more sophisticated and accurate models, as the existing models are identified with few challenges in overcoming the projection accuracy through reduction in geometric and semantic losses. The earlier works to achieve three-dimensional perception based on LiDAR-only data include the single-stage 3D Object detectors[11], [8], [12], [13], [14], which provided the platform for the evolution of many robust and sophisticated models. The model is enhanced using PointNets[15] and SparseConvNet[16] for extracting the flattened point-cloud features. Nevertheless, the restriction offered through the bounding box in the earlier models is overcome by introducing the anchorless models[17], [18], [19], [20]. Further investigations have led to the development of two-stage models through the amalgamation of the Region-based Convolutional Neural Network(R-CNN) architecture with the existing one-stage-based object detection model[21], [22], [23], [24], [25], [26]. The most crucial task for the offline construction of High-Definition(HD) maps is the three-dimensional segmentation of the semantic features. The models[16], [27], [28], [29], [30] developed to address the seminal task, analogous to U-Net, are note-worthy.

To replace the expensive LiDAR sensors for 3D object detection, commendable efforts are made to achieve three-dimensional perception based on Camera-only data. The FCOS3D[31] model utilizes three-dimensional regression branches suitably coupled with the image detectors[32], which is later enhanced to achieve greater depth in detection[33], [34]. Irrespective of perspective view-based object detection, models that learn from the object queries in the three-dimensional space coupled with the Deformable Transformer(DETR)[35] detection model, *viz.* DETR3D[36], PETR[37] and Graph-DETR3D[38], are also developed. The view transformer-based camera-only three-dimensional perception models explicitly transform camera data to perspective bird's eye view[6], [39], [40], [7]. The state-of-art models such as BEVDet[41] and M^{2}BEV[42], utilize Lift-Splat- Shoot(LSS)[7] and Orthographic Feature Transform(OFT)[40] for three-dimensional object detection. Also, the three-dimensional object detection models through time-dependent cues using multiple cameras *viz.* BEVDet4D[43], BEVFormer[44] and PETRv2[37] are some salient developments in single-frame methods. However, the models such as BEVFormer[44], CVT[9], and EGO3RT[45] also perform exceptionally well through multi-head attention for view transformation.

Lastly, efforts are put forth to study the models for multi-task learning. Simultaneous detection of objects and instant segmentation form the key aspects of multi-task learning[46], [47]. Further, the simultaneous detection and segmentation is extended to human-object interaction[43], [48], [49], [50]. The models for detection of object and instance segmentation, simultaneously, *viz* M^{2}BEV[42], BEVFormer[44] and BEVerse[51], are not developed by considering multi-sensor data fusion. Also, the activities are performed simultaneously, which demands longer computational time and higher hardware requirements, thereby significantly increasing the computational cost.

On the other hand, the MMF model[52] though performs detection and segmentation simultaneously, it is object-centric, which cannot be extended to BEV Segmentation.
The most recent attempts aim to significantly improve the detection performance by fusing sensors of different modalities. The methods can be categorized into *proposal-level* and *point-level*. The proposal-level methods are *object-centric* and therefore do not support map segmentation effectively, whereas point-level techniques are both *object-centric* and *geometric-centric*. Some of the exceptional contributions towards *proposal-level* techniques include IS-FUSION[**?**], SparseFusion[**?**], ObjectFusion[**?**] MV3D[53], F-PointNet[54], F-ConvNet[49], CenterFusion[55]. FUTR3D[56] and TransFusion[57], while point-level techniques include PointPainting[2], PointAugmenting[3], MVP[5], FusionPainting[58], AutoAlign[59], DeepContinuousFusion[52], Deep Fusion[4], and FocalSparseCNN[60]. Not all techniques can be incorporated to process the camera and LiDAR data. LiDAR data processing can be carried out very effectively through input-level decoration models *viz.* PointPainting[2], PointAugmenting[3], MVP[5], FusionPainting[58], AutoAlign[59], and FocalSparseCNN[60], while camera images require feature-level decoration *viz.* DeepContinuousFusion[52], Deep Fusion[4].
Contrary to the aforementioned models, the proposed model has following points, which render it unique: label=()

1) It is a point-level fusion approach that performs multi-sensor fusion in a shared space by providing weightage to both semantic and geometric information equally, both in the foreground and background
2) Faster computation is ensured alongside object detection, facilitated through a Residual Network(ResNET) based Feature Pyramid Network(FPN) model.
3) EKF is incorporated to handle the non-linearity of the camera and LiDAR data more effectively.
4) EKF fuses the camera and LiDAR data from ResNET-FPN by minimizing the accumulation error generated due to the usage of ResNET-FPN.
5) The framework is more generic with multi-sensor(3 Cameras(C) and 3 LiDAR(L)) perception and multitasking.

3. Method

The three crucial activities that have direct implications on the model performance are listed in the section 3-1, section 3-2, and section 3-3.

1) Unified Representation

Distinct qualities may be present in various viewpoints. LiDAR and radar features, for example, are usually in the three-dimensional bird's-eye view, whereas camera features are in the perspective view. Every camera function, such as front, back, left, and right, has a unique viewing angle. Due to this perspective mismatch, feature fusion becomes challenging because the same element may correspond to entirely different spatial locations in distinct feature tensors (naïve element-wise feature fusion will not operate in this scenario). Thus, it is imperative to identify a shared representation that is easily convertible to it without sacrificing information and appropriate for various purposes[1].

2) To Camera

One option is to project the LiDAR point cloud onto the camera plane and display the $2.5D$ sparse depth driven by RGB-D data. This conversion is geometrically lossy. In the 3D space, two neighbors on the depth map may be very far apart. For activities like 3D object detection that rely on the

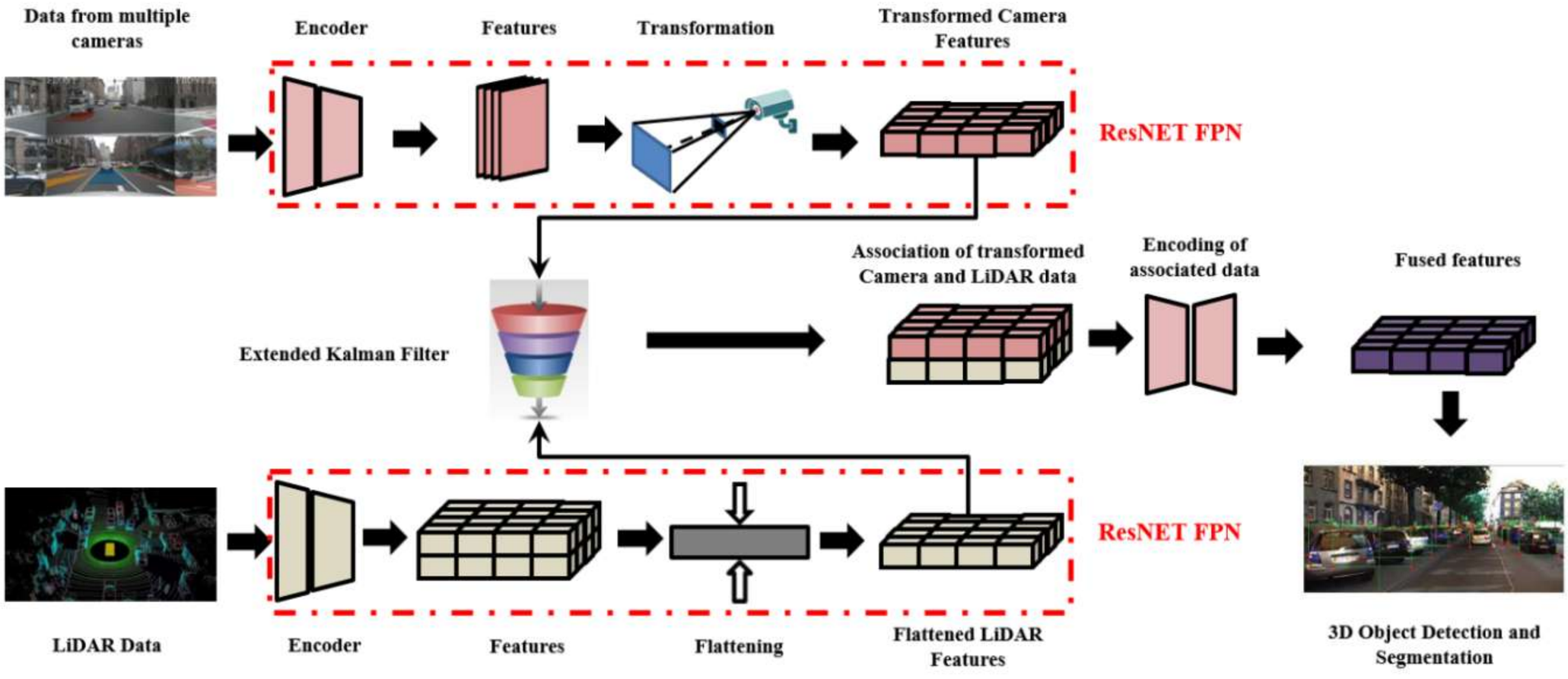


Figure 2. Framework

geometry of the item or scene, this reduces the effectiveness of the camera view.

3) To LiDAR

The majority of cutting-edge sensor fusion techniques [2], [5], [4] embellish LiDAR points with the matching camera features (e.g., virtual points, CNN features, or semantic labels). But this projection from the camera to LiDAR is semantically lossy. Because of the stark differences in densities between LiDAR and camera features (for a 32-channel LiDAR scanner), $< 5\%$ of camera features match a LiDAR point. On semantic-oriented tasks (such as BEV map segmentation), the model's performance is significantly affected by giving up the semantic density of camera features. More modern fusion techniques in the latent space, including object query, have comparable demerits[57], [19].

4) To BEV

The lossy identified and explained through Figure.1a and Figure.1b are considered during the transformation. The projection of LiDAR data to BEV evens out the sparse features in the height dimension, thereby eliminating the aspect of geometric lossy. On the contrary, the transformation of camera images to BEV is non-trivial due to its inherent depth.

The depth distribution of the pixels of the camera images is predicted using LSS[7] and BEVDet[41], [61]. The features are re-scaled upon scattering each feature's pixels to $\mathcal{D}$ discrete points along the ray of the camera. A cloud of the feature points is generated with a size of $\mathcal{NHWD}$, where $\mathcal{N}$ is the number of the cameras(in the present study, it is 3 numbers), while, $\mathcal{H}$ and $\mathcal{W}$ are the height and the width of the image, respectively. The grid size considered in the cartesian coordinate system is $0.35m \times 0.35m$, which is evened out in the z-direction.

The transformation of camera-to-BEV consumed a computational time of $\approx 452ms$ with a Quadro $P6000$ Graphics processing. This can be attributed to the large number of grid points generated per frame of the camera feature. The LiDAR features are, therefore, less dense and computationally inexpensive. Nevertheless, curtailing the computational time for the camera features demands a pre-computation and reduction in the interval considered earlier.

The *Pre-computation* involves associating the camera features to BEV grid points. From the calibration of the camera, the intrinsic and extrinsic stay the same, permitting locating coordinates of the feature cloud of the camera. Pre-computation is performed by segregating the grid points based on indices, and the ranks of the points are recorded. This permits the reordering of the feature points based on the pre-computed ranks. This task alone reduces grid-association latency by $\approx 24\%$, with the remaining latency reduction achieved through interval reduction.

The *Interval Reduction* aggregates the grid-points generated during the pre-computation through symmetric functions *viz. mean, maximum, and summation*, within the BEV grid. The assigned Graphics Processing Unit(GPU) thread accelerates the feature aggregation and eliminates the dependency between the outputs. This enables the reduction in latency by $\approx 44\%$

A. Multi-tasking

Practically, most 3D perception activity is carried out under detection and segmentation. The object center is evaluated based on the size, velocity, and rotation of the earlier 3D detection articles[57], [17], [5]. On the other hand, the segmentation is carried out by classifying the features and associating the binary segments with each. The training of the segmentation head is carried out through CVT[9], with the focal loss being treated using Lin *et.al* model[62].

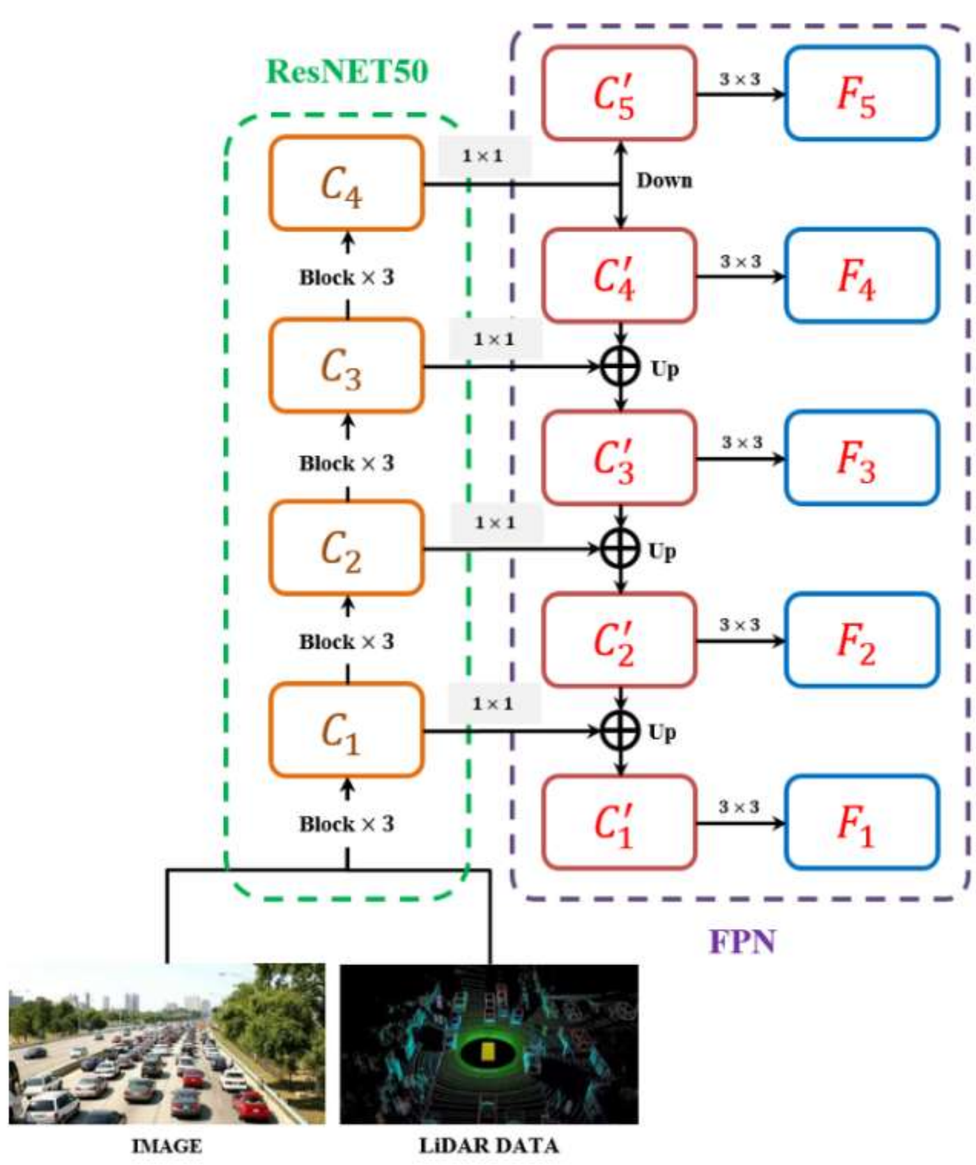
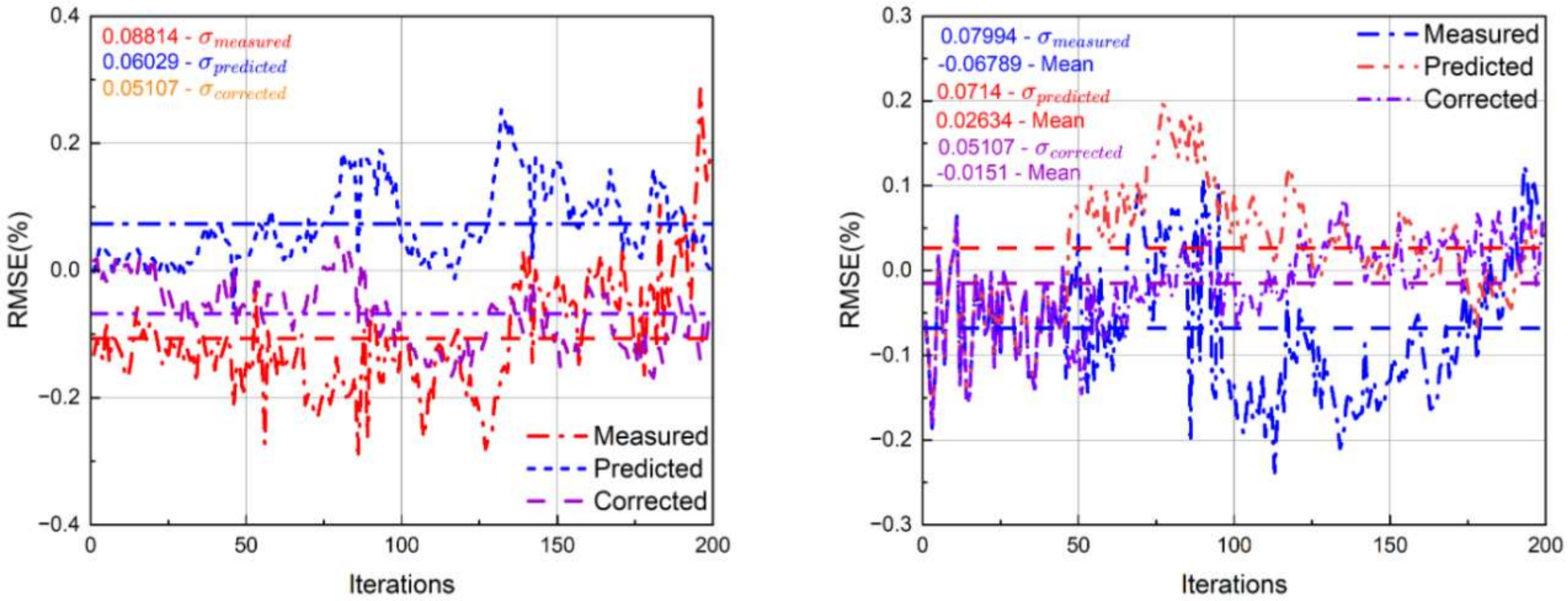


Figure 3. ResNET-FPN Architecture

4. Experiments

A. Model

ResNET50(Figure.3) is considered the backbone of the FPN due to its superior performance in extracting features with fewer parameters. The bottom-up feature extraction is handled by the CNN layers $\{C_1, C_2, C_3, C_4\}$, with the dimensionality of the output of each CNN layer being 64, 256, 512, and 1024, respectively. The intermediate CNN layers $\{C_1', C_2', C_3', C_4', C_5'\}$, obtained by 1×1 convolution and $2\times$ downsampling, eliminate the effects of aliasing between convolutional layers and transfer 3×3 convolutional kernel. Lastly, the FPN layers $\{F_1, F_2, F_3, F_4, F_5\}$, which are obtained by top-down operation and 1×1 convolution, are responsible for multi-scale information fusion, emanating from different convolutional layers. It generates the feature map by fusing the multi-scale camera images, downsampled to 256×704. The settings for FPN are made as



(a) Detected Signal

(b) Tracking Signal

Figure 4. Treatment of detection and tracking signals

per[62], with the dimensionality of each FPN layer being set to 256. Similarly, the LiDAR data, handled by ResNET-FPN, is down-sampled to 0.075 and 0.1 for detection and segmentation, respectively.

B. Framework

The framework for the present study is depicted in Figure.2, where 3 cameras and 3 LiDAR sensor data are input to the respective encoders. The convolutional encoders follow the ResNET-FPN architecture depicted in Figure.3. The features of the camera images are extracted and transformed, which forms the key aspect to achieve higher accuracy. LSS[7] and BEVDet[61] models are followed to achieve the transformation of camera images. The transformed images are filtered through an Extended Kalman Filter(EKF) and fused. EKF is a non-linear time-invariant state model represented by the Equation.1[63].

$$\psi(i+1) = \phi(\psi(I)) + \chi(i)\zeta(i) = \gamma(\psi(i)) + \nu(i) \tag{1}$$

where $\chi(i)$ and $\nu(i)$ are non-correlated processes with zero-mean, while ϕ and γ are the operators. The state $\psi(i+1)$ is predicted based on the $\zeta(i)$ measurement. EKF is effective in handling error back-propagation[64], with considerably shorter time for training in comparison with second-order gradient models such as the Gauss-Newton[10] and Least Mean Squares(LMS)[65] algorithms. Therefore, EKF transforms inherently non-linear LiDAR and Camera data, which generates a system matrix and evaluates noise-covariance by compensating for the quadratic effects of the data.
The filtered data is mapped by estimating the error through the update-and-predict of the EKF input state. The root mean squared error(RMSE) estimated for a single target is ≈ 0.32 during the present study. EKF predicts the model's state ($\psi(i+1)$), while Mahalanobis distance(MD) matches the states of multiple sensors[66], and the EKF updates the state based on the error generated by the MD calculation.

$$D^2(x) = (x - \overline{X_i}) \times M_i (x - \overline{X_i}) \tag{2}$$

Equation.2 is used to evaluate the MD, where x is the observations to be made, X_i is the calibration data set for the corresponding i^{th} sensor, while $\overline{X_i}$ is mean and M_i is the RMSE of the i^{th} sensor calibration data. Figure.4a and Figure.4b depict the error minimization, where corrected data is the output from the EKF.

C. Dataset

The present model is trained using Waymo Dataset[67], which has 798 sequences for training and 202 sequences for validation of vehicles and pedestrians. 64 lanes of LiDAR, or $180,000$ points per 0.1 seconds, make up the point clouds. The dataset for the present study comprises $20,156$ annotated samples of three monocular Camera RGB images capturing a 180° field-of-view and three 32-beam LiDAR data. The camera images are well-nourished with semantic information, while the LiDAR data precisely provide the spatial information.

D. Training

The model's training is carried out end-to-end to avoid camera-encoder freezing, as observed in earlier models[2], [3], [57]. The weight decay is ≈ 0.001, and optimization is achieved through AdamW[68] model.

E. Metrics

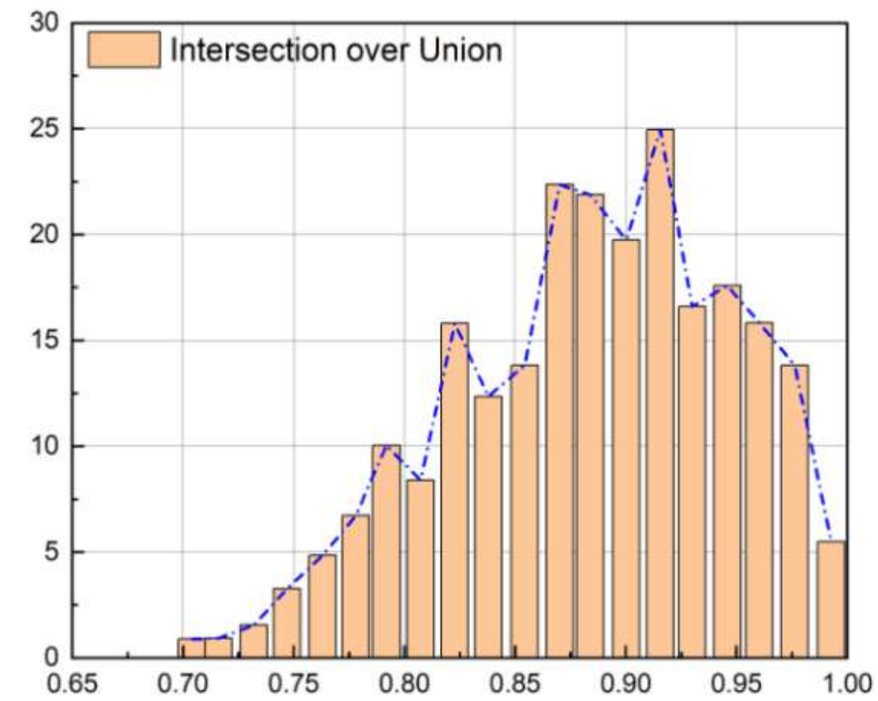
The evaluation of the model is made based on the following parameters discussed in section 4-E1 and section 4-E2.

1) Intersection over Union(IoU)

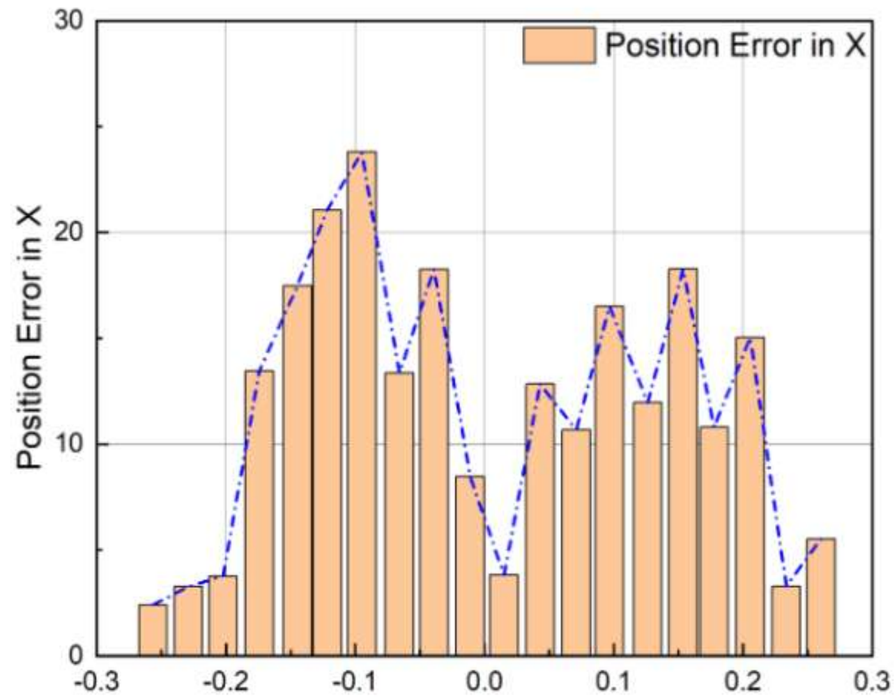
The accuracy with which the data is predicted can be obtained through Intersection over Union*(IoU)*, which

(a) IoU = 0.954(Excellent) (b) IoU = 0.739(Good) (c) IoU = 0.453(Poor)

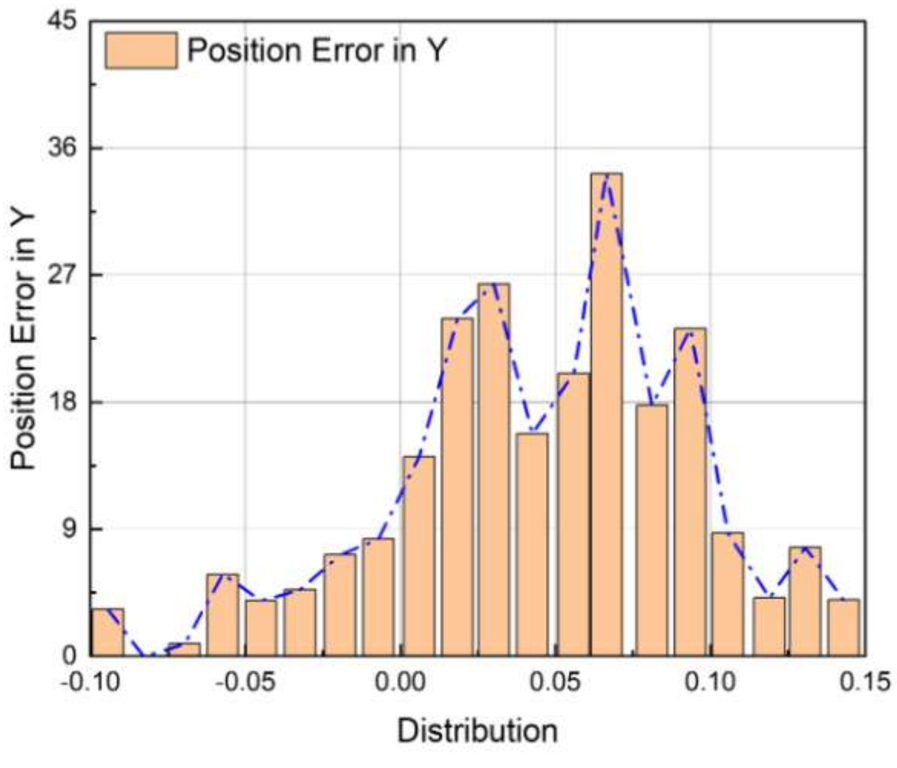
Figure 5. Intersection over Union



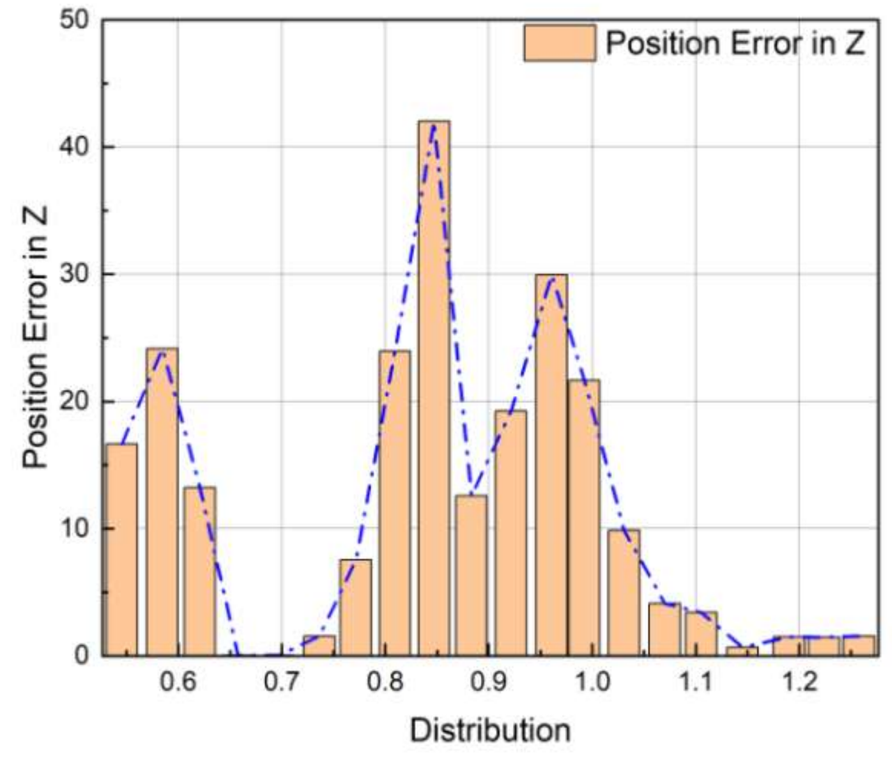
(a) Intersection Over Union



(b) X-Error in Signal



(c) Y-Error in Signal

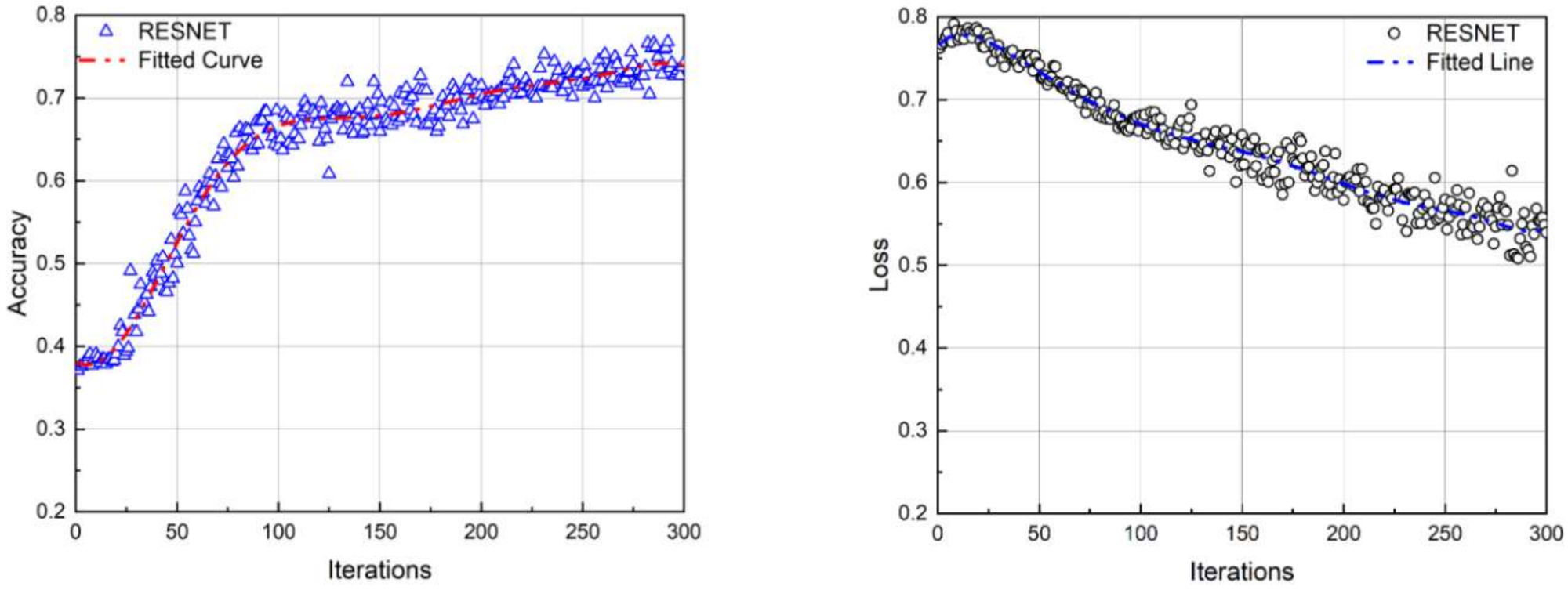


(d) Z-Error in Signal

Figure 6. Evaluation of Object Detection Performance

is defined as the percentage of overlap between the actual value(ground-truth) and the predicted value(Figure.5), mathematically represented by Equation.3.

$$IoU = \frac{|A \cap B|}{|A| \cup |B|} \tag{3}$$



(a) Accuracy

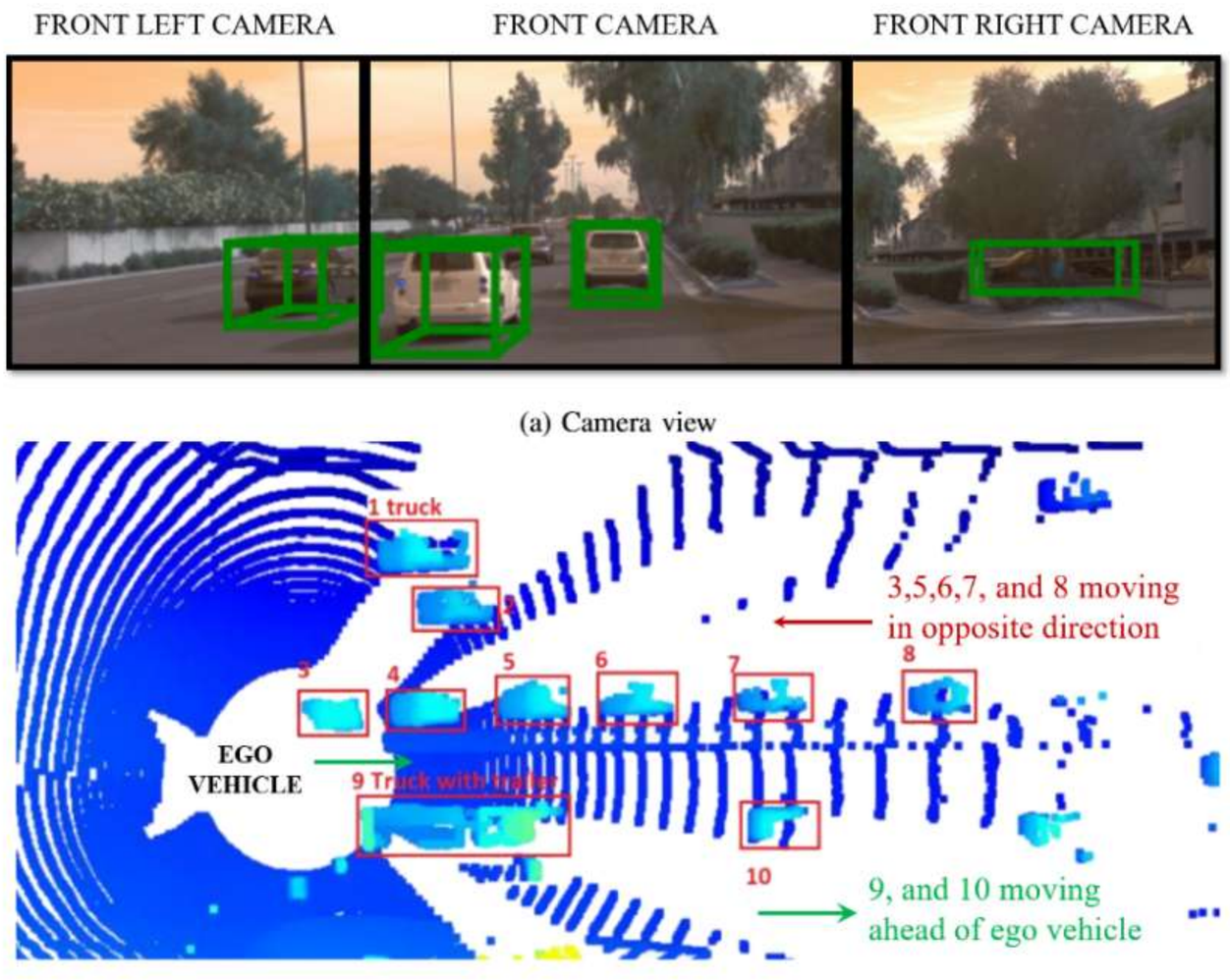
(b) Loss

Figure 7. Performance of FPN-ResNET based MSDF model

where A is the ground truth and B is the predicted value. Evaluation and comparison of the present model is facilitated by reporting the IoU on the defined background classes *viz.* Drivable space, Pedestrian crossing, walkway, stop-line, car parking, and lane divider. The mean IoU is calculated, forming the basis for comparing other models. Table.I lists the performance of the existing models, which are used for assessing the ResNET-FPN model(present study) based on the identified background classes. It can be observed that the present model outperforms with a mean IoU of $\approx 3.1\%$ greater than the BEVFusion model. The ResNET-FPN model's performance is promising for 3 Cameras and 3 LiDAR sensor data. However, it can be further enhanced by considering a larger dataset with 6 Cameras and 6 LiDAR sensors.

2) Mean Average Precision(mAP)

The mAP is calculated based on the Average Precision(AP) obtained from the area under the precision-recall curve. The AP is averaged for N samples as indicated by



(a) Camera view

(b) Bird's Eye View

Figure 8. Qualitative results of Camera and LiDAR data indicating object recognition

Models	Modality	Drive	Crossing	Walkway	Stop Line	Car Parking	Divider	Mean
OFT[40]	C	74	35.3	45.9	27.5	35.9	33.9	42.1
LSS[7]	C	75.4	38.8	46.3	30.3	39.1	36.5	44.4
CVT[9]	C	74.3	36.8	39.9	25.8	35	29.4	40.2
BEVFusion[1]	C	81.7	54.8	58.4	47.4	50.7	46.4	56.6
PointPillars[8]	L	72	43.1	53.1	29.7	27.7	37.5	43.8
CenterPoint[17]	L	75.6	48.4	57.5	36.5	31.7	41.9	48.6
PointPainting[2]	C+L	75.9	48.5	57.1	36.9	34.5	41.9	49.1
MVP[5]	C+L	76.1	48.7	57	36.9	33	42.2	49
BEVFusion[1]	C+L	85.5	60.5	67.6	52	57	53.7	62.7
ResNET-FPN	**C+L**	**88.4**	**65.2**	**67.1**	**51.7**	**61.1**	**54.2**	**64.6**

TABLE I. Comparison with the existing models based on IoU. *Camera(C); LiDAR(L)*

Equation.4 to obtain mAP.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{4}$$

5. Results and Discussion

Based on the methodology defined in the earlier section, LiDAR signals are processed for detection and tracking. The predicted signal generated through the EKF is compared with the measured signal, which is corrected based on the root-mean-squared error(RMSE) represented in percentage. The standard deviation between measured and predicted data for the detected signals is $\approx$ 6%, which is corrected to achieve a standard deviation between measured and corrected signal as $\approx$ 5%(Figure.4a). Also, for tracking signal, the RMSE for predicted values $\approx$ 7%, which is corrected to achieve an error of $\approx$ 5.1% (Figure.4b).
Figure.6a represents the IoU for the model, which demonstrates a good performance with a minimum score of 0.7, while most of the distribution is within the range of $84 - 99\%$. The error plots Figure.6b, and Figure.6c show symmetricity about zero with the maximum distribution close to zero, whereas in the case of error plot in the Z-direction, the range is between 0.5 to 1, with maximum peaks between to $0.8 - 1$. The mean position errors in X, Y, and Z directions are calculated to be $0.0041, 0.0363$, and 0.7243, respectively. The detection performance can be evaluated through accuracy and loss data plots as depicted in Figure.7a and Figure.7b, respectively. The model's accuracy is $\approx$ 72% while the loss is calculated to be $\approx$ 55%. The detection precision and recall are ≈ 0.9684 and ≈ 0.9436, respectively, and mAP is 74.3.
A qualitative result of the object detection is demonstrated in Figure.8b, while the quantitative evaluation is performed through accuracy and loss data plots as depicted in Figure.7a and Figure.7b, respectively. It is observed from Figure.7a that the training accuracy of the model reaches $\approx$ 72% at the end of training without significant variation thereafter. Similarly, the training loss curve depicted in Figure.7b demonstrates a value $\approx$ 51% and indicates an insignificant change at the end of the model's training. The training accuracy and loss curve also demonstrate that further training of the model fetches a meager improvement in the model's performance for the given dataset. The detection precision and recall are ≈ 0.9546 and ≈ 0.9344, respectively, and mAP is 71.2.
The results are compared with the existing models, as demonstrated in Table.II. It can be observed that the model's performance is better than the BEVFusion[1] by $\approx$ 1.4%. Usage of ResNET-FPN reduces the latency by $\approx$ 68%, with a corresponding increase in uncertainty due to error accumulation[10]. The effect of error accumulation on the model accuracy is mitigated by using EKF for the data fusion, which reduces the standard deviation by correcting the predicted signal based on the measured data[63]. Further, unlike BEVFusion, the non-linearity of camera and LiDAR data is handled by the introduction of EKF. A better comparison is possible between the present model and BEVFusion[1] by considering 6 cameras and 1 LiDAR data, which is left for future work. The results are compared with the existing models, as demonstrated in Table.II. It can be observed that the model's performance is better than the BEVFusion, which is $\approx$ 5%. However, for the present study, 3 cameras and 3 LiDAR data are used, unlike 6 Cameras and 1 LiDAR data in the case of BEVFusion model[1].

6. Conclusion

It is evident from the earlier discussion that many MSDF models have demonstrated greater accuracy in the recent past. However, challenges persist that can be attributed to the environmental or operating conditions that induce errors in the data as discussed in section 1. A formidable correction has to be incorporated, which otherwise can affect the model's accuracy. The model presented in this paper attempts to fuse the multi-modal data from different sensors to enhance object detection for future autonomous driving purposes.
The model demonstrates performance fairly well placed against the existing models, particularly BEVFusion. The BEVFusion model is developed by considering 6 cameras and 1 LiDAR data, while the present model considers 3 cameras and 3 LiDAR data for fusion. Hence, there are differences in the modalities handled in the course of development of the model. However, the present model is observed to have an accuracy of 72% with detection precision and recall of ≈ 0.9684 and ≈ 0.9436, respectively. The mean Average Precision is 74.3%, which is better than

Models	Modality	mAP
BEVDet[41]	C	42.2
M^2BEV[42]	C	42.9
BEVFormer[44]	C	44.5
BEVDet4D[61]	C	45.1
PointPillars[8]	L	-
SECOND[69]	L	52.8
CenterPoint[17]	L	60.3
PointPainting[2]	C+L	–
PointAugmenting[3]	C+L	66.8
MVP[5]	C+L	66.4
FusionPainting[58]	C+L	68.1
AutoAlign[59]	C+L	–
FUTR3D[56]	C+L	–
TransFusion[57]	C+L	68.9
BEVFusion[1]	C+L	70.2
FPN-ResNET(present work)	C+L	**74.3**

TABLE II. Comparison with the existing models. *Camera(C); LiDAR(L)*

BEVFusion by $\approx 5.6\%$.

Though the present model outperforms BEVFusion and other point-level methods, there is still great scope for developing multi-modal 3D object detection models with inherent challenges associated with accurate depth estimation. The model can be improved by utilizing ground truth to supervise the view-transformer[70], [71] that can be considered for future developments. Also, the present model has considered 180° FoV, and there is a scope for improvement by introducing 360° FoV with data from both front and rear cameras and LiDARs data. Further, the Single Nearest Neighbour(SNN) association is considered for tracking in the present work, which can be improved by introducing a Global Nearest Neighbour(GNN) or Joint Probabilistic Data Association(JPDA).

References

[1] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2774–2781.

[2] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4604–4612.

[3] C. Wang, C. Ma, M. Zhu, and X. Yang, "Pointaugmenting: Cross-modal augmentation for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 794–11 803.

[4] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le *et al.*, "Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 182–17 191.

[5] T. Yin, X. Zhou, and P. Krähenbühl, "Multimodal virtual point 3d detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 494–16 507, 2021.

[6] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4867–4873, 2020.

[7] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 194–210.

[8] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[9] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 760–13 769.

[10] G. Rigatos and S. Tzafestas, "Extended kalman filtering for fuzzy modelling and multi-sensor fusion," *Mathematical and computer modelling of dynamical systems*, vol. 13, no. 3, pp. 251–266, 2007.

[11] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[12] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 040–11 048.

[13] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.

[14] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," in *Conference on Robot Learning*. PMLR, 2020, pp. 923–932.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[16] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9224–9232.

[17] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[18] R. Ge, Z. Ding, Y. Hu, W. Shao, L. Huang, K. Li, and Q. Liu, "1st place solutions to the real-time 3d detection and the most efficient model of the waymo open dataset challenge 2021," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol. 1, 2021.

[19] Q. Chen, L. Sun, Z. Wang, K. Jia, and A. Yuille, "Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer, 2020, pp. 68–84.

[20] C. R. Qi, Y. Zhou, M. Najibi, P. Sun, K. Vo, B. Deng, and D. Anguelov, "Offboard 3d object detection from point cloud sequences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6134–6144.

[21] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770–779.

[22] C. Yilun, L. Shu, S. Xiaoyong, and J. Jiaya, "Fast point r-cnn," in *ICCV*, 2019.

[23] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2647–2664, 2020.

[24] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 529–10 538.

[25] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, "Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. arxiv 2021," *arXiv preprint arXiv:2102.00463*.

[26] Z. Li, F. Wang, and N. Wang, "Lidar r-cnn: An efficient and universal 3d object detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7546–7555.

[27] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3075–3084.

[28] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European conference on computer vision*. Springer, 2020, pp. 685–702.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[30] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9939–9948.

[31] T. Wang, X. Zhu, J. Pang, and D. Lin, "Fcos3d: Fully convolutional one-stage monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 913–922.

[32] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[33] T. Wang, Z. Xinge, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Conference on Robot Learning*. PMLR, 2022, pp. 1475–1485.

[34] H. Chen, P. Wang, F. Wang, W. Tian, L. Xiong, and H. Li, "Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2781–2790.

[35] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[36] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *Conference on Robot Learning*. PMLR, 2022, pp. 180–191.

[37] Y. Liu, T. Wang, X. Zhang, and J. Sun, "Petr: Position embedding transformation for multi-view 3d object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 531–548.

[38] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao, "Graph-detr3d: rethinking overlapping regions for multi-view 3d object detection," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5999–6008.

[39] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 138–11 147.

[40] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3d object detection," *arXiv preprint arXiv:1811.08188*, 2018.

[41] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, "Bevdet: High-performance multi-camera 3d object detection in bird-eye-view," *arXiv preprint arXiv:2112.11790*, 2021.

[42] E. Xie, Z. Yu, D. Zhou, J. Philion, A. Anandkumar, S. Fidler, P. Luo, and J. M. Alvarez, "M^2 bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation," *arXiv preprint arXiv:2204.05088*, 2022.

[43] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[44] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *European conference on computer vision*. Springer, 2022, pp. 1–18.

[45] J. Lu, Z. Zhou, X. Zhu, H. Xu, and L. Zhang, "Learning ego 3d representation as ray tracing," in *European Conference on Computer Vision*. Springer, 2022, pp. 129–144.

[46] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[47] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[48] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5693–5703.

[49] Z. Wang and K. Jia, "Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1742–1749.

[50] G. Gkioxari, R. Girshick, P. Dollár, and K. He, "Detecting and recognizing human-object interactions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8359–8367.

[51] Y. Zhang, Z. Zhu, W. Zheng, J. Huang, G. Huang, J. Zhou, and J. Lu, "Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving," *arXiv preprint arXiv:2205.09743*, 2022.

[52] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7345–7353.

[53] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[54] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918–927.

[55] R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1527–1536.

[56] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "Futr3d: A unified sensor fusion framework for 3d detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 172–181.

[57] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1090–1099.

[58] S. Xu, D. Zhou, J. Fang, J. Yin, Z. Bin, and L. Zhang, "Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 3047–3054.

[59] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, F. Zhao, B. Zhou, and H. Zhao, "Autoalign: pixel-instance feature aggregation for multi-modal 3d object detection," *arXiv preprint arXiv:2201.06493*, 2022.

[60] Q. Chen, S. Vora, and O. Beijbom, "Polarstream: Streaming object detection and segmentation with polar pillars," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 871–26 883, 2021.

[61] J. Huang and G. Huang, "Bevdet4d: Exploit temporal cues in multi-camera 3d object detection," *arXiv preprint arXiv:2203.17054*, 2022.

[62] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[63] E. W. Kamen and J. K. Su, *Introduction to optimal estimation*. Springer Science & Business Media, 2012.

[64] K. Watanabe and S. G. Tzafestas, "Learning algorithms for neural networks with the kalman filters," *Journal of Intelligent and Robotic Systems*, vol. 3, pp. 305–319, 1990.

[65] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[66] J. L. Crowley and Y. Demazeau, "Principles and techniques for sensor data fusion," *Signal processing*, vol. 32, no. 1-2, pp. 5–27, 1993.

[67] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[68] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[69] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[70] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8555–8564.

[71] D. Park, R. Ambrus, V. Guizilini, J. Li, and A. Gaidon, "Is pseudo-lidar needed for monocular 3d object detection?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3142–3152.