This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2967788, IEEE Internet of Things Journal

Blockchain-Enabled Cross-Domain Object Detection for Autonomous Driving: A Model Sharing Approach

Xiantao Jiang, F. Richard Yu, *Fellow, IEEE,* Tian Song, Zhaowei Ma, Yanxing Song, and Daqi Zhu

***Abstract*—Object detection for autonomous driving is a huge challenge in the cross-domain adaptation scenario, especially for the time- and resource-consuming task. Distributed deep learning (DDL) has demonstrated a considerably good balance between efficiency and computation complexity. However, the reliability of distributed deep learning is low. Moreover, the cost of training data and model is not priced well. In this work, a novel blockchain-enabled model sharing approach is proposed to improve the performance of object detection with cross-domain adaptation for autonomous driving systems. Based on blockchain and mobile edge computing (MEC) technology, a domain-adaptive *you only look once* (YOLOv2) model is trained across nodes, which can reduce significantly the domain discrepancy for different object categories. Furthermore, smart contracts are developed to perform data storage and model sharing tasks efficiently. The reliability of model sharing is ensured with blockchain consensus. We evaluate the proposed method under public datasets. The simulation results demonstrate that the efficiency and reliability of the proposed approach are better than the reference model.**

***Index Terms*—Autonomous Driving, Object Detection, Domain Adaptation, Blockchain, Model Sharing.**

I. Introduction

IN the Internet of Vehicles (IoV) system, autonomous cars are capable of sensing the environment and planning without human intervention, which is made possible by artificial intelligence (AI). Moreover, autonomous vehicles (AVs) combined a variety of sensors, such as cameras, radar, Lidar, and GPS, can share information with other vehicles to increase the reliability of its autonomous driving systems further. Object detection is to detect instances of semantic objects of a certain class, which is a basic task of autonomous driving. Recently, deep learning (DL) has been extensively used to improve the performance of object detection [1].

However, object detection for autonomous driving still faces challenges. Firstly, models cannot be generalized well to new domains, and the efficiency of cross-domain object detection with unsupervised learning is low [2]. For autonomous driving, external environment such as different background, image quality, and illumination can all lead to the domain shift. The training datasets have different quality when set by different in-vehicle camera parameters. There is some difference of object appearance between different cities. Moreover, most of the datasets are collected under the dry weather condition and daytime, while the autonomous driving system needs to be adapted to the bad weather condition and night time. Therefore, cross-domain object detection has attracted great attention [3]. In [4], the squeezeDet model training of object detection was accelerated by transfer learning method, and additionally a new empirical method was used to optimize the hyper-parameter. Moreover, Chen *et al.* proposed a domain-adaptive faster region based convolutional neural network (R-CNN) approach to improve the cross-domain robustness of object detection [5]. In Chen's work, two levels of domain shifts that include image-level and instance-level were designed to reduce the domain discrepancy. Inoue *et al.* presented a cross-domain weakly supervised object detection task, and a two-step progressive domain-adaptive method was developed for model training [6]. However, previous works mainly focus on data-driven transfer learning, while the pattern of model sharing for transfer adaptation learning is not perfect.

Secondly, with regards of the object, the local computation power and storage of the autonomous driving system are limited. With the increase of training datasets, the network model of object detection is getting more and more deep and complex. Moreover, when the computation complexity of model training is higher, the training phase of the model requires more memory. Thus, deeper model training is a time- and resource-consuming task, which needs high computational power along with efficient data processing and storage. Distributed training of the neural network is an effective way to balance the computation complexity and learning efficiency [7]. In [8], a survey about parallel and distributed deep learning was reported, and the strategy of achieving the trade-off between concurrency and accuracy was discussed. There are two common approaches of distributed deep training: model parallelism and data parallelism [9]. However, the reliability of distributed deep learning is low.

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61701297, in part by the China Postdoctoral Science Foundation under Grant No. 2018M641982, and in part by the Mitacs. (Corresponding author: F. Richard Yu)

Xiantao Jiang is with Department of Information Engineering, Shanghai Maritime University, NO.1550, Haigang Ave., Shanghai 201306, China (e-mail: xtjiang@shmtu.edu.cn).

F. Richard Yu and Zhaowei Ma are with Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: richard.yu@carleton.ca; zhaoweima@cmail.carleton.ca).

Tian Song is with Department of Electrical and Electronics Engineering, Tokushima University, 2-24, Shinkura-cho, Tokushima, Japan (e-mail: tiansong@ee.tokushima-u.ac.jp).

Yanxing Song is with Department of Information, Beijing Wuzi University, Beijing, 101149, China. (e-mail: songyanxing@bwu.edu.cn).

Daqi Zhu is with Department of Information Engineering, Shanghai Maritime University, NO.1550, Haigang Ave., Shanghai, 201306, China (e-mail: dqzhu@shmtu.edu.cn).

Therefore, in order to address the above issues, a novel object detection framework for autonomous driving is proposed based on blockchain and mobile edge computing (MEC) technologies, which provide distributed, secure reliable services at the edge of the network [10]–[12]. Firstly, MEC is a network architecture concept that enables cloud computing capabilities and service environment to work at edge network. MEC can provide low delay and real-time computation capabilities, which is suitable for computations-intensive services such as model training [13]. Secondly, blockchain is a decentralized system, which is a potential technology for secure model sharing. As the P2P network, the information on-blockchain is impossible to be modified without network consensus [14]. The advantages of applying MEC and blockchian to deep learning are: (1) The integrated MEC and deep learning give the birth of edge intelligence [15]. For one thing, deep learning can exploit the potential for data produced at the network edge. For another, MEC can prosper deep learning with richer data and emerging applications. (2) The integration of blockchain and deep learning is inevitable [16]. On one hand, blockchain can provide trust, privacy to deep learning. On the other, deep learning can bring security and scalability for blockchain-based learning system. Therefore, the data, model, and computing resources of deep learning can be powered by blockchain-enabled decentralized marketplaces and coordination platforms. Different from the traditional object detection methods, a domain-adaptive based object detection approach is used to train the model with distributed learning. Moreover, the blockchain is used to manage the model sharing across MEC nodes. The contributions to our work are summarized as follows:

- We present a novel blockchain-enabled distributed deep learning (DDL) framework to improve the performance of object detection for autonomous driving. Blockchain can reduce failure, at the same time it can increase security, effectiveness of the distributed deep learning system. MEC nodes are responsible for model sharing and consensus. To our best knowledge, this work is the first article to propose a blockchain-enabled object detection approach.
- In this framework, a domain-adaptive *you only look once* (YOLOv2) model is proposed to train the model with distributed transfer learning. A domain classifier is designed with adversarial training to reduce the domain discrepancy. The training data collected by vehicle users is uploaded to the nearby MEC node. The proposed object detection model is trained with distributed learning approach, and blockchain rewards are returned to the miner vehicles.
- The proposed approach is evaluated under public datasets. Simulation results demonstrate that the efficiency and reliability of the proposed object detector are better than the reference model in the domain shift scene.

The rest of this paper is organized as follows. The related works are reviewed in Section II. In Section III, we introduce the technical background. In Section IV, a novel object detection approach is proposed for driving systems based on blockchain and MEC. Simulation results are discussed in Section V. In Section VI, a conclusion of this work is given.

II. Related Works

In this section, we present the previous works about deep learning based object detection, domain-adaptive deep learning, and blockchain enabled vehicle data sharing.

A. Deep Learning based Object Detection

Since the competition of AlexNet, CNN has been used extensively to improve the accuracy of image classification, and DL-based object detection method has attracted a wide range of attention from numerous researchers. There are two main types of DL-based methods for object detection: two-stage method and single stage method. Firstly, the two-stage approach is one that is based on R-CNN that combines region proposal and CNN together. Girshick *et al.* proposed the R-CNN model [17]. R-CNN develops the selective search to obtain region proposal, and each of their features are computed by using CNN. After normalization, the features of each region proposal are computed by using CNN. At last, the support vector machine (SVM) classifier is used to predict each region. In regard to the disadvantage of R-CNN, He *et al.* developed spatial pyramid pooling network (SPP-Net) to extract the feature all at once [18], and the complexity of extracting feature was reduced. However, SPP-Net brings the consumption of storage space significantly. Girshick *et al.* proposed a fast region-based convolutional network (Fast R-CNN) method [19]. Based on SPP-Net, the SPP layer is simplified as the region of interest (ROI) pooling layer, and the output of the whole connection layer is decomposed by singular value decomposition (SVD). In Ren's work [20], a faster R-CNN was used to achieve end-to-end object detection by using region proposal networks (RPN). Moreover, some other works were proposed to improve the detection efficiency [21]. However, the real-time performance of R-CNN based approaches is relatively lower.

Secondly, the single stage approach is an approach that transforms object detection into a regression problem. Redmon *et al.* presented the YOLO model with a single network to improve the speed of object detection [22]. YOLO is good for real-time processing. Liu *et al.* proposed a single-shot detector (SSD) and models to balance the detection efficiency and detection speed [23]. However, SDD is still not ideal for a small object. Therefore, a deconvolutional single shot detector (DSSD) model is proposed to improve performance especially for small objects, which can improve the previous SSD with deconvolutional path [24]. In Redmon's latest work [25], [26], YOLOv2 and YOLOv3 were used to achieve a better trade-off between the detection efficiency and detection speed. Moreover, some other single stage works were proposed to improve the performance of object detection [27], [28]. However, those above works ignored the domain adaptation issue for object detection in the wild. Moreover, the pre-training model is unable to be migrated to other domain. There is a difference between the source domain and the target domain, and the training model is probably not the best model for the target domain.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2967788, IEEE Internet of Things Journal

B. Domain-Adaptive Deep Learning

Domain adaptation is a field associated with machine learning and transfer learning. For the unsupervised domain adaptation (UDA), model learning is from a source domain, while perform model on a different target domain. The data labels in the source domain are available, while the data labels in the target domain are not available [29]. Focusing on deep domain adaptation approaches, there are two main schemes for aligning feature representations namely (1) the maximum mean discrepancy (MMD) based methods and (2) the domain classifier network-based methods. On one hand, MMD is used to measure the distance between source and target domain distributions in a regenerative Hilbert space. Gong *et al.* proposed an approach to automatically discover latent domains by using maximum distinctiveness and maximum learning ability [30]. Haeusser *et al.* presented an associative domain adaptation with neural networks [31], in which the discrepancy can be reduced by an associative loss. Tzeng *et al.* proposed a new CNN architecture by using an adaptation layer along with MMD-based domain confusion loss [32]. Long *et al.* presented a novel deep adaptation network (DAN) framework [33], in which the multiple kernels maximum mean discrepancy (MK-MMD) was used to domain discrepancy further.

On the other hand, the domain classifier network is used to find common feature space between the source and target domains. In Ganin's work [34], a new approach was presented to domain adaptation in deep architecture, and the model can be trained by using standard back propagation. Ganin *et al.* proposed a new approach to domain adaptation by using domain-adversarial neural networks (DANN) [35], [36]. The proposed DANN can suit to the context of domain adaptation. Moreover, Ghifary *et al.* designed a deep reconstruction classification network (DRCN) for visual object recognition [37]. It is well noted that most of the above works are for image classification, while this work focuses on the object detection task, and it is a great challenge to inference the object location and category with cross-domain adaptation.

C. Blockchain Enabled Vehicle Data Sharing

Blockchain works as a distributed ledger that uses several servers in order to track the networks through which cryptocurrencies are shared. There are three major advantages for blockchain, security, transparency, and reliability [38], [39]. Moreover, blockchain has been introduced in terms of legitimate concern on vehicles data sharing, which can improve further can improve further autonomous driving function. The ordinary vehicles can gain information from other miner vehicles with built-in sensors such as cameras, Radar and Lidar. However, for vehicle data sharing, security is a great challenge during communication. Therefore, the marriage of blockchain and automotive technologies provides exciting solutions to vehicle data sharing [40]–[42].

Recently, the way of blockchain enabled vehicle data sharing has attracted the attention of researchers. Kaiser *et al.* developed an open vehicle data ecosystem with a privacy-preserving way [43], and the blockchain technology is used to protect the privacy of both the driver and vehicle owner. Sharma *et al.* proposed a blockchain based vehicular data management (B2VDM) framework [44], and the access control and load distribution can achieve a better trade-off with the proposed approach. Singh *et al.* presented a reward based vehicle data sharing framework with blockchain technology [45]. In Sharma's work [46], a distributed blockchain based vehicle network (Block-VN) framework was proposed to build the distributed data management system, and vehicles can work together to share their resources to provide the value-added services. Moreover, Singh *et al.* used blockchain technology to support secure and trust in-vehicle communication [47]. However, the above methods focus on safe and reliable data sharing, while the way of blockchain-enabled model sharing is not discussed in the previous works.

In this paper, an efficient blockchain-enabled transfer learning approach is proposed to improve the performance of object detector for autonomous driving systems. Based on the base detector YOLOv2, a domain-adaptive model is trained with distributed learning. With cross-domain adaptation, a domain classifier is used to reduce the domain discrepancy. Different from previous works, the trained model can be shared across MEC nodes.

III. Technical Background

In this section, an overview of YOLO based object detection and domain measure in domain adaptation are described.

A. YOLO based Object Detection

YOLO is a state-of-the-art, real-time object detection framework, which can apply a signal neural network to the full images [22]. YOLO network divides the image into different regions and further predicts the bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

One of the advantages of YOLO is that it looks at the whole image during the test time, and its predictions are informed by the global context in the image. Unlike R-CNN, YOLO makes predictions with a single network, which makes this algorithm extremely fast, over 1000x faster than R-CNN and 100x faster than Fast R-CNN. YOLO network model has 24 convolutional layers and two fully connected layers. Alternating 1×1 convolutional layers reduce the features space form preceding layers. YOLO divides the input image into $S \times S$ grids. Each grid can predict B bounding box (BBOX), confidence score, and C class probabilities.

In order to improve the recall rate and locating ability, five import improvements were proposed for YOLOv2 [25]: (1) *batch normalization*: Batch normalization can accelerate the convergence and improve the performance in mean average precision; (2) *high resolution classifier*: In YOLOv2 model, the full 448×448 resolution is used to fine-tune the classifier for 10 epochs on ImageNet, and it can increase the mean average precision; (3) *anchor boxes*: The anchor boxes are used to predict bounding boxes, and it can improve the performance in recall means with little the mean average precision loss; (4) *fine-grained feature*: A pass-through layer is added to connect the higher resolution features, hence, it increases the model performance; (5) *multi-scale training*: YOLOv2 can adapt images with different sizes, and the network can check features at different resolutions. YOLOv2 can achieve a better speed and accuracy. Compared with YOLOv2, YOLOv3 makes a bunch of little design changes [26], while YOLOv3 network model of is complicated. Therefore, YOLOv2 (YOLO9000) is used to train the model in this work.

B. Domain Measure in Domain Adaptation

Let us define two domain distributions D_S and D_T, and a hypothesis class $\mathcal{H}$. The feature vector is denoted by $\mathbf{x}$, and the labeled source sample and unlabeled target samples are denoted by (x_i^s, y_i^s), and x_i^t, respectively. A domain classifier $d : x \to \{0, 1\}$ can be built with a low target risk. The $\mathcal{H}$-divergence measure the ability of a hypothesis class $\mathcal{H}$ to discriminate between source D_S and target D_T distributions [5], and the domain distance is defined as follow:

$$disc_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) = 2\left(1 - \min_{d\in\mathcal{H}}\left(\mathrm{err}_{\mathcal{D}_S}(d(\mathbf{x})) + \mathrm{err}_{\mathcal{D}_T}(d(\mathbf{x}))\right)\right) \quad (1)$$

where $err_{\mathcal{D}_S}$ and $err_{\mathcal{D}_T}$ are the prediction errors of $d(\mathbf{x})$ on source and target domain samples, respectively.

In convolutional neural network, f denotes the feature extractor, which is enforced to output feature vector $\mathbf{x}$. Therefore, the minimum domain distance can be rewritten as follow:

$$\min_f disc_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) \Leftrightarrow \max_f \min_{d\in\mathcal{H}}\left\{\mathrm{err}_{\mathcal{D}_S}(d(\mathbf{x})) + \mathrm{err}_{\mathcal{D}_T}(d(\mathbf{x}))\right\} \quad (2)$$

Ganin *et al.* developed a gradient reverse layer (GRL) to optimize the minimum domain distance by adversarial training manner [32]. Moreover, the GRL can be integrated into the deep neural networks in the case of unsupervised domain adaptation.

C. Delegated Proof-of-Stake (DPoS) Consensus Algorithm

The core blockchain technology allows nodes in the system to compete in record and keep its criterion, which is known as the consensus mechanisms. So far, there are three widely accepted consensus mechanisms: Proof-of-Work (PoW) [48], Proof-of-Stake(PoS) [49], and Delegated Proof-of-Stake (DPoS) [50].

DPoS means the system chooses a representative from the holder that has the coins and let them do the record. The DPoS consensus algorithm consists of two parts: electing a group of block producers and scheduling production. The election process makes sure that stakeholders are ultimately in control. In comparison with Pow and PoS, the advantages of DPoS are less energy waste and faster speed. In this work, the DPoS consensus algorithm is used for the model sharing.

IV. Proposed Scheme

In this section, the system framework and the domain adaptation model of object detection are described. The main notions and abbreviations in this article are listed in Table I and Table II, respectively.

TABLE I: The main notations.

Notation	Explanation
D_S	Source domain
D_T	Target domain
(x_i^s, y_i^s)	A labeled source sample
x_i^t	An unlabeled target sample
$\mathbf{x}$	The feature vector
f	The feature extractor
y	The label predictor
d	The domain classifier
$\mathcal{H}$	A hypothesis class
$disc_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)$	The domain distance
$err_{\mathcal{D}_S}$	The prediction errors of $d(x)$ on source sample
$err_{\mathcal{D}_T}$	The prediction errors of $d(x)$ on target sample
$\mathcal{L}_{yolo}$	The training loss function of YOLOv2 model
$\mathcal{L}_{da}$	The training loss function of DA-YOLO model

TABLE II: The main abbreviations.

Abbreviation	Explanation
AV	Autonomous Vehicles
CNN	Convolutional Neural Networks
DA	Domain Adaptation
DANN	Domain-adversarial Neural Networks
DA-YOLO	Domain-Adaptive YOLO
DL	Deep Learning
DLG	Deep Learning Group
DPoS	Delegated Proof of Stake
DSSD	Deconvolutional Single Shot Detector
GRL	Gradient Reverse Layer
MEC	Mobile Edge Computing
PoS	Proof-of-Stake
PoW	Proof-of-Work
R-CNN	Regional Convolutional Neural Networks
SSD	Single Shot MultiBox Detector
YOLO	You Only Look Once

A. System Framework

Domain shift refers to the existing difference between the two distributions that will lead to unreliable predictions. Fig. 1 shows the typical dataset shifts for autonomous driving. Domain adaptation approaches are used to reduce the harmful effects of domain shift. The main task of this work is to train an efficient object detection model with the unsupervised domain adaptation. The target data collected by the ordinary vehicle is referred as the target domain D_T, and the source data collected by the miner vehicles is referred as the source domain D_S. In the source domain, the location and category are available, however, in the target domain, the image labels are not available. Therefore, based on distributed deep learning (data parallelism), the domain-adaptive YOLO (DA-YOLO) model is proposed to improve the efficiency of the object detector. Fig. 2 illustrates the blockchain-enabled object detection framework of autonomous driving systems. The main workflow of the blockchain-based object detection is as follow:

(1) The source and target data are uploaded to the nearby MEC nodes. The ordinary vehicle sends an object detection task request that can be regarded as transaction.

(2) It is to set the smart contract and form a deep learning group (DLG) for the object detection task across MEC nodes.

(3) Based on the distributed deep learning with domain adaptation YOLO (DA-YOLO) model, it is to train the feature extractor and class predictor on the source data, and train the feature extractor and domain classifier on the source and target data.

(4) When the training model converges, the address of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2967788, IEEE Internet of Things Journal

(a) BDD100K (b) Foggy Cityscapes

(c) KITTI

Fig. 1: The typical dataset shifts: BDD100K [51], Foggy Cityscapes [52], and KITTI [53].

model cached is returned. Feature extractor and class predictor are used at the test time.

Different from the previous works, a set of MEC nodes are used to process the task of DA-YOLO model with distributed data parallelism pattern. The MEC nodes can serve as storage data and model sharing. Moreover, in order to achieve the secure and reliable storage and sharing model, the MEC nodes are responsible for both model sharing and consensus performing.

Fig. 3 shows a pattern of the model sharing process. Each MEC node within the DLG is used to train the DA-YOLO model, and the model parameters can be shared across MEC nodes by using blockchain technology. In addition, the fine-tuning method is used in the model training process [34].

B. Domain-Adaptive YOLO for Object Detection

Domain adaptation is a particular case of transfer learning, which trains the model by using labeled data in source domains and inferences the task in the target domain. The main domain adaptation methods can be categorized into five classes: instance adaptation, feature adaptation, classifier adaptation, deep network adaptation and adversarial adaptation [3]. (1) Instance adaptation method trains the model on weighted source samples. (2) Feature adaptation method can achieve the feature imitation through the competition between the classifier and the discriminator. (3) Classifier adaptation method can learn the generic classifier by train on the source samples and few labeled target samples. (4) Deep network adaptation method can update the network structure by adapting or adding network weights. (5) Adversarial adaptation method is a potential technology, which can generate pixel level target sample or feature level target representations. Different from other methods, adversarial adaptation method can minimize the distance between the source and target distribution. In this work, the adversarial-based approach is used to train the model.

There are three approaches of adversarial adaptation, namely gradient reversal-based, minimax optimization, and GANs-Based [54], [55]. In this work, the gradient reversal-based adversarial strategy is used for domain adaptation. The network structure of the domain-adaptive YOLO is shown in Fig. 4. The DA-YOLO network consists of three parts: feature extractor f, label predictor y, and domain classifier d. It is noted that the standard YOLO model includes feature extractor and label predictor. Moreover, the GRL is used to connect the feature extractor and domain classifier, and no parameters are associated with GRL. The details about YOLO and GRL are described in Section III.

For the feature extractor f, the activation function is Sigmoid. $\mathbf{x}$ is input with feature vector and the network parameters of all layers are $(\mathbf{W}, \mathbf{b})$. Then the output of feature extractor $h(\mathbf{x})$ is defined as:

$$h(\mathbf{x}) = sigm(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{3}$$

with

$$sigm(z) = \frac{1}{1+\exp(-z)}$$

For the label predictor y, the activation function is Softmax. The network parameters of all layers are $(\mathbf{U}, \mathbf{c})$, and the output of label predictor $g(\mathbf{x})$ is given as:

$$g(\mathbf{x}) = softmax(\mathbf{U}h(\mathbf{x}) + \mathbf{c}) \tag{4}$$

with

$$softmax(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, for\ j = 1, 2, , K$$

The training loss function of YOLO consists of three parts: the classification loss, localization loss and confidence loss, respectively. For the input source domain data $D_S = \{(x_i^s, y_i^s)\}_{i=1}^m$, The loss function of YOLO is calculated as:

$$\mathcal{L}_{yolo} = \sum_{i=1}^{m} \mathcal{L}^y(g(x_i^s), y_i^s) \tag{5}$$

For the domain classifier d, the activation function is Sigmoid. The network parameters of all layers are $(\mathbf{V}, \mathbf{d})$, and the output of domain classifier $o(\mathbf{x})$ is defined as:

$$o(\mathbf{x}) = sigm(\mathbf{V}h(\mathbf{x}) + \mathbf{d}) \tag{6}$$

Therefore, for the input source domain data $D_S = \{(x_i^s, y_i^s)\}_{i=1}^m$, the flag of domain D_i is equal to 1. For the input source domain data $D_T = \{x_i^t\}_{i=1}^m$, the flag of domain D_i is equal to 0. The loss function of a domain-adaptive module is given as:

$$\mathcal{L}_{dom} = -\sum_{i=1}^{m} [\mathcal{L}^d(o(x_i^s), 1)] - \sum_{i=1}^{m} [\mathcal{L}^d(o(x_i^t), 0)] \tag{7}$$

with

$$\mathcal{L}^d(o(\mathbf{x}), D_i) = -D_i \log(o(\mathbf{x})) - (1 - D_i)\log(1 - o(\mathbf{x}))$$

The final training loss of the proposed network consists of YOLO and domain-adaptive module, which can be written as:

$$\mathcal{L}_{da} = \mathcal{L}_{yolo} + \lambda \mathcal{L}_{dom} \tag{8}$$

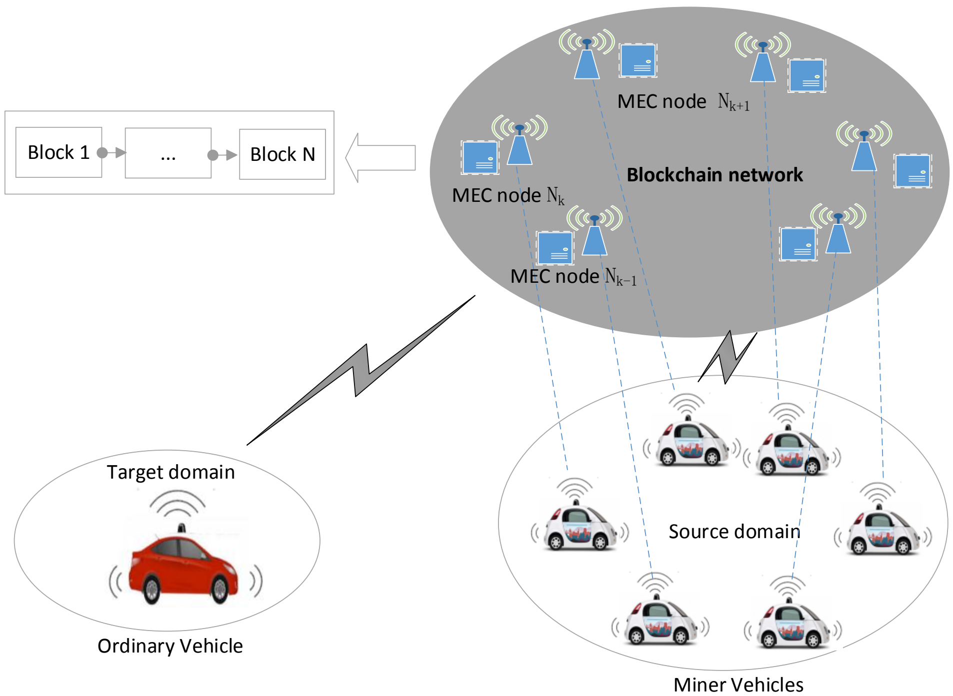


Fig. 2: Architecture of blockchain-enabled object detection.

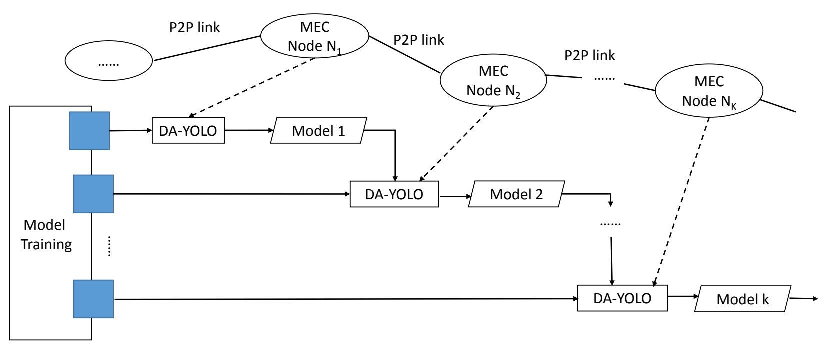


Fig. 3: The process of the model sharing.

where parameter λ (λ>0) is a weight factor that can achieve a trade-off between the YOLO loss and domain adaptation module. GRL acts as the reversal gradient with negative multiplier λ. The DA-YOLO network is trained in end-to-end learning by using stochastic gradient descent (SGD) algorithm [56]. The source data can be trained in feature extractor and label predictor, and the source data and domain data can be trained in feature extractor and domain classifier. In oder to improve the performance of the domain adaptation model, the DA-YOLO network can be fine-tuned under a particular constraint. In addition, the feature extractor and label predictor are used to inference.

C. Blockchain Enabled Model Sharing

1) Vehicle Blockchain: In the proposed blockchain-enabled object detection framework of connected vehicles, the miner vehicles are used to collect the images and annotate the object's location and categories respectively. After that, the training data (including source domain and target domain) are synchronized to the nearby MEC nodes. Until the model converges, the proposed DA-YOLO model will be trained nodes with the distributed learning approach. Therefore, the MEC node can be used for data storage, model training and the controller for smart contract. Finally, the ordinary vehicle can request the trained DA-YOLO model for the object detection task. However, data privacy and price are un-guaranteed for information sharing. Therefore, the blockchain is the key to provide privacy and reward for miner vehicles and MEC nodes. There are mainly two benefits of blockchain-enabled object detection approach: (1) The reliability of the training data can be guaranteed by using the blockchain consensus. (2) The spare computing power of MEC nodes can be priced through the reward.

2) Smart Contract for Model Sharing: A smart contract is an agreement between two or more parties, encoded in such a way that the correct execution is guaranteed by the blockchain. Moreover, the data and model can be priced via smart contracts. The structure of vehicle blockchain is shown

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2967788, IEEE Internet of Things Journal

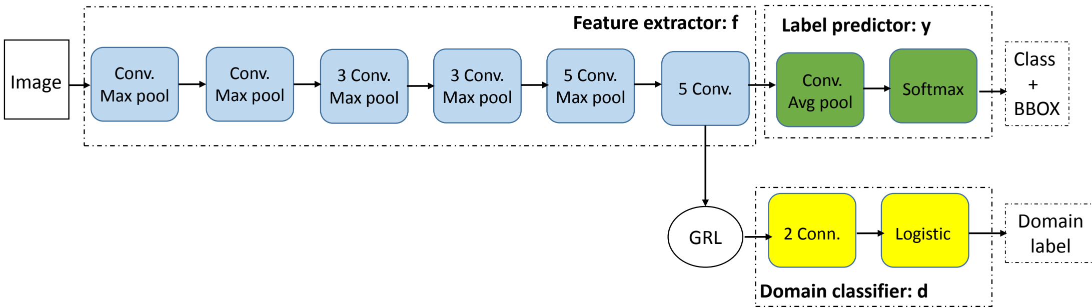


Fig. 4: An overview of the proposed DA-YOLO model. (1) The first 13 convolutional layers of the feature extraction network (including conv., conv., 3 conv., 3 conv., and 5 conv.) are general, hence these layers are frozen, (2) The last 5 convolutional layers of the feature extraction network (5 conv.) are slightly less transferable, hence these layers are learned via fine-tuning, where conv. represents convolution layer, conn. represents the full connection layer.

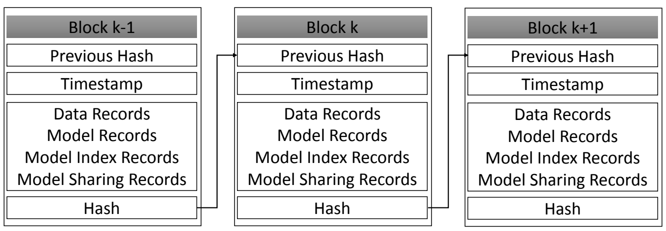


Fig. 5: The structure of vehicle blockchain.

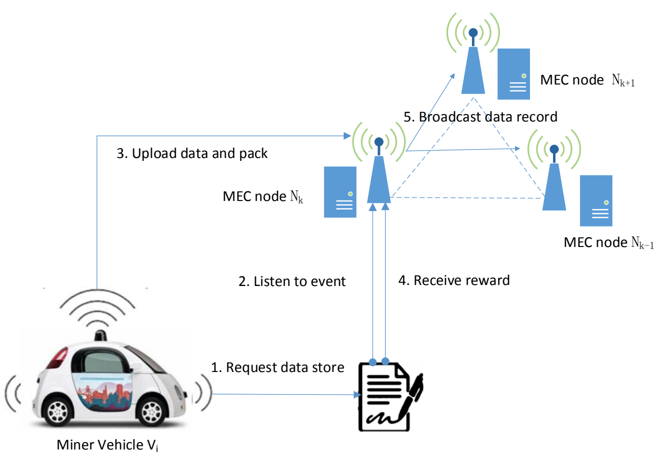


Fig. 6: Smart contract for data storage.

as Fig. 5, and the transaction data includes four types of records: data records, model records, model index records, and model sharing records. These smart contracts can be designed for these records as follows.

Firstly, Fig. 6 illustrates the smart contract for data storage of miner vehicles. The miner vehicles upload the encrypted data to the nearby MEC nodes. The InterPlanetary File System (IPFS) is used to store data [57], and IPFS can store and share the data with a content-addressable and peer-to-peer (P2P) method. When the data file is verified correctly, the current MEC node $N_k(k = 1, 2, ...)$ will broadcast the data record to other MEC nodes. Finally, the smart contract can be executed and synchronize to the vehicle blockchain. In addition, the smart contract of adding data records for the ordinary vehicle

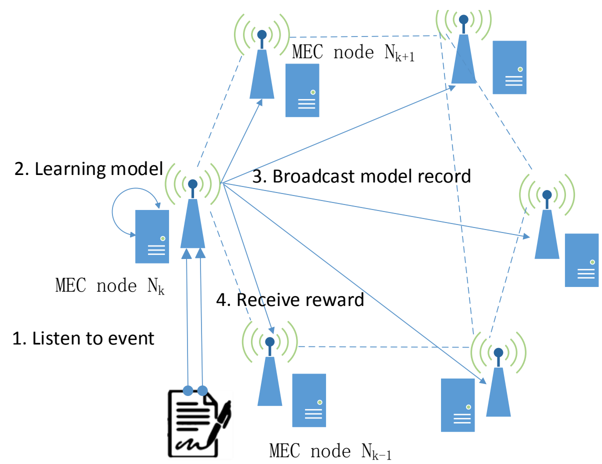


Fig. 7: Smart contract for model training.

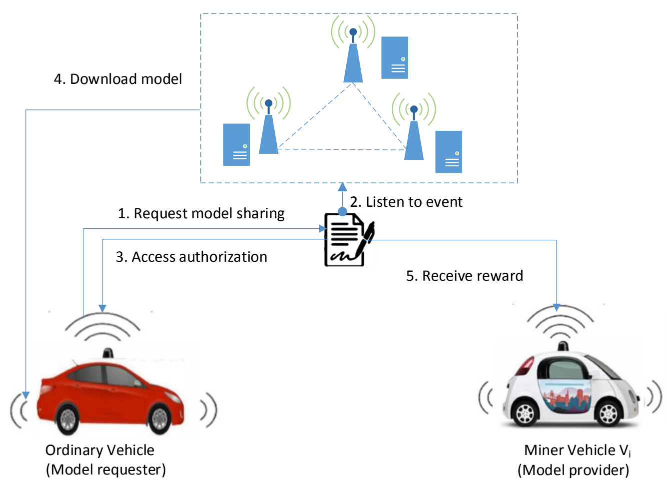


Fig. 8: Smart contract for model sharing.

to the blockchain is similar to the process of miner vehicles.

Secondly, Fig. 7 shows the smart contract for model training. After the miner vehicles and ordinary vehicle uploading the training data to the nearby MEC node, the deep learning group for the object detection task is formed by using DPoS consensus method, and the DA-YOLO model is trained across

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2967788, IEEE Internet of Things Journal

MEC nodes. Until the model converges, the transaction including the model index is broadcast to the nearby MEC nodes.

Finally, the smart contract for model sharing is shown as Fig. 8. When a ordinary vehicle requests the model sharing, the model cached is returned, and the trained model can be downloaded for the inference. Then, the smart contract will send the reward to the miner vehicles immediately.

3) Consensus Process for Blockchain: After generating the above four records by these smart contracts, the MEC controller node can package these records into blocks. By using DPoS algorithm, the delegates can add the block to the blockchain during the consensus stage [58]. Each MEC node has the right to vote the delegate, and the delegates with the highest number of votes are selected to produce new blocks.

TABLE III: The DA-YOLO setting.

Loss function	$\mathcal{L}_{da}$
Parameter λ	λ = 0.01
Optimization algorithm	SGD
Learning rate	lr=0.01
Early stopping rule	Loss in generality
Epoch	20
The number of MEC nodes k	k=6

TABLE IV: The mean average precision comparison.

Approach	B $\rightarrow$ K
YOLOv2 [25]	25.93%
DSSD [24]	27.40%
DP-YOLO [61]	28.60%
Proposed method	32.25%

V. Simulation Results and Discussions

In this section, the performance of the proposed approach is evaluated under public datasets. Two main different dataset shifts have been considered. At last, the reliability of the blockchain enabled approach is discussed.

A. Benchmarks and Training Settings

With the unsupervised domain adaptation approach, the training data includes the source domain data and the target domain data. The source domain data consists of images and annotations, which is uploaded by miner vehicles in this work. The target data includes unlabeled images, which is uploaded by ordinary vehicle. In this work, the two main dataset shifts have been considered: (1) cross camera transfer (BDD100K to KITTI as B $\rightarrow$ K), (2) normal to foggy transfer (BDD100K to Foggy Cityscapes as B $\rightarrow$ F). The BDD100K dataset is as the source domain D_S, and KITTI and Foggy Cityscapes datasets are as the target domain D_T. The training data is distributed evenly across MEC nodes. The classes are “car”, “bus”, “person”, “bike”, and “truck”.

The BDD100K and KITTI are datasets comprised of real driving data [51], [53]. The settings of BDD100K and KITTI dataset have city, rural area, and highway. The diversity of BDD100K dataset includes multiple cites, multiple weather conditions, daytime and nights, however, the diversity of KITTI dataset includes one city, one weather condition, and daytime. The Foggy Cityscape is a synthetic foggy dataset, which is generated by Cityscape data [52].

The original YOLOv2 (YOLO9000) model is as the reference model, and it is trained on the source domain data. The DA-YOLO setting is shown in Table III. The SGD algorithm is used to optimize the DA-YOLO model, which is a simple and computational efficient algorithm of gradient-based optimization. In addition, the model is initialized by using pre-trained YOLOv2 on COCO image dataset [59]. The mean average precision (mAP) is used to evaluate the performance of the proposed method. The intersection over union (IoU) that is greater than 0.5 with the bounding box is selected as the output.

In this work, the vehicle blockchain is based on the service-oriented blockchain system vDLT [60]. Different quality of service (QoS) requirements are fulfilled, moreover, the DPoS mechanism is used to manage nodes in the vDLT system. In addition, for each MEC node, the configuration of CPU is inter Core i5-8400, and the configuration of GPU is Geforce GTX 1050.

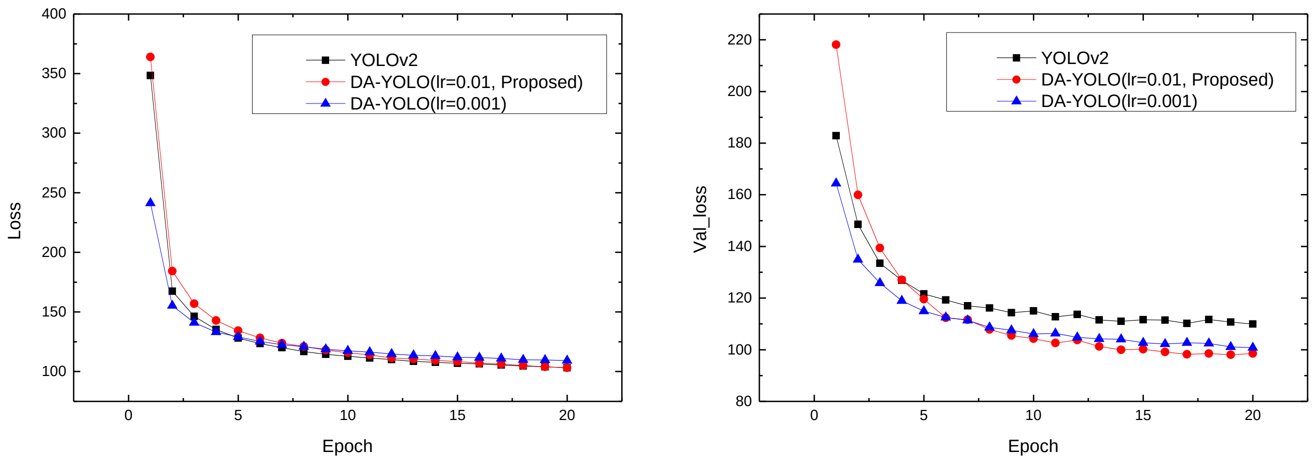
B. Results Comparison of Cross Camera Transfer

1) Training Results Analysis: Fig. 9 illustrates the training/validation loss-epoch curve of the proposed method, compared with the reference model. Fig. 9.(a) shows the training loss curves with different learning rates. It can be observed that the performance of DA-YOLO method with lr=0.01 (proposed) is better than the performance of DA-YOLO method with lr = 0.001. The loss of the YOLO baseline converges faster to fall during the first epoch. Moreover, during the later epochs, the convergence rate of the proposed method is similar as YOLO baseline. Fig. 9.(b) shows the validation loss curves for different methods. It can be seen that the validation loss of the DA-YOLO method with lr=0.001 converges faster to fall during the first five epochs. However, during the later epochs, the DA-YOLO method with lr=0.01 converges faster to fall, which verifies the effectiveness of the present loss function further.

Fig. 10 shows the six MEC nodes perform the task at a rate of one epoch every 1330 seconds, which is about 5.4 times faster than the case with only one node. Therefore, the computation overhead of each node can be reduced significantly, and accordingly the risk of interrupting training task caused by the failed occupation of resource can be eliminated.

2) Inference Results Analysis: For B $\rightarrow$ K transfer, the mAP comparison results of the proposed method is shown in Table IV, compared with [23], [24], [61]. YOLOv2 and DSSD belong to one stage methods. DSSD can achieve the accuracy-vs-speed trade-off, which is the more efficient than YOLOv2 model. Moreover, Sergeev *et al.* proposed a data parallel approach for distributed deep learning method. In this context, the same sample of source domain is trained multiple nodes in parallel, however, each node reads in a different chunk of data at each iteration. The gradients by all copies are averaged for multiple nodes. After that, the model is to be updated for each nodes. Based on Sergeev's work, the data parallel-enabled YOLO (DP-YOLO) approach is added to comparison experiments.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2967788, IEEE Internet of Things Journal



(a) Training (b) Validation

Fig. 9: The training/validation loss-epoch curves.

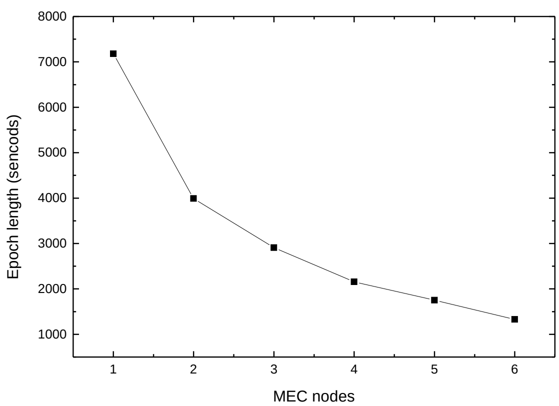


Fig. 10: Mean training epoch length vs. the number of MEC nodes.

From the Table IV, it is noted that DSSD approach can improve mAP by 1.47% and DP-YOLO approach can improve mAP by 2.67%, compared with YOLOv2 model. Moreover, the proposed method can achieve 32.25% mAP which are 6.32%, 4.85% and 3.65% better than YOLOv2, DSSD and DP-YOLO approaches. Therefore, the data parallel approach can improve the efficiency of YOLO model, and the performance of the proposed method is higher than YOLOv2, DSSD and DP-YOLO approaches. For further comparison analysis, Fig. 11 shows the comparison of quantifying results, compared with YOLOv2, DSSD and DP-YOLO approaches. For each component, the average precision (AP) of the proposed method is higher than the average precision of YOLOv2, DSSD and DP-YOLO approaches. Moreover, the log-average miss rate of the proposed method is lower than the log-average miss rate of YOLOv2, DSSD and DP-YOLO methods. Thus, the present method can reduce the domain discrepancy of different object classes significantly.

Fig. 12 further presents the precision-recall curves of the proposed method and the reference model. In general, the larger the area enclosed by the precision-recall curve, the higher the performance of the object detector. Therefore, the proposed detector achieves a better performance than the YOLO reference model.

TABLE V: The mean average precision comparison.

Approach	B $\rightarrow$ F
YOLOv2 [25]	20.78%
DSSD [24]	22.00%
DP-YOLO [61]	23.20%
Proposed method	25.80%

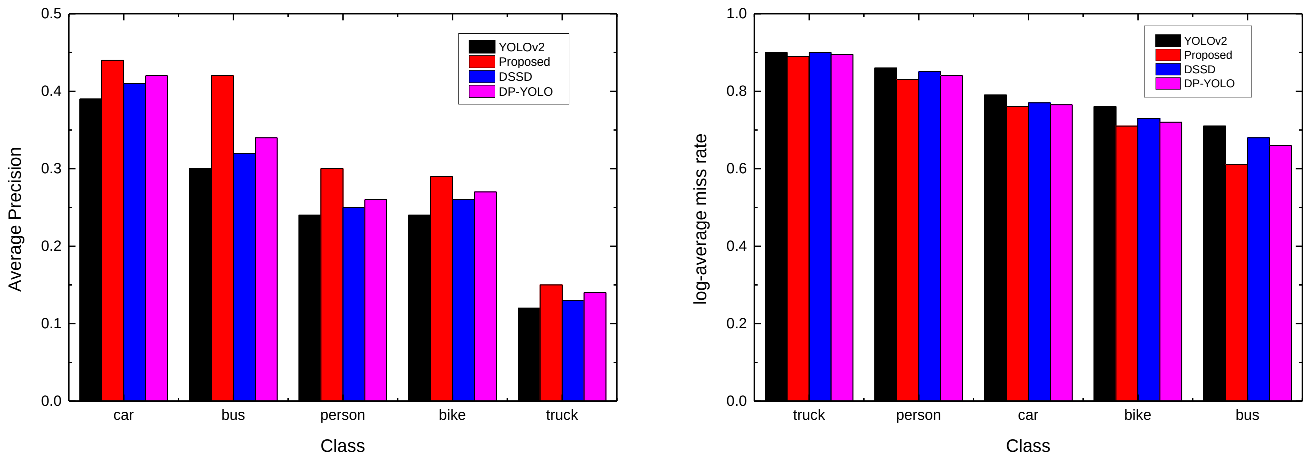
C. Results Comparison of Normal to Foggy Transfer

In order to verify the performance of the proposed method on Foggy Cityscapes data validation set for B $\rightarrow$ F transfer, the comparison results are shown as Table V. It can be seen that DSSD and DP-YOLO approaches can increase mAP by 1.22% and 2.42%, in comparison with YOLOv2 model. Moreover, the proposed method can achieve 25.80% mAP which is 3.80% and 2.60% better than DSSD and DP-YOLO approaches. Therefore, the proposed object detector has good performance for across-domain detection.

In addition, Fig. 13 illustrates the average precision results of each component. From this figure, it can be seen that the proposed method can increase the average precision for different categories. Moreover, the proposed method can increase the best performance by 10% for the object "car", compared with YOLOv2 model. The proposed method can reduce the domain discrepancy significantly, and the domain-adaptive YOLO detector can achieve a better performance.

D. Reliability Analysis

Blockchain provides higher reliability with data synchronization, and hence protects trained models. Tremendous resource is contributed by the participants in blockchain.



(a) The average precision of each component

(b) The log-average miss rate of each component

Fig. 11: The comparison of quantify results.

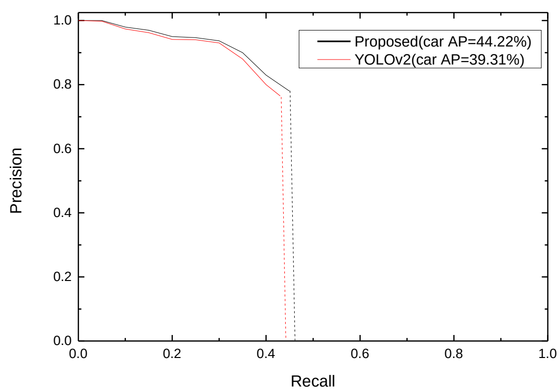


Fig. 12: The precision-recall curves of class "car".

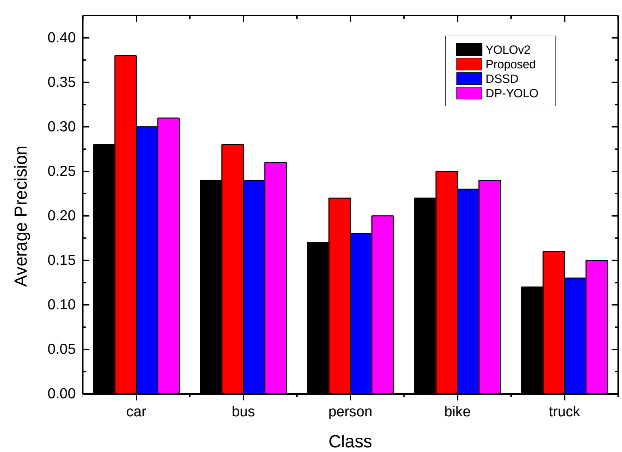


Fig. 13: The average precision of each component.

The combination of blockchain and object detection has the potential of establishing a trusted, reliable autonomous driving system. Moreover, blockchain encourages model sharing as it can offer transparency and reliability, and it makes the control of data back into vehicle user's hands.

Furthermore, smart contract supports flexible structure of message, and it can make model training automatically. Moreover, in comparison with conditional distributed learning, blockchain enabled distributed learning can reduce the computation overhead of each node.

Based on the above discussions, the proposed blockchain-enabled approach can improve the performance of object detection across-domain. With the domain adaptation, the proposed method can reduce domain discrepancy across different object classes. Moreover, the reliability of the model sharing can be guaranteed by using blochchain technology.

VI. Conclusions and Future Work

In this work, a blockchain-enabled object detection framework was proposed for autonomous driving. In order to develop cross-domain object detector, the domain-adaptive YOLO model was trained with distributed learning. Moreover, blockchain technology was used to achieve the model sharing across MEC nodes. The advantage of the proposed method was that the reliable object detector can predict efficiently the object class and bounding box for different domain shift scenarios. The simulation results showed that the proposed method can improve the mean average precision by 6.32% with cross camera transfer, compared with the reference model.

In order to reduce significantly the computation complexity and memory cost, the model compression and acceleration methods for convolutional neural networks will be studied further. The effectively structured prune networks can accelerate the model training significantly, and at the same time the performance of the object detector is not decreased.

REFERENCES

[1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv e-prints*, p. arXiv:1905.05055, May 2019.

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, no. 1-2, pp. 151–175, May 2010.

[3] L. Zhang, "Transfer adaptation learning: A decade survey," *arXiv e-prints*, p. arXiv:1903.04687, Mar. 2019.

[4] I. Athanasiadis, P. Mousouliotis, and L. Petrou, "A framework of transfer learning in object detection for embedded systems," *arXiv e-prints*, p. arXiv:1811.04863, Nov. 2018.

[5] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster R-CNN for object detection in the wild," in *Proc. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, UT, USA, June 2018, pp. 3339–3348.

[6] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proc. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, UT, USA, June 2018, pp. 5001–5009.

[7] S. Gupta, W. Zhang, and F. Wang, "Model accuracy and runtime tradeoff in distributed deep learning: A systematic study," in *Proc. IEEE 16th International Conference on Data Mining*, Barcelona, Spain, Dec. 2016, pp. 171–180.

[8] T. Ben-Nun and T. Hoefler, "Demystifying parallel and distributed deep learning: An in-depth concurrency analysis," *arXiv e-prints*, p. arXiv:1802.09941, Feb. 2018.

[9] Z. Zhang, L. Yin, Y. Peng, and D. Li, "A quick survey on large scale distributed deep learning systems," in *Proc. IEEE 24th International Conference on Parallel and Distributed Systems*, Singapore, Singapore, Dec. 2018, pp. 1052–1056.

[10] Y. He, F. R. Yu, N. Zhao, and H. Yin, "Secure social networks in 5g systems with mobile edge computing, caching, and device-to-device communications," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 103–109, June 2018.

[11] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Trans. Vehicular Technology*, vol. 67, no. 11, pp. 11 008–11 021, Nov. 2018.

[12] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[13] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[14] F. R. Yu, J. Liu, Y. He, P. Si, and Y. Zhang, "Virtualization for distributed ledger technology (vDLT)," *IEEE Access*, vol. 6, pp. 25 019–25 028, 2018.

[15] Z. Zhou and X. Chen and E. Li and L. Zeng and K. Luo and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *arXiv e-prints*, p. arXiv:1905.10083, May 2019.

[16] T. N. Dinh and M. T. Thai, "AI and blockchain: A disruptive integration," *Computer*, vol. 51, no. 9, pp. 48–53, Sep. 2018.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, OH, USA, June 2014, pp. 580–587.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[19] R. Girshick, "Fast R-CNN," in *Proc. The IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, Montral Canada, Dec. 2015, pp. 91–99.

[21] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. The IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, NV, USA, June 2016, pp. 779–788.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *arXiv e-prints*, p. arXiv:1512.02325, Dec. 2015.

[24] C-Y. Fu and W. Liu and A. Ranga and A. Tyagi and A. C. Berg, "DSSD : Deconvolutional single shot detector," *arXiv e-prints*, p. arXiv:1701.06659, Jan. 2017.

[25] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, HI, USA, July 2017, pp. 6517–6525.

[26] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv e-prints*, p. arXiv:1804.02767, Apr. 2018.

[27] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Beijing, China, June 2018, pp. 4203–4212.

[28] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2Det: A single-shot object detector based on multi-level feature pyramid network," *arXiv e-prints*, p. arXiv:1811.04533, Nov. 2018.

[29] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. The IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 5543–5551.

[30] B. Gong, K. Grauman, and F. Sha, "Reshaping visual datasets for domain adaptation," in *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, Dec. 2013, pp. 1286–1294.

[31] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers, "Associative domain adaptation," in *Proc. The IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2765–2773.

[32] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv e-prints*, p. arXiv:1412.3474, Dec. 2014.

[33] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. the 32nd International Conference on Machine Learning*, vol. 37, Lille, France, July 2015, pp. 97–105.

[34] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv e-prints*, p. arXiv:1409.7495, Sep. 2014.

[35] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, Apr. 2016.

[36] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *arXiv e-prints*, p. arXiv:1505.07818, May 2015.

[37] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. European Conference on Computer Vision*, Amsterdam, The Netherlands, Oct. 2016, pp. 597–613.

[38] J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, and Y. Liu, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Comm. Surveys Tutorials*, vol. 21, no. 3, pp. 2794–2830, thirdquarter 2019.

[39] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Comm. Surveys Tutorials*, vol. 21, no. 2, pp. 1508–1532, Secondquarter 2019.

[40] D. Zhang, F. R. Yu, and R. Yang, "Blockchain-based distributed software-defined vehicular networks: A dueling deep Q -learning approach," *IEEE Trans. Cognitive Comm. Networking*, vol. 5, no. 4, pp. 1086–1100, Dec. 2019.

[41] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Performance optimization for blockchain-enabled industrial internet of things (IIoT) systems: A deep reinforcement learning approach," *IEEE Trans. Industrial Informatics*, vol. 15, no. 6, pp. 3559–3570, June 2019.

[42] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial internet of things: A dueling deep Q -learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4627–4639, June 2019.

[43] C. Kaiser, M. Steger, A. Dorri, A. Festl, A. Stocker, M. Fellmann, and S. Kanhere, "Towards a privacy-preserving way of vehicle data sharing – a case for blockchain technology?" in *Proc. Advanced Microsystems for Automotive Applications 2018*, Berlin, Germany, Aug. 2019, pp. 111–122.

[44] R. Sharma and S. Chakraborty, "B2VDM: Blockchain based vehicular data management," in *Proc. International Conference on Advances in Computing, Communications and Informatics*, Bangalore, India, Sep. 2018, pp. 2337–2343.

[45] M. Singh and S. Kim, "Blockchain based intelligent vehicle data sharing framework," *arXiv e-prints*, p. arXiv:1708.09721, July 2017.
[46] P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-VN: A distributed blockchain based vehicular network architecture in smart city," *Journal of Information Processing Systems*, vol. 13, no. 1, pp. 184–195, Feb. 2017.
[47] M. Singh and S. Kim, "Branch based blockchain technology in intelligent vehicle," *Computer Networks*, vol. 145, no. 9, pp. 219 – 231, Nov. 2018.
[48] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, 2016, pp. 3–16.
[49] P. Vasin, "Blackcoins proof-of-stake protocol v2," 2014. [Online]. Available: https://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf
[50] D. Larimer, "DPOS consensus algorithm - the missing white paper," 2017. [Online]. Available: https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper
[51] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, HI, USA, 2017, pp. 3530–3538.
[52] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *arXiv e-prints*, p. arXiv:1708.07819, Aug 2017.
[53] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013.
[54] W. M. Kouw and M. Loog, "An introduction to domain adaptation and transfer learning," *arXiv e-prints*, p. arXiv:1812.11806, Dec 2018.
[55] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, May 2018.
[56] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv e-prints*, p. arXiv:1609.04747, Sep 2016.
[57] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," *arXiv e-prints*, p. arXiv:1407.3561, Jul 2014.
[58] J. Kang and Z. Xiong and D. Niyato and D. Ye and D. I. Kim and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.
[59] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. European conference on computer vision*, Zurich, Switzerland, Sep. 2014, pp. 740–755.
[60] F. R. Yu, "vDLT: A service-oriented blockchain system with virtualization and decoupled management/control and execution," *arXiv e-prints*, p. arXiv:1809.00290, Sep 2018.
[61] A. Sergeev and M. Del Balso, "Horovod: fast and easy distributed deep learning in TensorFlow," *arXiv e-prints*, p. arXiv:1802.05799, Feb. 2018.

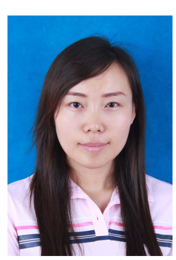
Xiantao Jiang (S'14-M'16) received M.E. degree from Shanghai Maritime University (SMU) in 2012, and the Dual Ph.D. degrees from Tongji University of China and Tokushima University of Japan in 2016. After that, he joined in the college of information engineering of Shanghai Maritime University as an Assistant Professor. He is currently a post-doc in Carleton university, Ottawa, Canada. His research interests include blockchain, multimedia networking and deep learning.

F. Richard Yu (S'00-M'04-SM'08-F'18) received the PhD degree in electrical engineering from the University of British Columbia (UBC) in 2003. From 2002 to 2006, he was with Ericsson (in Lund, Sweden) and a start-up in California, USA. He joined Carleton University in 2007, where he is currently a Professor. He received the IEEE TCGCC Best Journal Paper Award in 2019, Distinguished Service Awards in 2019 and 2016, Outstanding Leadership Award in 2013, Carleton Research Achievement Award in 2012, the Ontario Early Researcher Award (formerly Premiers Research Excellence Award) in 2011, the Excellent Contribution Award at IEEE/IFIP TrustCom 2010, the Leadership Opportunity Fund Award from Canada Foundation of Innovation in 2009 and the Best Paper Awards at IEEE ICNC 2018, VTC 2017 Spring, ICC 2014, Globecom 2012, IEEE/IFIP TrustCom 2009 and Int'l Conference on Networking 2005. His research interests include connected/autonomous vehicles, security, artificial intelligence, distributed ledger technology, and wireless cyber-physical systems.

He serves on the editorial boards of several journals, including Co-Editor-in-Chief for Ad Hoc & Sensor Wireless Networks, Area Editor for IEEE Communications Surveys & Tutorials, Lead Series Editor for IEEE Transactions on Vehicular Technology, and IEEE Transactions on Green Communications and Networking. He has served as the Technical Program Committee (TPC) Co-Chair of numerous conferences. Dr. Yu is a registered Professional Engineer in the province of Ontario, Canada, an IEEE Fellow, IET Fellow, and Engineering Institute of Canada (EIC) Fellow. He is an IEEE Distinguished Lecturer of both Vehicular Technology Society (VTS) and Comm. Society. He is an elected member of the Board of Governors of the IEEE VTS.

Tian Song received his B.E. degree from Dalian University of Technology, China, in 1995, and his M.E. and Dr.E. degrees from Osaka University in 2001 and 2004, respectively. He joined Tokushima University in 2004 as an Assistant Professor. Presently, he is an Associate Professor of the Department of Electrical and Electronic Engineering, Graduate School of Advanced Technology and Science, Tokushima University. He is a member of IEICE and IEEE. His current research interests include video coding algorithms.

Zhaowei Ma received M.E. degree from Carleton University, Ottawa, Canada in 2018. He is currently pursuing the Ph.D. degree with Carleton University. His current research interests include blockchain and deep learning.



Yanxing Song received the Ph.D. degree in instrument science and technology from Harbin Institute of Technology, China, in 2009. She conducted her Post-Doctoral Research in Beijing Jiaotong University, China. She had been a Visiting Scholar with Carleton University. She is currently an Associate Professor with the Information SchoolBeijing Wuzi University. Her research interests include blockchain and image processing.

Daqi Zhu received the B.Sc. degree in physics from the Huazhong University of Science and Technology, Wuhan, China, in 1992 and the Ph.D. degree in electrical engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2002. His current research interests include neural networks and control of AUV.