Road Scene Data Annotation with Semi-Automated Active Learning Framework for Convolutional Neural Networks

Mohd Hafiz Hilman Mohammad Sofian and Toshio Ito
Shibaura Institute of Technology, Tokyo, Japan
Email: {nb19108, tosi-ito}@shibaura-it.ac.jp

***Abstract*—Autonomous driving vehicles are considered the future of mobility as they can reduce the mortality rate owing to traffic accidents. This can also be achieved using cameras and a Convolutional Neural Network (CNN) to detect objects on the road and take necessary actions to prevent life-threatening occurrences. However, the current form of CNN needs to be trained using large amounts of annotated data, which is time consuming, expensive, and requires extensive manpower. These limitations can be overcome by using Active Learning (AL) systems, which only select a subset of informative data from the big data for annotation by humans. Although AL reduces the amount of data being used for CNN training, humans are still needed to annotate the data. This study proposes a Semi-Automated Active Learning system (SAAL) to further reduce the need for manpower for data annotation. SAAL uses AL and a new algorithm called Machine Teachers (MTs), which are stacked algorithms of pre-trained CNN and optical flow that use the temporal-spatial information video data from cameras on vehicles to help humans annotate images. This allows SAAL to be partially automated and further reduces human effort while roughly maintaining the accuracy of CNN to that of AL.**

***Index Terms*—active learning, convolutional neural network, image annotation, optical flow**

I. Introduction

Automated driving vehicle technology, a sophisticated and accessible technology comprising multiple complicated and sensitive systems that are harmonious and communicate with each other, is being developed rapidly. One of the most crucial components of automated driving vehicles is the perception system, which uses multiple sensors such as cameras and LiDAR to enable the vehicle to sense its surroundings while navigating.

The camera is considered the most important component of the perception system as it closely resembles the human eye and provides rich information in the form of color and shape. Additionally, given that cameras are inexpensive, automated driving vehicles using only cameras also become inexpensive. An affordable automated driving vehicle can be developed by creating a perception system that uses a camera and a Convolutional Neural Network (CNN) to help avoid obstacles while driving. Tesla Inc. applies this technology, and now, its vehicles are common on the road. This technology demonstrates the importance of CNNs in the development of practical automated driving vehicles.

Two major tasks are involved when using CNN for the image domain: Image Classification Task (ICT), which classifies objects or determines the "what" in an image, and Object Detection Task (ODT), which detects multiple objects or determines the "what" and "where" in an image. Fig. 1 shows the differences between the ICT and ODT results. The annotating images for ODT are cost-hungry compared to ICT owing to multiple bounding boxes and class labels that need to be annotated. Autonomous driving vehicle systems apply ODT, which will also be discussed in this study.

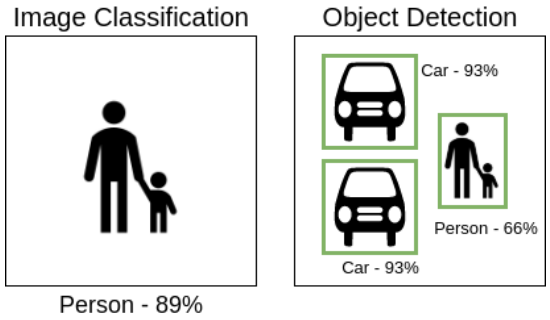

Figure 1. Difference between image classification and object detection tasks.

Before the development of CNN or deep learning, conventional image processing methods were widely used to realize ICT and ODT. These methods, which include a Histogram of Oriented Gradient (HoG) [1], [2], a Bag of Visual Words (BOVW) [3]-[5], SIFT [6], and SURF [7], need to be defined manually to extract image features, which makes them significantly different from CNN. One of their advantages is that their internal working is transparent and can be easily understood. In addition, these methods are considered mature considering they have been used for a long time. However, given that they need to be defined manually makes it difficult for them to be generalized for extensive applications. Furthermore, they cannot utilize the big data of the current world [8].

Compared to conventional image processing methods, CNN, which mimics the biological brain to a certain extent,

Manuscript received March 29, 2022; revised June 9, 2022.

doi: 10.12720/jait.13.5.441-449

can automatically extract features from images through backpropagation [9], thereby eliminating the need for manpower to manually define an extraction algorithm. This is highly advantageous considering CNNs can be scaled according to the size of the dataset [8], making it more accurate to the task to be tackled [10]. CNNs for ICT include GoogleNet [11] and ResNet [12], and ODT include such CNNs are the You Only Look Once (Yolo) object detector family [13]-[15] and Single Shot multibox Detector (SSD) [16] (Refer to [8] for more details). However, CNN requires a large number of annotated datasets to perform tasks. Annotating the image dataset for ODT is time-consuming, expensive, and requires manpower. This problem can be partially solved by Active Learning (AL) [17]-[19].

The AL system is designed to reduce manual image annotations based on the idea is that not all data are the same; that is, some data are more informative than others. For example, only half the data in the X number data are beneficial to the CNN during training. The beneficial or informative data is selected by the model to be trained, and a query strategy is implemented within the AL, as shown in Fig. 2. However, this oracle still needs to annotate the data in the AL system, which does not solve the problem of costs.

Therefore, this study proposes an improved AL system called the Semi-Automated Active Learning (SAAL) system that comprises AL and a novel algorithm called Machine Teachers (MTs) to further reduce the cost. (MTs) combine a pretrained CNN and conventional image-processing techniques to help the oracle by partially and automatically annotating images for ODTs. The proposed MT was implemented using an optical flow and a pre-trained CNN, which allowed us to use temporal-spatial information from video data recorded by the camera of the vehicle to automate the annotation process. This is advantageous considering data involving road scenarios can easily be used in the form of a video. In particular, a certain frame (a particular image from video) is selected, and the MTs use the spatial-temporal information revolving around that frame to annotate. To the best of our knowledge, this method has not been implemented in any other research. While ViewAL [20] used video with AL, it dealt with multiple videos depicting the same scene from different angles, making it unsuitable for the development of autonomous driving vehicles.

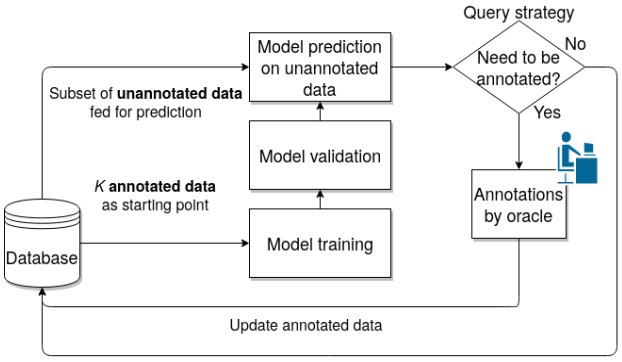

Figure 2. Loop of a typical AL system.

SAAL was evaluated by conducting three experiments: Experiment 1, Experiment 2, and Experiment 3. In Experiment 1, we trained the CNN model using a normal AL system to demonstrate that all annotations are made by humans. In Experiment 2, we trained the CNN model using the proposed SAAL system. Herein, the annotations were partially annotated by the MTs, while humans helped the annotations result by the MTs. Finally, Experiment 3 was conducted using 100% annotations by the MTs without human intervention. One can even refer to this as fully automated active learning. Results showed that SAAL was able to produce a CNN model comparable to AL in terms of accuracy while reducing the number of annotations work by the oracle.

II. Machine Teachers

In this section, we discuss the implementation of MTs and SAAL. First, we discuss the parts comprising MTs and their inner working. The next subsection discusses the current state of AL and the inner working of its algorithm. Lastly, we discuss the working of our proposed SAAL system.

A. Structure

Generally, MTs are a set of algorithms comprising a pre-trained CNN (hereon referred to as CNN_{MTs}) and conventional image processing techniques acting as the helper or teacher to automatically annotate image data. Note that the CNN_{MTs} are different from the CNN that will be trained later. MTs uses both the pre-trained CNN and conventional methods owing to their individual advantages, as mentioned in the Introduction section. Fig. 3 illustrates the concept of MTs. The objective of MTs is to automatically annotate some of the image data to reduce the workload of the oracle.

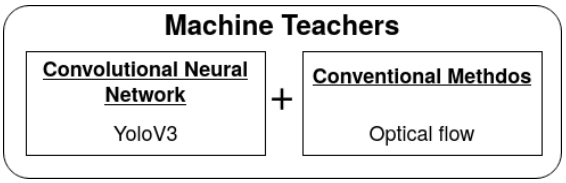

Figure 3. Implementation of MTs.

The proposed MTs are implemented to correspond with video data. It uses the spatial-temporal information of the video to annotate a single frame extracted from the video. The annotated frame is then forwarded to the oracle for repair before being transferred to the CNN for training. This was achieved by implementing an optical flow algorithm along with CNN_{MTs}. The concept of optical flow was introduced by a psychologist named James J. Gibson [21], and has been widely used in applications such as robotics and motion tracking. In this study, we used the standard optical flow of the Lucas-Kanade method [22], wherein the optical flow tracks the motion of neighboring pixels across frames, thereby allowing us to keep track of a particular region in a video. While other tracking methods are available, we chose optical flow considering it is easy to use.

For CNN_{MTs}, any pre-trained CNN available on the Internet can be used. Therefore, for our CNN_{MTs}, we used a network called YoloV3 [15] that has a fast inference or prediction time with high precision for object detection tasks. Published in 2018, YoloV3 is a stable CNN model for the proposed study. However, considering MTs are designed to be flexible, other CNNs can also be used depending on the usage. One can implement multiple CNN_{MTs} and conventional image processing methods over the optical flow to obtain improved annotation results. However, in this research, we do not go to that extent because we aim to build a working prototype of MTs.

The purpose of using a pre-trained CNN model in MTs (CNN_{MTs}) to detect object classes and use them as annotations to train another model, instead of using the model itself for our application considering it can detect object classes, is because we can train or retrain a particular CNN to be good and optimize it to predict only certain object classes in certain contexts, given that pre-trained models on the Internet are designed generally for a wide range of applications. Furthermore, MTs function more as an oracle helper than an object detection system considering they can be built in complicated ways, such as using multiple CNNs, which requires more computational power. This can be allowed in the object annotation process, such as in the proposed SAAL system, because unlike hardware usage in autonomous driving systems, hardware that can be used during object annotation does not have any constraints in terms of energy usage, size, and capacity.

B. Algorithm

We will now discuss the inner working of the proposed MTs. First, we detect the presence of objects using CNN_{MTs}. False positives or falsely detected objects can be expected, which can be suppressed and deleted using conventional image processing methods (in this case, optical flow).

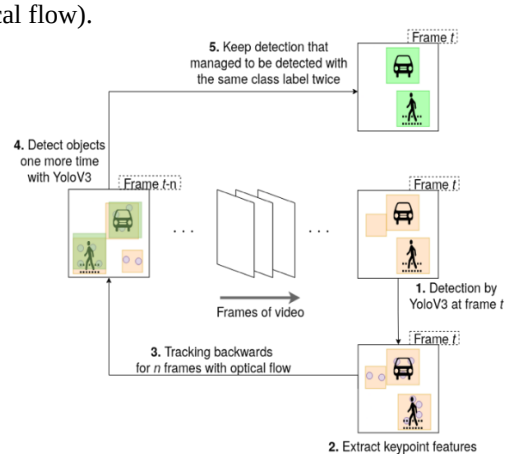

Figure 4. The proposed MTs implementation algorithm.

The algorithm is as follows: First, YoloV3 detects a single frame t of a video to obtain bounding boxes BB_t. The key point features from every BB_t were then extracted and tracked backward by n frames. At frame t-n, YoloV3 once again carries out detection to obtain bounding boxes BB_{t-n}. If BB_{t-n} overlaps with the tracked BB_t and the detected class labels are the same, the corresponding BB_t can be true positives. Therefore, the BB_t will be kept as an annotation. In other words, we use the spatial-temporal information existing in the video to suppress false positives using the first detection at frame t. In our implementation, t is set to the 10^{th} second and n is set to 10. Fig. 4 illustrates the entire process.

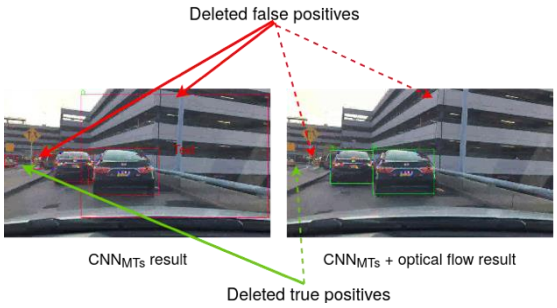

Figure 5. Experimental results of the MTs.

Fig. 5 shows the experimental results of the MTs for vehicle detection when used on a real video. Result showed that the optical flow was able to suppress the detection of two false positives by the CNN_{MTs}. However, true positive detection (correctly detected objects) was also deleted, which is inevitable and cannot be completely prevented. Because the oracle is required to annotate the remaining objects in the image, we called our version of AL semi-automated. These results can be improved by implementing more algorithms in MTs to improve accuracy. In this study, we first adhere to the simple implementation.

III. Semi-Automated Active Learning System

A. Active Learning

Research on AL has been ongoing for several decades. The idea is to reduce the amount of data needed to train a good model, which in this case, is the CNN. In other words, the AL system allows the model to be trained and the implemented query strategy framework to select the subset of data from our dataset. Only data that are considered "informative" are selected, which could lead to a lower but acceptable accuracy in the trained model. However, annotating all data is expensive.

Query strategy frameworks, such as uncertainty sampling [23], [24] and Query-by-Committee (QBC) [25], [26], mentioned above can be considered as the heart of the AL system, considering it decides how the system selects the data that is considered "informative." Uncertainty sampling is straightforward, where the framework selects data that the model is currently least confident about using the posterior probability or prediction score of the predicted class. The idea is that the less confident a model is on a particular data, the more informative it is. Additionally, the framework can use the difference or margin between the predicted and second predicted classes [27], [28]. Conversely, QBC is more complex as it involves the use of multiple models trained on the current annotated set while having different hypotheses. These models can give data instances they disagree with, which will then be considered as

informative, and selected. Interested readers can review these strategies and others from [17].

As shown in Fig. 2, an oracle first needs to annotate n data from the database to obtain *xanno* data, which should not be large. *xanno* is then be used to train our CNN (hereafter referred to as CNN_{AL}). This training phase should be conducted considering the maximum number of epochs are fulfilled. After training, validation data was used to validate and measure the performance of CNN_{AL}, and a subset of unannotated data *x* was obtained from the database. The query strategy framework within AL uses the prediction result to determine the informative data to be annotated. Finally, the oracle annotates the chosen data, and *xanno* is updated with the new data. This cycle is repeated until the number of cycles are fulfilled. It should be noted that CNN_{AL} and CNN_{MTs} are of different CNNs.

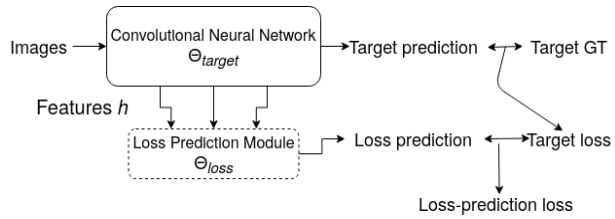

Figure 6. Loss prediction module.

However, this study does not use the aforementioned classic query strategy frameworks considering they are not suitable for use in ODTs. ODT is more complicated than ICT, where each image comprises multiple objects of different sizes and positions, as shown in Fig. 1, resulting in hurdles when applying normal query strategies for ODTs. However, research has come along considering the different approaches developed to solve this problem. In this study, we used the loss prediction module proposed in [29], wherein a small deep learning module is attached to our CNN_{AL} and trained together. The learning loss module is responsible for learning how to choose data that can be useful for CNN_{AL}. Furthermore, it learns how to predict the loss output by the unannotated image, followed by selecting the image with the highest predicted loss. The higher the loss is of an image, the more valuable and informative it is to the model. Fig. 6 shows how the loss prediction module is attached to the CNN_{AL}.

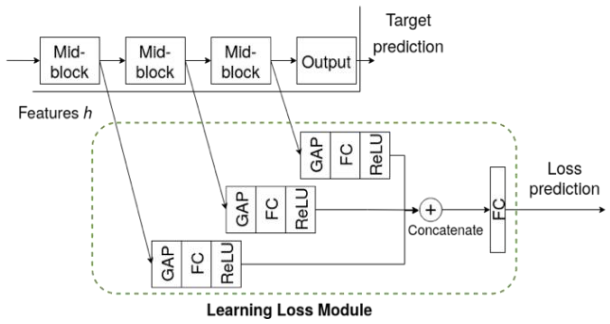

Figure 7. Architecture of the loss prediction module.

Fig. 7 shows the CNN architecture of the learning loss module *when attached to another CNN*. The loss module considers three feature maps from the main CNN as input. Each input is reduced to a fixed dimensional feature vector using a Global Average Pooling (GAP) layer and a Fully Connected layer (FC) before the ReLU activation function. Finally, all layers are concatenated and passed through another fully connected layer, which results in a scalar value for loss prediction.

During the selection for the informative data process, the AL system first takes a predefined number of data *J* from the unannotated database. Then, it passes the data through the CNN_{AL} and learning loss module. Because we do not have the ground truth of the data (they are unannotated), we cannot calculate the real loss of those data. However, the learning loss module can output the loss prediction considering it is trained to do so. Then, *K* data with the highest loss prediction is selected and sent to the oracle for annotation. This method is simple and task agnostic, which is highly suitable for ODT.

The loss prediction module algorithm can be summarized as follows: We have CNN_{AL} θ_{target} and loss prediction module θ_{loss}. The model outputs $\hat{y} = \theta_{target}(x)$, where *x* is the data point propagated across the model. Simultaneously, features of *x* extracted from several hidden layers of θ_{target}, h is passed to θ_{loss} and the loss of that particular image can be predicted as $\hat{l} = \theta_{loss}(h)$. With the target annotation *y* of *x*, the loss of θ_{target} is calculated as $l = L_{target}(\hat{y}, y)$. *l* is then used to calculate the loss of θ_{loss} as $L_{loss}(\hat{l}, l)$. Therefore, the final loss function is defined as $L_{target}(\hat{y}, y) + \lambda \cdot L_{loss}(\hat{l}, l)$. $L_{loss}(\hat{l}, l)$ is calculated from, the difference between the pairs of loss predictions, given as:

$$L_{loss}(\hat{l}^p, l^p) = max(0, -\mathbb{1}(l_i, l_j) \cdot (\hat{l}_i - \hat{l}_j) + \zeta)$$

$$\text{s.t.} \quad \mathbb{1}(l_i, l_j) = \begin{cases} +1, & if\ l_i > l_j \\ -1, & otherwise \end{cases} \tag{1}$$

where ζ is a pre-defined positive margin and *p* is the pair of loss predictions. With the number of batches *B*, the final loss function is determined as:

$$\frac{1}{B} \sum_{(x,y) \in B} L_{target}(\hat{y}, y) + \lambda \frac{2}{B} \cdot \sum_{(x^p, y^p) \in B} L_{loss}(\hat{l}^p, l^p)$$

$$\hat{y} = \theta_{target}(x)$$

$$\text{s.t.} \quad \hat{l}^p = \theta_{loss}(h^p)$$

$$l^p = L_{\text{target}}(\hat{y}^p, y^p) \tag{2}$$

We highly recommend the readers to read the paper [29] for a detailed insight on the algorithm.

B. Semi-Automated Active Learning

Here, we discuss the semi-automated active learning system. First, we built our own MT algorithm using CNN_{MTs} (in this case YoloV3) and optical flow. We then plugged the MTs into a typical AL system. In addition, we changed the conventional query strategy framework of the AL to the learning loss module to easily deal with the ODT. SAAL works the same as in usual AL, except that the oracle works alongside MTs during data annotations.

IV. Implementation

In this section, we discuss the dataset used to evaluate the proposed SAAL system. Furthermore, we discuss the implementation of our target model (to be trained or CNN_{AL}). Lastly, we discuss the implementation of the proposed SAAL system.

A. Dataset

We used the Berkeley Deep Drive (BDD) dataset [30] by the University of California-Berkeley for the implementation. BDD is a huge dataset comprising 100,000 driving videos of places such as New York, San Francisco Bay Area, and Berkeley, each being 40 s in length. The dataset provides ground truth annotations for certain tasks, such as lane marking, semantic instance segmentation, and object detection.

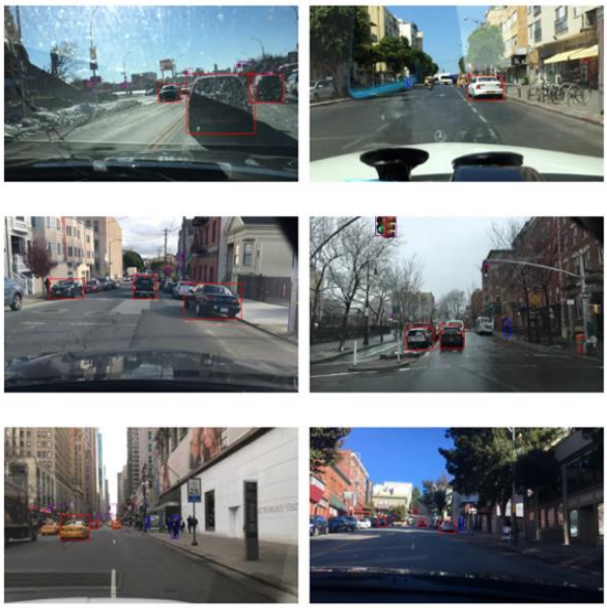

Figure 8. Example of image data from the BDD dataset.

Furthermore, this dataset provides images extracted from 100,000 videos. The images were extracted at the 10th second of each video, and the corresponding ground truth annotations of the images were also provided, which is why we chose this dataset for our experiments. Fig. 8 shows examples of images and annotations of this dataset. We used MTs to extract annotations at the 10th second of the videos, whereas the provided ground truth annotations were used to add up missed annotations by MTs to simulate the act of oracle adding the annotations. Because our data comprised driving videos, we assumed that our experiment would involve object detection for an autonomous driving vehicle.

For experimental purposes, we only used part of the data based on object class cars, persons, and traffic lights, which are objects essential to autonomous driving vehicles. Fig. 9 shows the height and width distributions of the subset dataset used. By constraining our dataset, we were able to focus on developing the SAAL to understand its results and performance.

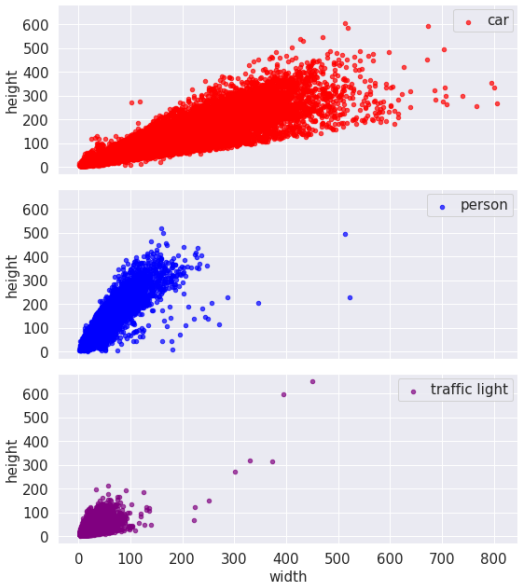

Figure 9. Object classes size distribution of the subset of BDD dataset.

B. Target Model

We created a custom model CNN_{AL} by combining MobileNetV2 [31], a mobile CNN architecture where the inference time is the primary concern, with the head of YoloV3 [15]. However, because MobileNetV2 is an architecture for ICT, the head of YoloV3, responsible for the ODT, was considered. In other words, MobileNetV2 was used as a backbone network considering YoloV3 is a network in itself comprising two parts: the image classification task backbone and the object detection head. In this study, MobileNetV2 is an already pre-trained model using the ImageNet dataset [10], whereas the head of YoloV3 was randomly initiated prior to training.

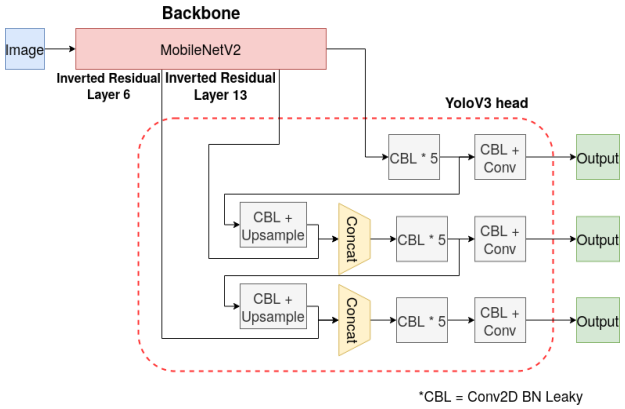

Figure 10. Proposed custom MobileNetV2 + YoloV3.

Considering the nature of our dataset indicates the usage for autonomous vehicle object detection, given that it will operate with constrained computing resources during inference time, we might also use an architecture that deals with constrained resources. Fig. 10 shows the implemented model. The CBL in the figure is an abbreviation for "Conv2D BN Leaky," a basic component in YoloV3 comprising a convolutional (Conv2D) layer, a Batch Normalization (BN) layer, and an activation

function (in this case, ReLU6 [32]). We will refer to the MobileNetV2 [31] and YoloV3 [15] papers for details of the models themselves.

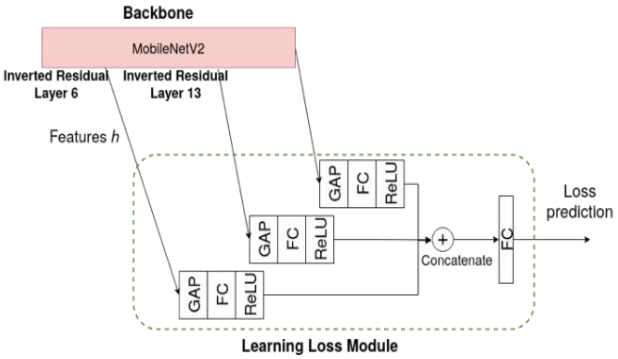

Figure 11. MobileNetV2 connected to the learning loss module.

Features *h* from MobilenetV2 were extracted and propagated through the learning loss module during training. As shown in Fig. 11, the features *h* are from the inverted residual layers 6 and 13, and the final output of MobileNetV2, which is the same place where the head of YoloV3 is connected. We extracted features from MobileNetV2 instead of YoloV3 because MobileNetV2 is responsible for image feature extraction, which is required by the learning loss model. This is different from the YoloV3 head considering the YoloV3 head is responsible for the localization of objects, and hence, will not be helpful to the learning loss module. This approach is similar to that of the original study [29].

C. Semi-Automated Active Learning Setup

Fig. 12 shows the implementation of the proposed SAAL system, which can be contrasted with the normal AL shown in Fig. 2. Considering the learning loss module, following the implementation in [29], all fully connected layers except the last one were set to output 128-dimensional features. Considering the last fully connected layer must output the loss prediction, it is set to output a 1-dimensional feature. The general workflow of the SAAL is as follows: First, we assume that we have already collected *N* number of unannotated data from a source, denoted as database *Y*. Although the data can be anything from videos to audio, but in this case, it was images. The database is then be referred to as Y_N^0, where 0 refers to the initial stage of the database. As a prerequisite, the oracle needs to annotate some of the data from the database. The annotated number of data is *K*, which is considered as the initial data used to train CNN_{AL}. By referring to our annotated training data as *T*, we acquire T_K^0 of the annotated training data and Y_{N-K}^0 database of the unannotated data at our disposal. The first data is annotated manually because the CNN model should first be trained with the subset of our dataset to understand its nature before selecting the informative data during data querying.

CNN_{AL} was trained using the transfer learning method [33] considering MobileNetV2 is a pre-trained network. The entire training was divided into three stages, where all of them received images of size 416×416 as the input. With a batch size of 64, the mini-batch size was set to 32 and *K* was set to 960. In the first stage, our model was trained with T_K^0 data for 30,000 batches with stochastic gradient descent as the optimization algorithm, where the learning rate, momentum, and weight decay were set to 0.001, 0.9, and 0.0005, respectively. In this stage, we only trained the head layer of our CNN, which is responsible for detecting objects in the image. We tracked the performance of the model and saved the weight that resulted in the highest mAP reading during validation.

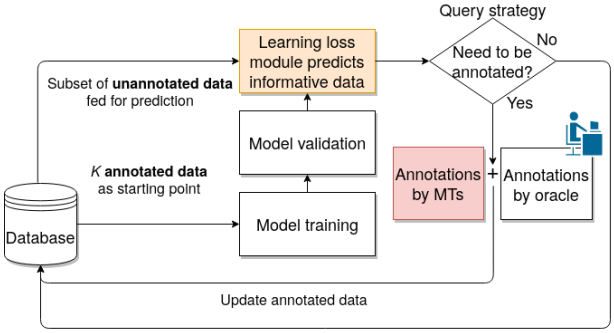

Figure 12. Proposed SAAL system.

In the second stage, the model was continuously trained using the saved weight from the first stage. While it was trained similarly as in the first stage, all of the layers were unfrozen and with a batch number of 50,000. The weight that resulted in the highest mAP reading during validation was also saved. These two stages familiarize the CNN with the new data that will be fed.

Finally, the final crucial third stage of SAAL includes the training process similarly to that in [29]. We assumed one AL cycle is completed every 300 epochs during training. Once an AL cycle is completed, we add another K data selected by the loss prediction module from the subset of data *J*. Therefore, every time the AL cycle is completed, our annotated training data will be $T_{2K}^1, T_{3K}^2, T_{4K}^3$ and so on. Furthermore, we trained the model with a batch size of 64 and mini-batch size of 32. After 240 epochs, the learning rate was reduced from 0.001 to 0.0001. Additionally, the loss prediction module was frozen to ensure it is not trained until the next AL cycle. In this experiment, we used $J = K * 10$ and $K = 960$ images, and trained them for 50,000 batches, which resulted in four AL cycles. As for the learning loss module hyperparameters, margins ζ and λ were set to 1.0, similar to that in [29] during evaluation of the learning loss module with ODT. As for the MTs, we extracted images at the 10th second of the videos and tracked 10 frames backward with the optical flow to suppress false positives. This framework is mainly built using PyTorch [34] and LightNet [35] libraries.

The difference between the SAAL and AL becomes clear during image annotation. The proposed framework SAAL first feeds the corresponding video of the image into the MTs. By using the video, MTs conduct annotation to the image as mentioned in the MT algorithm section. Furthermore, the oracle adds annotations as needed. Although we mentioned that the oracle will perform the annotations, we simulate the process by considering the ground truth annotations provided with the dataset.

V. Results and Discussion

To the best of our knowledge, no other methods directly help the oracle to annotate images. Therefore, we only compared the results of the proposed SAAL system with the previous method of normal AL.

Fig. 13 and Fig. 14 show the Average Precision (AP) and Mean Average Precision (mAP) results of the three experiments, respectively. The mAP reading is the mean of the AP reading from the three object classes. While experiment 1 shows the result of pure AL, where 100% of the data was annotated by the oracle, experiment 2 shows the result of the proposed SAAL, where human intervention was included to add needed annotations that went undetected by MTs. Experiment 3, which shows the result of SAAL but with data annotations only provided by the MTs without human intervention (a fully automated SAAL), was conducted to ensure the completeness of the evaluation.

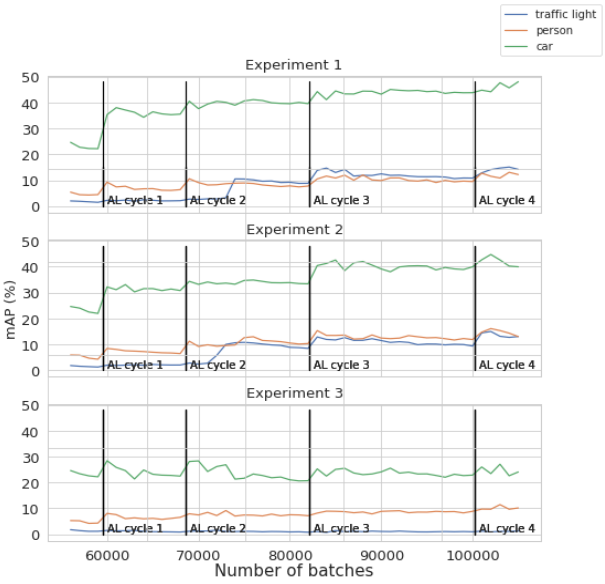

Figure 13. Average precision of each experiment.

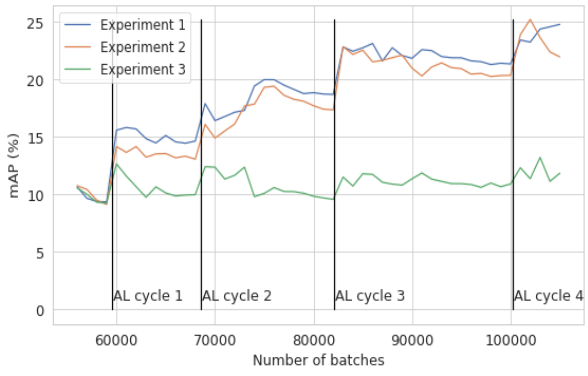

Figure 14. Mean average precision of each experiment.

Experiments 1 and 2 showed promising results. Both mAP values increased in every AL cycle and were comparable to each other. As shown in Fig. 15, we compared the number of works that the oracle needed to manually add annotations to the images in both experiments for every AL cycle. Results showed that the number of works the oracle needed with SAAL was lower compared to normal AL owing to MTs ability to perform annotations automatically. Furthermore, we found out that the total annotations needed to be performed by the oracle in SAAL and AL were 33,848 and 41,202, respectively. Therefore, compared to AL, the proposed SAAL system was able to reduce oracle work by 17.85%. Full-scale data acquisition and annotation from video sources can be tedious, and the effect of lowering oracle work can be very beneficial. However, we believe that MTs can be further improved by stacking more algorithms. For example, multiple CNNs can be used for CNN parts of MTs. Another conventional image processing method can also be implemented alongside optical flow to ensure false positives can be suppressed without mistakenly deleting true positives. Based on one usage, it could take some time to implement better MTs.

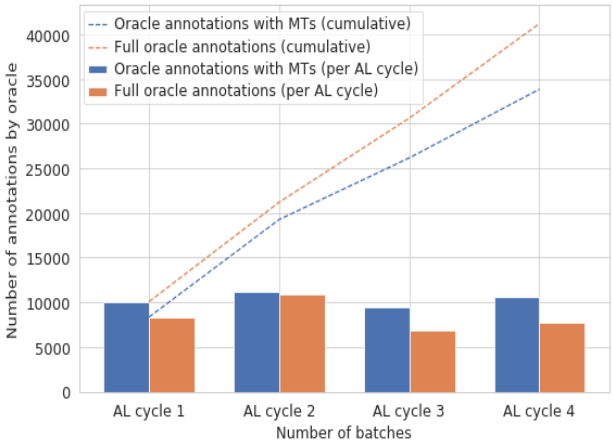

Figure 15. Number of annotations by the oracle.

Conversely, in Experiment 3, every new AL cycle did not result in an increase in mAP reading, as shown in Fig. 13 and Fig. 14. Further inspection showed that the problem was attributed to the low number of annotations produced by the MTs. Although all experiments were constantly being added with K=960 images after each AL cycle, the annotations associated with the images in Experiment 3 were much lower than those in Experiments 1 and 2. For example, compared to Experiment 1, Experiment 3 used 27,371 lesser annotations in total, which explains the failure of Experiment 3 to produce good results. We can understand that human intervention is still needed in the SAAL system; however, it can be lowered by improving the MT performance.

Considering the relatively low mAP reading in three of the experiments, results indicated that this can be attributed to the nature of the dataset, which are videos taken from the video camera on vehicles. Naturally, cars are abundantly projected in the video, which results in an imbalanced distribution of object classes. Traffic lights and people will have a much lower number of instances and be very small, as shown in Fig. 9. Other object classes in the BDD dataset, but not included in this study, also has the same problem where the number of annotations is too low or the size is too small, which is challenging for datasets that are addressed separately.

One might argue that the complexity introduced by MTs is worth a ~18% reduction in workloads. We believe this is justified, considering the larger the total number of datasets is, the larger the number of works that can be reduced. For example, if there are 100,000 annotations that need to be done, the work can be reduced to approximately 82,000. In addition, MTs are dynamic, which means that their performance can be improved over time. Because the CNN model can be retrained with new data over time, they can exhibit improved performance with time and can manage to help the oracle significantly.

VI. Limitations and Future Works

However, SAAL has several limitations. The plugged-in MTs are dependent on the CNN used. By using a pre-trained CNN, common objects such as cars and human beings can be easily detected, followed by reducing the number of false positives. However, this method will not work when rare object classes are involved, considering a CNN pre-trained for rare object classes may not exist. This problem can be fixed by manually annotating object classes in interest followed by manually training a CNN for that particular object class until an acceptable accuracy is obtained. The CNN can then be improved over time using annotations extracted during SAAL.

Although MTs can suppress false positives, they can also delete true positives, resulting in an increased number of false negatives. MTs should reduce the oracle's work by presenting the oracle with already annotated images during annotation, and not just blank un-annotated images. Therefore, images should be presented in a form where the false positives are suppressed as much as possible. The deletion of true positives by mistakes is unavoidable, where the oracle needs to carry out his/her job and annotate all the false negatives. This can be improved by optimizing the MT algorithm for the specific task and the dataset.

Additionally, building MTs and optimizing their performance is challenging and requires trial and error efforts, such as obtaining the best combination of conventional image processing methods and adjusting their parameters. However, if the MTs and SAAL are built in a modular fashion in a framework that can be scaled, it could benefit in the long run. MTs can be improved regularly based on feedback from the oracle or based on new project requirements. Therefore, we are currently researching a methodology to develop MTs.

Lastly, the proposed MTs and SAAL can be improved. Once our CNN_{AL} is sufficiently trained, it can be used as a CNNMT. Then, instead of the loss prediction module, MTs can be directly used for query strategy. This can be achieved by comparing the detection at frame t by CNN_{MTs} with frame t-n, where the false positives have been suppressed with optical flow. Furthermore, we can quantize the informativeness of frame t by comparing the number of false positives present in frame t. The more the number of false positives is, the more informative the frame would be. This can be explored in a separate study. Although this research focuses on the usage of MTs and SAAL with the road scene video dataset, it can easily be used to evaluate other types of datasets involving videos.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

M. H. H. Mohammad developed the idea of the research, built the necessary tools, processed the data, conducted the experiments, and analyzed the data. T. Ito supported and provided necessary feedback and ideas for the entire research paper.

Acknowledgment

The authors wish to thank Yasutaka Okada, Hiromi Rei, and Ogishima Aoi from Denso Ten Limited for their support, ideas, feedback, and assistance in developing MTs and the SAAL framework.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886-893.
[2] L. Weixing, S. Haijun, P. Feng, G. Qi, and Q. Bin, "A fast pedestrian detection via modified HOG feature," in *Proc. 34th Chinese Controlled Conference*, 2015, pp. 3870-3873.
[3] Z. Sivic and V. Google, "A text retrieval approach to object matching in videos," in *Proc. Ninth IEEE International Conference on Computer Vision*, 2003, pp. 1470-1477.
[4] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of Keypoints," in *Proc. Workshop on Statistical Learning in Computer Vision*, 2004, pp. 1-22.
[5] L. Zhi-Jie, "Image classification method based on visual saliency and bag of words model," in *Proc. 8th International Conference on Intelligent Computation Technology and Automation*, 2015, pp. 466-469.
[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
[7] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. European Conference on Computer Vision*, 2006, pp. 404-417.
[8] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, *et al*., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 53, 2021.
[9] D. E. Rumelhart and J. L. McClelland, "Learning internal representations by error propagation, parallel distributed processing: Explorations in the microstructure of cognition," *Foundations*, pp. 318-362, 1987.
[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, *et al*., "Imagenet large scale visual recognition challenge," *Clinical Orthopaedics and Related Research*, 2014.
[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, *et al*., "Going deeper with convolutions," *Clinical Orthopaedics and Related Research*, 2014.
[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Clinical Orthopaedics and Related Research*, 2015.
[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Clinical Orthopaedics and Related Research*, 2015.
[14] J. Redmon and A. Farhadi, "Yolo 9000: Better, faster, stronger," *Clinical Orthopaedics and Related Research*, 2016.
[15] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *Clinical Orthopaedics and Related Research*, 2018.
[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, *et al*., "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science*, pp. 21-37, 2016.
[17] B. Settles, "Active learning literature survey," *Computer Sciences Technical Report*, 2009.

[18] N. Rubens, D. Kaplan, and M. Sugiyama, "Active learning in recommender systems," *Recommender Systems Handbook*, 2011, pp. 735-767.
[19] S. Das, W. K. Wong, T. Dietterich, A. Fern, and A. Emmott, "Incorporating expert feedback into active anomaly discovery," in *Proc. 16th International Conference on Data Mining*, 2016, pp. 853-858.
[20] Y. Siddiqui, J. Valentin, M. Nießner, and A. L. View, "Active learning with viewpoint entropy for semantic segmentation," *Clinical Orthopaedics and Related Research*, 2019.
[21] J. J. Gibson, *The Perception of the Visual World*, Houghton Mifflin, 1950.
[22] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th International Joint Conference on Artificial Intelligence*, 1981, pp. 674-679.
[23] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proc. Eleventh International Conference on Machine Learning*, 1994, pp. 148-156.
[24] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. SIGIR*, 1994, pp. 3-12.
[25] A, K, Mccallum, "Employing EM in pool-based active learning for text classification," in *Proc. 15th International Conference on Machine Learning*, 1998, pp. 350-358.
[26] H. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proc. COLT*, 1992, pp. 287-294.
[27] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2372-2379.
[28] D. Roth and K. Small, "Margin-Based active learning for structured output spaces," *Machine Learning: ECML*, pp. 413-424, 2006.
[29] D. Yoo and I. S. Kweon, "learning loss for active learning," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 93-102.
[30] Y. Fisher, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, *et al*., "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," *Clinical Orthopaedics and Related Research*, 2018.
[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and C. Liang-Chieh, "Mobilenetv2: Inverted residuals and linear bottlenecks," *Clinical Orthopaedics and Related Research*, 2018.
[32] A. Krizhevsk, "Convolutional deep belief networks on CIFAR-10," unpublished manuscript, 2010.
[33] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345-1359, 2010.
[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, *et al*., "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024-8035, 2019.
[35] L. T. Ophoff, "Building blocks to recreate darknet networks in Pytorch," 2018.

Mohd Hafiz Hilman Mohammad Sofian received the B.S. degree in Engineering, M.S. degree in Systems Engineering and Science, and Ph.D. degree in Functional Control Systems from Shibaura Institute of Technology, Japan in 2017, 2019 and 2022 respectively.
He has experience collaborating with multiple Japanese companies in applying image processing techniques and Deep Learning in image domain for vehicles use. His research is mainly involving application of Convolutional Neural Network for autonomous driving system.

Toshio Ito received his B.A. in Engineering in 1982 and awarded Ph.D. in 1995 degrees in system engineering from Kobe University.
He joined Daihatsu Motor Co., Ltd, Japan in 1982, and had been working on the research and development of advanced driver assistance systems, and commercialized pre-crash safety system. He retired from the company on 2013. Currently, he is a Professor with the Department of Machinery and Control Systems, Shibaura Institute of Technology, Omiya Campus, Saitama City Japan. Prof. Ito current interests include the driver behavior and machine vision for advanced driving support systems.