Data Quality Assurance in Autonomous Driving Systems

by

Seyed Matin Tavakoli Afshari

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science
in
Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

University of Alberta

Abstract

In recent years, autonomous driving systems (ADSs) using deep learning-based modules have significantly attracted the attention of researchers from different communities, such as computer vision. These intelligent systems require a precise and accurate training process before their deployment to real-life situations. The performance and reliability of ADSs are dependent on two important factors, namely, training dataset and model components, each of which must be carefully taken into consideration. Since in most of the realistic cases, the models of ADSs are released in a black-box form, and access to their components (e.g., loss functions and hyper-parameters) is not granted, therefore, ensuring the quality of the samples in the ADSs training datasets is of paramount importance. In view of these explanations, in this work, we focus on developing an efficient scheme for cleaning the training datasets of ADSs that employ deep image object detectors, by identifying the samples in the dataset with erroneous bounding boxes. In this regard, we leverage the visual signals associated with the bounding boxes, in addition to their spatial coordinates, for predicting the erroneous status of the bounding boxes in an accurate manner. Moreover, we incorporate confident learning in the proposed scheme in order to prune the predictions of the erroneous statuses of the bounding boxes, and, further contribute to developing secure and reliable ADSs. The results of the extensive experiments demonstrate the effectiveness of various ideas employed in the design of the proposed erroneous bounding box detection scheme for the ADSs datasets. Further, it is shown that the proposed scheme could significantly outperform the other state-of-the-art data selection methods in cleaning the training datasets of ADSs.

Preface

Part of this thesis has been submitted as A Esmaeilzehi, SMT Afshari, Q Guo, F Juefei-Xu, and L Ma ”MIPE: Towards Cleaning the Datasets of Autonomous Driving Systems using Multi-modal Information Processing and Confident Learning” to IEEE Transactions on Intelligent Vehicles (T-IV) Journal (Impact Factor: 8.2).

To my parents

Who have always graciously supported me throughout my life.

He who has a why to live can bear almost any how.

– Friedrich Nietzsche

Acknowledgements

I would like to take the time to express my gratitude towards a handful of people/individuals.

First, I would like to thank my supervisor, Dr. Lei Ma, for his invaluable guidance throughout my research path.

Next, I would like to thank Alireza Esmaeilzehi, a Postdoctoral Fellow who, during his stay at University of Alberta, heavily collaborated in our work.

I would like to thank two of my very good friends, Hossein Zaredar and Arad Firouzkouhi, for genuinely accepting to proofread my thesis during its development.

Lastly, I would like to thank my wonderful partner, Bahar Kaviani, for her moral support all through my master's.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Neural networks have been a crucial and decisive part of modern society in the past 10-15 years. As these networks continue to progress and scale up more and more, a critical issue begins to be unveiled. Namely, it is a matter of the black-box nature of such networks, meaning that the essential learned representations are, for the most part, completely unknown. Numerous studies throughout the years have suggested that one major caveat of this black-box behavior arises when the inputs to the already trained network are carefully crafted to deliberately fool these intelligent systems. This has been perfected to such an extent that the inputs can even dictate what the output of the network shall be. In fact, this area of research has expanded so much that it received its own name: Adversarial machine learning.

Thus, it is clinical to investigate the functionality of such networks and identify their misbehaviors. Specifically, in the area of autonomous driving, such scrutinies become crucial when considering the fact that in this domain, the cost of failure is virtually insufferable. Each erroneous behavior in the functioning of these intelligent systems can result in intolerable consequences, including crashing into other vehicles and objects. Moreover, unlike software-related products, testing cannot be done as frequently, since each failure results in a wasted product, that requires going back to the manufacturing pipeline.

Having shed light on these insights, it becomes of paramount importance to carefully observe the behavior of such intelligent systems with respect to their training process. The training process essentially boils down to two important factors: **training dataset** and **model components**. While meticulously inspecting and analyzing the model components of intelligent systems is in itself a very interesting topic, our focus is mainly on the training datasets. Specifically, this work centers around investigating on **assuring the quality of the data** in the form of the training samples of the dataset utilized by ADSs.

1.2 Background

Investigating the functionality of ADSs and identifying their misbehaviors are fundamental tasks in the community of intelligent vehicles. In recent years, deep neural networks (DNNs) have revolutionized the design of ADSs [1], [4], [7], [9], [24], [30], [32], [37], [45], [48], [53], in view of their high performances and real-time processing capabilities. The safety-critical considerations of ADSs necessitate a meticulous training process for their DNN modules before being deployed to real-life scenarios.

Many ADSs utilize cameras for acquiring visual signals and information from their interacting environments. Hence, employing high-performance image processing tools for extracting important information from the acquired visual signals is crucial for the proper functioning of ADSs. Deep learning-based image object detection schemes [40], [29] are important image processing methods employed by ADSs, which enable these intelligent systems to detect vital objects, such as vehicles, pedestrians, and cyclists, and help them in planning safe and secure driving.

As mentioned, each erroneous behavior in the functioning of ADSs can result in intolerable consequences. For instance, some case studies [11], [34], [35], [42] illustrate several malfunctions of ADSs, in which despite the huge amount of funding provided by the big technology companies, such as Tesla, Uber, and Google, there are still serious concerns regarding their security and reliability, part of which can be emanated from the training process and quality of training samples. It is worth mentioning that since ADSs are considered large-scale complex machine learning systems, they suffer from a lack of transparent troubleshooting. Therefore, it is vital to perform the training process of such systems reliably by using training samples that possess correct annotations and labels.

While in recent years, a vast amount of research has been carried out on designing high-performance models for deep image object detectors [5], [28], [29], [40], [51], not much attention has been paid to developing methods for selecting suitable data for the training process of the deep image object detectors. In view of these explanations, and the importance of designing secure ADSs, in this work, we aim to develop a high-performance erroneous label detection scheme for cleaning the training datasets of deep image object detection blocks employed by ADSs.

1.3 Problem Definition

Existence of erroneous labels in the DNNs' large-scale training datasets can frequently occur due to the exhaustive human annotation process. It has been demonstrated in [33] that even very popular datasets used for training DNNs, such as MNIST [21], contain some samples with erroneous labels. Since the value of the loss of DNNs to the training samples with erroneous labels is higher than that to the samples with clean labels, DNNs are vulnerable to overfitting the erroneously labeled samples in the training process, which negatively affects their performance and reliability. Therefore, handling the training samples with erroneous labels is of paramount importance in the training process of DNNs.

Deep image object detectors employed by ADSs carry out the regression and classification tasks simultaneously by, respectively, localizing the objects of the acquired visual signals and detecting their classes. Therefore, the training datasets of these intelligent systems contain labels for both the classification and regression tasks, i.e., the class labels of the objects, as well as their bounding boxes, which are obtained from the human annotation process. In view of the fact that for ADSs, the number of object classes is small, and the classes are completely distinguishable from one another (for example, one class is “vehicle” and the other is “cyclist”), it is unlikely that a human annotator makes a mistake in determining the classes of the objects. On the other hand, due to the changes in object sizes and occlusions, determining the coordinates of the objects bounding boxes is significantly more challenging for the annotators, and thus, occurring errors in the bounding box annotation process cannot be ruled out. Therefore, the design of an erroneous bounding box detection scheme that is able to faithfully discriminate between clean and erroneous bounding boxes in the training datasets of ADSs is necessary.

The information associated with each bounding box is represented as tabular data, i.e., a vector containing the spatial coordinates of the bounding box. In order to detect the erroneously annotated bounding boxes, one could adopt the existing methods in the literature of data selection (we discuss them in detail in Chapter 2). However, since these schemes are developed for generic machine learning systems, they do not exploit the crucial information that is encoded in the datasets of ADSs. This adversely impacts identifying the erroneous bounding boxes and leads to an improper pruning of the object detection datasets used for training ADSs. In this regard, we aim to develop a novel scheme for detecting erroneous bounding boxes by exploiting the necessary information existing in the datasets of ADSs.

In order to exploit the training datasets of ADSs for the task of erroneous bounding box detection, we propose to extract the useful information from the *visual signals* associated with the bounding boxes, and fuse this information with that obtained from the original representation of the bounding boxes, i.e., the vector containing their spatial coordinates.

1.4 Contribution

The main contributions of this work can be summarized as follows:

- We develop a novel feature extraction technique for obtaining useful information from the visual signals associated with the bounding boxes whose quality and correctness must be assessed.

- We fuse the features obtained from the original representation of the bounding boxes, i.e., the spatial coordinates vector, with those extracted from the corresponding visual signals, in an end-to-end manner, to determine the erroneous status of the bounding boxes in an accurate fashion.

- We determine the uncertainty of the erroneous status of the bounding boxes estimated by our proposed scheme using confident learning (CL), and reduce the risk of not identifying the erroneous bounding boxes in the datasets of ADSs.

- We investigate the impact of our proposed erroneous bounding box detection scheme on the development of more secure and reliable ADSs that employ deep image object detectors.

1.5 Outline

The rest of the thesis is organized as follows: In Chapter 2, we review the existing data selection methods that can be adopted for the task of erroneous bounding box detection. In Chapter 3, we describe the proposed scheme for identifying the erroneously annotated bounding boxes of the datasets of ADSs. In Chapter 4, we carry out extensive experimentations to verify the effectiveness of the various components of the proposed scheme. Furthermore, in this Chapter, the performance of the proposed erroneous bounding box detection scheme for ADSs is compared with those of the other state-of-the-art data selection methods that exist in the literature. Finally, in Chapter 5, we summarize the concluding remarks on the work carried out in this paper.

Chapter 2

Related Work

In Chapter 1, we introduced the problem we focus on (erroneous label detection) and the scope of this work. There are a handful of works focusing on both identifying erroneous labels and alleviating their influence on the training process of intelligent systems (e.g., ADSs). In this Chapter, we will review the related work to locate our problem among similar works.

The techniques used for handling the training samples with erroneous labels can be broadly categorized into two groups. In the first group, the methods, such as [14], [17], [19], aim at modifying the training algorithms of DNNs to reduce the influence of samples with erroneous labels on DNNs training process, and therefore, making DNNs robust to these types of samples. In the second category, the schemes, such as [33], [54], perform data selection on the DNNs training datasets by identifying and eliminating the erroneously labeled samples.

In many real-life situations, the model components of DNNs, including their loss functions and their parameter updating algorithms, have already been optimized after extensive experimentations. Furthermore, access to the model components of DNNs is not granted in many realistic situations, since these intelligent systems are released as black-box models. Based on these points, and the explanations given in the above paragraph, the use of erroneous label detection schemes that are solely dependent on the training datasets and do not require any knowledge about the DNN model components, is preferable in the community of intelligent vehicles for developing secure and reliable systems.

2.1 Scenarios Engineering

Since ADSs employ various AI-enabled modules, such as deep learning-based image object detection blocks, the use of scenarios engineering-based methods is crucial to the development of such systems. Notably, in [27], it has been mentioned that verification and validation (V&V) are two critical stages in the development of AI-enabled systems in a trustworthy fashion. In [25], the authors have developed a novel scenarios engineering-based framework, that includes six layers, namely, infrastructure, operation, knowledge, intelligence, management, and interaction layers, in order to develop trustworthy intelligent systems.

In [26], a new scheme for generating realistic visual signals from synthetic images based on the image-to-image translation method has been proposed. Next, the proposed scheme is utilized for generating an abundant amount of visual signals with their corresponding annotations in order to develop reliable and high-performance ADSs. Given the importance of scenarios-based engineering, it can be concluded that the design of ADSs that are able to perform reliably is necessary in realistic situations.

2.2 Robust Training against Erroneous Labels

The existence of samples with erroneous labels in training datasets leads to deteriorating DNNs performance. In recent years, many schemes have been developed to improve the robustness of DNNs against training samples with erroneous labels. These methods focus on identifying the erroneously labeled samples in the training datasets and reducing their impact on the training process of deep neural networks, to provide superior performances. In the following, we first review the methods that are developed for the robust training of DNNs against samples with erroneous labels. We then discuss the schemes that are specifically developed for training the deep image object detectors robustly against samples with erroneous labels. It is worth mentioning that in this work, we focus solely on the task of erroneous label detection. Therefore, even though the generalizability of DNNs and improving their out-of-domain generalization capability is itself an interesting domain of research, we believe it falls beyond the scope of this work.

In [19], training of a DNN (referred to as the base network) is supervised by another neural network (referred to as the mentor network), which by producing a curriculum, encourages the base DNN to learn only the samples that are associated with the correct labels. In [17], a novel training scheme, called cyclical training, has been proposed in order to identify the samples with erroneous labels. Specifically, in cyclical training, the status of the training of the DNN alternates between overfitting and underfitting by changing the learning rate of the optimization process. It has been shown that by optimizing the parameters of a DNN with the cyclical training process, its loss to the samples with erroneous labels becomes significantly larger than that to the samples with clean labels, which in turn contributes to identifying the erroneously labeled samples. In [52], the training of the DNN is first started by the samples with clean labels and then gradually continued by those whose labels are corrected based on the consistency between the values of the loss in the previous training iterations. In [14], two DNNs with the same architecture have been employed to teach each other by using the training samples with small loss values. In [39], a meta-learning-based method has been proposed to weigh the loss values of the DNN for different training samples based on their erroneous label status. By this, the influence of samples with erroneous labels on the performance of the DNN is decreased, and a stable training process is achieved. In [36], a new metric, referred to as *TracIn*, has been developed to measure the influence of each sample on the training process of DNNs. Specifically, *TracIn* is obtained by multiplying the gradient of the DNN with respect to a given sample by its gradients with respect to the training samples at specific training checkpoints. It has been shown that the samples with erroneous labels lead to negative *TracIn* values.

In addition to the above-mentioned works existing in the literature for handling the erroneously labeled samples in the training datasets of DNNs, there exists several schemes, that focus on improving the robustness of the DNN-based image object detectors against training images with erroneous bounding boxes. In [23], a new algorithm has been proposed for correcting the erroneous bounding boxes in a training dataset of deep image object detectors. Specifically, the parameters of the deep image object detectors are updated during the training process in such a way that the discrepancy between the two detection heads is minimized. Recently, the authors of [47] have proposed a novel scheme, referred to as Meta-Refine-Net (MRNet), for handling the erroneous bounding boxes in the training process of the deep image object detectors, in which two additional low-complexity neural networks have been employed. The first neural network is responsible for reweighting the loss of the classification head based on the erroneous status of the labels, and the second neural network strives to refine the spatial coordinates of the erroneous bounding boxes.

It is seen that all the above-mentioned schemes focus on reducing the influence of samples with erroneous labels on the training process of DNNs. However, the functioning of these schemes relies on the availability of the DNN model components, such as the loss functions and gradients, which restricts the applicability of such schemes in realistic situations. For example, in the case of several ADSs, the model components are not accessible for any adjustment. Hence, for handling the erroneous bounding boxes in the training datasets of ADSs, the development of methods that are able to identify these bounding boxes without requiring any knowledge about the ADSs model components is crucial. In the next Section, we review the approaches that can be employed to detect the erroneous bounding boxes in ADSs training datasets without taking their models into consideration.

2.3 Data Selection Methods

Even though there exists a few schemes in the literature for detecting samples with erroneous labels in the training datasets of DNNs [33], [54], to the best of our knowledge, an **explicit algorithm** for identifying the erroneously labeled bounding boxes of ADSs still has not been developed. Given this, one could adopt the existing data selection methods as the baselines for the task of erroneous bounding box detection. We summarize all these methods in the following categories:

2.3.1 Supervised Learning-based Schemes

When the erroneous status of a number of bounding boxes in the dataset is determined, i.e., they are labeled as *clean* or *erroneous*, several supervised learning-based methods that are devised for processing and classifying tabular data can be employed. These methods can be broadly categorized into two groups: *learning-centric schemes* and *data-centric schemes*. The algorithms, ranging from the conventional high-performance classifiers (e.g. SVM and XGBoost [8]) to the neural networks employing transformers that are specified for processing tabular data (RTDL) [13], can be considered as the learning-centric methods. On the other hand, methods such as SimiFeat [54], which use the neighboring information of each sample for detecting its erroneous status, are data-centric.

2.3.2 Semi-supervised Learning-based Schemes

When the number of bounding boxes in the dataset whose erroneous statuses are available is not sufficient, training a high-performance erroneous bounding box detector is difficult. In this case, semi-supervised learning-based schemes such as S3VM [3] and VIME [49], could be utilized for increasing the amount of the labeled data, and hence, improving the accuracy and efficiency of the erroneous bounding box detection task.

2.3.3 Active Learning-based Schemes

In many realistic situations, the human annotator is available to check the quality of the bounding boxes and correct those that are annotated erroneously. However, due to the large cost of this process and the limited available budget, the process can only be carried out on a portion of the bounding boxes of the dataset. Active learning-based methods [6], [22], could be employed to first obtain a number of the most informative bounding boxes in the dataset, and then pass them to the human annotators for performing the reannotation process.

2.3.4 Confident Learning-based Schemes

The scheme of [33], referred to as confident learning (CL), is a recent state-of-the-art method in the literature of data selection, which aims at determining the samples with unreliable labels in large-scale datasets by using its statistical information. It has been shown in [33] that pruning the datasets with confident learning results in enhancing the performance of the machine learning systems that employ these datasets for training.

It should be noted that in this work, we assume that the erroneous status of some bounding boxes in the ADSs training dataset is available. This assumption is valid, since in real-life situations, human annotators could easily assess the quality of a number of the bounding boxes and determine whether they are erroneous or not. Hence, in the above categorization, we do not consider the *unsupervised learning-based schemes* for detecting erroneous bounding boxes.

Chapter 3

Methodology

In Chapter 2, we reviewed the literature regarding the task of erroneous bounding box detection. In this Chapter, we will go through the proposed scheme, explaining in great detail how it is structured, and how it can be employed to identify the erroneously annotated bounding boxes of the datasets of ADSs. As mentioned, the main novelty of our proposed scheme can be summed up into two essential points:

1. **Multi-modal information processing**: Fusing features from different domains in order to determine the erroneous status of the bounding boxes in an accurate fashion.
2. **Confident learning**: In order to prune the uncertainty of the erroneous status of the bounding boxes

3.1 Motivation

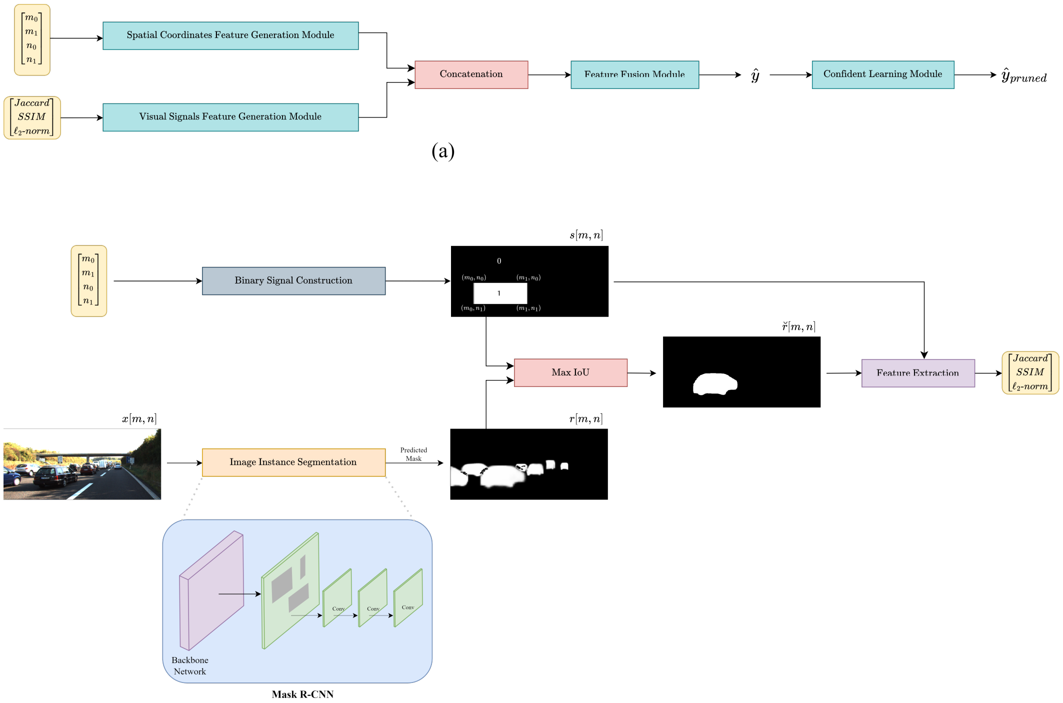
As mentioned in Section 1.3, the bounding boxes in the training datasets of ADSs can be represented as vectors containing the spatial coordinates. In many real-life situations, the spatial coordinates of a clean bounding box and its erroneous version could be very similar to each other. For example, the erroneous bounding box can be obtained by applying the translation operation of only 1 unit (here, pixel) to the original clean bounding box. Therefore, estimating the erroneous status of the bounding box reliably cannot be achieved by merely considering its spatial coordinates.

The bounding box is considered as *clean*, if it precisely encompasses an object in the image. On the other hand, the bounding box is labeled as *erroneous*, if it fails to fit to its corresponding object in the image accurately. Based on these assumptions, it can be argued that the image, to which a given bounding box belongs, contains a set of information that could guide the process of identifying the erroneous bounding boxes accurately. In this regard, we propose to employ the interaction between the two *visual signals*, in which the first one is associated with the image that the bounding box belongs to, and the second corresponds to the spatial coordinates of the bounding box, as a discriminative factor between the clean and erroneous bounding boxes for the task of erroneous bounding box detection. We describe the details of how to obtain these two visual signals in the next Section.

3.2 Algorithm

Let $\mathbf{v} = [m_0, m_1, n_0, n_1]^T$ denote a bounding box belonging to the image $x[m, n]$ ($0 \leq m \leq M - 1$ and $0 \leq n \leq N - 1$. Here, M and N are the width and height of the image, respectively) from the ADSs training dataset, where (m_0, n_0) and (m_1, n_1), respectively, denote the spatial coordinates of the upper left corner and the lower right corner of the bounding box. Let also y denote the erroneous status of the bounding box $\mathbf{v}$. Specifically, if the bounding box $\mathbf{v}$ is clean/erroneous, its corresponding erroneous status becomes $y = 1/y = 0$. Our objective in this work is to confidently identify the bounding boxes that are erroneously labeled in the ADSs training dataset.

Recent advances in developing DNNs have resulted in obtaining promising performances in many information processing tasks, such as the task of tabular data classification [13]. In view of the emergence of novel neural network-based processing techniques, e.g., sophisticated activation functions [31], [16], dropout [2] and transformers [44], one could design a neural network architecture to learn a direct end-to-end mapping from the input vector $\mathbf{v}$ to its corresponding erroneous status y. Given these explanations and those given in the previous Section, we now develop a neural network that employs the spatial coordinates information of the bounding boxes encoded in $\mathbf{v}$, as well as that from their corresponding visual signals, in order to identify the erroneous bounding boxes in the datasets of ADSs.



(a)

(b)

Figure 3.1: High-level block representation of the proposed MIPE. (a) Overall architecture. (b) Details of obtaining the feature vector $\mathbf{w}$ corresponding to the visual signals of the bounding boxes.

Figure 3.1 depicts the overall architecture of the proposed erroneous bounding box detection scheme. It is seen from this figure that the proposed scheme consists of four main modules, namely, *spatial coordinates feature generation* module, *visual signals feature generation* module, *feature fusion* module, and *confident learning* module. It should be noted that we generate features from two different modalities, i.e., the spatial coordinates of the bounding boxes, and the visual signals corresponding to them. Hence, our scheme effectively is a multi-modal information processing system for detecting erroneously labeled bounding boxes in an automated fashion. In the following, we describe each of the four modules utilized in our erroneous bounding box detection scheme in detail.

3.2.1 Spatial coordinates feature generation module

We first extract the useful information of the bounding box from the spatial coordinates vector $\mathbf{v}$. In this regard, we apply a cascade of three fully-connected layers to the vector $\mathbf{v}$ to obtain the feature vector $\mathbf{f}_1$ (output of the module) as:

$$\mathbf{f}_1 = W_1(\mathbf{v}) \tag{3.1}$$

where W_1 represents a cascade of three fully-connected layers that employ, respectively, 16, 32, and 64 hidden units. All these fully-connected layers are followed by batch normalization [18] and the ReLU activation function.

3.2.2 Visual signals feature generation module

In this module, we aim at producing the visual features, which their fusion with the feature vector $\mathbf{f}_1$ obtained from the spatial coordinates feature generation module, facilitates finding the erroneous bounding boxes in the training datasets of ADSs. Since the visual signals associated with the bounding boxes contain a rich set of semantic and geometrical information, one could consider using them for the task of erroneous bounding box detection. As seen from Figure 3.1(b), in order to obtain the visual signals associated with the bounding box $\mathbf{v}$, we first pass it through the *binary signal construction* stage to produce the two-dimensional signal $s[m, n]$ ($0 \leq m \leq M - 1$ and $0 \leq n \leq N - 1$) as:

$$s[m, n] = \left\{ \begin{array}{ll} 1 & m_0 \leq m \leq m_1 \;\; and \;\; n_0 \leq n \leq n_1 \\ 0 & O.W. \end{array} \right. \tag{3.2}$$

The two-dimensional signal $s[m, n]$ possesses the geometrical information of the bounding box $\mathbf{v}$. Hence, it is expected that if $\mathbf{v}$ is a clean bounding box, the corresponding object in the image $x[m, n]$ fits in the area of $s[m, n]$ specified with the values 1. Next, we feed the image $x[m, n]$ (which the bounding box $\mathbf{v}$ belongs to) to the *image instance segmentation* stage to estimate all its object components. Specifically, we pass the image $x[m, n]$ through Mask R-CNN [15], which is a deep learning-based image instance segmentation scheme, and obtain the two-dimensional signal $r[m, n]$ as:

$$r[m, n] = MaskRCNN(x[m, n]) = \left\{ \begin{array}{ll} 1 & (m, n) \in O_i \\ 0 & O.W. \end{array} \right. \tag{3.3}$$

where O_i represents the i-th object of the image $x[m, n]$. For the network Mask R-CNN, we employ ResNet50-FPN [28] as the backbone for the feature extraction process. ResNet50-FPN employs several skip connections in its network architecture, which facilitates the flow of information in both forward and backpropagation. Further, it produces features at multiple spatial scales, which are crucial for segmenting objects and instances with various spatial sizes. We use 256 hidden units in the classification layer of Mask R-CNN. The signal $r[m, n]$ produced as the output of Mask R-CNN contains the I objects of the image $x[m, n]$. Then, we obtain the IoU (intersection over union) values between the area of the signal $s[m, n]$ specified with the values 1 and each of the I objects in the signal $r[m, n]$. We refer to these IoU values as $a_1, a_2, ..., a_I$. Next, the maximum value among a_i's, which corresponds to the most correlated object in the image $x[m, n]$ to the bounding box $\mathbf{v}$, is obtained for constructing the signal $\check{r}[m, n]$ as:

$$\check{r}[m,n] = \begin{cases} 1 & (m,n) \in O_{i_{max}} \\ 0 & O.W. \end{cases} \tag{3.4}$$

where $i_{max} = \underset{i}{\arg\max}(a_i)$, $\big(i = 1, ..., I\big)$. It is seen from Equation (3.4) that the signal $\check{r}[m, n]$ now possesses only one object, that is the one with the maximum correlation with the bounding box $\mathbf{v}$.

After yielding the two visual signals $s[m, n]$ and $\check{r}[m, n]$, we can obtain the vector $\mathbf{w}$, whose entities represent the interactions between these two signals. Specifically, we form the vector $\mathbf{w}$ as:

$$\begin{aligned} &\mathbf{w} = [b_1, b_2, b_3]^T \\ &b_1 = Jaccard(s[m,n], \check{r}[m,n]) \\ &b_2 = SSIM(s[m,n], \check{r}[m,n]) \\ &b_3 = \|s[m,n] - \check{r}[m,n]\|_2 \end{aligned} \tag{3.5}$$

where $Jaccard$ and $SSIM$ denote, respectively, the Jaccard similarity index and the structural similarity index measure [46] between the two visual signals. It should be noted that in addition to these three metrics, we have also attempted to include some other similarity metrics, such as D^2 pinball metric, D^2 Tweedie metric, and cosine similarity metric, in the vector $\mathbf{w}$. However, we

Algorithm 1 Obtaining Feature Vector $\mathbf{w}$

```
**Input**: Bounding box $\mathbf{v} = [m_0, m_1, n_0, n_1]^T$, Image $x[m, n]$, $\mathbf{v}$ *belongs to* $x[m, n]$.
**Output**: Feature vector $\mathbf{w}$.
Obtain $s[m, n]$ from $\mathbf{v}$ using eq. (3.2).
Apply Mask R-CNN to $x[m, n]$ in order to obtain $r[m, n]$ $\big($eq. (3.3)$\big)$.
**for** $i = 1 : I$ (total number of objects in $x[m, n]$) **do**
    Obtain $a_i = IoU(Keep_{O_i}(r[m, n]), s[m, n])$.
    $\#\#\#Keep_{O_i}(r[m, n])$ keeps only the $i$-th object in $r[m, n]$.
**end for**
Obtain $i_{max} = \underset{i}{\arg\max}(a_i)$.
Obtain $\check{r}[m, n]$ using eq. (3.4).
**Return** $\mathbf{w}$ using eq. (3.5).
```

have observed that the inclusion of other metrics in the vector $\mathbf{w}$ not only does not improve the accuracy of the task of erroneous bounding box detection, but sometimes even deteriorates it. Hence, we only use the Jaccard, SSIM, and ℓ_2-norm metrics for constructing the vector $\mathbf{w}$, as we empirically observed that they better exploit the similarity between the two visual signals in our task.

It should be noted that the vector $\mathbf{w}$ obtained from the visual signals associated with the bounding box now has a rich set of semantic and geometrical information, which its fusion with the original representation of the bounding box, $\mathbf{v}$, could further enhance the performance of detecting erroneously labeled bounding boxes. Finally, we apply a cascade of three fully-connected layers to the vector $\mathbf{w}$ to obtain the feature vector $\mathbf{f}_2$ (output of the visual signals feature generation module) as:

$$\mathbf{f}_2 = W_2(\mathbf{w}) \tag{3.6}$$

where W_2 denotes a cascade of three fully-connected layers that utilize, respectively, 16, 32, and 64 hidden units. All these fully-connected layers are followed by batch normalization [18] and the ReLU activation function.

The summary of the operations carried out in obtaining feature vector $\mathbf{w}$ is given in Algorithm 1.

3.2.3 Feature fusion module

In this module, we fuse the information obtained from the above two feature generation modules, in order to produce a rich set of features for the task of erroneous bounding box detection. Specifically, the feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ are first concatenated, and the feature vector $\mathbf{f}_3$ is yielded. Next, the feature vector $\mathbf{f}_3$ is passed through a cascade of four fully-connected layers to obtain the estimated erroneous status $\hat{y}$ of the bounding box as:

$$\hat{y} = W_3(\mathbf{f}_3) \tag{3.7}$$

where W_3 represents a cascade of four fully-connected layers that utilize, respectively, 64, 32, 16, and 1 hidden units. All these fully-connected layers, except the final layer, are followed by batch normalization and the ReLU activation function. The final fully-connected layer is only followed by batch normalization.

3.2.4 Confident learning module

Even though the proposed neural network for the task of erroneous bounding box detection learns a nonlinear end-to-end mapping from the features of both the bounding boxes and visual signals to their erroneous statuses, it is still possible that some of the estimated erroneous statuses $\hat{y}$ obtained by the proposed neural network would not be precise. To identify the bounding boxes whose erroneous status $\hat{y}$ (here, $\hat{y} = 0$ and $\hat{y} = 1$, respectively, denote the erroneous and clean bounding boxes) do not have high confidence, we employ the confident learning technique. Algorithm 2 summarizes the confident learning module employed by our proposed erroneous bounding box detection scheme.

Let y denote the true erroneous status of the bounding box $\mathbf{v}$, and $\mathbf{V}$ represent the set of all the bounding boxes in the training dataset of the ADS. We first obtain the *joint confidence* statistic $c_{y,\hat{y}}$ between y and $\hat{y}$ as:

$$\begin{aligned} & c_{y,\hat{y}}[i,j] = Count(\mathbf{V}_{y=i,\hat{y}=j}) \\ & \mathbf{V}_{y=i,\hat{y}=j} = \{\mathbf{v} \in \mathbf{V} | \hat{p}(\hat{y} = i | \mathbf{v}, \boldsymbol{\theta}) \geq t_i\} \quad i \in \{0,1\},\ j \in \{0,1\} \end{aligned} \tag{3.8}$$

where $Count$ counts the number of bounding boxes with a specific property, $\hat{p}(\hat{y} = i | \mathbf{v}, \boldsymbol{\theta})$ represents the estimated probability that the bounding box $\mathbf{v}$ is

Algorithm 2 Confident Learning Module

```
Input: List of bounding boxes $\mathbf{V}$ of the training dataset of the ADS, List of erroneous status $\hat{y}$ estimated by the proposed neural network for the bounding boxes $\mathbf{V}$.
Output: List of reliable bounding boxes $\mathbf{V}_{reliable}$.
for $i = 0 : 1$ do
    for $j = 0 : 1$ do
        Obtain the joint confidence statistic $c_{y,\hat{y}}[i,j]$ using eq. (3.8).
        Obtain the joint probability distribution $\mathbf{Q}_{y,\hat{y}}[i,j]$ using eq. (3.10).
    end for
end for
$\mathbf{S} = Sort_{c_{y,\hat{y}}}(\mathbf{V})$.
### Sorts the bounding boxes in $\mathbf{V}$ based on their self-confidence values in a descending order.
Return $\mathbf{V}_{reliable}$ by removing the last $n \sum_{i \in \{0,1\}: i \neq j} \mathbf{Q}_{y,\hat{y}}[i,j]$ bounding boxes from $\mathbf{S}$.
```

identified with the erroneous status $\hat{y} = i$ (where $i = 0, 1$) when the predictor employs the set of parameters $\boldsymbol{\theta}$, and t_i is the expected self-confidence of the i-th erroneous status and is given by:

$$t_i = \frac{1}{Count(\mathbf{V}_{\hat{y}=i})} \sum_{\mathbf{v} \in \mathbf{V}_{\hat{y}=i}} \hat{p}(\hat{y} = i | \mathbf{v}, \boldsymbol{\theta}) \tag{3.9}$$

We estimate $\hat{p}(\hat{y} = i | \mathbf{v}, \boldsymbol{\theta})$ using the random forest classifier with the set of parameters $\boldsymbol{\theta}$. Random forest is a fast classification algorithm that by employing a small complexity, can provide reliable estimations. We set the maximum depth and the number of trees in the random forest classifier as 7 and 100, respectively. After obtaining the joint confidence statistics for the erroneous status $\hat{y}$ of the bounding boxes using Equation (3.8), we estimate their joint probability distribution matrix $\mathbf{Q}_{y,\hat{y}}[i,j]$ as:

$$\mathbf{Q}_{y,\hat{y}}[i,j] = \frac{\frac{c_{y,\hat{y}}[i,j]}{\sum_{i \in \{0,1\}} c_{y,\hat{y}}[i,j]} Count(\mathbf{V}_{\hat{y}=j})}{\sum_{i \in \{0,1\}} \sum_{j \in \{0,1\}} \left(\frac{c_{y,\hat{y}}[i,j]}{\sum_{i' \in \{0,1\}} c_{y,\hat{y}}[i',j]} Count(\mathbf{V}_{\hat{y}=j}) \right)} \tag{3.10}$$

The off-diagonal entities of the joint probability distribution matrix $\mathbf{Q}_{y,\hat{y}} \in \mathbb{R}^{2 \times 2}$ demonstrate the probability that the bounding box $\mathbf{v}$ identified by our proposed scheme as clean/erroneous was indeed erroneously/correctly labeled. Hence, in order to identify the bounding boxes in the training dataset of the

ADS, whose erroneous status estimated by our neural network is not reliable, for each erroneous status $\hat{y} = j$, we remove the $n \sum_{i \in \{0,1\}: i \neq j} \mathbf{Q}_{y,\hat{y}}[i, j]$ bounding boxes with the lowest joint confidence values (in this work, $n = 4361$, which is 10% of the total number of bounding boxes).

Since the proposed erroneous bounding box detection scheme employs two modalities (the spatial coordinates of the bounding boxes and the visual signals corresponding to the bounding boxes) for obtaining the erroneous status of the bounding boxes, we refer to it as ***M**ulti-modal **I**nformation **P**rocessing network for **E**rroneous bounding box detection (MIPE).*

After obtaining the erroneous bounding boxes in the dataset, the following two scenarios can happen. First, the human annotator may not be available to correct the erroneous bounding boxes. In this case, the identified erroneous bounding boxes must be removed from the ADSs training dataset to avoid jeopardizing the training process of these intelligent systems. Second, access to the reannotation process by human experts is granted. In this case, the detected erroneous bounding boxes can be passed to the human annotators to be corrected. Considering that the latter scenario requires more budget than the former, and that the availability of human annotators is not always guaranteed in real-life situations, in this work, our main objective is to focus on the first scenario for cleaning the ADSs training datasets. However, in order to carry out a comprehensive study, we performed several experiments for the second scenario as well.

3.3 Training Process

We now describe the details of the training process of the proposed erroneous bounding box detection scheme.

3.3.1 Erroneous bounding box generation

KITTI [12] is one of the most popular datasets that is utilized for training ADSs. Specifically, this dataset contains 6373 training images and 1108 validation images (both of size 370×1224 pixels), all acquired by a vehicle driving in and around Karlsruhe, Germany. We extract the 43614 bounding boxes from the 6373 training images of the KITTI dataset for the training and evaluation processes of the proposed MIPE. In this regard, and because the proposed MIPE is employed for cleaning the training datasets of deep image object detectors, we employ cross-validation, i.e., we first train the proposed MIPE with half of the bounding boxes in the training set of KITTI (21807 bounding boxes), and then apply it to the remaining half of the bounding boxes for obtaining their erroneous statuses, and vice versa. Finally, the cleaned version of the KITTI dataset obtained by the proposed MIPE and cross-validation is employed in the training process of deep image object detectors.

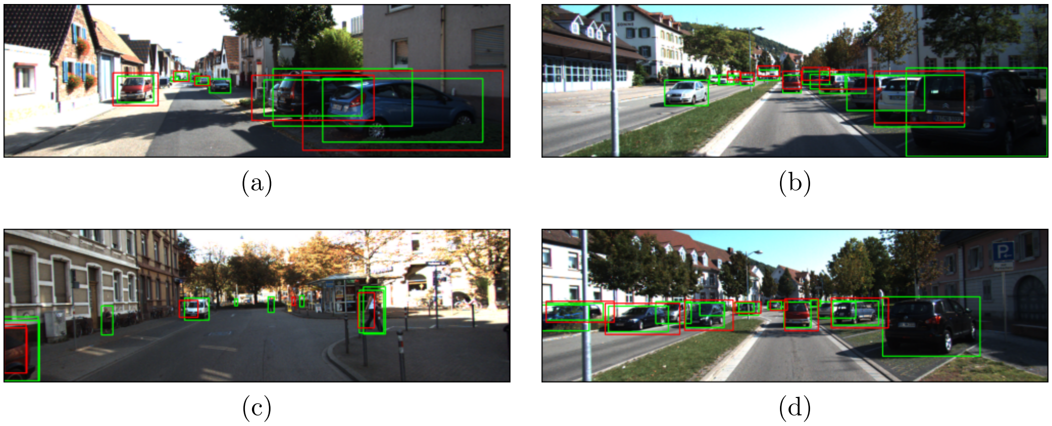
The 43614 bounding boxes of the KITTI training dataset specify objects from 9 different classes, namely, *car*, *van*, *truck*, *pedestrian*, *person* (*sitting*), *cyclist*, *tram*, *misc.*, and *don't care*. The number of bounding boxes for each of these 9 classes are, respectively, 23861, 2615, 934, 4299, 222, 1445, 410, 952, and 8876. The sizes of these bounding boxes vary in the range of [3,510] and [3,374] pixels along the x- and y-axes, respectively.

To synthetically generate erroneous bounding boxes for the KITTI dataset, we first create a pool of corrupting operations consisting of translation, scaling, and their combinations. Specifically, we use 20 translation operations (translating the bounding box with 10, 20, 30, 40, and 50 pixels in one of the four directions of left, right, up, or down), 4 scaling operations (scaling the bounding box symmetrically by factors of 0.5, 0.75, 1.25, and 1.5), and their combinations, i.e., scaling followed by the translation operation, mounting to the total number of 104 operations. It is worth noting that since the spatial size of the images in the KITTI dataset is 370 × 1224 pixels, we observe that the parameters used for the translation and scaling operations in our pool result in producing a diverse set of realistic erroneous bounding boxes. We then randomly select an operation from our corrupting pool thus created and apply it to a random bounding box chosen from the KITTI dataset. We carry out this process on the α percentage of the bounding boxes of the KITTI dataset (in this work, we set $\alpha = 25, 50,$ and 75). We assign the erroneous status $y = 0$ to the bounding boxes that are changed by the operations of the pool, and

$y = 1$ to those that are kept unchanged. Figure 3.2 shows several erroneous bounding boxes generated by the above process.

3.3.2 Training details of MIPE

As seen from Figure 3.1, the proposed scheme consists of two trainable parts, i.e., the image instance segmentation stage, which is Mask R-CNN, and the network used to map the vectors $\mathbf{v}$ and $\mathbf{w}$ to the erroneous status y of the bounding box. We first train Mask R-CNN with 200 images of the KITTI Instance Segmentation dataset [12] and their corresponding instance segmentation maps. Specifically, this dataset consists of two parts, training part and testing part, each containing 200 images (of size 370×1224 pixels), where only the images of the training part are associated with the ground truth instance segmentation maps. The stochastic gradient descent (SGD) optimizer with a learning rate of 0.005 is utilized for training Mask R-CNN. The batch size in each training iteration of Mask R-CNN has been chosen as 1. The training process of Mask R-CNN is continued for 6000 iterations. When training Mask R-CNN is finished, we employ it as a *fixed* stage (freeze its parameters) in the architecture of the proposed MIPE. Finally, we train the proposed MIPE


(a)

(b)

(c)

(d)

Figure 3.2: Examples of erroneous bounding boxes that are synthetically produced for our experiments. Corrupting operations are: (a) scaling by a factor of 1.25, (b) translating up by 10 pixels, (c) scaling by a factor of 0.75, followed by translating left by 20 pixels, (d) scaling by a factor of 1.25, followed by translating right by 10 pixels. All operations are applied to 50% of the bounding boxes.

with the bounding boxes of the KITTI dataset (this dataset is different than that used for training Mask R-CNN) and their erroneous status y using the Adam optimizer [20] with an initial learning rate of 0.1. The batch size in each training iteration of MIPE is 6. The training process of the proposed MIPE is continued for 191220 iterations. We decrease the learning rate by a factor of 10 after each 60000 iterations.

Chapter 4

Experimental Results

In Chapter 3 we proposed a new scheme for erroneous bounding box detection, and went through the algorithmic procedure. In this Chapter, we will dive deep into the experimental results conducted, and further demonstrate the effectiveness and superiority of the proposed erroneous bounding box detection scheme in cleaning the datasets of autonomous driving systems, compared to the state-of-the-art data selection schemes. We will perform several ablation studies to verify the usefulness of different modules employed in the proposed erroneous bounding box detection scheme. We will also compare the performance of the proposed scheme with those of the state-of-the-art data selection methods for both the tasks of erroneous bounding box detection and image object detection. Finally, we will examine the effectiveness of MIPE in domains other than ADSs.

4.1 Effectiveness of Employing Multi-modal Information Processing by MIPE

To investigate whether the processing of both the spatial coordinates of the bounding boxes and their associated visual signals is effective in identifying the erroneous bounding boxes accurately, we form the two following variants of the proposed erroneous bounding box detection scheme:

Proposed scheme without employing spatial coordinates feature generation module In this variant, we remove the spatial coordinates feature generation module from the proposed erroneous bounding box detection

Table 4.1: Effectiveness of employing the multi-modal information processing technique by the proposed MIPE. All performances are in terms of accuracy (%).

Method	*25%* Err. Rate	*50%* Err. Rate	*75%* Err. Rate
W/O Spatial Coordinates	90.23	82.15	73.57
W/O Visual Signals	88.56	68.05	64.79
Proposed Scheme	93.66	88.72	82.46

The values in red indicate the best performance.

scheme.

Proposed scheme without employing visual signals feature generation module In this variant, we remove the visual signals feature generation module from the proposed scheme.

We train the proposed scheme and its two above-variants with the bounding boxes of the KITTI training dataset with 25%, 50%, and 75% error rates. Table 4.1 gives the accuracy of the proposed scheme and its two variants obtained using cross-validation. It is seen from the results of this table that indeed processing both the spatial coordinates of the bounding boxes and the visual signals associated with them have a positive impact on providing a superior erroneous bounding box detection performance. Since the features generated by the visual signals exploit the geometrical and structural information associated with each bounding box efficiently, their incorporation with the features obtained from the spatial coordinates enhances the performance of the erroneous bounding box detection task.

It is seen from the results of Table 4.1 that the network, which merely processes the visual signals associated with the bounding boxes is able to provide superior performance to the one employing only the spatial coordinates of the bounding boxes. This shows that the distributions of the features generated from the visual signals associated with the bounding boxes are more discriminable for a classifier than those produced by the spatial coordinates of the bounding boxes.

Table 4.2: Effectiveness of employing the confident learning technique by the proposed MIPE. All performances are in terms of accuracy (%).

Method	*25%* Err. Rate	*50%* Err. Rate	*75%* Err. Rate
W/O CL Module	89.42	85.12	79.58
Proposed Scheme	93.66	88.72	82.46

The values in red indicate the best performance.

4.2 Effectiveness of Employing Confident Learning by MIPE

The final module employed by the proposed erroneous bounding box detection scheme is confident learning, which identifies the bounding boxes that are *unreliably* estimated by our neural network as clean/erroneous. To determine the impact of the confident learning module used in our scheme on enhancing the accuracy of the task of erroneous bounding box detection, we form a variant of the proposed scheme by removing the confident learning module. It should be pointed out that both the proposed scheme and its variant that does not employ the confident learning module utilize both the features of the spatial coordinates of the bounding boxes and the visual signals associated with them for identifying the erroneous status of the bounding boxes. Table 4.2 gives the accuracy of the proposed scheme and its variant without the confident learning module.

It is seen from the results of this table that employing the confident learning module in the proposed scheme indeed contributes to enhancing the performance of the task of erroneous bounding box detection. Specifically, the accuracy of identifying the erroneous status of the bounding boxes correctly is improved by 4.24%, 3.60%, and 2.88%, in the cases of the three different error rates, when the confident learning module is incorporated in the proposed erroneous bounding box detection scheme. In fact, the confident learning module leverages the statistical information associated with the estimated erroneous statuses of the bounding boxes, and identifies those that are predicted with low confidence. Hence, by pruning the low-confident estimations, a superior erroneous bounding box detection performance is yielded.

Table 4.3: Comparison between different classifiers employed by the confident learning module of the proposed MIPE. All performances are in terms of accuracy (%).

Method	*25%* Err. Rate	*50%* Err. Rate	*75%* Err. Rate
Base Accuracy (W/O CL)	89.42	85.12	79.58
CL (Naïve Bayes)	91.41	86.29	81.24
CL (Neural Network)	91.57	86.69	80.44
CL (SVM)	93.21	87.73	82.14
CL (Random Forest)	93.66	88.72	82.46

It is mentioned in Section 3.2.4 that we estimated $\hat{p}(\hat{y} = i|\mathbf{v}, \boldsymbol{\theta})$ using the random forest classifier. To investigate the impact of using the random forest classifier in the confident learning module on the performance of MIPE, we now replace random forest with three other fast classification algorithms, namely, naïve Bayes, SVM, and neural network with 50 hidden units, for estimating $\hat{p}(\hat{y} = i|\mathbf{v}, \boldsymbol{\theta})$. Table 4.3 gives the results of this experiment.

As seen from this table, the best performance of the proposed MIPE is achieved when its confident learning module employs the random forest classifier for estimating the probability $\hat{p}(\hat{y} = i|\mathbf{v}, \boldsymbol{\theta})$.

4.3 Comparison between MIPE and an Ensemble of Deep Image Object Detectors

It has been shown in Section 4.1 that the use of the visual signals associated with the bounding boxes has a significant impact in assessing their erroneous statuses correctly. Given this observation, one could argue that there exist other ways of using the visual signals corresponding to bounding boxes to determine their erroneous statuses. One such way is to employ an ensemble of deep learning-based image object detection schemes and take their agreement with the given bounding box as a metric in order to determine its erroneous status. Following this, we design an experiment, in which the performance of the proposed MIPE is compared with that of the ensemble of deep image object detectors. Specifically, we apply the three deep image object detectors, Faster R-CNN [40], RetinaNet [29], and FCOS [43], to the image $x[m, n]$ that

Table 4.4: Effectiveness of the proposed MIPE against an ensemble of deep image object detectors. All performances are in terms of accuracy (%).

Method	*25%* Err. Rate	*50%* Err. Rate	*75%* Err. Rate
Ensemble of Detectors	73.51	54.97	41.42
Proposed Scheme	93.66	88.72	82.46

The values in red indicate the best performance.

the given bounding box **v** belongs to. Then, for each of the deep image object detectors, we obtain the bounding box estimated by it that has the maximum IoU with the given bounding box **v**. Next, we calculate the Euclidean distance between the estimated bounding box thus obtained from each deep image object detector and the given bounding box **v**. Finally, we compare the average Euclidean distances of the three deep image object detectors with a pre-defined threshold. If the average of the Euclidean distances is above this threshold, it can be concluded that the given bounding box is erroneous. Otherwise, the average value lower than the threshold would indicate that the bounding box **v** is clean.

Table 4.4 gives the results of the above experiment. It should be pointed out that the value of the threshold is empirically set to 0.5. It is seen from the results of this table that the proposed MIPE indeed is a more effective way for cleaning the datasets of ADSs in comparison to the ensemble method, even though both of the schemes employ visual signals for carrying out the task of erroneous bounding box detection. For example, it is seen from Table 4.4 that the performance of MIPE is 20.15%, 33.75%, and 41.04% superior to that of the ensemble method in the cases of the 25%, 50%, and 75% error rates, respectively. This superiority of the proposed MIPE can be attributed to multiple factors, including the end-to-end learning capability between the input features of the bounding boxes and their erroneous statuses, multi-modal information processing from both the bounding boxes' spatial coordinates and their corresponding visual signals, and the confident learning module for pruning the low-confident estimated erroneous statuses.

4.4 Comparative Study of Performance of MIPE for Erroneous Bounding Box Detection

As mentioned in Chapter 1, the proposed erroneous bounding box detection scheme is necessarily a data selection method that is designed for cleaning the training datasets of ADSs. To investigate the effectiveness of the proposed scheme in a comprehensive manner, we compare its performance with those of XGBoost [8], RTDL (FT-Transformer) [13], SimiFeat [54] (all are *supervised learning-based* data selection methods), Ranked Batch AL [6] (*active learning-based* data selection method), S3VM [3] and VIME [49] (*semi-supervised learning-based* data selection methods), and CL [33] (*confident learning-based* data selection method). Specifically, we compare the accuracy of these schemes in identifying the clean and erroneous bounding boxes correctly.

We also evaluate the impact of the data cleaning process performed by the proposed scheme on the performance of the deep image object detectors utilized in ADSs. In this regard, we compare the performances of the two deep image object detectors, namely, Faster R-CNN [40] and RetinaNet [29], in different cases that they are trained with the KITTI dataset whose bounding boxes are cleaned using the proposed scheme and the other data selection methods.

In order to scrutinize the effectiveness of the proposed MIPE, we compare its accuracy for the task of erroneous bounding box detection with those of the other state-of-the-art data selection methods (mentioned in the previous paragraph). Table 4.5 gives the accuracy of these different methods in identifying the erroneous status of the bounding boxes correctly. We cannot obtain the accuracy of Ranked Batch AL [6], since this scheme is completely carried out by accessing annotators (however, we investigate its effectiveness on the performance of deep image object detectors in the next Section).

It is seen from the results of this table that the proposed erroneous bounding box detection scheme, MIPE, is able to discriminate between the clean and erroneous bounding boxes more accurately than the other state-of-the-art

Table 4.5: Comparative study of the proposed MIPE for the task of erroneous bounding box detection. All performances are in terms of accuracy (%).

	Dataset	KITTI [12]			BDD100K [50]			Waymo [41]		
	Err. Rate	*25%*	*50%*	*75%*	*25%*	*50%*	*75%*	25%	*50%*	*75%*
Method	SVM	80.57	53.86	47.41	80.38	53.00	67.26	80.36	55.56	67.00
	XGBoost [8]	80.97	57.47	49.16	80.92	55.08	68.12	80.86	56.76	67.53
	S3VM [3]	73.67	50.35	61.31	74.62	51.39	61.27	73.91	50.28	60.99
	VIME [49]	81.84	59.75	58.26	80.98	55.95	68.38	80.94	57.16	67.90
	Ranked Batch AL [6]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	RTDL [13]	82.02	59.23	51.77	81.38	56.10	68.53	81.05	57.46	68.39
	SimiFeat [54]	84.48	69.99	71.00	82.07	56.32	69.20	81.20	57.54	69.30
	CL [33]	84.59	71.26	70.87	82.43	55.34	71.23	81.55	58.12	70.31
	Proposed MIPE	93.66	88.72	82.46	84.91	68.88	75.64	84.46	65.73	73.41

The values in red indicate the best performance.

data selection methods do. For example, the accuracy of the proposed MIPE is 9.07%, 17.46%, and 11.59%, higher than that of the second best-performing data selection method, CL [33], in the cases of the 25%, 50%, and 75% error rates of the KITTI dataset. This significant improvement of the accuracy of MIPE over those of the other data selection methods is achieved in light of the processing of the visual signals corresponding to the bounding boxes, since they are more interpretable to the classifier for identifying the clean and erroneous bounding boxes. Further, the results of Table 4.5 confirm the importance of the objective of this work, i.e., the development of a novel data selection method that is specified for the task of erroneous bounding box detection, as the performances of the existing data selection methods for such a task are limited.

By comparing the results of Tables 4.2 and 4.5, it is seen that the performance of the variant of the proposed MIPE that does not employ the confident learning module is still significantly higher than those of the other state-of-the-art data selection methods. Therefore, it can be concluded that even though the use of the confident learning module by the proposed MIPE helps enhance its performance, the proposed scheme without employing the confident learning module is still able to significantly outperform the existing data selection methods in cleaning the training datasets of ADSs.

We now investigate the effectiveness of the proposed MIPE in the case of the training datasets with bounding boxes that are misannotated by different levels of severity. In this regard, we consider 5 levels of severity for corrupting the bounding boxes based on the IoU metric. Specifically, we randomly select an operation from the corrupting pool and apply it to a bounding box that is randomly chosen from the KITTI dataset. We next measure the IoU metric between the new erroneous bounding box generated by the corrupting operation and its clean version. Since the value of the IoU metric lies within the range of $[0, 1]$, we divide this range into 5 equally-spaced bins to determine the level of severity of the erroneous bounding box. We consider the first bin, i.e., $[0.8, 1]$, as severity level 1, and similarly, the last bin, i.e., $[0, 0.2)$, as severity level 5. Therefore, the level of severity of the erroneous bounding box is considered to be the same as the bin number corresponding to the IoU value with its clean version. We form a version of the KITTI training dataset with 50% erroneous bounding box rate for each of the 5 levels of the misannotation severities. Figure 4.1 shows the accuracy of the proposed MIPE and the other data selection methods as a function of the severity level of the misannotation. The following remarks can be made from this figure. First, it is seen that the accuracy of the proposed MIPE in detecting erroneous bounding boxes is significantly superior to those of the other data selection methods in the cases of all the 5 severity levels of the misannotation. Second, it is observed that, except for the proposed MIPE, whose performance degrades consistently by increasing the level of the severity of misannotation, the performance of the other schemes does not always decrease as a function of the severity level. Specifically, the performance of the other data selection methods in the case of severity level 4 is slightly superior to those in the case of severity level 3. We argue that since the other data selection methods employ only the spatial coordinates of the bounding boxes, their sensitivity to the change in distribution of the erroneous bounding boxes is not as high as that of the proposed scheme, which processes the visual signals in addition to the spatial coordinates. Therefore, these schemes could provide some potentially unreliable results when the distribution of the erroneous bounding boxes is changed in a slight regime.

4.5 Effectiveness of MIPE in Developing more Secure and Reliable ADSs

As we have mentioned in Chapter 3, the main objective of this work is to clean the training datasets of ADSs that employ deep image object detection blocks, since training these intelligent systems with erroneously labeled training samples leads to a reduction in both their performance and security in real-world situations. To investigate the impact of the proposed MIPE in developing more reliable ADSs, we carry out an experiment, in which the two deep image object detectors, namely, Faster R-CNN [40] and RetinaNet [29], are trained by different versions of the KITTI, BDD100K [50], and Waymo [41] training datasets, each containing different numbers of erroneously labeled bounding boxes. The KITTI training dataset consists of 6373 images and 43614 bounding boxes. These bounding boxes specify objects from 9

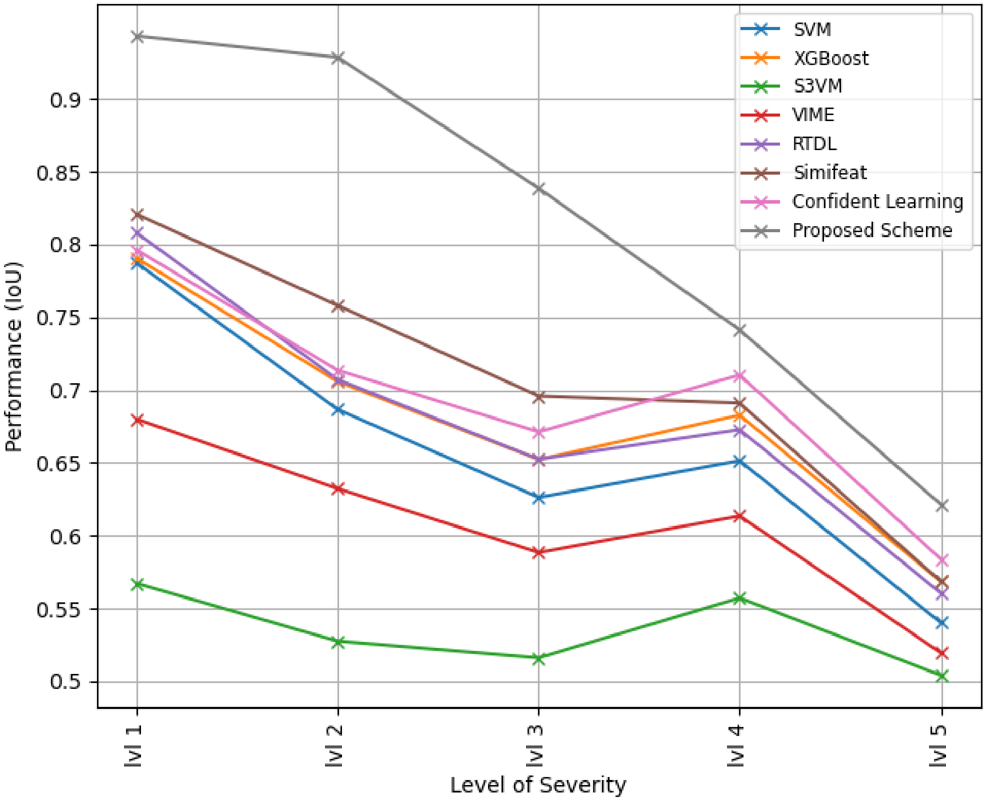


Figure 4.1: Accuracy of various data selection methods as a function of the severity level of misannotation.

different classes, namely, *car*, *van*, *truck*, *pedestrian*, *person (sitting)*, *cyclist*, *tram*, *misc.*, and *don't care*. The BDD100K training dataset has 70000 images and 1273707 bounding boxes, where the objects related to the bounding boxes are from 10 different classes, namely, *pedestrian*, *rider*, *car*, *truck*, *bus*, *train*, *motorcycle*, *bicycle*, *traffic light*, and *traffic sign*. Finally, the Waymo training dataset contains 798 images and 19625 bounding boxes. The bounding boxes are from classes *vehicle*, *pedestrian*, *cyclist*, and *sign*. Similar to the task of erroneous bounding box detection, we use the KITTI [12], BDD100K [50], and Waymo [41] Object Detection datasets for training and evaluation of the two deep image object detection networks, Faster R-CNN [40] and RetinaNet [29]. The description of the training set of the KITTI dataset [12] is thoroughly explained in Section 3.3, and for BDD100K and Waymo is explained above. For evaluating the performance of the deep image object detectors, we use the validation sets of these datasets (since their test datasets do not consist of ground truth bounding boxes). The KITTI validation dataset contains 1108 images and 8251 bounding boxes. Specifically, the validation images of the KITTI dataset are annotated based on 8 classes, namely, *car*, *van*, *truck*, *pedestrian*, *cyclist*, *tram*, *misc.*, and *don't care*. The number of bounding boxes in each of these 8 classes are, respectively, 4881, 299, 160, 188, 182, 101, 21, and 2419. The sizes of these bounding boxes vary in the range of [7,444] and [4,360] pixels along the x- and y-axes, respectively. The BDD100K validation dataset contains 10000 images and 185945 bounding boxes from 10 classes (similar to its training dataset), where the number of bounding boxes in each of these 10 classes are, respectively, 13425, 658, 102837, 4243, 1660, 15, 460, 1039, 26884, and 34724. Finally, for the Waymo validation dataset, we have 202 images and 4534 bounding boxes, where each class consists of, respectively, 3463, 1039, 32, and 0 number of samples (no objects of class *sign* included). We train the two image object detectors with each of the training datasets in the following cases:

1) Clean dataset In this case, all the bounding boxes of each training dataset are clean. This is the ideal case and provides the upper bound of the performance for the object detectors.

2) Dataset with erroneously labeled samples For this case, we consider three versions of each training dataset, namely, dataset with 25%, 50%, and 75% error rates, that correspond to applying the corrupting operations from the pool to 25%, 50%, and 75% of the bounding boxes of each training dataset, respectively.

3) Dataset cleaned by different data selection methods In this case, we first apply the corrupting operations to 25%, 50%, and 75% of the bounding boxes of each training dataset, and then clean them by the various data selection methods mentioned in Section 4.4. Specifically, in this case, we assume that the annotators are not available to correct the bounding boxes that are identified as erroneous by each of the data selection methods. Therefore, we remove these erroneous bounding boxes from the training dataset of the ADS.

The training process of Faster R-CNN has been carried out using the stochastic gradient descent optimizer with the learning rate 0.01 for 174000 iterations. We decrease the learning rate by a factor of 10 after each 58000 iterations. The batch size in each training iteration of Faster R-CNN has been chosen as 1. In the training process of RetinaNet, the stochastic gradient descent optimizer with the learning rate 0.001 has been employed. We decrease the learning rate by a factor of 10 after each 11600 iterations. The training process of RetinaNet has been continued for 29000 iterations with batches of size 1. After training the two deep image object detectors, we evaluate their performances in terms of the IoU between the estimated and ground truth bounding boxes of the KITTI, BDD100K, and Waymo validation datasets. It should be pointed out that as Faster R-CNN is a two-stage deep image object detector, while RetinaNet is a one-stage detector, we use two different threshold values for their IoU metrics at inference time. Specifically, we use the IoU threshold values of 0.35 and 0.45, respectively, for Faster R-CNN and RetinaNet. The results of these experiments are given in Table 4.6.

It is seen from this table that cleaning the training datasets of ADSs with the proposed MIPE leads to providing deep image object detectors with higher detection performances. For example, it is seen from the results of Table 4.6 that, in the case of the KITTI dataset with 50% error rate, Faster R-CNN and

RetinaNet trained with the datasets cleaned by MIPE can provide IoU values, that is, respectively, 5.39% and 3.49% higher than those obtained by cleaning the dataset with CL [33] (second best-performing data selection method). Since our scheme, MIPE, processes the crucial information associated with the bounding boxes, such as their geometrical interaction with the visual signals, it is able to provide superior IoU metrics for the deep image object detection blocks of ADSs. In other words, the cleaning process of the training datasets of ADSs is performed in a more suitable fashion by the proposed MIPE, compared to that carried out by the other data selection methods, which in turn better suppresses the bounding boxes with incorrect sizes, whose inclusion in the training process leads to a reduction in the performance of deep image object detectors.

Figure 4.2 shows examples of the outputs of Faster R-CNN, when it is trained with the clean KITTI training dataset, KITTI training dataset with 50% error rate, and KITTI training dataset with 50% error rate that is then cleaned by the proposed MIPE. It is seen from this figure that training deep image object detection blocks of ADSs with samples contaminated with erroneous labels leads to predicting very imprecise bounding boxes, and therefore, increases the risk of colliding the vehicle with other objects. On the other

Table 4.6: Comparative study of the proposed MIPE in developing more secure and reliable ADSs.

		Dataset	KITTI [12]				BDD100K [50]				Waymo [41]			
		Err. Rate	*0%**	*25%*	*50%*	*75%*	*0%*	*25%*	*50%*	*75%*	*0%*	25%	*50%*	*75%*
Object Detector	Faster R-CNN [40]	No Data Selection	78.11	70.60	62.69	55.12	80.27	77.43	75.14	71.25	75.71	72.31	70.87	68.73
		SVM	-	72.25	67.47	59.94	-	78.09	75.73	72.19	-	72.61	71.11	69.45
		XGBoost [8]	-	72.67	68.34	63.47	-	78.39	75.87	74.59	-	72.69	71.28	69.59
		S3VM [3]	-	71.06	65.03	64.73	-	77.91	76.15	74.64	-	72.47	70.95	70.11
		VIME [49]	-	73.14	67.71	64.27	-	78.58	76.44	74.72	-	73.11	71.43	70.18
		Ranked Batch AL [6]	-	71.89	69.94	62.59	-	78.72	76.62	74.81	-	73.34	71.67	70.41
		RTDL [13]	-	73.77	69.28	64.26	-	78.64	76.53	74.73	-	73.23	71.54	70.24
		SimiFeat [54]	-	74.47	70.33	69.58	-	78.83	76.69	74.93	-	73.89	71.85	70.63
		CL [33]	-	74.83	70.79	70.52	-	78.92	76.83	74.96	-	74.52	72.04	70.82
		Proposed MIPE	-	77.10	76.18	72.37	-	79.41	78.21	77.08	-	75.22	74.20	72.93
	RetinaNet [29]	No Data Selection	78.91	68.22	62.29	55.97	79.51	73.79	72.04	70.72	75.82	73.65	71.07	68.70
		SVM	-	73.50	64.71	59.22	-	74.11	72.81	70.97	-	73.95	72.34	70.65
		XGBoost [8]	-	73.57	65.14	60.04	-	74.25	72.99	71.02	-	74.12	72.38	70.65
		S3VM [3]	-	70.03	63.67	61.67	-	73.95	72.44	71.08	-	73.80	71.29	70.70
		VIME [49]	-	73.38	64.56	62.45	-	74.89	72.55	71.11	-	74.31	73.24	70.87
		Ranked Batch AL [6]	-	74.67	65.91	58.14	-	75.27	72.78	71.23	-	74.45	73.56	71.23
		RTDL [13]	-	73.84	65.85	60.53	-	75.13	72.74	71.19	-	74.39	73.41	71.15
		SimiFeat [54]	-	75.51	66.15	63.25	-	75.42	72.92	71.28	-	74.67	73.62	71.50
		CL [33]	-	75.77	68.16	63.64	-	76.60	73.04	71.33	-	74.98	73.90	71.88
		Proposed MIPE	-	76.21	71.65	68.20	-	78.53	75.11	73.54	-	75.26	74.31	72.27

The values in red indicate the best performance.

∗ All Clean.

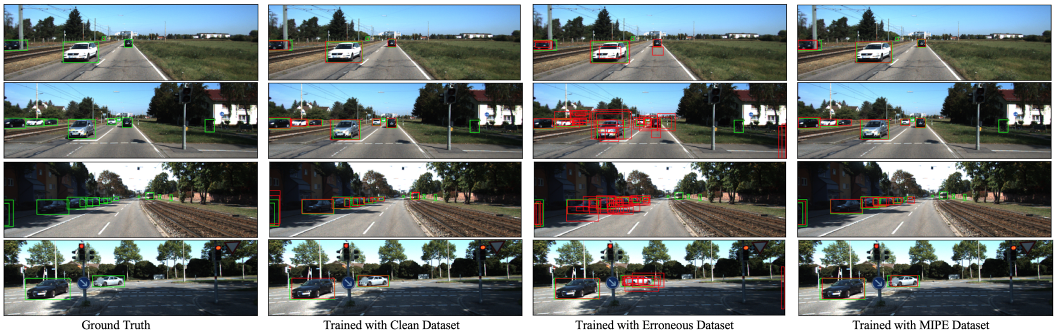

Ground Truth
Trained with Clean Dataset
Trained with Erroneous Dataset
Trained with MIPE Dataset

Figure 4.2: Bounding boxes predicted by Faster R-CNN in the cases where the network is trained with different training datasets. Green bounding boxes denote the ground truth and red bounding boxes represent the predictions obtained by Faster R-CNN.

hand, applying the proposed MIPE to the training datasets of ADSs, to clean them from erroneously labeled samples, results in predicting bounding boxes that are very similar to the ground truth. This could remarkably improve the safety and reliability of ADSs by helping them navigate more securely.

It should be noted that we have targeted two objectives in order to evaluate the effectiveness of the proposed MIPE, i.e., evaluating the effectiveness of the proposed MIPE in identifying the erroneous bounding boxes in the training dataset of ADSs, and evaluating the impact of cleaning the dataset performed by MIPE on the performance of the deep image object detection blocks of ADSs. It can be seen that the former is the task of erroneous bounding box detection, while the latter is the task of image object detection. Each of these two tasks has its specific evaluation metrics. Tables 4.1, 4.2, 4.3, 4.4, and 4.5 contain the results of the erroneous bounding box detection task. Since this task effectively is a binary classification problem, we have used the accuracy metric (in %) of identifying the correct labels for reporting the results of these tables. On the other hand, Table 4.6 includes the results of the image object detection task. In this regard, we have used the intersection over union (IoU) metric for obtaining the results of these tables. It should be noted that the IoU metric is a geometrical measurement that evaluates the performance of the task of image object detection in terms of pixels. Therefore, the performance improvement obtained by the cleaning process of the proposed MIPE has indeed been reported in terms of pixels. Based on the above explanations, by reporting the performances of the two tasks of erroneous bounding box detection and image object detection using the accuracy and IoU metrics, respectively, we can evaluate both how much overall the proposed MIPE is successful in identifying the total number of misannotated bounding boxes, and how much it accomplished in discerning the misannotated bounding boxes with larger units of corrupting operations. If the proposed MIPE cannot detect the bounding boxes with the larger units of corrupting operations, they will be kept in the training process of the deep image object detector, and hence, negatively impact its performance in terms of IoU. In this case, it is legitimate to suppose that the failure of the proposed MIPE in identifying such misannotated bounding boxes results in making the network proportionally estimate the bounding boxes with larger geometrical errors. This error can be effectively measured by the IoU metric, which is itself a metric for measuring the geometrical information.

As we have mentioned above, in order to perform our experiments realistically, for obtaining the results of Table 4.6, we assume that access to annotators is not granted, and therefore, we clean the datasets by removing the erroneous bounding boxes from them. Hence, the performance of the deep image object detectors, especially in the case of the 75% error rate, drops significantly, as in this case, the majority of the bounding boxes are erroneous, and removing them results in producing a training dataset with a small number of bounding boxes. To further investigate the merits of the proposed MIPE in cleaning the training datasets of ADSs, we also carry out an experiment, in which we suppose that the annotators are available for correcting the bounding boxes that are identified as erroneous. In this regard, we first apply the proposed MIPE to the bounding boxes of the dataset and determine their erroneous statuses. We then keep the bounding boxes that are detected clean as they are, and replace those that are identified as erroneous by their clean versions. Table 4.7 gives the performance of Faster R-CNN trained with the KITTI dataset cleaned by the proposed MIPE, when access to the annotators is also granted.

Table 4.7: Effectiveness of cleaning the dataset with the proposed MIPE in tandem with the annotators. All performances are in terms of accuracy (%).

Access to Annotator	*25%* Err. Rate	*50%* Err. Rate	*75%* Err. Rate
No	77.10	76.18	72.37
Yes	77.41	77.10	76.07

The values in red indicate the best performance.

It is seen from the results of this table that, as expected, the proposed MIPE in tandem with the annotators results in designing ADSs that are more secure and reliable, compared to the case where annotators are not available. The results of this table confirm the fact that correcting the misannotated bounding boxes by human experts is a better way to clean the training datasets of ADSs, in comparison to the case of simply discarding such bounding boxes. However, it should be pointed out that the human annotation process is always accompanied by increased costs and the need for further budget.

It is also to be noted that the accuracies of the deep image object detection networks reported in Table 4.6 for the case of the clean (accurate) datasets are obtained when such networks are trained with the original datasets, in which no corrupting operation is synthetically applied to the bounding boxes. For this case, we have not employed our proposed erroneous bounding box detection scheme, MIPE, to clean the training datasets of the deep image object detectors. To investigate the impact of the proposed MIPE in identifying low-quality bounding boxes, we now perform a new experiment, in which the proposed MIPE is applied to the original datasets of the deep image object detectors for identifying the bounding boxes, whose qualities are not as high as expected. The performances of the two deep image object detection networks, Faster R-CNN and RetinaNet, trained with the original datasets of KITTI, BDD100K, and Waymo that are cleaned by the proposed MIPE, are given in Table 4.8.

It is seen from this table that identifying low-quality bounding boxes in the original datasets of ADSs and removing them from the training process undoubtedly has a positive impact on enhancing the performance of the task of image object detection.

Table 4.8: Effectiveness of MIPE on cleaning the original training datasets of ADSs. All performances are in terms of accuracy (%).

Dataset	Object Detector	No Data Selection (All Clean)	Proposed MIPE
KITTI [12]	Faster R-CNN [40]	78.11	79.47
	RetinaNet [29]	78.91	79.85
BDD100K [50]	Faster R-CNN [40]	80.27	81.61
	RetinaNet [29]	79.51	80.72
Waymo [41]	Faster R-CNN [40]	75.71	77.84
	RetinaNet [29]	75.82	77.93

The values in red indicate the best performance.

Table 4.9: Effectiveness of MIPE in cleaning the dataset of domains other than ADSs. All performances are in terms of accuracy (%).

Object Detector	PASCAL VOC Err. Rate	No Data Selection	Proposed MIPE
Faster R-CNN [40]	*0%* (All Clean)	70.42	-
	25%	67.23	67.71
	50%	65.40	66.66
	75%	60.89	62.11
RetinaNet [29]	*0%* (All Clean)	70.84	-
	25%	68.73	69.57
	50%	66.76	68.53
	75%	63.71	64.65

The values in red indicate the best performance.

4.6 Effectiveness of MIPE in Other Domains

The effectiveness of the proposed MIPE in developing secure and reliable ADSs using deep learning-based image object detection blocks is confirmed in the previous Section. To investigate whether the proposed MIPE is also effective for the cleaning process of the datasets of the deep image object detectors used in domains other than autonomous driving systems, we perform another experiment. Specifically, we train Faster R-CNN and RetinaNet with the training images of the PASCAL VOC [10] dataset, which contains visual signals from generic images (including *person*, *animal*, *vehicle*, etc.), in the following 3 cases: training with the clean dataset, training with the dataset with erroneously labeled samples, and training with the dataset cleaned by the proposed MIPE. The descriptions of these 3 cases are thoroughly explained in the previous Section. The results of these experiments on the 17125 images with cross-validation are given in Table 4.9.

Table 4.10: Execution time of the proposed MIPE for the task of erroneous bounding detection.

Method	Execution time per bounding box
SVM	0.6619 (ms)
XGBoost [8]	0.8462 (ms)
S3VM [3]	190.1983 (ms)
VIME [49]	289.3744 (ms)
Ranked Batch AL [6]	N/A
RTDL [13]	37.0182 (ms)
SimiFeat [54]	128.7240 (ms)
CL [33]	0.5683 (ms)
Proposed Scheme	369.6431 (ms)

It is seen from this table that the proposed MIPE is able to enhance the performance of Faster R-CNN and RetinaNet by 0.48% & 0.84%, 1.26% & 1.77%, and 1.22% & 0.94%, in the cases of the 25%, 50%, and 75% error rates, respectively. Hence, it can be concluded from the results of Table 4.9 that the use of the proposed MIPE is not restricted to only ADSs, and our method can be generally employed for cleaning various image object detection datasets.

4.7 Time Complexity of MIPE

It is seen from the previous sections that the proposed MIPE is indeed a very effective method for cleaning the training datasets of ADSs employing deep learning-based image object detection blocks. We now investigate the execution time of the proposed MIPE to evaluate its complexity. It should be noted that since the proposed MIPE is a multi-modal information processing system that is specifically developed for the task of erroneous bounding box detection, we expect that its time complexity is higher than those of the other *uni-modal* data selection methods, which merely process the spatial coordinates of the bounding boxes. Therefore, our main objective in this Section is to investigate whether the execution time of the proposed MIPE is reasonable enough for real-life applications. Table 4.10 gives the execution time of the proposed MIPE and the other data selection methods used in our comparison for determining the erroneous status of a single bounding box.

It is seen from the results of this table that even though the proposed MIPE requires a larger execution time for obtaining the erroneous status of a given bounding box compared to the other methods, its time complexity is still acceptable in many real-world situations. Since the proposed MIPE is only used for cleaning the training datasets of ADSs prior to their training stages, its time complexity does not have any impact on the execution speed of ADSs.

Chapter 5

Conclusion & Future Work

5.1 Conclusion

Developing secure and reliable ADSs is one of the crucial tasks in the community of intelligent vehicles, since these systems must be applied in safety-critical scenarios. The reliability of ADSs is very much dependent on their model components and the datasets used for their training process. In several realistic situations, the models of ADSs are meticulously optimized to work well under various conditions. However, often not much attention has been paid to ensure that the training datasets of ADSs contain samples with clean and correct labels. The existence of training samples that are labeled erroneously in the training datasets of ADSs could lead to a serious impact on both their performance and reliability. Given these points, in this work, we have developed a high-performance erroneous bounding box detection method for cleaning the datasets of ADSs employing deep image object detectors. Specifically, we have focused on identifying the bounding boxes in the dataset that are erroneously annotated. In this regard, we have incorporated two techniques, namely, multi-modal information processing and confident learning, in order to detect the erroneously labeled bounding boxes in the dataset accurately prior to the training process of ADSs. The results of various experiments have shown the effectiveness of different ideas used in the development of the proposed scheme. It has been shown that the proposed scheme significantly outperforms the other state-of-the-art data selection methods in cleaning the datasets of ADSs employing deep image object detectors. Finally, it has been shown that this cleaning process, in turn, leads to the development of more secure and reliable ADSs that employ deep image object detectors.

5.2 Future Work

As mentioned in Section 2.3, even though there exists a few schemes in the literature for detecting samples with erroneous labels in the training datasets of DNNs, to the best of our knowledge, an explicit algorithm for identifying the erroneously labeled bounding boxes of ADSs prior to our work has not been developed. The following is a list of potential research directions that can be further investigated succeeding this work:

- Datasets: In this work, we aimed to evaluate our dataset cleaning scheme on three popular ADS datasets: KITTI [12] , BDD100K [50], and Waymo [41]. The performance of our scheme can be further assessed under other ADS datasets, and even extended to domains other than ADSs (e.g., PASCAL VOC [10], which has been partly investigated in this work).

- Object detectors: Regarding image object detectors, our primary focus was on two, namely, Faster R-CNN [40] and RetinaNet [29]. This work can be further expanded to include other image object detectors, for example, YOLO [38] architectures.

- Data domain: To tackle the task of erroneous label detection, we focused our attention on the image data available in the ADS community. However, ADSs operate in other data domains as well, such as point cloud data, which is attributed to LiDAR. The work carried out here can be studied in other data domains.

- Algorithm: As mentioned in Chapter 3, our scheme was centered around two key parts: **multi-modal information processing** and **confident learning**. Considering the former, fusing can be further extended to other data domains, enabling a potentially more viable scheme for cleaning the datasets of ADSs. Regarding the latter, it can be substituted, should there be a more impactful module for estimating the uncertainty of the erroneous status of the labels in the upcoming future.

References

[1] S. Aradi, “Survey of deep reinforcement learning for motion planning of autonomous vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, 2020.

[2] P. Baldi and P. J. Sadowski, “Understanding dropout,” *Advances Neural Inf. Process. Syst.*, vol. 26, 2013.

[3] K. Bennett and A. Demiriz, “Semi-supervised support vector machines,” *Advances Neural Inf. Process. Syst.*, vol. 11, 1998.

[4] D. Cao, X. Wang, L. Li, *et al.*, “Future directions of intelligent vehicles: Potentials, possibilities, and perspectives,” *IEEE Trans. Intell. Vehicles*, vol. 7, no. 1, pp. 7–10, 2022.

[5] J. Cao, H. Cholakkal, R. M. Anwer, F. S. Khan, Y. Pang, and L. Shao, “D2det: Towards high quality object detection and instance segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 485–11 494.

[6] T. N. Cardoso, R. M. Silva, S. Canuto, M. M. Moro, and M. A. Gonçalves, “Ranked batch-mode active learning,” *Inf. Sciences*, vol. 379, pp. 313–337, 2017.

[7] L. Chen, Y. Li, C. Huang, *et al.*, “Milestones in autonomous driving and intelligent vehicles: Survey of surveys,” *IEEE Trans. Intell. Vehicles*, 2022.

[8] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[9] F. Ding, K. Yu, Z. Gu, X. Li, and Y. Shi, “Perceptual enhancement for autonomous vehicles: Restoring visually degraded images for context prediction via adversarial training,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9430–9441, 2021.

[10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, 2010.

[11] F. M. Favarò, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, "Examining accident reports involving autonomous vehicles in california," *PLoS one*, vol. 12, no. 9, e0184952, 2017.

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[13] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," *Advances Neural Inf. Process. Syst.*, vol. 34, pp. 18 932–18 943, 2021.

[14] B. Han, Q. Yao, X. Yu, *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Advances Neural Inf. Process. Syst.*, vol. 31, 2018.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[16] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[17] J. Huang, L. Qu, R. Jia, and B. Zhao, "O2u-net: A simple noisy label detection approach for deep neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3326–3334.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. Mach. Learn.*, PMLR, 2015, pp. 448–456.

[19] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 2304–2313.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Mach. Learn. Proc.* Elsevier, 1994, pp. 148–156.

[23] J. Li, C. Xiong, R. Socher, and S. Hoi, "Towards noise-resistant object detection with noisy annotations," *arXiv preprint arXiv:2003.01285*, 2020.

[24] L. Li, W.-L. Huang, Y. Liu, N.-N. Zheng, and F.-Y. Wang, "Intelligence testing for autonomous vehicles: A new approach," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 2, pp. 158–166, 2016.

[25] X. Li, Y. Tian, P. Ye, H. Duan, and F.-Y. Wang, "A novel scenarios engineering methodology for foundation models in metaverse," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 4, pp. 2148–2159, 2022.

[26] X. Li, K. Wang, X. Gu, F. Deng, and F.-Y. Wang, "Paralleleye pipeline: An effective method to synthesize images for improving the visual intelligence of intelligent vehicles," *IEEE Trans. Syst., Man, Cybern., Syst.*, 2023.

[27] X. Li, P. Ye, J. Li, Z. Liu, L. Cao, and F.-Y. Wang, "From features engineering to scenarios engineering for trustworthy AI: I&I, C&C, and V&V," *IEEE Intell. Syst.*, vol. 37, no. 4, pp. 18–26, 2022.

[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[29] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[30] J. Liu, H. Wang, L. Peng, Z. Cao, D. Yang, and J. Li, "Pnnuad: Perception neural networks uncertainty aware decision-making for autonomous vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24 355–24 368, 2022.

[31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Int. Conf. Mach. Learn.*, 2010.

[32] J. Nie, J. Yan, H. Yin, L. Ren, and Q. Meng, "A multimodality fusion deep neural network and safety test strategy for intelligent vehicles," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 2, pp. 310–322, 2020.

[33] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *J. Artif. Intell. Res.*, vol. 70, pp. 1373–1411, 2021.

[34] npr, *Nearly 400 car crashes in 11 months involved automated tech, companies tell regulators*, `https://www.npr.org/2022/06/15/1105252793/nearly-400-car-crashes-in-11-months-involved-automated-tech-companies-tell-regul`, Jun. 2022.

[35] PolicyAdvice, *25 Astonishing Self-Driving Car Statistics For 2023*, `https://policyadvice.net/insurance/insights/self-driving-car-statistics/`, Last modified: Mar 23, 2023, Mar. 2023.

[36] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan, "Estimating training data influence by tracing gradient descent," *Advances Neural Inf. Process. Syst.*, vol. 33, pp. 19 920–19 930, 2020.

[37] B. Ranft and C. Stiller, "The role of machine vision for intelligent vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 8–19, 2016.

[38] 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*, 2016, pp. 779–788.

[39] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 4334–4343.

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances Neural Inf. Process. Syst.*, vol. 28, 2015.

[41] P. Sun, H. Kretzschmar, X. Dotiwalla, *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.

[42] TheVerge, *Waymo's driverless cars were involved in two crashes and 18 'minor contact events' over 1 million miles*, `https://www.theverge.com/2023/2/28/23617278/waymo-self-driving-driverless-crashes-av`, Feb. 2023.

[43] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.

[44] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances Neural Inf. Process. Syst.*, vol. 30, 2017.

[45] F.-Y. Wang, R. Song, R. Zhou, *et al.*, "Verification and validation of intelligent vehicles: Objectives and efforts from china," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 2, pp. 164–169, 2022.

[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[47] Y. Xu, L. Zhu, Y. Yang, and F. Wu, "Training robust object detectors from noisy category labels and imprecise bounding boxes," *IEEE Trans. Image Process.*, vol. 30, pp. 5782–5792, 2021.

[48] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, and Q. Dai, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 284–295, 2017.

[49] J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar, "Vime: Extending the success of self-and semi-supervised learning to tabular domain," *Advances Neural Inf. Process. Syst.*, vol. 33, pp. 11 033–11 043, 2020.

[50] F. Yu, H. Chen, X. Wang, *et al.*, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[51] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, “Varifocalnet: An iou-aware dense object detector,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8514–8523.

[52] T. Zhou, S. Wang, and J. Bilmes, “Robust curriculum learning: From clean label detection to noisy label self-correction,” in *Int. Conf. Learn. Represent.*, 2020.

[53] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, “Overview of environment perception for intelligent vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2584–2601, 2017.

[54] Z. Zhu, Z. Dong, and Y. Liu, “Detecting corrupted labels without training a model to predict,” in *Int. Conf. Mach. Learn.*, PMLR, 2022, pp. 27 412–27 427.