Proceedings Paper:
Chen, R., Hawes, M., Isupova, O. et al. (2 more authors) (2017) Online Vehicle Logo Recognition Using Cauchy Prior Logistic Regression. In: 2017 20th International Conference on Information Fusion (Fusion). 20th International Conference on Information Fusion, 10-13 Jul 2017, Xi'an, China. IEEE . ISBN 978-0-9964-5270-0

https://doi.org/10.23919/ICIF.2017.8009720

Online Vehicle Logo Recognition Using Cauchy Prior Logistic Regression

Ruilong Chen[a], Matthew Hawes[a], Olga Isupova[a], Lyudmila Mihaylova[a] and Hao Zhu[b]

[a]Department of Automatic Control and Systems Engineering, University of Sheffield, S1 3JD, UK
[b] Chongqing University of Posts and Telecommunications, Chongqing, 400065, PR China
{rchen3, m.hawes, o.isupova, l.s.mihaylova}@sheffield.ac.uk, haozhu1982@gmail.com

***Abstract*—Vehicle logo recognition is an important part of vehicle identification in intelligent transportation systems. State-of-the-art vehicle logo recognition approaches typically consider training models on large datasets. However, there might only be a small training dataset to start with and more images can be obtained during the real-time applications. This paper proposes an online image recognition framework which provides solutions for both small and large datasets. Using this recognition framework, models are built efficiently using a weight updating scheme. Another novelty of this work is that the Cauchy prior logistic regression with conjugate gradient descent is proposed to deal with the multinomial classification tasks. The Cauchy prior results in a quicker convergence speed for the weight updating process which could decrease the computational cost for both online and offline methods. By testing with a publicly available dataset, the Cauchy prior logistic regression deceases the classification time by 59%. An accuracy of up to 98.80% is achieved when the proposed framework is applied.**

***Index Terms*—Vehicle Logo Recognition, Cauchy Prior, Online Learning, Conjugate Gradient Descent, Logistic Regression**

I. Introduction

As vehicle logos are among the most distinguishable marks on vehicles, recognising vehicle logos can help with vehicle identification [1]. Recently Vehicle Logo Recognition (VLR) has become a popular research topic in Intelligent Transportation Systems (ITSs) for traffic monitoring and vehicle management. For example, VLR can detect fraudulent plates if the combination does not match the data stored on the police security database [2]. In addition, identifying the vehicle logos around can give guidance for autonomous driving systems [3].

To the best knowledge of the authors, existing VLR frameworks train models on large fixed image training datasets. In practice, there may only be an initially small training dataset, with additional images becoming available during the implementation of the classification scheme. In order to take advantage of these additional images, new models can be built independently when the images become available.

However, retaining new models increases the computational cost, especially when the new models are updated frequently. In order to deal with this problem, this paper proposes a novel online framework for model learning, in which models are rebuilt efficiently using a weight updating scheme when dealing with datasets of an increasing size.

In image recognition, features rather than the raw pixel values are often used to represent an image. Features can be separated as global features and local features, where global features are generated from the whole image while local features only consider partial information of an image. In general, local features are more robust to image noise, distortions and scale variations [4] when compared with global features. They have been applied on VLR recently. For example the Scale invariant Feature Transform (SIFT) has been applied with various classifiers [1, 5–8].

However, local features need a representation process such as bag of words. The representation process involves a clustering process for the dictionary generation. When there are more images for training, the number of interest points increases which causes different clustering results. Hence, in different training stages an image is represented as different vectors. In the classification stage, for example using Support Vector Machine (SVM) and Logistic Regression (LR), weights are associated with the input vector. If an image is represented by irrelevant vectors from a different training stage, the corresponding weights will also be irrelevant. Therefore, local features cannot be applied to the online weights updating scheme.

The Histogram of Oriented Gradients (HOG) [9] feature is a global feature algorithm which has been applied to VLR [10, 11]. Unlike local features requiring the bag-of-words representation model before the classification stage, the HOG algorithm does not need this process and always gives the same vector length regardless of the training dataset size. Therefore, it can be used for online model updating in the classification stage.

Previous work [8] shows that the LR outperforms the SVM and K Nearest Neighbours classifiers in terms of accuracy.

In addition, LR can be easily extended for online model updating and it explores the confidence level of the decision that the data has been correctly classified [12]. However, when all of the training data can be perfectly classified, the LR suffers a common problem called separation. This is when the maximum likelihood gives infinite number of estimates which results in the regression becoming unstable [13, 14]. In order to have a generalised LR classifier without the separation problem, Gelman et al. suggest a Cauchy prior for LR and the posterior can be computed using Gibbs sampling [14]. However, this approach involves a high computational cost.

Carpenter proposes using the Stochastic Gradient Descent (SGD) to solve this problem and the Cauchy gradient is derived without considering a bias term in logistic regression [15]. However, the key disadvantage of SGD is that it requires manual tuning of parameters such as learning rates and stopping criteria [16].

Meanwhile, the Conjugate Gradient Descent (CGD), [17], automatically choose a learning rate which could avoid this problem [16]. This work combines the CGD with LR for both online and offline classification. The novelties of this work are as follows: 1) This paper gives a derivation of the maximum a posterior expression for the Cauchy prior LR based on CGD and extends it to a multinomial classifier with bias term. 2) An online classification scheme considering increasing training images is developed using the proposed classifier.

The rest of this paper is organized as follows. In Section II, a review of the HOG algorithm and the LR classifier are provided. Section III.A presents the proposed Cauchy prior on LR with CGD and how it is extend to multinomial classification. Section III.B presents the proposed online VLR framework for increasing dataset sizes. A performance evaluation is provided in Section IV and a summary is given in Section V.

II. Related Works

A. HOG descriptor

The gradient, illumination value difference between adjacent pixels, can be used to describe an image. HOG calculates the horizontal gradient G_x and the vertical gradient G_y on every pixel in the image using a 1-D filter [-1, 0, 1]:

$$G_{x(i,j)} = f(i+1,j) - f(i-1,j), \tag{1}$$
$$G_{y(i,j)} = f(i,j+1) - f(i,j-1), \tag{2}$$

where $f(i,j)$ is the intensity value at pixel location (i,j). Then the horizontal gradient and vertical gradient are used to calculate the orientation of gradients $\theta(i,j)$ and the magnitude of gradients $H(i,j)$ for every pixel in the image:

$$\theta(i,j) = arctan(G_{x(i,j)}/G_{y(i,j)}), \tag{3}$$
$$H(i,j) = \sqrt{G^2_{x(i,j)} + G^2_{y(i,j)}}. \tag{4}$$

The image is then divided into cells, where a certain number of cells form a block. A quantization process is applied, in which the orientations are quantized into bins in 0^o to 180^o. Hence, each cell can be represented as a histogram using the quantized orientations as the histogram bins and the magnitude of gradients as the weights. In each block, the histogram is normalized in order to be invariant to illumination, shadowing, etc. The HOG feature is the concatenation of the histogram vectors from all cells [9].

B. Logistic Regression

LR uses the maximum likelihood model and explores the confidence level of the decision that the data has been correctly classified [18]. Given a training image $(\mathbf{x}, y)$, where $\mathbf{x} \in \mathbb{R}^{M\times 1}$ is an image represented by the HOG algorithm and y is a scalar label. Note, $\mathbf{x}$ is a vector rather than a matrix as the images are represented using the HOG algorithm, as described above in Section II.A. The relationship between $\mathbf{x}$ and y is given by:

$$y = \mathbf{w}^T\mathbf{x} + b, \tag{5}$$

where $\mathbf{w} \in \mathbb{R}^{M\times 1}$ is the weight vector and the scalar b is the bias associated with the linear regression.

In binary classification y is a scalar which can either be '1' (positive) or '0' (negative). Using the 'logistic' function $f(x) = 1/(1+e^{-x})$, the probability that the training image belongs to class '1' can be expressed as:

$$s = p(y=1|\mathbf{w}, b) = f(\mathbf{w}^T\mathbf{x} + b) = \frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}}. \tag{6}$$

Therefore, the probability of a negative outcome is $1-s$, which is given by:

$$p(y=0|\mathbf{w}, b) = 1 - s = \frac{e^{-(\mathbf{w}^T\mathbf{x}+b)}}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}}. \tag{7}$$

The likelihood of all of the training labels is therefore given by the product:

$$p(\mathbf{y}|\mathbf{w}, b) = \prod_{i=1}^{N} s_i^{y_i}(1-s_i)^{1-y_i}, \tag{8}$$

where $\mathbf{y} \in \mathbb{R}^{N\times 1}$ is a vector representing all the training labels and s_i represents the probability that the i^{th} image belongs to the positive class. Maximizing the likelihood in Eq. (8) is equivalent to minimizing the negative of its Logarithm, which is given by:

$$\begin{aligned}E &= -ln\ p(\mathbf{y}|\mathbf{w}, b)\\ &= -\sum_{i=1}^{N} y_i ln s_i + \sum_{i=1}^{N}(1-y_i)ln(1-s_i)\\ &= -\sum_{i=1}^{N} y_i ln\ f(\mathbf{w}^T\mathbf{x}_i + b)\\ &\qquad - \sum_{i=1}^{N}(1-y_i)ln(1-f(\mathbf{w}^T\mathbf{x}_i + b)). \end{aligned} \tag{9}$$

The gradients with respect to $\mathbf{w}$ and b can be used for minimizing Eq. (9).

Note, that the logistic function has the following property:

$$\begin{aligned}f'(x) &= \frac{\partial}{\partial x}\left(\frac{1}{1+e^{-x}}\right)\\ &= \frac{1}{(1+e^{-x})^2}(e^{-x})\\ &= \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right)\\ &= f(x)(1-f(x)). \end{aligned} \tag{10}$$

This gives the gradient with respect to $\mathbf{w}$ as:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{w}} &= -\sum_{i=1}^{N}\frac{y_i}{f(\mathbf{w}^T\mathbf{x}_i + b)}f'\boldsymbol{x}_i + \sum_{i=1}^{N}\frac{1-y_i}{1-f(\mathbf{w}^T\mathbf{x}_i + b)}f'\boldsymbol{x}_i\\ &= -\sum_{i=1}^{N} y_i(1-f(\mathbf{w}^T\mathbf{x}_i + b))\mathbf{x}_i\\ &\qquad + \sum_{i=1}^{N}(1-y_i)f(\mathbf{w}^T\mathbf{x}_i + b)\mathbf{x}_i\\ &= \sum_{i=1}^{N}(f(\mathbf{w}^T\mathbf{x}_i + b) - y_i)\mathbf{x}_i, \end{aligned} \tag{11}$$

where f' represents the partial derivative of $f(\mathbf{w}^T\mathbf{x}_i + b)$ with respect to $\mathbf{w}$. In the same way taking the gradient with respect to b gives:

$$\frac{\partial E}{\partial b} = \sum_{i=1}^{N}(f(\mathbf{w}^T\mathbf{x}_i + b) - y_i). \tag{12}$$

The minimisation of (11) and (12) are usually solved by gradient descent method such as SGD [18] and Newton's method [19]. Notice that the LR is a maximum likelihood model which does not involve any prior information. However, when the maximum likelihood perfectly separates the training dataset, there are infinite possible solutions caused by the separation problem.

III. Proposed Framework for online VLR

A. Cauchy prior on LR and multinomial LR with CGD

A Cauchy prior on LR can avoid the separation problem. It assumes that the coefficients in LR are sparse, this could provide a quicker convergence in the gradient descent process. A zero mean Cauchy prior is assumed for the weights, this gives:

$$p(\mathbf{w}) = \frac{1}{\pi}\left(\frac{\gamma}{\mathbf{w}^2 + \gamma^2}\right), \tag{13}$$

where γ is a scale parameter. According to the Bayes' rule:

$$p(\mathbf{w}, b|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{w}, b)p(\mathbf{w})p(b), \tag{14}$$

where $\mathbf{w}$ and b are independent.

The weights are assumed sparse which makes the majority of the weights zero (or close to zero) valued. However, b is the intercept of the decision line which does not have any prior knowledge associated with it. As a result, here assume b is controlled by a non-informative prior. Therefore, maximizing the posterior $p(\mathbf{w}, b|\mathbf{y})$ is equivalent to maximising:

$$p(\mathbf{y}|\mathbf{w}, b)p(\mathbf{w}) = \frac{1}{\pi}\left(\frac{\gamma}{\mathbf{w}^2 + \gamma^2}\right)\prod_{i=1}^{N} s_i^{y_i}(1-s_i)^{1-y_i}. \tag{15}$$

Maximizing the likelihood in (15) is equivalent to minimizing the negative of its logarithm, which is given by:

$$\begin{aligned}E &= -ln\,(\ p(\mathbf{y}|\mathbf{w}, b)p(\mathbf{w}))\\ &= -\sum_{i=1}^{N} y_i ln s_i - \sum_{i=1}^{N}(1-y_i)ln(1-s_i)\\ &\qquad - ln(\gamma) + ln((\mathbf{w}^T\mathbf{w} + \gamma^2)\pi)\\ &= -\sum_{i=1}^{N} y_i ln f(\mathbf{w}^T\mathbf{x}_i + b) + ln((\mathbf{w}^T\mathbf{w} + \gamma^2)\pi)\\ &\qquad - \sum_{i=1}^{N}(1-y_i)ln(1-f(\mathbf{w}^T\mathbf{x}_i + b)) - ln(\gamma). \end{aligned} \tag{16}$$

In order to minimize (16), taking the gradient with respect to $\mathbf{w}$ gives:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{w}} &= -\sum_{i=1}^{N}\frac{y_i}{f(\mathbf{w}^T\mathbf{x}_i + b)}f'(\mathbf{w}^T\mathbf{x}_i + b)\mathbf{x}_i + \frac{2\mathbf{w}}{\mathbf{w}^T\mathbf{w} + \gamma^2}\\ &\qquad + \sum_{i=1}^{N}\frac{(1-y_i)}{1-f(\mathbf{w}^T\mathbf{x}_i + b)}f'(\mathbf{w}^T\mathbf{x}_i + b)\mathbf{x}_i\\ &= -\sum_{i=1}^{N} y_i(1-f(\mathbf{w}^T\mathbf{x}_i + b))\mathbf{x}_i + \frac{2\mathbf{w}}{\mathbf{w}^T\mathbf{w} + \gamma^2}\\ &\qquad + \sum_{i=1}^{N}(1-y_i)(f(\mathbf{w}^T\mathbf{x}_i + b))\mathbf{x}_i\\ &= \frac{2\mathbf{w}}{\mathbf{w}^T\mathbf{w} + \gamma^2} + \sum_{i=1}^{N}(f(\mathbf{w}^T\mathbf{x}_i + b) - y_i)\mathbf{x}_i. \end{aligned} \tag{17}$$

In the same way taking the gradient with respect to b gives:

$$\begin{aligned}\frac{\partial E}{\partial b} &= -\sum_{i=1}^{N} \frac{y_i}{f(\mathbf{w}^T\mathbf{x}_i + b)} f'(\mathbf{w}^T\mathbf{x}_i + b) \\ &\quad + \sum_{i=1}^{N} \frac{(1-y_i)}{1 - f(\mathbf{w}^T\mathbf{x}_i + b)} f'(\mathbf{w}^T\mathbf{x}_i + b) \\ &= -\sum_{i=1}^{N} y_i(1 - f(\mathbf{w}^T\mathbf{x}_i + b)) \\ &\quad + \sum_{i=1}^{N} (1-y_i)(f(\mathbf{w}^T\mathbf{x}_i + b)) \\ &= \sum_{i=1}^{N} (f(\mathbf{w}^T\mathbf{x}_i + b) - y_i). \end{aligned} \tag{18}$$

For minimising equations (17) and (18) we apply the same methods as for equations (11) and (12). This can be solved using gradient descent which update the weights iteratively:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta \frac{\partial E}{\partial \mathbf{w}^{(k)}}, \tag{19}$$

$$b^{(k+1)} = b^{(k)} - \eta \frac{\partial E}{\partial b^{(k)}}, \tag{20}$$

where η is a fixed learning rate which controls the speed of convergence and $k \in \{0, 1, \cdots, K\}$ is the iteration index.

One key disadvantage of gradient descent methods such as batch gradient descent and SGD is that a good learning rate is difficult to find [16]. In order to avoid this problem, this work proposes using the Cauchy prior LR with CGD, which automatically choose a learning rate in each iteration.

Denote all the variables as $\mathbf{v} = (b, \mathbf{w}^T)$. Therefore E in Equation (16) can be written as a function of $\mathbf{v}$, giving $E = g(\mathbf{v})$. For the first iteration, the gradient update for all the variables is:

$$\mathbf{v}_1 = \mathbf{v}_0 - \eta_0 \frac{\partial E}{\partial \mathbf{v}_0}, \tag{21}$$

where $\mathbf{v}_0$ represent the initial bias and weights (initialised as zero values) when $k = 0$. A line search is applied to find the initial learning rate [17]:

$$\eta_0 = \operatorname*{argmin}_{\eta} g\left(\mathbf{v}_0 - \eta \frac{\partial E}{\partial \mathbf{v}_0}\right). \tag{22}$$

For the following iterations where $k > 0$, gradients are along the conjugate directions. In order to avoid a zig-zagging path, the new gradient direction combines the gradient $-\frac{\partial E}{\partial \mathbf{v}_k}$ and the previous direction:

$$\mathbf{d}_{k+1} = -\eta_k \frac{\partial E}{\partial \mathbf{v}_k} + \beta_k \mathbf{d}_k, \tag{23}$$

with $\mathbf{d}_0 = -\frac{\partial E}{\partial \mathbf{v}_0}$. According to the Polak-Ribiere rule [20], the value of β_k is given by:

$$\beta_k = \frac{\left(\frac{\partial E}{\partial \mathbf{v}^k}\right)^T \left(\frac{\partial E}{\partial \mathbf{v}_k} - \frac{\partial E}{\partial \mathbf{v}_{k-1}}\right)}{\left(\frac{\partial E}{\partial \mathbf{v}_k}\right)^T \frac{\partial E}{\partial \mathbf{v}_k}}. \tag{24}$$

The gradient update process is:

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \eta_k d_k \tag{25}$$

and a line research is applied to find the optimal learning rate:

$$\eta_k = \operatorname*{argmin}_{\eta} g\left(\mathbf{v}_k + \eta \mathbf{d}_k\right). \tag{26}$$

For a new testing image $\mathbf{x}^*$, the probability that it belongs to the positive class is:

$$p(y^* = 1|\mathbf{w}, b) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}^* + b)}} \tag{27}$$

and the probability that it belongs to the negative class is therefore:

$$p(y^* = 0|\mathbf{w}, b) = 1 - p(y^* = 1|\mathbf{w}, b). \tag{28}$$

Here y^* represents the predicted label for the testing image. Hence, the testing image can be allocated into the class which has the higher probability.

The Cauchy prior LR in binary classification can be easily extended to multinomial classification. The training images are from C categories $y_i \in \{1, 2, \cdots, C\}$. In multinomial classification, the probability of $p(y_i = c|\mathbf{W}, \mathbf{b})$ for each $c = (1, 2, \cdots, C)$ can be denoted as:

$$\begin{bmatrix} p(y_i{=}1|\mathbf{W},\mathbf{b}) \\ p(y_i{=}2|\mathbf{W},\mathbf{b}) \\ \vdots \\ p(y_i{=}C|\mathbf{W},\mathbf{b}) \end{bmatrix} = \frac{1}{\sum_{c=1}^{C} e^{(\mathbf{w}_c^T\mathbf{x}_i + b_c)}} \begin{bmatrix} e^{(\mathbf{w}_1^T\mathbf{x}_i + b_1)} \\ e^{(\mathbf{w}_2^T\mathbf{x}_i + b_2)} \\ \vdots \\ e^{(\mathbf{w}_C^T\mathbf{x}_i + b_C)} \end{bmatrix}, \tag{29}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_C]$ is a matrix consisting of the weights and $\mathbf{b} = [b_1, b_2, \cdots, b_C]$ is the bias of the multi-class LR models. The term $\sum_{c=1}^{C} e^{(\mathbf{w}_c^T\mathbf{x}_i + b_c)}$ normalizes the distribution so that all of the probabilities sum up to one. Hence, for a testing image $\mathbf{x}^*$, the probability that its label y^* equals c is :

$$p(y^* = c|\mathbf{W}, \mathbf{b}) = \frac{e^{(\mathbf{w}_c^T\mathbf{x}^* + b_c)}}{\sum_{c=1}^{C} e^{(\mathbf{w}_c^T\mathbf{x}^* + b_c)}}. \tag{30}$$

The incoming testing image is then assigned to the class which has the highest probability.

B. Cauchy prior LR for increasing VLR training dataset size

In order to deal with training datasets that increase in size, the classifier needs to be retrained as more training images become available. However, rather than retraining different classifiers independently, the classifiers trained for the previous stages can be useful. Figure 1 shows the general process of retraining models when the size of training images are increasing. Using the HOG algorithm, each image is represented by a vector $\mathbf{x}$ and its label y. Algorithm 1 shows the offline method, which retrains a new model independently when additional training images arrive. More specifically, the

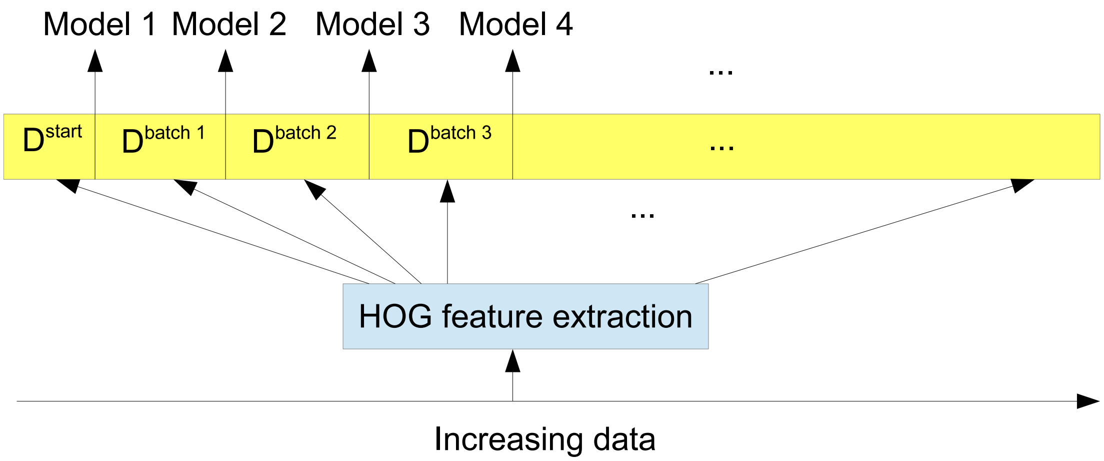

Fig. 1: The online recognition framework of VLR.

```
Algorithm 1 Framework of offline Cauchy prior LR
Input:
    The initial training images $\mathbf{D}^{start}$
    The sequential training images, $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_i, y_i), \cdots\}$, with $i$ is the index of the $i^{th}$ image
Output:
    The model parameters in LR and accuracies on the testing dataset
Apply the Cauchy prior LR on the initial training images $\mathbf{D}^{start}$ and save the initial model (Model1 in Figure 1)
for each $i = 1, i++$ do
    if i/(batch size)==int then
        Retrain a new model using the all available training images $\mathbf{D}^{ava} = \{(\mathbf{x}_1, y), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_i, y_i)\}$ with LR
    end if
    Use the retrained model to classify the testing images
end for
return The model parameters and accuracies
```

```
Algorithm 2 Framework of online Cauchy prior LR
Input:
    The initial training images $\mathbf{D}^{start}$
    The sequential training images, $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_i, y_i), \cdots\}$, with $i$ is the index of the $i^{th}$ image
Output:
    The model parameters in Cauchy prior LR and accuracies on the testing dataset
Apply the logistic regression on the initial training images $\mathbf{D}^{start}$ and save the initial model (Model1 in Figure 1)
for each $i = 1, i++$ do
    if i/(batch size)==int then
        Update the model using all available training images $\mathbf{D}^{ava} = \{(\mathbf{x}_1, y), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_i, y_i)\}$ with the previous $\mathbf{w}$ and $b$ are used as the initial start point of the model parameters.
    end if
    Use the updated model to classify the testing dataset
end for
return The weight vector and accuracies
```

initial model was trained using a small amount of training images $\mathbf{D}^{start}$. When there are extra training images available, $\mathbf{D}^{batch}$, the model is retrained using all the available images $\mathbf{D}^{ava}$. This now includes both the additional images and the previously available images. This process is repeated each time additional training images become available. The batch size is a parameter which controls how often the model is updated, i.e. the number of additional images required before retraining occurs.

Using the offline methods, models are retrained independently as $\mathbf{W}$ and $\mathbf{b}$ are initialized to zero. However, $\mathbf{W}$ and $\mathbf{b}$ from the previous models might be good initial points which could help the current model converges faster. Therefore, the current model can be updated based on previously trained model rather than a model retained independently. Algorithm 2 show the general process of online model updating.

IV. Performance Evaluation

In this section the open dataset provided by Huang et al [21] is used to evaluate the proposed classification framework. It has 10 categories and each category contains 1000 training images and 150 testing images, all with a size of 70×70 pixels. Figure 2 shows an example for each of the 10 vehicle categories by randomly choosing one image from each category in the training dataset. Figure 3 shows some challenging test images which can be easily misclassified.

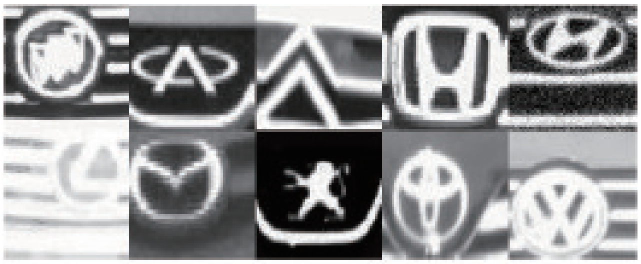

Fig. 2: Example logos from the dataset.

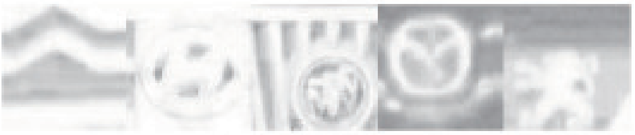

Fig. 3: Examples of some challenge images in the testing dataset.

The performance evaluation of the online Cauchy LR with HOG feature is conducted in Matlab 2015 on a computer with the following specification: Intel CPU I5-4590 (3.4Ghz) and 24GB of RAM. The proposed Cauchy prior LR is compared with LR and the Cauchy prior is evaluated for training datasets that increase in size. The performance of each method is measured in terms of accuracy (percentage of correctly classified images), total number of misclassified images and the computation time (to indicate the relative computational complexities). Accuracies and computational times are given as average values taken from 30 simulation runs.

A. Comparison of logistic regression and logistic regression with Cauchy prior

TABLE I: Accuracy comparisons between LR and Cauchy prior LR with different size of dataset (average value from 30 simulation runs).

Training size	100	500	1000	2000	3000	4000
LR (%)	67.05	91.02	96.07	98.11	98.56	98.67
misclassified images	494.24	135.70	58.95	28.35	21.60	19.95
Time (s)	4	23	49	94	135	179
Cauchy LR (%)	66.24	90.35	95.33	97.59	98.06	98.35
misclassified images	506.40	144.75	70.05	36.15	29.10	24.75
Time (s)	2	8	17	36	54	74
Training size	5000	6000	7000	8000	9000	10000
LR (%)	98.72	98.84	98.83	98.76	98.72	98.80
misclassified	19.20	17.40	17.55	18.60	19.20	18
Time (s)	216	261	302	339	386	437
Cauchy LR (%)	98.38	98.42	98.42	98.51	98.48	98.80
misclassified images	24.30	23.70	23.70	22.35	22.80	18
Time (s)	92	119	149	175	180	182

In our implementation, the HOG feature is different from the original HOG method in [9]. Here a histogram vector is built for each block rather than each cell and 12 bins with uniform spacings are applied on the angular range from 0^o to 180^o. The block window scans the whole image taking the size of a cell as the sliding size and the block window is made up by 2 by 2 cells and each cell is make from 5 by 5 pixels. These techniques give an improvement of accuracy more than 3% when the model is trained on the whole training dataset (from 93.53% to 97.13% when LR with CGD is applied). Each HOG feature vector is normalized with zero mean and the standard derivation is set to 1. This process is able to increase accuracy about 2% (from 97.13% to 98.80% when LR with CGD is applied).

Finding the learning rate is a difficult issue in SGD. Using the whole training dataset with the testing dataset as the validation data, the best accuracy SCD (95.35%) archived is about 3% lower when compared with CGD (98.80%). However, when applied in practice the testing dataset is not known in advance. As a result, it is not possible to find the learning optimal learning rate for use in classification. This means a further degradation in performance would be expected for methods based on SGD. In the following, the optimised HOG feature with normalization and CGD are applied in order to compare LR and Cauchy prior LR.

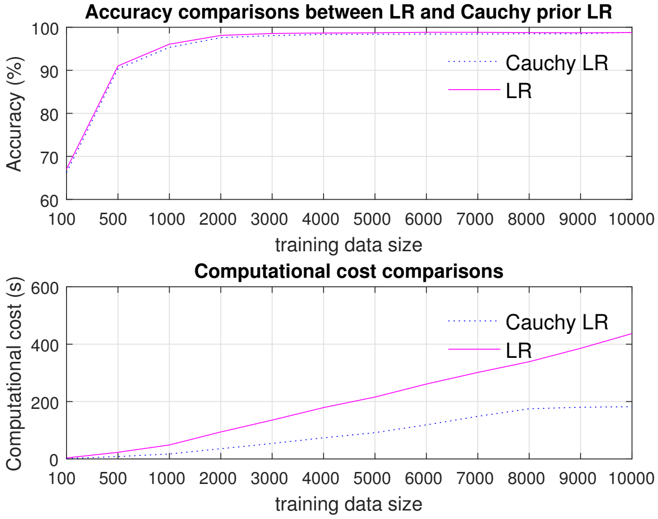

Fig. 4: Accuracy comparisons between LR and Cauchy prior LR with different size of dataset (average value of 30 simulation runs).

Different training dataset size are tested and the accuracies are evaluated on the complete testing data. The results are given in Table I and Figure 4. The accuracy of both classifier are close while the Cauchy prior LR has a significant reduction on computational cost. Take the training size equals 10000 as

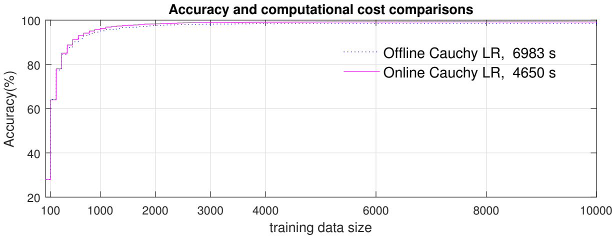

Fig. 5: Accuracy and computational cost results up to 10000 training images ($\mathbf{D}^{batch}$ = 100).

an example, the Cauchy prior is able to decrease the computational cost from 7 minutes (437 seconds) to approximately 3 minutes (179 seconds) when the whole dataset is applied, i.e, 59% reduction in computational cost. This can be explained by the prior information resulting in a quicker convergence. As a result, only LR with the Cauchy prior will be considered in the remaining comparisons of online and offline training that follow below.

B. Comparison of online and offline Cauchy prior LR

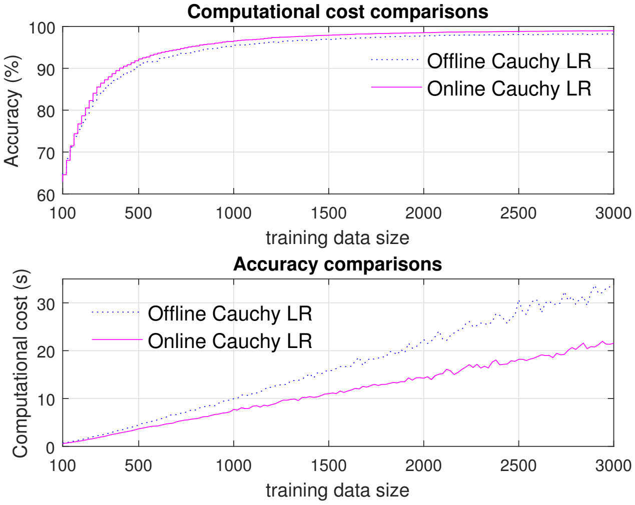

Fig. 6: Accuracy results up to 3000 random training images ($\mathbf{D}^{batch}$ =20).

In this section, the Cauchy prior LR is implemented for online learning. Figure 5 shows the performances of the online method and the offline method using the Cauchy prior LR. A random set of training images for each class are picked and used for the initial classifier training (the same initial set for each method). When the training size is increasing, training models are updated when the available number of training images meets the requirement described in Algorithm 1. Here the batch size $\mathbf{D}^{batch}$=100 is used. The accuracy is evaluated on the whole testing dataset. Offline method means the weights are retrained on the available images with all weights initialized to 0, while online method involves a weight initialization from the previous models.

It is shown in Figure 5 that the HOG features can achieve a good accuracy when there is a small dataset (90% accuracy is achieved when the training size around 500). After the training size above 2000, both accuracies become high and stable. The time in the figure shows the computation cost for the whole process, which includes the testing scenario and model updating process when the training dataset size is increased by 100. The online scheme reduced the computational cost by 33% which indicates that the weights initialization can help with the convergence in CGD.

In Figure 5 the dataset size is increased up to 10000, this involve high computation cost if the model updates frequently. However, Figure 5 indicates the accuracy becomes stable when the training size is above 2000. Therefore a more detailed comparison can be made by a smaller dataset while updating the model more frequently as a smaller batch size gives more comparison results. Figure 6 shows the more detailed results by setting $\mathbf{D}^{batch}$ =20 and the training size varies from 100 to 3000. It indicates that the online method provides a slightly higher accuracy and a quicker convergence speed.

V. Summary

VLR is important for vehicle identification in ITS and has many potential applications in traffic monitoring and vehicle management systems. The existing VLR systems in literature build models using large training dataset which might not be available in real applications. This paper proposes a novel classification method, which incorporates a Cauchy prior for LR combined with CGD, for multinomial image recognition tasks. This paper also proposes an novel online VLR framework using the proposed classifier, which provides solutions for both small and large datasets. The proposed classifier results in a quicker convergence speed as compared to LR while giving a similar accuracy. By testing with the publicly available dataset, the proposed classifier decreases the computational cost by 59% when compared with LR and an accuracy of up to 98.80% is achieved. The proposed online VLR framework is tested for training datasets of an increasing size, this further decreases the computational cost and slightly increases the recognition accuracy when compared with the offline method.

In the future, the proposed method will be compared with the deep learning methods such as the Convolutional Neural Networks (CNN). CNN features are more representative than HOG features if very large training dataset is available. However, these methods have a high computational cost associated with them. Hence, a combined solution could be built to cope with different training dataset sizes.

VI. Acknowledgements

We appreciate the support of the SETA project funded from the European Unions Horizon 2020 research and innovation programme under Grant Agreement No. 688082. The authors are also grateful to the UK-China Mobility grant: Multi-vehicle tracking and classification for intelligent transportation systems (Reference number E150823) from the UK Royal Society fund. We acknowledge also the EC Seventh Framework Programme [FP7 2013-2017] TRAcking in compleX sensor systems (TRAX) Grant agreement no.: 607400.

References

[1] A. P. Psyllos, C.-N. E. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a SIFT-based enhanced matching scheme," *IEEE Trans. on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 322–328, 2010.

[2] L. Figueiredo, I. Jesus, J. T. Machado, J. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *Proc. Intelligent Transportation Systems*, vol. 88, 2001, pp. 1206–1211.

[3] Z. Zhang, X. Wang, W. Anwar, and Z. L. Jiang, "A comparison of moments-based logo recognition methods," in *Proc. Abstract and Applied Analysis*, vol. 2014. Hindawi Publishing Corporation, 2014.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] R. Lipikorn, N. Cooharojananone, S. Kijsupapaisan, and T. Inchayanunth, "Vehicle logo recognition based on interior structure using SIFT descriptor and neural network," in *Proc. International Conf. on Information Science, Electronics and Electrical Engineering*, vol. 3, April 2014, pp. 1595–1599.

[6] S. Yu, S. Zheng, H. Yang, and L. Liang, "Vehicle logo recognition based on bag-of-words," in *Proc. 10th IEEE International Conf. on Advanced Video and Signal Based Surveillance*. IEEE, 2013, pp. 353–358.

[7] Q. Gu, J. Yang, G. Cui, L. Kong, H. Zheng, and R. Klette, "Multi-scale vehicle logo recognition by directional dense SIFT flow parsing," in *Proc. 2016 IEEE International Conf. on Image Processing (ICIP)*, Sept 2016, pp. 3827–3831.

[8] R. Chen, M. Hawes, L. Mihaylova, J. Xiao, and W. Liu, "Vehicle logo recognition by spatial-SIFT combined with logistic regression," in *Proc. 2016 19th International Conf. on Information Fusion*, July 2016, pp. 1228–1235.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[10] D. Llorca, R. Arroyo, and M. Sotelo, "Vehicle logo recognition in traffic images using HOG features and SVM," in *Proc. Intelligent Transportation Systems*. IEEE, 2013, pp. 2229–2234.

[11] Q. Sun, X. Lu, L. Chen, and H. Hu, "An improved vehicle logo recognition method for road surveillance images," in *Proc. Seventh International Symposium on Computational Intelligence and Design*, vol. 1, Dec 2014, pp. 373–376.

[12] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5, pp. 352–359, 2002.

[13] C. Rainey, "Dealing with separation in logistic regression models," *Political Analysis*, vol. 24, no. 3, pp. 339–355, 2016.

[14] A. Gelman, A. Jakulin, M. G. Pittau, and Y.-S. Su, "A weakly informative default prior distribution for logistic and other regression models," *The Annals of Applied Statistics*, pp. 1360–1383, 2008.

[15] B. Carpenter, "Lazy sparse stochastic gradient descent for regularized multinomial logistic regression," *Alias-i, Inc., Tech. Rep*, pp. 1–20, 2008.

[16] J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, Q. V. Le, and A. Y. Ng, "On optimization methods for deep learning," in *Proc. of the 28th International Conf. on Machine Learning*, 2011, pp. 265–272.

[17] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Pittsburgh, PA, USA, Tech. Rep., 1994.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, USA: Springer, 2006.

[19] R. Battiti, "First- and second-order methods for learning: Between steepest descent and Newton's method," *Neural Computation*, vol. 4, no. 2, pp. 141–166, March 1992.

[20] L. Grippo and S. Lucidi, "A globally convergent version of the polak-ribiere conjugate gradient method," *Mathematical Programming*, vol. 78, no. 3, pp. 375–391, 1997.

[21] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Trans. on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1951–1960, 2015.