2016 IEEE Second International Conference on Multimedia Big Data

CNUSVM: Hybrid CNN-Uneven SVM Model for Imbalanced Visual Learning

Mengyue Geng[1,2,3], Yaowei Wang[4*], Yonghong Tian[2,3*], Tiejun Huang[2,3]
{mygeng,yhtian,tjhuang}@pku.edu.cn, yaoweiwang@bit.edu.cn
[1]School of Electronic and Computer Engineering,
Shenzhen Graduate School at Peking University, Shenzhen, China
[2]National Engineering Laboratory for Video Technology, School of EE&CS,
Peking University, Beijing, China
[3]Cooperative Medianet Innovation Center, China
[4]School of Information and Electronics,
Beijing Institute of Technology, Beijing, China

***Abstract*—Recently, deep Convolutional Neural Networks (C-NNs) have been used to achieve state-of-the-art performance on a wide range of visual learning tasks. However, when facing some imbalanced learning tasks where the training samples are unevenly distributed among different classes, CNNs tend to produce performance bias toward the majority class, making them not suitable for applications in which the recognition ability on the minority class is highly valued. To address the problem, this paper proposes a hybrid classification model by combining CNN with Support Vector Machine (SVM) that has uneven margins. In this model, CNN works as a feature extractor and the extracted features are then sent into a L2-SVM with linear uneven margins. We also develop a gradient-descent learning approach for this hybrid CNN-uneven SVM (CNUSVM) model by minimizing an uneven margin based L2-hinge loss. Our experiments on two benchmark datasets show that the CNUSVM model can make more favorable decisions for imbalanced visual learning tasks in comparison with the standard CNN and the hybrid CNN-SVM model.**

I. Introduction

Deep learning methods using Convolutional Neural Networks (CNNs) [11] have achieved state-of-the-art performance in many visual learning tasks such as image classification [9] and object detection [7]. As a multi-layered back propagation neural network, CNN minimizes a cross-entropy loss using the classical back propagation learning algorithm. However, when training data are highly imbalanced, BP networks (including CNNs) often perform better on the majority class, yet the minority class examples often tend to be misclassified [8], [15]. This is a serious problem in many real-world applications such as anomaly detection and vision-aided medical diagnosis, where the minority class is what we really care about. With the growing interest in imbalanced learning among researchers, many methods have been proposed to tackle the problem both at the data and algorithmic levels [6]. Some of them focus on re-balancing the training data using sampling strategies, such as oversampling [3], [5] and subsampling [10]. Oversampling artificially generates new samples that belong to the minority class so as to bridge the gap between the numbers of two classes, while subsampling selects a reasonably-sized subset of examples that belong to the majority class. Other methods try to develop new learning algorithms to fit imbalance data distribution, such as cost-sensitive learning [23] and genetic programming [2].

It should be noted that when using neural networks (NNs) for imbalanced learning tasks, most researches focus on sampling the training data. Due to the disadvantages associated with the use of sampling methods, such as discarding the potentially useful data (caused by subsampling) and increasing the learning time (caused by oversampling) [25], we still stand on the algorithmic aspect to solve the imbalanced neural classification problem.

As a well-known margin-based learning algorithm, Support Vector Machines (SVMs) [24] are widely used in many machine learning tasks. Several variation algorithms that adapt the SVM to different imbalanced classification problems have also been proposed and demonstrated remarkable performance gains over the traditional SVM [1], [22], [27]. In recent years, some hybrid classification models by combining NNs with SVMs have also been developed, in which NN works as an automatic feature extractor and SVM works as a classifier [13], [27].

Inspired by these models, this paper develops a hybrid classification model, called CNUSVM, which combines CNN with uneven-margin-based SVM proposed in [12]. This CNUSVM model automatically generates feature vectors using the CNN, and these features are then sent to an uneven L2-uneven SVM for further classification. We also develop a gradient-descent learning approach by minimizing an uneven margin based L2-hinge loss. Our experimental results show that the proposed CNUSVM model can fit the imbalanced data distribution well and the learning algorithm is suitable for joint training and incremental learning. Compared with the standard CNN and the hybrid CNN-SVM model, the CNUSVM model can make more favorable decisions for imbalanced visual learning tasks.

The rest of the paper is organized as follows. Sec. II provides background knowledge of the last layer linear classifier in CNN, standard SVM and the combination models of CNNs

*Yaowei Wang and Yonghong Tian are corresponding authors.

978-1-5090-2179-6/16 $31.00 © 2016 IEEE
DOI 10.1109/BigMM.2016.19

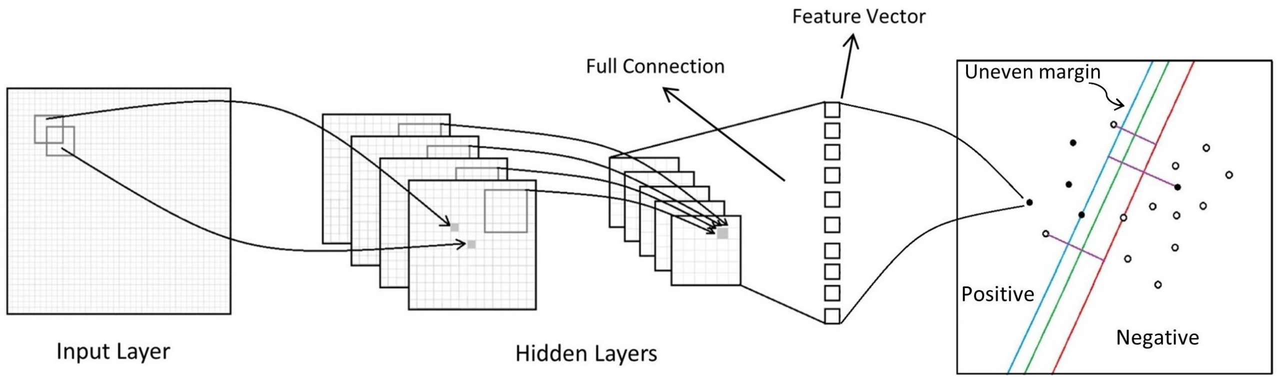

Fig. 1. The architecture of the CNUSVM model.

and SVMs. Sec. III presents the proposed CNUSVM model. Experimental results and analysis are presented in Sec. IV. The conclusion is drawn in Sec. V.

II. Background

We now give a brief background description of the softmax classification function of the standard CNN and SVM. Since our model is tightly-related to the recently proposed hybrid models which combine CNN with the standard linear SVM, this CNN-SVM model will also be introduced.

A. Softmax

In most deep neural networks (including CNNs), it is typical to implement the softmax function at the top layer for classification, and train the network under a cross-entropy regime. For each training sample in an N-class classification problem, the total input a_k to the k^{th} softmax layer unit is

$$a_k = \sum_j o_j w_{jk} \tag{1}$$

where $\boldsymbol{o} = \{o_j\}$ is the activation of the penultimate layer and $\boldsymbol{w} = \{w_{jk}\}$ is the weight matrix between the last two layers. The output of the k^{th} unit p_k can be calculated as

$$p_k = \frac{\exp(a_k)}{\sum_{n=1}^{N} \exp(a_n)} \tag{2}$$

The predicted class $\hat{i}$ would be

$$\hat{i} = \underset{i}{\operatorname{argmax}}\, p_i \tag{3}$$

B. Support Vector Machines

Support Vector Machines (SVMs) were originally designed for the binary classification problem, given training set $S = \{(\boldsymbol{x_i}, y_i) \mid \boldsymbol{x_i} \in \mathbb{R}^m, y_i \in \{-1, +1\}\}, i = 1, 2, \ldots, N$. SVM finds a hyperplane to separate the input data by maximizing the margin in the feature space. The linear soft margin SVM, which is a commonly used SVM model, results in the following optimization problem:

$$\begin{aligned} &\text{minimize}_{\boldsymbol{w},\xi_i} \left\{ \frac{\|\boldsymbol{w}\|^2}{2} + C \sum_{i=1}^{N} \xi_i \right\} \\ &s.t. \quad y_i (\boldsymbol{w} \cdot \boldsymbol{x_i}) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0 \quad \forall i \end{aligned} \tag{4}$$

where ξ_i are slack variables that measure the degree of misclassification of the input sample $\boldsymbol{x_i}$, and C is called the cost factor which controls the trade-off between training error minimization and margin maximization. The corresponding unconstrained form of the optimization problem is

$$\text{minimize}_{\boldsymbol{w}} \quad \frac{\|\boldsymbol{w}\|^2}{2} + C \sum_{i=1}^{N} \max(1 - y_i (\boldsymbol{w} \cdot \boldsymbol{x_i}), 0) \tag{5}$$

Eq.5 is also known as the primal problem of L1-SVM with the standard hinge loss. A least squares version of L1-SVM, also known as L2-SVM has also been proposed by minimizing the squared hinge loss (L2-loss) [20]:

$$\text{minimize}_{\boldsymbol{w}} \quad \frac{\|\boldsymbol{w}\|^2}{2} + C \sum_{i=1}^{N} \max(1 - y_i (\boldsymbol{w} \cdot \boldsymbol{x_i}), 0)^2 \tag{6}$$

Typically, the optimization problem of SVM is solved in the dual form by introducing Lagrange multipliers, which however is not suitable for the incremental learning process of CNN. To combine SVM with CNN and achieve the goal of joint training, optimization with gradient descent based approaches has been developed in this study, as discussed in Sec. II-C.

C. Hybrid CNN-SVM models

Motivated by the fact that the linear classifier used in the standard CNN has a very limited classification ability, several hybrid models have been developed recently by replacing the linear classifier of CNN with SVM [16], [21]. In these hybrid CNN-SVM models, the original softmax classifier in the standard CNN is replaced, and the lower layer weights are learned by backpropagating the gradients (partial derivatives) from the top layer linear SVM. Let the SVM objective in Eq.5 be $l(\boldsymbol{w})$, and the input $\boldsymbol{x}$ is replaced by the activation $\boldsymbol{o}$ of the penultimate layer, the gradient of $l(\boldsymbol{w})$ with respect to $\boldsymbol{o}$ is

$$\frac{\partial l(\boldsymbol{w})}{\partial \boldsymbol{o}_i} = -C y_i \boldsymbol{w} \left(\mathbb{I}\left\{1 > y_i \left(\boldsymbol{w} \cdot \boldsymbol{o}_i\right)\right\}\right) \tag{7}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function, and $y_i \in \{-1, +1\}$ is the corresponding label of $\boldsymbol{o}_i$. Likewise, for L2-SVM, we have

$$\frac{\partial l(\boldsymbol{w})}{\partial \boldsymbol{o}_i} = -2C y_i \boldsymbol{w} \left(\max\left(1 - y_i \left(\boldsymbol{w} \cdot \boldsymbol{o}_i\right), 0\right)\right) \tag{8}$$

The calculated gradients can then be backpropagated to learn the lower layer weights.

In essence, the combination of CNN and SVM can be seen as a change of cost function, and many researches have shown that such a combination can be successfully applied to visual learning tasks and often yield better performance than the original CNN [21], [27].

III. The Proposed Method

Despite its high performance, the hybrid CNN-SVM model is still not suitable for imbalanced learning problems. In recent years, variation models of SVM have been developed to address the problem of imbalanced learning. In this context, we focus our attention towards combining CNN with such an uneven-margin-based SVM to gain better performance on the imbalanced visual learning tasks. Thus in this section, we will first briefly introduce the uneven margin based SVM used in our model, and then present the proposed hybrid CNN-uneven SVM model (CNUSVM).

A. SVM with uneven margins

In [12], a SVM with uneven margins was proposed for imbalanced classification, which significantly outperformed the standard SVM with respect to the document categorization for small categories. By introducing a margin parameter τ into Eq.4 to control the ratio of the positive margin over the negative margin in SVM, the primal problem can be changed to the following optimization problem:

$$\begin{aligned} &\text{minimize}_{\boldsymbol{w},\xi} \left\{ \frac{\|\boldsymbol{w}\|^2}{2} + C\sum_{i=1}^{N} \xi_i \right\} \\ s.t. \quad & \boldsymbol{w} \cdot \boldsymbol{x}_i + \xi_i \geq 1 \quad if \quad y_i = +1 \\ & \boldsymbol{w} \cdot \boldsymbol{x}_i - \xi_i \leq -\tau \quad if \quad y_i = -1 \\ & \xi_i \geq 0 \quad \forall i \end{aligned} \tag{9}$$

where τ is the ratio of the negative margin to the positive margin of the classifier. For imbalanced learning tasks, set $0 < \tau < 1$ and the classification hyperplane will be close to the negative margin, thus improving the classification performance towards the minority (positive) samples.

A set of transformations were given in [12] to obtain the uneven SVM with any margin parameter $0 < \tau < 1$ from its corresponding standard SVM. However, in our hybrid CNN-SVM model, it is easy to fall into local minimum if CNN and SVM are trained separately. This is mainly due to the features extracted by separately trained CNN are already vulnerable to the high imbalanced data ratio and the performance will not be optical if we further classify these features using unbalance SVM. To enable the joint training, we thus propose a new gradient-descent learning method to directly backpropagate the gradients of the uneven SVM's objective so as to train CNN.

B. Hybrid CNN-Uneven SVM model

In our CNUSVM model, the classification layer of the CNN is replaced by an uneven margin based linear L2-SVM (i.e. the activation of the penultimate fully-connected layer acts as an input of the uneven SVM). Its architecture is shown in Fig. 1. Firstly, the sample images are sent to the input layer of the network. CNN is trained to generate features and the extracted features are then sent to the uneven SVMs at the last layer. Note that we use the one-vs-rest strategy described in [24]. For binary classification problems, there are two different SVMs in the last layer with exactly opposite inputs (that is, the positive samples of the first SVM are the negative samples of the second), and the test samples would belong to the positive class of the SVM which has a bigger output than the other.

To use the objective of the uneven SVM to train our CNUSVM, we need to differentiate it with respect to the activation of the penultimate layer. This requires the corresponding unconstrained form of the optimization problem in Eq.9, as follows

$$\begin{aligned} \text{minimize}_{\boldsymbol{w}} \frac{\|\boldsymbol{w}\|^2}{2} + C\sum_{i=1}^{N} \max\Bigg(\frac{\tau+1}{2} \\ -y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i + \frac{\tau-1}{2}\right), 0\Bigg) \end{aligned} \tag{10}$$

Let the objective in Eq.10 be $l(\boldsymbol{w})$, and replace the input data sample $\boldsymbol{x}$ with the activation of the penultimate layer $\boldsymbol{o}$. Then we have

$$\begin{aligned} \frac{\partial l(\boldsymbol{w})}{\partial \boldsymbol{o}_i} = -C y_i \boldsymbol{w} \Bigg(\mathbb{I}\Bigg\{\frac{\tau+1}{2} \\ > y_i\left(\boldsymbol{w} \cdot \boldsymbol{o}_i + \frac{\tau-1}{2}\right)\Bigg\}\Bigg) \end{aligned} \tag{11}$$

In [16], the authors found that as the linear classifier at the last layer, L2-SVM is slightly better than L1-SVM in most cases. Since L2-SVM is differentiable and better in performance, we also use the L2-uneven SVM in our CNUSVM model, which optimizes the following problem:

$$\begin{aligned} \text{minimize}_{\boldsymbol{w}} \frac{\|\boldsymbol{w}\|^2}{2} + C\sum_{i=1}^{N} \max\Bigg(\frac{\tau+1}{2} \\ -y_i\left(\boldsymbol{w}\cdot\boldsymbol{o}_i + \frac{\tau-1}{2}\right), 0\Bigg)^2 \end{aligned} \tag{12}$$

The partial derivative of Eq.12 with respect to the activation of the penultimate layer $\boldsymbol{o}$ is

$$\begin{aligned} \frac{\partial l(\boldsymbol{w})}{\partial \boldsymbol{o}_i} = -2C y_i \boldsymbol{w} \Bigg(\max\Bigg(\frac{\tau+1}{2} \\ -y_i\left(\boldsymbol{w} \cdot \boldsymbol{o}_i + \frac{\tau-1}{2}\right), 0\Bigg)\Bigg) \end{aligned} \tag{13}$$

Algorithm 1 Gradient Descent Training of CNUSVM

Input: training image set S, α, τ_1 ,τ_2, C
Initialize: CNN layers, SVM weight vectors $\{\boldsymbol{w_1}, \boldsymbol{w_2}\}$
Output: trained CNN, updated SVM weight vectors

1) **while** not converged **do**
2)   **for each** $\boldsymbol{x} \in S$ **do**
3)   Feed forward CNN using $\boldsymbol{x}$;
4)   Get penultimate layer activation $\boldsymbol{o}$;
5)   **for each** SVM$_i$, $i \in \{1, 2\}$ **do**
6)   Get corresponding label $y_i \in \{+1, -1\}$;
7) $L = \frac{(1+\tau_i)}{2} - y_i \left((\boldsymbol{w_i} \cdot \boldsymbol{o}) - \frac{(1-\tau_i)}{2} \right)$;
8) $g_i = -2Cy_i\boldsymbol{w_i} \max(L, 0)$;
9) $\boldsymbol{w_i} = \boldsymbol{w_i} - \alpha g_i \, [\boldsymbol{o}./\boldsymbol{w_i}]$;
10)   **end for**
11)   Backpropagate g_1, g_2;
12)   **end for**
13) **end while**

From this point on, the backpropagation algorithm is exactly the same as the standard CNN and the calculated partial derivatives are also used to train the last layer SVMs through a gradient descent procedure. A detailed training procedure for binary classification problems are described in Algorithm 1, where L in Line 7 is a temporary variable used to calculate gradients g and the $[./]$ symbol in Line 9 denotes element-wise division of two vectors.

There are four user-defined parameters in Algorithm 1: learning rate α, margin parameters τ_1, τ_1 for the two SVMs and SVM cost factor C. For binary classification problems with highly imbalanced training data, we would like the classification hyperplanes of both SVMs in the last layer close to the majority samples. Thus, for the SVM which takes the minority data as positive class, τ will be set close to 0, and for another τ will be set larger than 1.

IV. Experiments

A. Experimental settings

Datasets. The effectiveness of the proposed CNUSVM model is evaluated on two benchmark datasets. Detailed data distribution of both datasets are shown in Table I.

TABLE I
DATA DISTRIBUTION OF THE EXPERIMENTAL DATASETS.

Dataset	Training		Testing		Validation	
	#Pos	#Neg	#Pos	#Neg	#Pos	#Neg
Pedestrian	4800	20000	4800	5000	4800	5000
GTSRB	210	1860	30	315	30	315

The first dataset is the Daimler Mono pedestrian classification benchmark (Pedestrian) [14], which contains five different sets, each with 4800 pedestrian and 5000 non-pedestrian (background) images (see Fig. 2). We randomly choose 4800 pedestrian and 20000 non-pedestrian images to train our model, with an imbalance ratio of about 1:4.

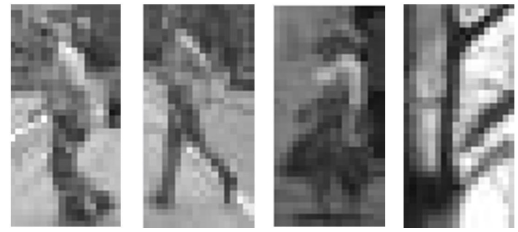

Fig. 2. Example images of Daimler Mono pedestrian classification benchmark: pedestrians (left three) and background (right).

The second dataset is the German traffic sign recognition benchmark(GTSRB) [19], with 43 classes and the sample number of each class ranges from 210 to 2250. Since in this study we only focus on the imbalanced binary classification tasks, we choose two classes in GTSRB in our experiments, i.e. the speed limit sign of 20 mph and 80 mph, each containing 210 and 1860 training samples. The imbalance ratio is approximately 1:9. Example images are shown in Fig. 3.

Fig. 3. Example images of GTSRB.

Evaluation metrics. Several metrics have been used to evaluate the effectiveness of our hybrid CNUSVM model. To test whether our model can gain a balanced performance on minority examples as well as on majority examples, we adopt Geometric Mean (G-mean) [10] as one of the main criteria. Given accuracies observed separately on positive examples a^+ and negative ones a^-, G-mean can be calculated as $g = \sqrt{a^+ \cdot a^-}$. As we can see, a very high a^- by a low a^- will still result in poor g.

We also adopt Area Under the ROC Curve (AUC) to evaluate the classification robustness of our model. By changing the decision threshold of the classifier, we can get different True Positive Rates (TPRs) and the False Positive Rates (FPRs), and then draw the ROC Curve by plotting all the (TPR,FPR) pairs. Area Under The ROC Curve can then be calculated to quantify the performance of the used model.

Despite the above two metrics, overall classification accuracy and accuracy on the minority and minority class are also took into consideration.

Methods for comparison. The proposed model is compared with several imbalanced learning methods. These approaches can be roughly categorized into three groups, including:

(1) Sampling Group. This group contains two sampling methods designed for imbalanced learning, including

TABLE II

PERFORMANCE OF VARIOUS APPROACHES ON THE PEDESTRIAN DATASET.

Method		G-mean	AUC	a^+	a^-	acc
Sampling Group	HOG + CUS + NN	0.9086	0.9725	96.35%	85.68%	90.91%
	HOG + SMOTE-RSB + NN	0.9413	0.9865	92.04%	96.26%	94.19%
Algorithmic Group	CSCNN	0.9366	0.9814	93.44%	93.88%	93.66%
	HOG + CSNN	0.9454	0.9877	94.50%	94.58%	94.54%
Baseline Group	CNSVM	0.9246	**0.9891**	86.94%	**98.34%**	92.76%
	HOG + NN	0.9323	0.9877	88.77%	97.94%	93.45%
	CNN	0.9168	0.9814	86.50%	97.18%	91.95%
Ours	CNUSVM	**0.9557**	**0.9891**	**97.02%**	94.14%	**95.55%**

TABLE III

PERFORMANCE OF VARIOUS APPROACHES ON THE GTSRB DATASET.

Method		G-mean	AUC	a^+	a^-	acc
Sampling Group	HOG + CUS + NN	0.8440	0.9637	73.33%	97.14%	95.07%
	HOG + SMOTE-RSB + NN	0.9071	0.9891	83.33%	98.73%	97.39%
Algorithmic Group	CSCNN	0.9212	0.9990	85.00%	99.84%	98.55%
	HOG + CSNN	0.9100	0.9889	83.33%	99.37%	97.97%
Baseline Group	CNSVM	0.9030	0.9972	81.67%	99.84%	98.26%
	HOG + NN	0.8916	0.9759	80.00%	99.37%	97.68%
	CNN	0.9037	0.9991	81.67%	**100.0%**	98.41%
Ours	CNUSVM	**0.9574**	**0.9994**	**91.67%**	**100.0%**	**99.28%**

SMOTE-RSB [17], which is an oversampling method based on the well-known SMOTE algorithm, and cluster-based under-sampling(CUS) [26], which is a subsampling method. Since these sampling methods often work better on manually extracted features than on raw images, we extract HOG features [4] of the original images and adopt fully connected Neural Network (NN) with backpropagation-based training [18] as the classifier.

(2) Algorithmic Group. This group contains two algorithmic methods for imbalanced learning problems, including Cost Sensitive Neural Network (CSNN) [28], which is trained using the extracted HOG features, and CNN with the same cost sensitive strategy (CSCNN), which is trained using the original images.

(3) Baseline Group. To compare above methods with the standard classification models, three baseline methods are used, including the standard CNN, the hybrid CNN-SVM model(CNSVM) [21] and the HOG + NN without any sampling or cost sensitive implementation.

B. Experimental results

We adopt all 3 groups of the comparing methods as well as our model (denoted as **Ours**) to see whether our model can make more favourable decision on the imbalanced datasets. The results on the two datasets are reported in Table II and Table III. where a^+, a^- stand for the classification accuracy on positive (minority) class and negative (majority) class and acc stands for the overall accuracy.

We use a six-layer convolutional neural network as the forepart feature extractor. The learning rate α is set to 0.01 and the penalty factor C of the SVM is set to 1. The uneven parameters τ for each SVM are manually adjusted to best fit each dataset, which will be discussed in Sec. IV-C.

1) Performance on the pedestrian dataset: From Table II, we can see that on the Pedestrian dataset, all methods from the Baseline group suffer a performance bias, getting high classification accuracy on the majority examples, yet accuracy on the minority examples are relatively low. Intuitively, all methods from the non-baseline Groups can improve the classification accuracy on the minority examples. Among them, our CNUSVM gets the best G-mean and a^+, followed by HOG + CSNN and HOG + SMOTE-RSB + NN. The best AUC score is gained by our model and CNSVM. Table II also presents that our model gets the best overall accuracy among all methods, thanks to the powerful classification ability of the combination of CNN and uneven SVM.

We can also see from Table II that all methods designed for imbalance learning have a performance drop on the majority data samples in comparison with their corresponding baseline methods. Among them, algorithmic methods perform better than sampling methods according to the experimental results. Surprisingly, although HOG + CUS + NN gets a high accuracy on the minority examples, the accuracy on the majority examples drops significantly, leading to a rather low G-mean value. This fact reveals that in some cases, the loss of useful information during the subsampling process may have great influence on classification results.

2) Performance on the GTSRB dataset: From Table III, we can see that on the GTSRB dataset, baseline methods suffer even more performance bias due to a extremely small number of positive samples in the training data. Another fact we can see is that the methods using HOG features get worse performances on the GTSRB dataset than on the Pedestrian dataset. The positive and negative examples in the former dataset have pretty small areas of difference and as a result, HOG features can not distinguish them effectively. Another reason is that HOG features are particularly fitted for human detection, so they work better on the Pedestrian dataset.

Table III also shows that our model outperforms the others on all criteria, getting the highest G-mean of 0.9574 and an ideal AUC of 0.9994. Among the two methods in the Sampling group, HOG + CUS + NN gets even worse performance compare to HOG + NN, this is mainly due to the lack of training samples sent to NN after undersampling.

C. Choosing parameters

From Table I, we can see that different datasets have different imbalance ratios. To use our CNUSVM model on a particular dataset, first we need to find a way to choose the best parameter for the uneven SVMs. In this section, we design several experiments for the purpose of finding the best uneven margin parameter τ. We take the GTSRB dataset as an example below to show the whole parameter choosing process. All parameter selection experiments use validation data and similar experiments are conducted for every dataset where our model is applied.

For binary classification problems, there are two uneven SVMs in our model, the first SVM takes the minority samples as it's own positive input, and the other takes the majority data as positive. Let the uneven parameter for the two SVMs be τ_1 and τ_2. In order to make both SVMs' classification hyperplanes close to the majority samples, τ_1 should be set smaller than 1, τ_2 should be set larger than 1. In our first experiment, we test the performance of our model on the GTSRB dataset with different values of $\tau_1 = 0.2, 0.4, 0.6, 0.8$ and $\tau_2 = 2.0, 4.0, 6.0, 8.0$. Results are shown in Table IV.

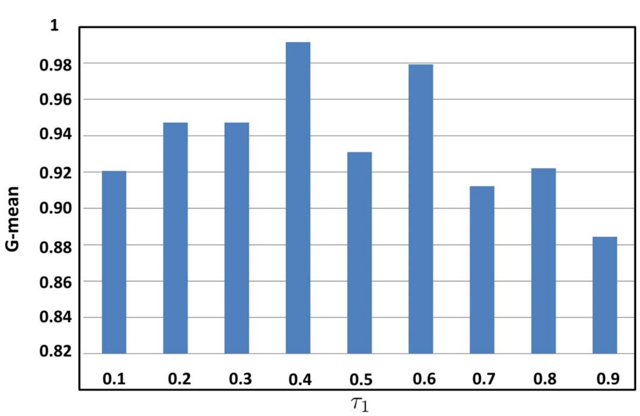

Fig. 4. The G-mean scores of our model on the GTSRB dataset when using different uneven parameter τ_1.

From Table IV, we notice that with the increasing of τ_2, the best record occurs at a gradually smaller value of τ_1. Through observing the relationship of τ_1 and τ_2 in these records, we found that approximatively they have a relationship of $\tau_2 \approx 10 \times (1 - \tau_1)$.

TABLE IV
PERFORMANCE ON THE GTSRB DATASET USING DIFFERENT MARGIN PARAMETERS. BEST RECORDS UNDER EACH VALUE OF τ_2 ARE BOLDED.

τ_2	τ_1	G-mean	a^+	a^-
2.0	0.2	0.8660	75.00%	100.0%
	0.4	0.7853	61.67%	100.0%
	0.6	0.9037	81.67%	100.0%
	0.8	**0.9220**	**85.00%**	**100.0%**
4.0	0.2	0.8944	80.00%	100.0%
	0.4	0.8944	80.00%	100.0%
	0.6	**0.9793**	**96.67%**	**99.21%**
	0.8	0.8660	75.00%	100.0%
6.0	0.2	0.9302	86.67%	99.84%
	0.4	**0.9916**	**98.33%**	**100.0%**
	0.6	0.9472	90.00%	99.68%
	0.8	0.9544	91.67%	99.37%
8.0	**0.2**	**0.9472**	**90.00%**	**99.84%**
	0.4	0.9449	90.00%	100.0%
	0.6	0.9376	88.33%	99.68%
	0.8	0.8944	80.00%	100.0%

By exploiting the approximate relationship, we run a second experiment on the dataset. In this time, the value of τ_1 ranges from 0.1 to 0.9, and τ_2 is decided through the above equation. Results are shown in Fig. 4. As we can see, when τ_1 = 0.4 (τ_2 = 6.0), our model gets the best G-mean value, and this would be the final parameters we use in the experiments.

D. Performance at various imbalance ratios

Sec. IV-B presents that our CNUSVM model can make more favorable decisions than the other methods when facing the the imbalanced data. To further investigate whether our model can work on datasets with a wide range of imbalance ratios, we test our model on several training subsets selected from the Daimler Mono Pedestrian Classification Benchmark. Each subset contains 18000 negative samples, and the number of positive samples varies from 360 to 3600, the imbalance ratio changes from 1:5 to 1:50. We also test the performance of the standard CNN and the CNSVM on these subsets for comparison. Result are shown in Fig. 5.

From Fig. 5 (a) and (c), we can see that with the decreasing of the imbalance ratio, the performance on the minority samples drops on all three models. However, our model shows more robustness towards the increasing gap between the numbers of the minority and majority samples, in comparison with CNSVM and CNN, both of which suffer a very rapid performance decline.

Fig. 5 (d) shows that on each training set, the accuracy on the majority samples of CNUSVM is slightly lower than CNN and CNSVM. However, since in most imbalance learning

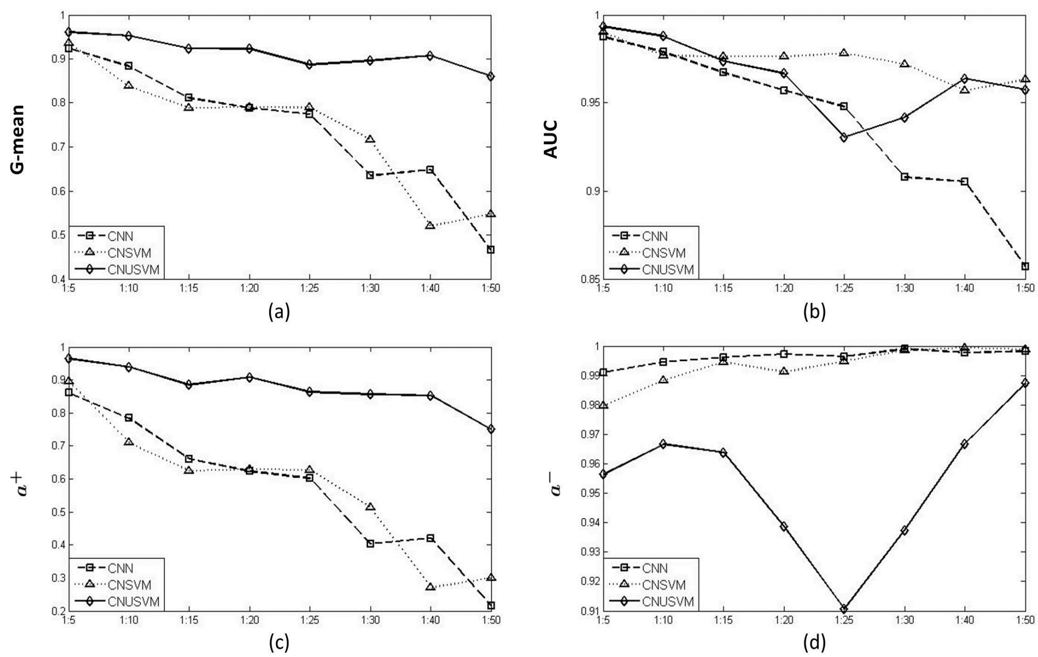

Fig. 5. Performance of the standard CNN, CNSVM and our CNUSVM model on training sets with various imbalance ratios.

tasks we care more about the minority samples rather than the majority samples, such a performance drop is often acceptable.

Finally, from Fig. 5 (b) we can see that the AUC score of the standard CNN drops at each ratio, while CNUSVM and CNSVM have relatively more stable AUC scores on most ratios. To sum up, our CNUSVM model is capable of fitting datasets with a wide range of imbalance ratios and can make better decision at each ratio than the standard CNN and CNSVM.

V. Conclusion

In this paper we propose a novel hybrid CNN-uneven SVM model for imbalanced visual learning problems. The CNN in our model is trained by minimizing the objective of an uneven linear L2-SVM instead of the standard mean-squared loss. From the experimental results, we find that by using an uneven L2-SVM as the classifier at the last layer, our CNUSVM can make more favourable decisions when facing imbalanced datasets.

In the future work, we will extend our model to tackle multi-class imbalanced learning problems. Since our model doesn't change the structure of CNN, it can naturally be extended to perform multi-class classification, the main focus will be the choosing of parameters. This will lead to the study on automatic parameter selection. We will also try to combine other imbalance learning approaches with our model, and we believe that its performance can be gradually improved by combining with other well-performed approaches for imbalance learning.

Acknowledgment

This work is partially supported by the National Basic Research Program of China under grant 2015CB351806, the National Natural Science Foundation of China under contract No. 61425025, No. 61390515, No. 61421062, No. 61471042 and Shenzhen Peacock Plan.

References

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *Proceedings of European Conference on Machine Learning*, pages 39–50. Springer, 2004.

[2] U. Bhowan, M. Zhang, and M. Johnston. Genetic programming for image classification with unbalanced data. In *Proceedings of 24th International Conference on Image and Vision Computing New Zealand*, pages 316–321, 2009.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, 2011.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[5] A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36, 2004.

[6] V. Ganganwar. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47, 2012.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[8] T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano. Supervised neural network modeling: an empirical investigation into learning from imbalanced data with labeling errors. *IEEE Transactions on Neural Networks*, 21(5):813–830, 2010.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] M. Kubat, S. Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of International Conference on Machine Learning*, volume 97, pages 179–186. Nashville, USA, 1997.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[12] Y. Li and J. Shawe-Taylor. The svm with uneven margins and chinese document categorization. In *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation*, pages 216–227, 2003.

[13] K. Mori, M. Matsugu, and T. Suzuki. Face recognition using svm fed with intermediate output of cnn for face detection. In *MVA*, pages 410–413, 2005.

[14] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1863–1868, 2006.

[15] Y. L. Murphey, H. Guo, and L. A. Feldkamp. Neural learning from unbalanced data. *Applied Intelligence*, 21(2):117–128, Sept. 2004.

[16] J. Nagi, G. A. Di Caro, A. Giusti, F. Nagi, and L. M. Gambardella. Convolutional neural support vector machines: Hybrid visual pattern classifiers for multi-robot systems. In *Proceedings of International Conference on Machine Learning and Applications*, volume 1, pages 27–32, 2012.

[17] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera. Smote-rsb*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. *Knowledge and information systems*, 33(2):245–265, 2012.

[18] R. Rojas. *Neural networks: a systematic introduction*. 1996.

[19] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0):–, 2012.

[20] J. A. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[21] Y. Tang. Deep learning using linear support vector machines. In *Proceedings of ICML Workshop on Challenges in Representation Learning*, 2013.

[22] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser. Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1):281–288, 2009.

[23] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme. Cost-sensitive learning methods for imbalanced data. In *Proceedings of the 2010 International Joint Conference onNeural Networks*, pages 1–8, 2010.

[24] V. N. Vapnik. *The nature of statistical learning theory*. Springer, New York, 1995.

[25] G. M. Weiss, K. McCarthy, and B. Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? In *DMIN*, pages 35–41, 2007.

[26] S.-J. Yen and Y.-S. Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In *Intelligent Control and Automation*, pages 731–740. 2006.

[27] S. Zhong and J. Ghosh. Decision boundary focused neural network classifier. 2000.

[28] Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006.