# A Multi-Label Classification Method for Vehicle Video

**Yanqiu Cao[1], Chao Tan[1] and Genlin Ji[1, *]**

**Abstract:** In the last few years, smartphone usage and driver sleepiness have been unanimously considered to lead to numerous road accidents, which causes many scholars to pay attention to autonomous driving. For this complexity scene, one of the major challenges is mining information comprehensively from massive features in vehicle video. This paper proposes a multi-label classification method MCM-VV (Multi-label Classification Method for Vehicle Video) for vehicle video to judge the label of road condition for unmanned system. Method MCM-VV includes a process of feature extraction and a process of multi-label classification. During feature extraction, grayscale, lane line and the edge of main object are extracted after video preprocessing. During the multi-label classification, the algorithm DR-ML-KNN (Multi-label K-nearest Neighbor Classification Algorithm based on Dimensionality Reduction) learns the training set to obtain multi-label classifier, then predicts the label of road condition according to maximum a posteriori principle, finally outputs labels and adds the new instance to training set for the optimization of classifier. Experimental results on five vehicle video datasets show that the method MCM-VV is effective and efficient. The DR-ML-KNN algorithm reduces the runtime by 50%. It also reduces the time complexity and improves the accuracy.

**Keywords:** Vehicle video, multi-label classification, dimensionality reduction.

## 1 Introduction

With the increase of vehicle crashes, threats in driving are rising. The most common use of vehicle video is detecting and identifying one important target in the video (e.g., target vehicles, traffic scene text, pedestrians, etc.). However, autonomous driving needs more information, such as pedestrian, lane line and others. Traditional classification, single-label classification is difficult to describe multiple information accurately contained in vehicle video. Therefore, multi-label learning [Zhang and Zhou (2014)] has gradually become the focus of research.

In label classification, for each training instance x , there is a corresponding label y∈Y. Let $\mathrm{Y} = \{y_1, y_2, \dots, y_q\}$ denote the label space consisting of q class labels. Given a training set $\mathrm{T} = \{(x_1, Y_1), (x_2, Y_2), \dots, (x_m, Y_m)\}$ , if $|Y_i| = 1$ , it means that each instance only corresponds to one label, which is a single-label classification; if $|Y_i| > 1$, it means each

[1] School of Computer Science and Technology, Nanjing Normal University, Nanjing, China.

[*] Corresponding Author: Genlin Ji. Email: glji@njnu.edu.cn.

Received: 05 January 2020; Accepted: 13 May 2020.

JBD. doi:10.32604/jbd.2020.01003

instance corresponds to multiple labels, which is a multi-label classification [Zhang and Zhou (2014)]. For example, a picture can be labelled as “blue sky”, “house”, “tree” and so on. Another example is that, a movie can belong to “action films”, “spy_war films” and may also involve suspense stories, thus can be labelled as “suspension films”. Different from multi-class classification, multi-label classification means that each one instance corresponds to multiple labels in label space, and multi-class classification refers to one instance corresponds to only one labels among multiple class labels. Essentially, multi-class classification falls into the category of single-label classification.

Multi-label learning algorithms are divided into two categories: “problem transformation” and “algorithm adaptation”. Problem transformation methods tackle multi-label learning problem by transforming it into other well-established learning scenarios, such as Binary Relevance [Tanaka and Macedo (2015)]. Algorithm adaptation methods tackle multi-label learning problem by adapting popular learning techniques to deal with multi-label data directly, such as ML-KNN [Zhang and Zhou (2007)].

This paper aims to identify multiple labels for vehicle video. We set villainous weather such as rain and snow, pedestrian blocking in front, and unclear lane lines as unusual events. The multi-label classification method can mine valuable information contained in vehicle video comprehensively and provide auxiliary information for unmanned system to improve driving safety. Actually, the feature of adjacent image frames of the vehicle video is easy to show high similarity, resulting in a particularly large feature quantity and redundancy in the data set. The existing multi-label classification method cannot handle this problem, and we choose to improve one of the efficient method ML-KNN (Multi-label K-nearest algorithm). The traditional ML-KNN needs to find K neighbors in the entire data set, which will take a lot of time to run. The major challenges are high time complexity and poor practical effect.

According to the requirements and problems mentioned above, a multi-label classification method MCM-VV is proposed in this paper. Firstly, we extract features after video preprocessing. Then algorithm DR-ML-KNN is used to learn the training set to obtain a multi-label classifier. The algorithm performs the PCA [Tan, Ji and Zhao (2017)] on the transposed matrix of feature matrix and uses maximum a posteriori principle to predict labels of test instance, and finally add this instance to the original training set for classifier optimization. Experimental results show that the algorithm can predict labels of road condition quickly, improve accuracy and reduce runtime by 50%.

In summary, the major contributions of this paper are as follows.

(1) Formulate the problem of multi-label classification for vehicle video.

(2) Devise a feature extraction method suitable for vehicle video, which can accurately extract the effective information in video. Also, we propose a multi-label classification algorithm based on dimensionality reduction, which is beneficial to predict labels quickly and accurately.

(3) Conduct comprehensive experiments on five real video datasets. The experimental results demonstrate the effectiveness and efficiency of our proposals.

## 2 Related work

In recent years, academia has carried out numerous research work on multi-label learning. Zhang proposed backpropagation for multi-label learning to classify gene functions and texts [Zhang and Zhou (2006)]. Jiang proposed a multi-label text classification method based on fuzzy similarity measure (FSM) and k-nearest neighbor (KNN) [Jiang, Tsai and Lee (2012)]. Yu proposed a multi-label classification framework MLNRS based on neighborhood rough set [Yu, Psdrycz and Miao (2013)]. Liu proposed a method of emotion analysis based on multi-label learning [Liu and Chen (2015)]. Ding proposed an algorithm by assessing cost of the majority classes and value of the minority classes to handle the multi-label imbalanced data classification problem [Ding, Yang and Lan (2018)]. Han proposed a new approach called multi-label learning with label-specific features using correlation information (LSF-CI) to learn label-specific features for each label with the consideration of both correlation information in label space and correlation information in feature space [Han, Huang, Zhang et al. (2019)]. Therefore, the multi-label learning has been widely applied in text classification, gene function reorganization, emotion analysis, and has achieved good research results.

For vehicle video, we usually do the following analysis: obtain the driving trajectory of the target vehicle [Tan, Ji and Zhao (2017)] through the tracking of profile and characteristics of the target vehicle in vehicle video; locate and recognize the traffic scene text [Zhang, Mao and Chen (2009)] in vehicle video; detect pedestrians in front of vehicle [Pan, Jin and Feng (2015)]. The above research only detects and identifies one important target in vehicle video. In this paper, we propose a multi-label classification method for vehicle video, it can mine and utilize multiple important information contained in vehicle video.

## 3 Preliminaries

### *3.1 Problem statement*

Let V denote vehicle video set with $p$ vehicle videos. Let $\mathrm{X}=\{x_1,x_2,\dots x_m\}$ denote instance space with m instances, and each $x_i\in\mathrm{X}$ represents a feature vector with n-dimension extracted from V. It is extracted from one frame image in vehicle video. Let $\mathrm{Y}=\{y_1,y_2,\dots y_q\}$ denote label space with q labels, each label describes one kind of the road condition. Given a training set $\mathrm{T}=\{(x_1,Y_1),(x_2,Y_2),\dots,(x_n,Y_m)\}(x_i\in\mathrm{X},\ Y_i\subseteq\mathrm{Y})$. The particular $i$-th instance is denoted by $(x_i,Y_i)$. $Y_i$ is the real label set $x_i$.

**Table 1:** Label information

| Label | Example |
|---|---|
| Driving scene | Highway, City, Country |
| Driving time | Day, Night |
| Weather | Sunny, Rain or snow |
| Pedestrian | Yes, No |
| Lane line | Yes, No |

The label in this paper is shown in Tab. 1. Multi-label classification method MCM-VV extracts features from vehicle video in V, converting it to X. Multi-label classification algorithm DR-ML-KNN learns a multi-label classifier $h: X \rightarrow 2^{y}$ which optimizes some specific evaluation metric. In most cases however, instead of outputting a multi-label classifier, the learning system tend to produce a real-valued function of the form $f: X \times Y \rightarrow R$ . It means that, for each instance $x_i$ and its real label set $Y_i$, the value for label $y \in Y_i$ is $f(x_i, y)$, and the value for label $y' \notin Y_i$ is $f(x_i, y')$, a successful learning system is tend to output $f(x_i, y) > f(x_i, y')$. What's more, the real-valued function can be transformed to a ranking function $rank_f(\cdot,\cdot)$, if $f(x_i, y) > f(x_i, y')$, it means that $rank_f(x_i, y) < rank_f(x_i, y')$.

Given a video set and test video. Our proposed method MCM-VV will first preprocess video to remove noise. Then it will extract features of the video. Each feature of one frame image in video is characterized by a matrix, fuse all feature matrices and calculate an eigenvector $x_i$. After that, video set is converted into feature set and label set are combined as the classifier input. Next, algorithm DR-ML-KNN produce a real-valued function $f$. When a test video arrives, our proposed DR-ML-KNN finds its neighbors in the training set according to features, and outputs the labels of test video by calculating real value $f$ for each label.

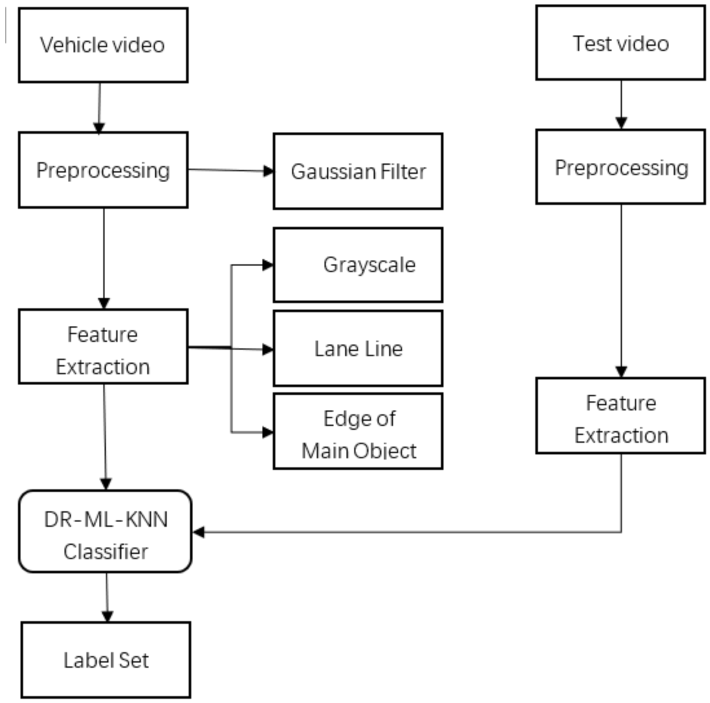

**Figure 1:** Framework of MCM-VV

### *3.2 Framework*

In this section, we present the framework of method MCM-VV, which is shown in Fig. 1. It shows us how to discover multiple labels from the test video. Method MCM-VV is a lazy machine learning method. It predicts multiple labels for the test instance online.

From Fig. 1, method MCM-VV is mainly divided into three steps: preprocessing, feature extraction and classification. The detail of each process is described in next section.

## 4 Methodology

### *4.1 Preprocessing*

From driving recorder shoots the vehicle video to the video is processed and analyzed, the image quality will be poor due to the type and version of the driving recorder. The same problem exists in the transmission media and storage devices, which tends to cause various noises in the formation and transmission of video.

According to the research, we find that noise exists in vehicle video is Gaussian noise, the Gaussian filter is selected in this paper, which is a kind of smooth linear filter and is suitable for removing Gaussian noise [Zhang (2014)]. So, for the image in vehicle video, two-dimensional mean Gaussian filtering is utilized, and the calculation is as shown in Eq. (1):

$$\mathrm{G}(x,y)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

Here, $\mathrm{G}(x,y)$ is the value filtered by Gaussian filter, and $\sigma$ is a parameter. Based on multiple comparison experiments, it can be seen that the noise reduction capability of the Gaussian filter is better.

### *4.2 Feature extraction*

Each image has its own features, such as grayscale, texture, edge… [Chen, Wang and Chen (2015); Wang, Liu and Hai (2017); Zhang and Bao (2002); Zou, Liu and Chen (2019)]. Select some appropriate features can accurately describe the image.

#### *4.2.1 Extract grayscale*

Now, almost all vehicles are equipped with a driving recorder in order to detect the driving situation in front of the vehicle effectively. Usually, the video collected by the driving recorder is a color video. If the information is mined from color video directly, it will require a large amount of memory and calculation time. Therefore, we should convert color video to grayscale video collected by the driving recorder firstly.

The popular methods used to do this are the maximum value method, the average value method, and the weighted average method. In this paper, the weighted average method [Chen, Wang and Chen (2015)] is selected to perform grayscale processing on the vehicle video. Depending on the sensitivity of human eyes to color, we know that the sensitivity to blue is relatively low, and the sensitivity to green is relatively high, so that different weighting coefficients $w_i$ are assigned to three components of R, G, and B to obtain the most suitable grayscale image. As shown in Eq. (2):

$$\mathrm{f}(x,y)=w_1\times R(x,y)+w_2\times G(x,y)+w_3\times R(x,y) \tag{2}$$

### *4.2.2 Extract lane line*

After we get the grayscale vehicle video, the lane line information in the video is blurred. However, according to the information of the lane line, it is possible to judge whether there is an abnormal event such as a pedestrian ahead of the vehicle, and whether it is possible to continue driving. Therefore, in this paper we detect the lane line and uses the lane line as an important feature to determine whether the area between the lane line is the travelable area of the vehicle. The line detection method based on Hough transform [Wang, Liu and Hai (2017)] considers each point in the image as the intersection of the corresponding parameter space curve. When the number of curves intersecting this point exceeds a threshold, this point is considered to be a point on a straight line in the image space. Given a point $\mathrm{M}(x, y)$ on the Line L in a Cartesian coordinate system, its distance to the origin O is r, the angle formed by the line OM and the x-axis is θ. The parameter equation [Qin (2012)] of line L is as shown in Eq. (3):

$$\mathrm{y} = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{1}{\sin\theta}\right)r \tag{3}$$

where $\mathrm{r} = \mathrm{x}\cos\theta + y\sin\theta,$ for any point$(x', y')$ in the image, the straight line passing through this point can be expressed in Eq. (3). And each group of different $(r, \theta)$ represents a different line through this point. Therefore, in the parameter space, the line can be detected by the number of curves intersecting at a certain point. The more curves intersect at a certain point, it means that the line which corresponds to this point is composed of many points.

### *4.2.3 Extract edge of main object*

The edge of the image refers to the portion of the image where the change in brightness is significant. It contains more important information in the image, which plays an important role in the analysis and classification of the whole image. Therefore, in feature extraction of vehicle video, edge information [Zhang and Bao (2002)] is also a very important feature. The edge information of the image can be determined by the grayscale gradient amplitude and direction of the image pixel. The calculation is as shown in Eq. (4):

$$\mathrm{D}(x, y) = \sqrt{f_x^2 + f_y^2},\ \ \theta = \arctan\left(\frac{f_y}{f_x}\right) \tag{4}$$

here, $\mathrm{D}(x, y)$ is the grayscale gradient magnitude at $(x, y)$; θ is the direction. $f_x$ and $f_y$ are the derivatives of $(x, y)$.

Canny operator is selected in this paper, which is not susceptible to noise interference and detect true edges. The weight coefficient matrix is shown in Eq. (5).

$$S_x = \begin{vmatrix} -1 & -1 \\ 1 & 1 \end{vmatrix},\quad S_y = \begin{vmatrix} 1 & -1 \\ 1 & -1 \end{vmatrix} \tag{5}$$

Use the weight coefficient matrix to find $f_x$ and $f_y$, then calculate the grayscale gradient amplitude and direction of the image according to Eq. (4).

By combining the features mentioned above, a huge feature set of vehicle video can be obtained [Zhang, Wang and Geng (2015)]. It can be found that, the number of feature set is particularly large. For feature extraction of vehicle video frame by frame, it can accurately obtain the label set. But at the same time, it is easy to have another condition that the difference of adjacent frames is extremely small. The obtained feature data set has a problem of high redundancy. And if we reduce the frequency of feature extraction violently, it will seriously affect the accuracy of the experiment, which is shown in 5.3. In order to solve the problem of high redundancy and ensure the accuracy of the forecast, we propose a multi-label classification algorithm DR-ML-KNN.

### *4.3 Multi-label K-nearest neighbor classification algorithm based on dimensionality reduction*

We present a Multi-label K-nearest Neighbor Classification Algorithm based on Dimensionality Reduction, named DR-ML-KNN. For high-dimensional data such as vehicle video, we select Principal Component Analysis (PCA) for dimensionality reduction of high-dimensional data [Han, Huang, Zhang et al. (2019)]. The basic idea of the PCA method is to map the original n-dimensional features to the k-dimension, that is, reconstruct the k-dimensional features using the original n-dimensional features. The new k-dimensional features are completely new orthogonal features, i.e. principal components. PCA aims to reduce the dimensions to remove similar features, and extract principal component features. One of the major challenges solved in this paper is to remove similar instances, so we apply PCA to transposed matrix of the feature matrix. As a result, similar samples are removed and data redundancy is reduced.

Step 1: The original training set X is a data set includes $m$ feature vectors. Each feature vector has n dimensions. To reduce the amount of calculation in this paper, the transposed matrix $X_{[n][m]}$ needs to be reduced to k-dimension.

$$x_i = \left[x_i^1, x_i^2, \dots x_i^m\right] \tag{6}$$

The training set is a matrix $X_{[m][n]}$ represents the feature vector set. The whole training set is composed of feature vectors of $m$ frame images, and dimension reduction is performed on the transposed matrix $X_{[n][m]}$.

Step 2: decentralization, that is, subtracting the characteristic mean $\bar{x}$ from each eigenvalue.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{m} x_i \,, x' = x_i - \bar{x} \tag{7}$$

At this point, the training set is still a matrix of $n$ rows and $m$ columns.

Step 3: calculate the covariance matrix Ω.

$$\Omega = \mathrm{X}X^T \tag{8}$$

Step 4: solve the eigenvalues and eigenvectors of the covariance matrix by using the singular value decomposition method.

$$[U, \Sigma, V] = svd(\Omega) \tag{9}$$

U is a n × n matrix, the orthogonal vector contained is a left singular vector; Σ is a n × m matrix, all elements except the elements on the principle diagonal are 0 and each element on the principal diagonal are singular value. V is a m × m matrix, the orthogonal vector is the right singular vector. U and V are both unitary matrices. In general, we will sort the values on the Σ in a descending order, and select the eigenvectors corresponding to the first k eigenvalues.

Step 5: the feature set is transformed into $X_{new}$ in the new space constructed by k feature vectors, and update the original training set X as the transposed matrix $X_{new}$.

Step 6: let $\vec{y}_x$ be the category vector for the instance $x$ ,where its $l-\text{th}$ components $\vec{y}_x(l)(l \in Y)$ takes the value of l if $l \in Y$ and 0 otherwise. let $N(x)$ denote a set of k nearest neighbors of instance x in training set X. In general, the similarity between instances is measured by the Euclidean distance. For the j-th class label, DR-ML-KNN chooses to calculate the following variables:

$$\vec{C}_x(y) = \sum_{a\in N(x)} \vec{y}_a(y), y \in Y \tag{10}$$

Algorithm DR-ML-KNN is described in Algorithm 1.

```
Algorithm 1: DR-ML-KNN

Input: a training set T, test instance $x_{new}$, smoothing parameter s，the number of neighbors  K
Output: the category vector $\vec{L}_{x_{new}}(y)$ of test instance
T' ← PCA(T); // Remove data redundancy
for $i$=1 to $m$ do
    Identify $K$ nearest neighbors $N(x_i)$ for $x_i$;
for each y ∈ Y do
    Estimate the prior probabilities $P(H_b^y)$;
for each y ∈ Y do
    for j=1 to $K$ do
        $\varphi[j] = 0$，$\bar{\varphi}[j] = 0$;
    for $i$=1 to $m$ do
        $\theta = \vec{C}_x(y)$;
        if ($\vec{y}_{x_i}(l) == 1$) then $\varphi[\theta] = \varphi[\theta] + 1$;
                         else $\bar{\varphi}[\theta] = \bar{\varphi}[\theta] + 1$;
for j=1 to $K$ do
    Estimate the posterior probabilities $P\left(E_{\vec{C}_x(y)}^y | H_b^y\right)$;
Identify $K$ nearest neighbors $N(x_{new})$ for $x_{new}$;
for j=1 to $K$ do
    Calculate statistic $\vec{C}_{x_{new}}(y)$;
Return $\vec{L}_{x_{new}}(y)$;
T' = T' ∪ {($x_{new}$, $Y_{new}$)};
// $Y_{new}$ is the label set of $x_{new}$ converted by $\vec{L}_{x_{new}}(y)$
```

Step 7: let $H_1^y$ denote the event that the instance x contains label y and $H_0^y$ denote the event that the instance x does not contain label y. Let $E_j^y$ denote that instance x has j neighbors containing label y. $\mathrm{P}\left(H_1^y|E_j^y\right)$ represents a probability that instance x has j neighbors containing label y and itself also contains label $y$. Correspondingly, $\mathrm{P}\left(H_0^y|E_j^y\right)$ represents the probability in the opposite case described above. According to MAP principle [Zhang and Zhou (2007)], the category vector $\vec{y}_x(y)$ for unseen instance x is determined using the following maximum a posteriori principle：

$$\vec{y}_x(y) = argmax_{b\in(0,1)}P\left(H_b^y|E_{\vec{C}_x(y)}^y\right), y \in \mathrm{Y} \qquad (11)$$

Based on Bayesian rule [16 Zhang and Zhou (2007)]:

$$\vec{y}_x(y) = argmax_{b\in(0,1)}P\left(H_b^y\right)P\left(E_{\vec{C}_x(y)}^y|H_b^y\right), y \in \mathrm{Y} \qquad (12)$$

In the process of algorithm DR-ML-KNN, we firstly extract the features from the vehicle video and performs the de-redundancy operation on the transposed matrix of the feature matrix, so that we get the first k principal instances. This operation removes similar instances and reduces the calculation of finding the neighbors of the test instance in DR-ML-KNN. Also, after predicting labels for the test instance, we add the test instance to original training set to enrich the training set and optimize the classifier. What's more, we can predict the label set of the test instance more accurately.

## 5 Experiment and analysis

### *5.1 Evaluation metrics*

We choose five evaluation criteria for multi-label learning algorithms mentioned in [Zhang and Zhou (2007)]: Hamming Loss, One-error, Coverage, Ranking Loss, Average Precision.

**Table 2:** Experimental data

| Name | Example | Label | Training Set | Test Set |
|---|---|---|---|---|
| BRVD1 | 70 | 5 | 50 | 20 |
| BRVD2 | 70 | 5 | 50 | 20 |
| BRVD3 | 69 | 5 | 50 | 19 |
| BRVD4 | 69 | 5 | 50 | 19 |
| BRVD5 | 119 | 5 | 80 | 39 |

### *5.2 Dataset*

In this paper, we choose five vehicle video data sets in Berkeley Road-Vehicle Dataset (BRVD) to verify the performance and advantages of the algorithm DR-ML-KNN. Our proposed DR-ML-KNN is compared with the traditional ML-KNN and the ML-KNN that reduce feature extraction frequency, recorded as V-ML-KNN. As far as we know, this paper applies the multi-label classification method on vehicle video label prediction for the first time and there is no other algorithm to compare.

**Table 3:** The result of Hamming Loss

| Dataset | ML-KNN | V-ML-KNN | DR-ML-KNN |
|---|---|---|---|
| BRVD1 | **0.1951** | 0.2846 | 0.1967 |
| BRVD2 | 0.2814 | 0.2802 | **0.2012** |
| BRVD3 | 0.3153 | 0.3199 | **0.1486** |
| BRVD4 | 0.2870 | 0.2803 | **0.1997** |
| BRVD5 | 0.2893 | 0.2823 | **0.1801** |

**Table 4:** The result of One-error

| Dataset | ML-KNN | V-ML-KNN | DR-ML-KNN |
|---|---|---|---|
| BRVD1 | 0.2057 | 0.2299 | **0.2048** |
| BRVD2 | 0.1469 | 0.1693 | **0.1290** |
| BRVD3 | **0.2698** | 0.2994 | 0.2908 |
| BRVD4 | 0.2269 | 0.1820 | **0.1426** |
| BRVD5 | 0.2057 | 0.2399 | **0.2037** |

**Table 5:** The result of Coverage

| Dataset | ML-KNN | V-ML-KNN | DR-ML-KNN |
|---|---|---|---|
| BRVD1 | 4.4229 | 4.3377 | **4.2986** |
| BRVD2 | 4.2540 | 4.3141 | **4.0471** |
| BRVD3 | 4.9138 | 4.9761 | **4.8534** |
| BRVD4 | 4.6713 | 4.5474 | **4.3216** |
| BRVD5 | 4.5775 | 4.3722 | **4.1886** |

### *5.3 Effectiveness on DR-ML-KNN*

The algorithm is implemented by MATLAB. The experiments are carried out on MATLAB 2016a, running on a PC with Inter(R) Core 3.2GHz CPU with 4GB of memory. DR-ML-KNN is compared with traditional ML-KNN and V-ML-KNN on 5 real-world multi-label datasets.

For average precision, the larger the value, the better the performance. For the other four metrics, the smaller the values, the better the performance.

**Table 6:** The result of Ranking Loss

| Dataset | ML-KNN | V-ML-KNN | DR-ML-KNN |
|---|---|---|---|
| BRVD1 | 0.2446 | **0.2184** | 0.2318 |
| BRVD2 | 0.2262 | 0.2327 | **0.2183** |
| BRVD3 | 0.2950 | 0.3091 | **0.2702** |
| BRVD4 | 0.2470 | 0.2380 | **0.1986** |
| BRVD5 | 0.2576 | 0.2385 | **0.2271** |

**Table 7:** The result of Average precision

| Dataset | ML-KNN | V-ML-KNN | DR-ML-KNN |
|---|---|---|---|
| BRVD1 | 0.6951 | 0.7019 | **0.7061** |
| BRVD2 | **0.7122** | 0.7102 | 0.7112 |
| BRVD3 | 0.6500 | 0.6505 | **0.6786** |
| BRVD4 | 0.7096 | 0.7279 | **0.7536** |
| BRVD5 | 0.6894 | 0.7018 | **0.7111** |

As can be seen from Tab. 3, Hamming Loss for our algorithm is lower than ML-KNN and V-ML-KNN. The similarity between predicted label set and real set is above 80%. From Tab. 4-6, One-error, Coverage and Ranking Loss for our algorithm are all lower than ML-KNN and V-ML-KNN. From Tab. 7, Average Precision for our algorithm higher than ML-KNN and V-ML-KNN.

In summary, it can be seen from the above five tables that the algorithm proposed in this paper works well and improves the accuracy of the classification label.

### *5.4 Efficiency on DR-ML-KNN*

In the next set of experiments, we validate the efficiency of the proposed DR-ML-KNN. The time unit is seconds, including training time and predicting time. Runtime comparison is shown in Fig. 2.

From Fig. 2, we can see that our algorithm is more efficient than traditional ML-KNN. Although the runtime of V-ML-KNN is the least, its accuracy in Tab. 3 is quite low so that it will not be considered. In summary, when our algorithm is applied to the real world, it will have a good performance.

## 6 Conclusion

The traditional multi-label classification method can only predict labels, which can-not extract features from the original video data. Moreover, when the traditional multi-label classification method is applied to vehicle video data, the runtime is high and the classification performance is poor. In order to solve the above defects, we propose a multi-label classification method MCM-VV. We achieve the ultimate goal of reducing time complexity and improving classification accuracy. The experimental results on five vehicle video data sets show that the algorithm achieves better multi-label evaluation indicators and have less running time.

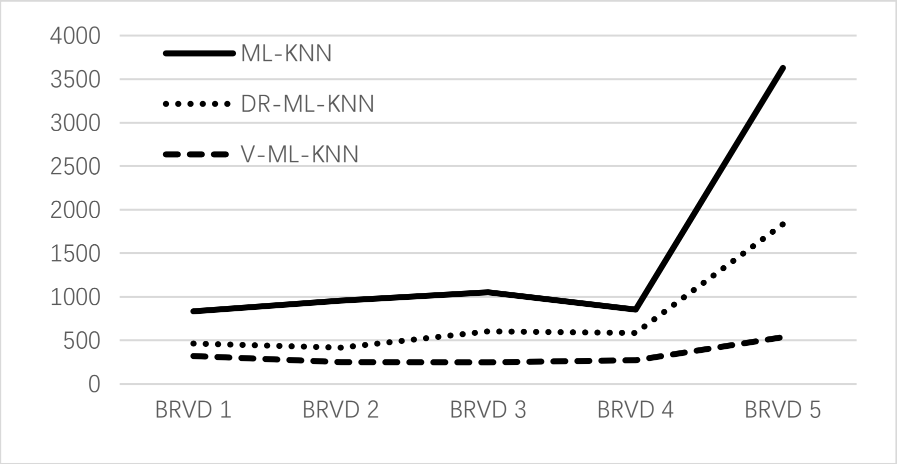

**Figure 2:** Runtime comparison of three algorithms

**Funding Statement:** This article was funded by the National Natural Science Foundation of China (Nos. 61702270, 41971343) and the China Postdoctoral Science Foundation (No. 2017M621592).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

**Chen, R.; Wang, M.; Chen, X.** (2015): Human detection based on HOG-PCA and LBP characteristics. *Information Technology*, no. 2, pp. 101-105.

**Ding, M. X.; Yang, Y. L.; Lan, Z. Q.** (2018): Multi-label imbalanced classification based on assessments of cost and value. *Springer Science+Business Media*, no. 48, pp. 3577-3590.

**Han, H. R.; Huang, M. X.; Zhang, Y.; Yang, X. G.; Feng, W. L.** (2019): Multi-label learning with label specific features using correlation information. DOI: 10.1109/ACCESS.2019.2891611.

**Jiang, J. Y.; Tsai, S. C.; Lee S. J.** (2012): FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors. *Expert Systems with Applications*, vol. 39, no. 3, pp. 2813-2821.

**Liu, S. M.; Chen, J. H.** (2015): A multi-label classification based approach for sentiment classification. *Expert Systems with Applications*, vol. 42, no. 3, pp. 1083-1093.

**Pan, W. S.; Jin, L. W.; Feng, Z. Y.** (2015): Recognition of Chinese characters based on multi-scale gradient and deep neural network. *Journal of Beijing University of Aeronautics and Astronautics*, vol. 41, no. 4, pp. 751-756.

**Qin, M.** (2012): *A Research of Lane Detection and Tracking Based on Machine Vision* (*Ph.D. Thesis)*. Ocean University of China.

**Tan, C.; Ji, G. L.; Zhao, B.** (2017): Self-Adaptive Streaming Big Data Learning Algorithm Based on Incremental Tangent Space Alignment. *Journal of Computer Research and Development*, vol. 54, no. 11, pp.2547-2557.

**Tanaka, E. A.; Macedo, A. A.** (2015): A multi-label approach using binary relevance and decision trees applied to functional genomics. *Journal of Biomedical Informatics*, vol. 54(C), pp. 85-95.

**Wang, X.; Liu, Y. C.; Hai, D.** (2017): Lane detection method based on double ROI and varied-line-spacing scanning. *Journal of Command and Control*, vol. 3, no. 2, pp. 154-159.

**Yu, Y.; Pedrycz, W.; Miao, D.** (2013): Neighborhood rough sets based multi-label classification for automatic image annotation. *International Journal of Approximate Reasoning*, vol. 54, no. 9, pp. 1373-1387.

**Zhang, H.** (2014): *Lane Line Detection Based on Digital Image Processing (Ph.D. Thesis)*. Xinjiang University.

**Zhang, J.; Mao, X. B.; Chen, T. J.** (2009): Survey of moving object tracking algorithm. *Application Research of Computer*, vol. 26, no. 12, pp. 4407-4410.

**Zhang, L.; Bao, P.** (2002): *Edge Detection by Scale Multiplication in Wavelet Domain.* Elsevier Science Inc.

**Zhang, M. L.; Zhou, Z.** (2006): Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338-1351.

**Zhang, M. L.; Zhou, Z.** (2014): A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, vol. 26 no. 8, pp. 1819-1837.

**Zhang, M. L.; Zhou, Z. H.** (2007): ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, vol. 40, no. 7, pp. 2038-2048.

**Zhang, Z. X.; Wang, M.; Geng, X.** (2015): Crowd counting in public video surveillance by label distribution learning. *Neurocomputing*, pp. 151-163.

**Zou, M. L.; Liu, Z. X.; Chen, X. Y.** (2019): A meaningful image encryption algorithm based on prediction error and wavelet transform. *Journal on Big Data*, vol. 1, no. 3, pp. 151-158.