Traffic Signal Classification with Cost-Sensitive Deep Learning Models

Tsz Shing Tsoi
College of Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida, USA

***Abstract*—Deep learning has many successful real-world applications including traffic signal recognition, which are used in driver assistance systems and autonomous vehicles. Accurate detection of traffic signal indications is critical to ensure safety under autonomous driving. Many past studies have been completed on traffic signal recognition including detection, classification and tracking with datasets which are typically highly imbalanced due to the nature of traffic signal displays. However, most studies simply ignored the minority classes and did not consider cost-sensitive information inherent to traffic signal indications. This paper evaluated several cost-sensitive techniques applicable to deep learning models in traffic signal classification. A convolutional neural network (CNN) was used in the evaluation as the baseline model. Cost-sensitive techniques including cost-proportionate rejection sampling and the use of cost-sensitive loss function was then applied to the baseline CNN model to evaluate and compare the effects of using cost information in traffic signal classification. Arbitrary cost information was assumed in this evaluation; however, it was found that cost information has the potential to alter predictions to minimize the overall cost when more appropriate cost information is used.**

Keywords—cost-sensitive, deep learning, traffic signal, traffic light, classification, sampling, loss function

I. Introduction

Deep learning has been applied to many industries and has had a lot of successes in many real-world applications such as image classifications, medical diagnosis, email spam filtering, fraud detection, online click prediction, and traffic sign recognitions, etc. The consequences of misclassifying one class into another class, such as false negative, may be significantly higher than if the misclassification is the other way around. For example, a cancer patient mis-identified as healthy may cause delay for the patient to seek treatment and the consequence could be life-threatening. Misclassification is also exacerbated by the inherent imbalanced nature of real-world datasets, as the number of positive class samples, e.g. cancer patients, spam emails, credit card frauds, is often significantly less than the negative class samples. Traffic signal recognition is an example of these real-world applications which suffers from these issues.

Automatic traffic signal recognition has practical applications in driver assistance systems (DAS) and autonomous vehicles [1]. Accurate detection and classification of traffic signals are critical in autonomous vehicle applications as vehicle control decisions depend on the detected traffic signal indications. While a green signal phase detected as red might cause vehicles to slow down or stop, the consequence of a red signal phase detected as green could be catastrophic. Unlike other image classification tasks, multiple traffic signals at different signal phases could be present in an image frame, and they typically occupy very small regions out of the entire image frame, which present another challenge in traffic signal recognitions.

As indicated in [1] – [6], traffic signal recognitions could be broken down into three stages: detection, classification and tracking. Detection concerns with identifying the locations of all traffic signals in an image frame, which are used in the classification stage to differentiate signal phases such as red, yellow or green. Tracking uses spatial information from previous image frames to improve detection and classification accuracies. Various techniques were used in past studies for traffic signal recognitions such as component, feature and blob (e.g. color, shape, edges) analyses, machine learning models and neural networks/deep learning models [1] – [13]. However, misclassification cost or imbalanced datasets were not considered during the classification stage in these studies.

Traditional machine and deep learning models are cost-insensitive models which treat each class equally with the assumption of balanced dataset as inputs [14] – [16]. These models do not assume higher misclassification cost of one class over another class and will favor the major class in the classification output if the datasets are imbalanced. Therefore, past researches focusing on cost-sensitive deep learning models are often associated with imbalanced datasets.

Cost-sensitive deep learning researches can be categorized into several different ways. Some researches focus on binary classification [14] – [18] while some focus on multi-class classifications [19]. The cost in deep learning models could be associated with each class with the use of cost matrix [16], or it could be associated with each example with the use of cost vector [19]. Techniques used in past researches to incorporate cost information into deep learning models could be broadly divided into three categories: 1) data sampling to increase the number of minor class samples and/or decrease the number of major class samples without modifying the deep learning models [15], [16]; 2) modifying the algorithms in deep learning models with cost-sensitive loss functions and/or weights [14] – [16], [18], [19]; and 3) incorporating cost information in class prediction to favor the minor class or minimize the expected cost instead of predicting the class with the highest probability [16], [17].

This paper evaluates and compares several cost-sensitive deep learning modeling techniques used in traffic signal classification. The rest of this paper is organized as follows:

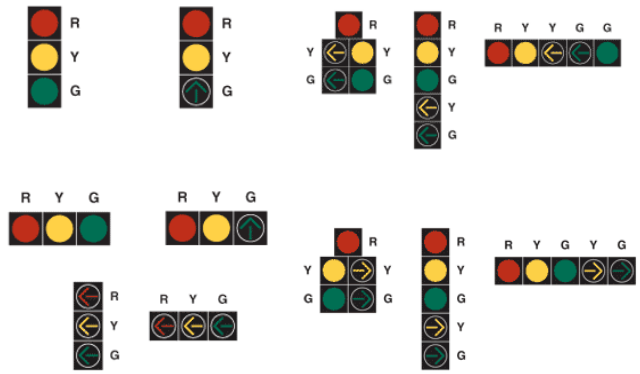
Section II provides a brief background on traffic signal displays; Section III summarizes related works in traffic signal classifications with deep learning models and cost-sensitive deep learning techniques; Section IV discusses the evaluation methodologies of traffic signal classification using cost-sensitive deep learning models; Section V summarizes the experiments and discusses the evaluation results; and Section VI includes the conclusion and discusses future works.

II. Background on Traffic Signal Displays

Although traffic signal displays vary across the world, they are typically standardized within the same region, and green indications typically represent "go", red indications represent "stop" and yellow indications represent "ready to go/stop". In the United States, traffic signals are standardized in Manual on Uniform Traffic Control Devices for Streets and Highways (MUTCD) [20], which governs the types, number, shapes, sizes, arrangements and placement of traffic signal indications (i.e. each individual section within a signal face) and signal faces (i.e. collection of signal indications in a signal display). Examples of possible traffic signal indications and arrangements of traffic signal faces are illustrated in Fig. 1. While most traffic signal indications are 12 inches in diameter, traffic signal indications with 8-inch diameter are allowed and used under certain circumstances.

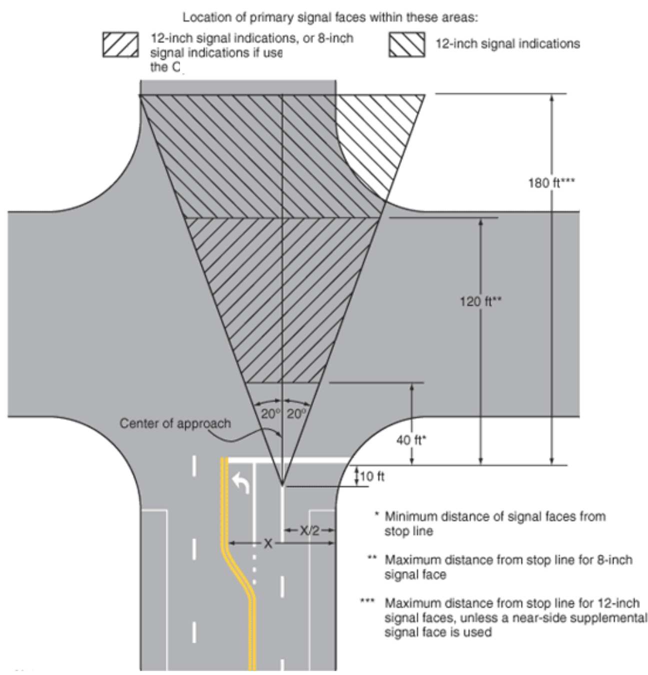
Most traffic signal displays have circular indications. Due to the characteristics of signal phasing operations, these indications are either red or green for majority of the time while yellow indications are shown for 3 to 6 seconds only for each traffic signal display in each traffic signal cycle. The number of traffic signal displays with left arrows are generally less than circular displays, and indications with left green or yellow arrows are even rarer, not to mention traffic signal displays with right or straight through arrows or other types of traffic signal displays used for bicycles and transit vehicles, and pedestrian [1] and railroad crossings. Therefore, most datasets in traffic signals are generally highly imbalanced [1] – [5], [12], [13].

Placements of traffic signal faces are also standardized by MUTCD and are located at the far side of the intersection between 40 feet and 180 feet from the stop line with a lateral viewing angle of approximately 20 degrees to the left and right of the center, as illustrated in Fig. 2. However, the use of supplemental traffic signal faces located at the near side of the intersection may not subject to these requirements. The mounting heights of traffic signal faces vary among states but are generally subject to the requirements in MUTCD with heights between approximately 15 feet and 22 feet when located above roadways or between 8 feet and 19 feet when located above sidewalks, as measured from the bottom of the traffic signal faces. These requirements on traffic signal displays mean that the locations of traffic signal display are limited to certain areas in an image frame. As illustrated in [1, Fig. 11], all traffic signal displays are located in the top two-third of the image frames with most gravitated towards the center and center-right for the traffic signal dataset collected in San Diego, California.



Sources: Manual on Uniform Traffic Control Devices for Streets and Highways [20]

Fig. 1. Examples of traffic signal indications and faces



Sources: Manual on Uniform Traffic Control Devices for Streets and Highways [20]

Fig. 2. Placement of traffic signal faces

III. Related Work

A. Traffic Signal Classification with Deep Learning

Past researches have proposed and used various techniques in traffic signal detection and classification. Many studies, as summarized in [1] and used in [3] and [9] – [11], used heuristic model-based methods such as color thresholding, connected component analysis, blob analysis, spot light detection, or even prior knowledge in traffic signal detection; while traffic signal classification was done based on color or machine learning models (e.g. support vector machine, k-nearest neighbor) with a few using neural networks. While these techniques have generally achieved great success in traffic signal detection and classification, other researches have attempted to use deep learning models, particularly convolutional neural networks

[1] Pedestrian hybrid beacons

(CNN) or their variations, to perform traffic signal detection, classification or both.

In [2], a small CNN was used for classification and detection was based on modified YOLO object detector with the dense network removed; while YOLO was used for both detection and classification in [8] and [12], and with prior knowledge map incorporated in [12]. In [4], a modified CaffeNet was used for classification on high dynamic range images and the results are compared with results evaluated from YOLO. In [7], a modified ResNet-50 version of Faster R-CNN was used for both traffic signal and traffic sign detection and classification. In [13], a CNN was proposed for both detection and classification based on the concepts of object localization and detection in the OverFeat feature extractor [21] with the network structure taken from AlexNet. In [5], LeNet and AlexNet were used for classification; and in [6], a small CNN was used for classification.

None of these researches on traffic signal classification considered cost information or issues with imbalanced datasets. In most cases, classes with limited sample sizes were either ignored or consolidated. In [3] and [5], only samples with circular red and green indications were included while other classes were ignored; while in [2], only samples with circular red, yellow and green indications and indications in the "off" state were included due to limited sample size in other classes. In [12], the datasets were consolidated to red (including circular and arrow red and yellow indications) and green (including circular and arrow green indications) classes, with other types of indications such as pedestrian and bicycle signals ignored even though they were available in the dataset. In [13], all types of red and green circular and arrow indications were categorized as red and green classes with the yellow indication ignored due to limited sample size.

B. *Cost-Sensitive Deep Learning*

One of the earlier papers [16] discussed the concept of cost-sensitive learning in a two-class scenario, which is based on cost matrix constructed for the true positive, true negative, false positive and false negative predictions. Several techniques in cost-sensitive learning were investigated, including 1) sampling to change the balance of positive and negative training examples; 2) use of an objective function which minimizes the expected cost; and 3) use of an adjusted probability threshold, calculated based on the cost matrix, to make the decision between positive and negative classes; and the author concluded that sampling has little effect on the learned classifier. Subsequent studies have proposed various techniques in cost-sensitive learning. Although with varying details, the techniques generally fall under these three categories:

1) Sampling: Sampling is a pre-processing technique which modifies the sample distribution to be used in cost-insensitive classification models to produce cost-sensitive classifers. However, straightforward sample-with-replacement typically has minimal effect on the classifers [15], [16]. In [15], the authors proposed cost-proportionate rejection sampling, which accepts each example with a probabilty proportional to the cost of each example, together with aggregation creating an ensemble of classifiers trained with different training samples.

2) Cost-sensitive loss functions: Cost information is incorporated in the loss function of the classification models so that parameters of the models are learned to consider cost of each training example or class. In [15], the authors proposed cost-weight adjustment to the objective function . In [19], the authors proposed a cost-sensitive loss function for mult-class classification with example-dependent cost vectors, incorporating the pre-training stage cross-entropy loss and training stage smooth one-sided regression loss. In [14], the authors proposed a cost-sensitive loss function for binary classification, which dynamically changes misclassification cost weight based on both global and local minibatch imbalances. In [18], the authors proposed two modified loss functions to minimize the mean false error and mean squared false error for training of imbalanced datasets for binary classification problems.

3) Class prediction with cost information: Incorporating cost information during class prediction is typically done at the output level to select the classification based on minimized cost instead of the highest probabilitiy. In [17], the authors proposed a cost-sensitive three-way decision modeling which determines the optimal decision (class) based on minimized misclassification and delay costs.

IV. Evaluation Methodologies

The objective of this paper is to evaluate and compare several cost-sensitive deep learning modeling techniques used for traffic signal classification. The evaluation is performed as follows: a baseline cost-insensitive model is first developed to serve as the benchmark, then several cost-sensitive techniques are applied to develop cost-sensitive classification models, and lastly the selected performance metrics are compared between these models.

Cost information is generally not available for traffic signal classification problems since the datasets are typically in the format of image sequences. The cost of misclassifying traffic signal indications could be associated with the potential safety cost. Classifying a green arrow indication as circular green may not create serious safety concerns, but classifying a yellow or red indication as green indication may result in potential crashes depending on the time of day and the traffic and roadway conditions. However, determination of safety cost requires substantial engineering efforts and supplemental data on driving conditions, roadway and traffic characteristics which may not always be available. Therefore, for the purpose of this evaluation, the cost information is derived based on the dataset imbalances.

Details of the proposed methodologies and performance metrics are discussed below.

A. *Baseline Model*

Although a number of past studies have use YOLO object detector or other deep learning object detection models for traffic signal recognition, the objective of this paper is to evaluate cost-sensitive traffic signal classification instead of object detection. Therefore, a simple CNN model, similar to the traffic signal classification model used in [2], is proposed in this evaluation as the baseline model. A simple classifier is also preferred since microcontrollers in vehicles generally have limited storage and processing power for real-time traffic signal classification for DAS and autonomous vehicle applications. Since traffic signals usually occupy very small regions in an image frame, the CNN used in this evaluation has input color images with size of 48 × 48 pixels. It consists of 3 convolutional layers with 2 max-pooling layers, and 3 fully connected layer with 2 drop out layers and a softmax layer as the output. A multi-class cross entropy loss function is used. Table I illustrates the CNN architecture.

B. *Cost-Sensitive Classification with Sampling*

The first cost-sensitive technique used for evaluation is sampling. Sampling is a preprocessing step used in the training phase to create a new dataset distribution to be used to train cost-insensitive classifiers [16]. A significant advantage with sampling is that no changes to any part of the cost-insensitive model are required to produce a cost-sensitive classifier. Cost-insensitive model typically favors major classes. Therefore, the approach to increase the weight on a particular class is through oversampling of that class or undersampling of other classes. However, oversampling would increase the training time and may cause with overfitting while undersampling may result in loss of information [14], [15], [18]. As indicated in [15] and [16], straightforward oversampling or undersampling generally does not have significant effect on the classifiers; therefore, cost-proportionate rejection sampling with ensemble aggregation, as suggested in [15], was used as the sampling methodology.

In cost-proportionate rejection sampling, a sample is accepted with a probability of c_s/Z where c_s is the cost of a sample *s* within the set of all samples *S*, and *Z* is a constant which satisfies the property $max_{s\in S}c_s \leq Z$. For the purpose of this evaluation, *Z* is assumed to be the maximum value of c_s as in:

$$Z = max_{s\in S}c_s. \tag{1}$$

The cost of each sample is derived based on the dataset imbalances, and is assumed to be inversely proportional to the number of samples for each class to which the sample belongs using (2), where c_k is the cost of sample *s* with label *y(s)* equal to class k within the set of all classes K, n_k is the number of samples which belongs to class *k*, *N* is the total sample size, and *a* is a positive constant used to adjust the relative cost ratio between classes.

$$c_s = c_k = \left(\frac{N}{n_k}\right)^a,\ where\ y(s) = k, \forall s \in S, k \in K \tag{2}$$

Cost-proportionate rejection sampling creates a sample set *S'* which is usually smaller than *S*; however, this reduced sample set *S'* is at least as informative as the original sample set *S* according to [15]. Also, this reduced sample set *S'* will contain approximately equal numbers of samples in each class when the cost for each sample is derived from (2) with the constant *a* equal to 1.

With reduced sample set *S'*, cost-proportionate rejection sampling allows the classification accuracy to be improved with ensemble aggregation without significantly increasing the computation time during training. The classification with ensemble learning is proposed to be based on (3), where *I* is the total number of classifiers, $P_i(k|s)$ is the probability of classifying sample s as class k in the *i*th classifier which is obtained from the softmax layer, and *h(k)* is the probability corresponding to class *k* with the highest confidence.

TABLE I. CNN ARCHITECTURE

Layer	Structure
Input	48 × 48 × 3
Convolution	Filters: 32, kernel: (7, 7), padding: 0, ReLU
Max-pooling	Kernel (2,2)
Covolution	Filters: 64, kernel: (3, 3), padding: 0, ReLU
Max-pooling	Kernel (2,2)
Convolution	Filters: 128, kernel: (3, 3), padding: 0, ReLU
Fully-conneccted	Units: 256, ReLU
Dropout	P=0.5
Fully-connected	Units: 128, ReLU
Dropout	P=0.5
Fully-conneccted	Units: 6, softmax

$$h(k) = \underset{k\in K}{\operatorname{argmax}} \sum_{i=1}^{I} P_i(k|s) \tag{3}$$

For comparison purpose, the aggregated number of samples used in this evaluation with cost-proportionate rejection sampling would be approximately equal to the total sample size in the original sample set. The ensemble aggregation learning algorithm is shown below.

Ensemble Aggregation (Sample Set *S*)

1 Set *m* = 0 and *i* = 1
2 While (*m* < *N*)
3   Create *S'* from *S* with rejection sampling
4   Train classifier *i* with *S'*
5   Set *m* = *m* + |*S'*| and *i* = *i* + 1
6 Classify sample *s* with (3)

C. *Classification with Cost-Sensitive Loss Function*

Another technique for cost-sensitive classification is to use a cost-sensitive loss function, which includes cost information to increase the emphasis on samples with higher cost. Since deep learning models are trained by minimizing an objective function, cost-sensitive loss function could easily be incorporated in deep learning models. Various cost-sensitive loss functions were proposed in past studies. However, the loss function *L(S)* shown in (4) is used for this evaluation, which is similar to the method proposed in [15] to incorporate weights directly into the learning algorithm. *l(s)* is the cross entropy for each sample *s*, and *Z* and c_s are previously defined in (1) and (2) respectively.

$$L(S) = \sum_{s\in S} \frac{c_s}{Z} l(s) \tag{4}$$

The overall evaluation workflow is identical to the baseline scenario except with the modified loss function.

D. Evaluation Metrics

In cost-sensitive classification, the goal is typically to minimize misclassification cost; and therefore, the overall misclassification cost would normally be the choice as the evaluation metrics. However, since the cost used in this evaluation is based on dataset imbalance, the metrics selected are related to the classification correctness with respect to each class. The most common evaluation metrics used in general machine or deep learning especially in traffic signal recognition include precision, recall and accuracy [1], [4], [5], [11] – [14], [21]. In this evaluation, the precision, recall and accuracy metrics for multi-class classifications defined in (5), (6) and (7) respectively are used as the evaluation metrics, where $n(j|k)$ is the number of samples with predicted class j given the actual class k.

$$\text{Precision} = \frac{1}{|K|}\sum_{j\in K}\frac{n(j|j)}{\sum_{k\in K} n(j|k)} \quad (5)$$

$$\text{Recall} = \frac{1}{|K|}\sum_{k\in K}\frac{n(k|k)}{\sum_{j\in K} n(j|k)} \quad (6)$$

$$\text{Accuracy} = \frac{\sum_{k\in K} n(k|k)}{N} \quad (7)$$

V. Experiments

The evaluation and comparison of various cost-sensitive techniques were performed using the Traffic Light Dataset from Laboratory for Intelligent and Safe Automobiles (LISA) at University of California, San Diego [1], [11] with the evaluation methodologies described in Section IV. The rest of this section discusses the dataset, evaluation tools, parameters and analysis results.

A. Dataset

Most datasets used in past studies on traffic signal recognition were local datasets and are not publicly available for research purposes. Although there are several annotated traffic signal datasets publicly available, LISA Traffic Light Dataset is one of the few which were recorded on roadways within the United States. The LISA dataset was from videos recorded in San Diego during both day and night times with over 100,000 traffic signal annotations. The dataset consists of 7 classes and is divided into training and test sets with over 50,000 annotations each. The distribution of this dataset is summarized in Table II, which shows that this dataset is highly imbalanced with each minor class comprising between 0.6% to 2.4% of the total training samples. Since the class "go forward" is only available in the test set but not in the training set, this class is ignored in this evaluation.

The entire image frame is 1,280 × 960 pixels, and a square image of the traffic signal is cropped out from each traffic signal annotation for training and testing. This allows various types of backgrounds to be included in the input images. The cropped images are relatively small. In the training set, approximately 27,000 cropped images out of approximately 52,000 traffic signal annotations are larger than 48 × 48 pixels; and in the test set, approximately 13,000 cropped images out of approximately 57,000 traffic signal annotations are larger than 48 × 48 pixels.

TABLE II. LISA Traffic Light Dataset Distributions

Dataset	**Classes**							**Total**
	Go	***Go Left***	***Stop***	***Stop Left***	***Warn-ing***	***Warn-ing Left***	***Go For-ward***	
Training	22,946	1,236	18,382	7,707	1,258	297	0	51,826
Test	23,777	1,240	25,936	5,027	1,411	53	205	57,649

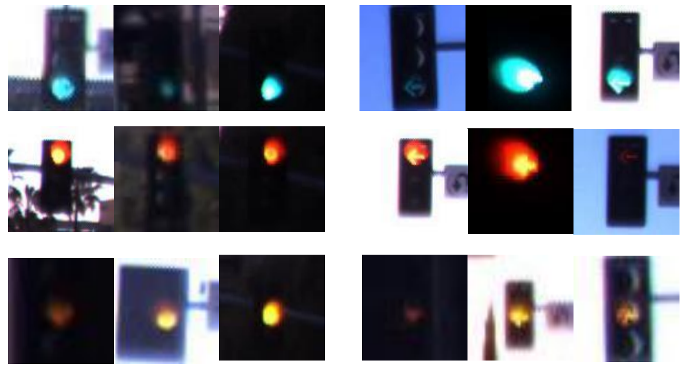


Fig. 3. Examples of traffic signal input data

All traffic signals in this dataset are vertically arranged. Fig. 3 shows some examples of the input data.

B. Evaluation Tools and Parameters

The CNN models developed for various scenarios were implemented with Keras 2.2.5.0 using Tensorflow 1.15 as the backend in RStudio with R version 3.6.1. The training set is divided into 80% for training and 20% for validation. Table III summarizes the key hyperparameters used in the CNN models for various scenarios. A constant *a* of 0.5 in (2) was used to derive the cost information.

C. Evaluation Results

A summary of the evaluation results for the 3 scenarios are included in Table IV. The confusion matrix for each scenario are included in Tables V, VI and VII. Accuracy is similar across all 3 scenarios. While the baseline model achieved the highest overall precision, recall is the highest in the scenario with rejection sampling. Comparison of the confusion matrices

TABLE III. Key Hyperparameters in CNN Models

Parameters	**Training Scenarios**		
	Baseline	***Rejection sampling***	***Cost-sensitive loss function***
Sample Size	41,457	~8,000[a]	41,457
Optimizer	Stochastic gradient descent		
Learning Rate	Schedule with initial learning rate of 0.001 and factored by 0.5 every 10 epochs		
Momentum	0.6 with Nesterov momentum applied		
Batch size	25	10	25
Epochs	40	50	40
Image Distortion	Rescale (1/255), rotation, vertical and horizontal shift, brightness adjustment, zoom, horizontal flip		

a. 7 ensemble classifiers are used with a sample size of approximately 8,000 to train each classifier

TABLE IV. SUMMARY OF EVALUATION METRICS

Scenarios	Evaluation Metrics		
	Precision	***Recall***	***Accuracy***
Baseline	**67.0%**	70.4%	**88.3%**
Rejection sampling	64.7%	**74.8%**	87.8%
Cost-sensitive loss function	60.4%	70.4%	87.4%

TABLE V. CONFUSION MATRIX FOR BASELINE MODEL

		Actual Class					
		Go	***Go Left***	***Stop***	***Stop Left***	***Warn-ing***	***Warn-ing Left***
Predicted Class	***Go***	**23,261**	784	16	10	0	0
	Go Left	390	357	0	0	0	0
	Stop	40	0	24,540	3,772	46	0
	Stop Left	12	2	1,348	**1,240**	8	0
	Warn-ing	64	97	32	4	**1,293**	8
	Warn-ing Left	10	0	0	1	64	45

TABLE VI. CONFUSION MATRIX WITH REJECTION SAMPLING

		Actual Class					
		Go	***Go Left***	***Stop***	***Stop Left***	***Warn-ing***	***Warn-ing Left***
Predicted Class	***Go***	22,434	611	25	7	0	0
	Go Left	1,153	**589**	11	6	0	0
	Stop	80	3	**24,970**	3,815	77	0
	Stop Left	17	0	838	1,180	4	0
	War n-ing	67	37	89	15	1,227	0
	War n-ing Left	26	0	3	4	103	**53**

TABLE VII. CONFUSION MATRIX WITH COST-SENSITIVE LOSS FUNCTION

		Actual Class					
		Go	***Go Left***	***Stop***	***Stop Left***	***Warn-ing***	***Warn-ing Left***
Predicted Class	***Go***	23,053	836	186	48	0	0
	Go Left	593	368	11	7	0	0
	Stop	50	6	24,584	3,872	33	0
	Stop Left	14	0	578	1,047	43	0
	Warn-ing	67	30	577	52	1,128	0
	Warn-ing Left	0	0	0	1	207	**53**

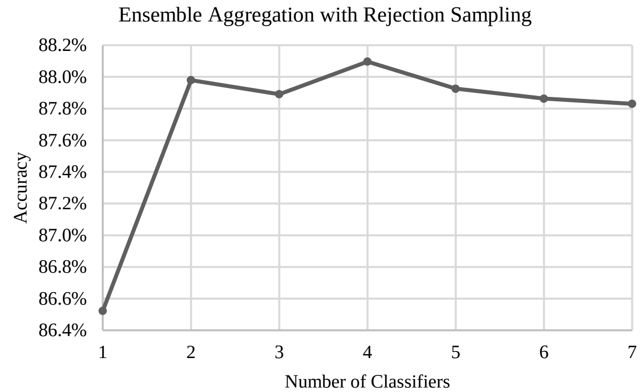


Fig. 4. Classification accuracy with ensemble aggregation

shows that cost-sensitive techniques generally have the effect to shift predictions to classes with higher costs. However, some of these shifts were not desirable, e.g. almost 5% of the "go" class samples were incorrectly predicted to be the "go left" classes with rejection sampling.

The scenario with rejection sampling is based on an ensemble of 7 classifiers with a training sample size of approximately 8,000 used to train each classifier. With one classifier, the precision is 61.9%, recall is 75.4% and accuracy is 86.5%, which are very comparable to the results with an ensemble of 7 classifiers. Fig. 4 illustrates the classification accuracy with respect to the number of classifiers with rejection sampling.

Considering each model is trained with limited sample size under the rejection sampling scenario, the model was able to generalize fairly well with limited but more balanced dataset. The selection of cost information in this evaluation was arbitrarily based on the imbalance ratios between these classes. More carefully crafted cost information and/or using other variations of cost-sensitive techniques may achieve more desirable results.

VI. CONCLUSION

This paper provides an overview of traffic signal recognition and various techniques used in past studies for traffic signal classification. It also summarizes various cost-sensitive modeling techniques used in past studies, which could be broadly divided into 3 categories: sampling, using cost-sensitive loss functions and class prediction with cost information. To evaluate and compare cost-sensitive techniques with traffic signal classification on imbalanced dataset, a simple CNN was used as the baseline model together with cost-proportionate rejection sampling and cost-sensitive loss function. Arbitrary cost information was assumed in this evaluation; however, it was found that cost information has the potential to alter predictions to minimize the overall cost when more appropriate cost information is used.

Future works include approaches to develop more appropriate cost information for traffic signal classification. Applying other cost-sensitive techniques and extending the cost-sensitive traffic signal classification approaches to traffic signal detection could be some of the research areas for future studies.

References

[1] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund and M. M. Trivedi, "Vision for Looking at Traffic Lights: Issues, Survey, and Perspectives," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800-1815, July 2016.

[2] K. Behrendt, L. Novak and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 1370-1377.

[3] X. Shen, H. Andersen, M. H. Ang and D. Rus, "A hybrid approach of candidate region extraction for robust traffic light recognition," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, 2017, pp. 1-8.

[4] J. Wang and L. Zhou, "Traffic Light Recognition With High Dynamic Range Imaging and Deep Learning," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1341-1352, April 2019.

[5] G. Lee and B. K. Park, "Traffic light recognition using deep neural networks," *2017 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2017, pp. 277-278.

[6] V. John, K. Yoneda, B. Qi, Z. Liu and S. Mita, "Traffic light recognition in varying illumination using deep learning and saliency map," *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, 2014, pp. 2286-2291.

[7] A. Pon, O. Adrienko, A. Harakeh and S. L. Waslander, "A Hierarchical Deep Architecture and Mini-batch Selection Method for Joint Traffic Sign and Light Detection," *2018 15th Conference on Computer and Robot Vision (CRV)*, Toronto, ON, 2018, pp. 102-109.

[8] M. B. Jensen, K. Nasrollahi and T. B. Moeslund, "Evaluating State-of-the-Art Object Detector on Challenging Traffic Light Data," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, 2017, pp. 882-888.

[9] A. Fregin, J. Müller and K. Dietmayer, "Feature detectors for traffic light recognition," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, 2017, pp. 339-346.

[10] T. H. Tran, C. C. Pham, T. P. Nguyen, T. T. Duong and J. W. Jeon, "Real-time traffic light detection using color density," *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, Seoul, 2016, pp. 1-4.

[11] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund and M. M. Trivedi, "Traffic Light Detection: A Learning Algorithm and Evaluations on Challenging Dataset," *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas, 2015, pp. 2341-2345.

[12] L. C. Possatti *et al*., "Traffic Light Recognition Using Deep Learning and Prior Maps for Autonomous Cars," *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 2019, pp. 1-8.

[13] M. Weber, P. Wolf and J. M. Zöllner, "DeepTLR: A single deep convolutional network for detection and classification of traffic lights," *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, 2016, pp. 342-348.

[14] Y. Geng and X. Luo, "Cost-sensitive convolutional neural networks for imbalanced time series classification", *Intelligent Data Analysis*, vol. 23, no. 2, pp. 357-370, 2019. Available: 10.3233/ida-183831.

[15] B. Zadrozny, J. Langford and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," *Third IEEE International Conference on Data Mining*, Melbourne, FL, USA, 2003, pp. 435-442.

[16] C. Elkan, "The foundations of cost-sensitive learning. In International joint conference on artificial intelligence," *Lawrence Erlbaum Associates Ltd.*, vol. 17, no. 1, pp. 973-978, August 2001.

[17] H. Li, L. Zhang, X. Zhou and B. Huang, "Cost-sensitive sequential three-way decision modeling using a deep neural network", *International Journal of Approximate Reasoning*, vol. 85, pp. 68-78, 2017.

[18] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, 2016, pp. 4368-4374.

[19] Y. A. Chung, H. T. Lin, and S. W. Yang, "Cost-aware pre-training for multiclass cost-sensitive deep learning," *arXiv preprint*, 2015.

[20] Federal Highway Administration, "Manual on Uniform Traffic Control Devices for Streets and Highways 2009 Edition", 2012.

[21] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint*, 2013.