Intelligent and Converged Networks
2024, 5(2): 134−146

ISSN 2708-6240
DOI: 10.23919/ICN.2024.0010

JudPriNet: Video transition detection based on semantic relationship and Monte Carlo sampling

Bo Ma, Jinsong Wu*, and Wei Qi Yan

Abstract: Video understanding and content boundary detection are vital stages in video recommendation. However, previous content boundary detection methods require collecting information, including location, cast, action, and audio, and if any of these elements are missing, the results may be adversely affected. To address this issue and effectively detect transitions in video content, in this paper, we introduce a video classification and boundary detection method named JudPriNet. The focus of this paper is on objects in videos along with their labels, enabling automatic scene detection in video clips and establishing semantic connections among local objects in the images. As a significant contribution, JudPriNet presents a framework that maps labels to "Continuous Bag of Visual Words Model" to cluster labels and generates new standardized labels as video-type tags. This facilitates automatic classification of video clips. Furthermore, JudPriNet employs Monte Carlo sampling method to classify video clips, the features of video clips as elements within the framework. This proposed method seamlessly integrates video and textual components without compromising training and inference speed. Through experimentation, we have demonstrated that JudPriNet, with its semantic connections, is able to effectively classify videos alongside textual content. Our results indicate that, compared with several other detection approaches, JudPriNet excels in high-level content detection without disrupting the integrity of the video content, outperforming existing methods.

Key words: video scene detection; Monte Carlo; object detection; Continuous Bag-of-Words

1 Introduction

Although previous studies have explored various attributes and semantic concepts related to video detection, classifying videos can be complex and challenging, particularly in the case of videos that portray human emotions. Videos involving human emotions, such as humor or historical shorts, present considerable difficulty in classification and defy straightforward mathematical modeling. It is noteworthy that human brain can readily discern video types based on language and visual expressions.

However, based on our prior observations and analysis, we have observed distinct variations in the shooting styles employed in romance and humor films. Furthermore, these differences are reflected in image features extracted from the videos and exhibit certain patterns. We aim to capture these features utilizing an object detection model[1] and classify the styles within given image datasets.

In this paper, our objective is to sample various types of videos by sampling methods and automatically assess their pixel characteristics for the purpose of pixel-based classification.

Motivation. Video instance segmentation is used in scene segmentation. In the case of video instance segmentation, tracking objects over time requires analyzing multiple video frames simultaneously to

• Bo Ma and Wei Qi Yan are with the School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1024, New Zealand. E-mail: rcn4743@aut.ac.nz; weiqi.yan@aut.ac.nz.
• Jinsong Wu is with the School of Artificial Intelligence, Guilin University of Electronic Technology, Guilin 540004, China, and also with the Department of Electrical Engineering, University of Chile, Santiago 8010037, Chile. E-mail: wujs@ieee.org.
∗ To whom correspondence should be addressed.

Manuscript received: 2023-10-09; revised: 2023-10-26; accepted: 2024-02-09

accurately locate segmentations[2]. This requires the association of object detection objects and label annotations to effectively understand the video content. This is also the motivation for the framework proposed in this article and the problem that JudPriNet wants to solve.

In 2020, Rao et al.[3] proposed a BNet method to segment videos based on their semantic content. However, BNet aims to cut videos of different scenes based on scene transitions, locations, actors, actions, and audio[3]. If any of the above five elements are wrong or missing, it will be difficult to edit the video according to the scene. On this basis, JudPriNet in this article hopes to extract labels through sampling, helping to eliminate possible obstacles to effectively extracting semantic information. This is also the second main motivation of this paper.

Contribution. In this paper, we propose a video scene transition detection method called JudPriNet. The contributions are as follows:

First, the method in this article is based on the time series of tags extracted from the video to form the vector relationship of the scene description to determine whether the video scene has been transformed through the vector relationship.

The second contribution is that this article classifies the labels using the video annotations from the annotation objects in the video and introduces the Continuous Bag of Visual Words (CBOVW) model into the classification to make the classification more accurate. Specifically, the method in this article maps labels to a "Continuous Bag Visual Words Model" and creates a relationship tree. The generated model will apply the Monte Carlo sampling method to classify video clips, so that after classification, it can accurately map the classification results back to the relationship tree to classify the scene accordingly. These two contributions are two most important innovations proposed by JudPriNet in this paper.

2 Related work

2.1 CBOVW

As a Nature Language Processing (NLP) method, Word2Vec includes a number of related models to produce word embedding to predict the probability of the current word based on the context[4], which takes use of two model architectures to generate a distributed representation of words, Continuous Bag-of-Words (CBOW) or Continuous Skip-Gram (SKIP-GRAM). Figure 1 illustrates traditional CBOW, which can represent the text document, whose goal is to predict the probability of the current word based on the context.

In computer vision, the researchers explored the Bag-of-Words in CBOW, and the CBOW can encapsulate the image statistics of hundreds of local features[5] into a vector, which has been applied to pre-trained convolutional networks and calculated image representation from intermediate feature maps[6]. However, a very limited number of works have integrated CBOW into the monastery training pipeline. Among them, NetVLAD[7] imitates the Vector of Locally Aggregated Descriptors (VLAD) via learning visual vocabulary and other layers and soft-quantified activation of the vocabulary.

2.2 Traditional deep learning multi-label text classification model

Multi-label Text Classification (MTC) can automatically adapt the parameters for the hidden layer

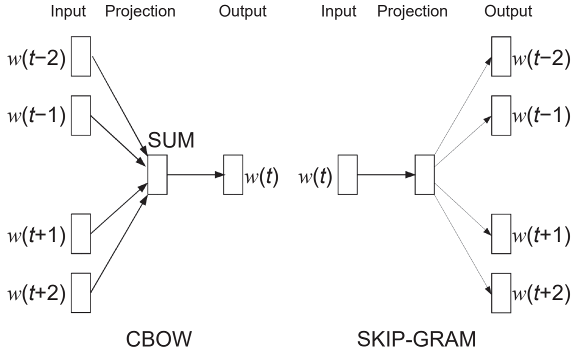


Fig. 1 Traditional CBOW model and SKIP-GRAM model[4]. The Word2Vec model[4] has two types of model, the first model is CBOW and the second model is SKIP-GRAM. Those two models are conducting the opposite operation. Input the CBOW the Word vector (as $w(t-1)$,$w(t)$, ..., $w(t+n)$), then calculate to SUM, and output is $w(t)$. w is the word vector and t is the time series. The SKIP-GRAM input $w(t)$ and put to projection, last output to $w(t-1)$,$w(t)$, ..., $w(t+n)$.

of the maximum pool to the information in the video via capturing more fine-grained information[8]. In 2017, Nam et al.[9] converted the relevant tags into a predicted binary sequence and took use of the Recurrent Neural Network (RNN) to complete tag prediction. In 2018, Wang et al.[10] trained joint words through word embedding, which can use context-sensitive methods to parse tags and form a Word2Vec. In 2018, Zhang et al.[11] proposed to perform label prediction by searching the label space. On this basis, Yao et al.[12] used Graph Convolutional Network (GCN) to jointly learn word and document embedding in graph text representation, and applied GCN to extract the label information. However, GCN methods can extract the semantics of labels with visual objects, but the classification features cannot describe the boundary of the scene. Our proposed methods take into consideration video clips and create a words (labels) vector relationship strategy to decide the relationship between the labels, and combine the Monte Carlo sampling method with our proposed JudPriNet to determine the boundary of video scene.

2.3 Object detection

Our proposed method integrates an existing object detection framework to recognize internal objects and features in the video, which requires collecting the features of detected objects and regenerating them into a new hierarchy tree with their semantic expressions. Note that the existing object detection integrates the combination of object classification and object localization in the scene. In terms of speed and accuracy, Redmon et al.[13] proposed “You Only Look One” (YOLO) model, which is one of the latest object detection methods based on Deep Neural Networks (DNNs). YOLO can accelerate object detection in videos in embedded devices in real-time, and use the evolved deep intelligence framework to evolve the YOLO network architecture and produce an optimized architecture.

In order to further reduce the power consumption of embedded devices while maintaining performance, a motion-adaptive inference method is introduced into the proposed JudPriNet combined with latest YOLO[14] framework to reduce the frequency of the deep inference of YOLO based on temporal motion characteristics.

3 Methodology

The name of JudPriNet reason has improved the works of BNet[3] and graph convolutional network[12], and further, a new perspective to understand the video contents and judge the boundaries of scenes in the video. Figure 2 shows the JudPriNet framework with MOT16[15] as a test case. After the labels for Shots 1, 2, and 3 are detected and collected, those labels are encoded via CBOVW, and then the number of words or labels has been counted by the proposed model. After the labels are processed via prepared stages and reconstructed, JudPriNet clusters the vector words via Mento Carlo sampling algorithm. After assessing loss with the cross-entropy loss function, the model will decide the recycling behavior or generate the prediction model and inference for the test video. While processing the label information, the JudPriNet model needs to collect and process the labels. Then, the labels are required to be associated with the vector and JudPriNet establishes the relationship tree according to their semantic relationship.

3.1 Model initialization and iterated training

In model initialization stage, the labels are grouped into a vector, and then a word vector structure can be established for different labels via detecting the similarity within the labelled objects. Each label represents a kind of words, and the similar words in the word vector space can be employed to the semantic similarity of the labels. In order to establish a context-sensitive Continuous Bag-of-Words structure, we combine object detection solutions to obtain the objects in the video and then make use of the proposed solution JudPriNet to detect the scene of the video.

$$\begin{gathered}\text{vector(“Person”)} - \text{vector(“Middle-aged man”)} - \\ \text{vector(“Elder man”)} - \text{vector(“Woman”)} = \\ \text{vector(“Yound man”)}, \\ \text{vector(“Boy”)} - \text{vector(“Man”)} + \\ \text{vector(“Woman”)} = \text{vector(“Girl”)}\end{gathered} \tag{1}$$

The words “Young man” and “Girl” are the results,

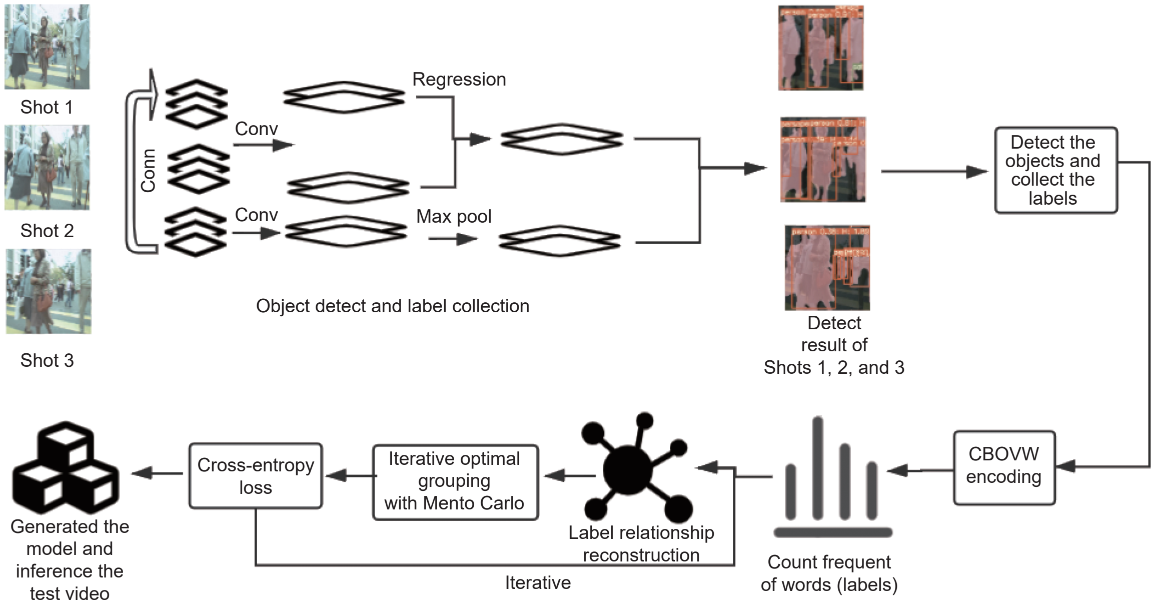

Shot 1
Shot 2
Shot 3
Conn
Conv
Conv
Regression
Max pool
Object detect and label collection
Detect result of Shots 1, 2, and 3
Detect the objects and collect the labels
CBOVW encoding
Count frequent of words (labels)
Label relationship reconstruction
Iterative optimal grouping with Mento Carlo
Cross-entropy loss
Iterative
Generated the model and inference the test video

Fig. 2 JudPriNet Framework (with MOT16 datasets as motivation case[15]). The top left side corresponds to the whole video timeline where the top right side regions represent the inference results of objects from YOLO[13]. In the left Shot 1, people in the video are walking in the street, which thus requires a semantic understanding of this scene to prevent it from categorizing them into different scenes. In the left Shot 3, the task becomes even more difficult, as objects have disappeared.

respectively. Moreover, this semantic relationship is not obtained using artificial prior knowledge such as WordNet[16] but using purely statistical methods such as Huffman coding to avoid the difficulty and workload of manual construction. CBOVW, in essence, can be considered as a distributed representation method in vocabulary vectorization, we propose all object labels in target video clips as the words and establish the semantic relationship with lexical vector for those labels.

After obtaining the optimal vector representation of each label (word), then the model needs to establish the lexical-semantic connection according to the feature of the label (word) vectors. One of the simplest solutions is to sum the values of document vocabulary vectors directly and obtain the average value from them as

$$v_d = \frac{1}{\sum\limits_{a_i \in d} \mathrm{fs}_{a_i}} \sum_{a_i \in d} \mathrm{fs}_{a_i} v_{a_i} \tag{2}$$

where fs_{a_i} means the frequency of the word a_i appears in the vector v_{a_i}, and subscript a_i means the word vector, v_d is characterization word (label) vector d. The construction of such a word vector only is applied to the appearance of words within the vector and the frequency with which these words (labels) appear inside the vector.

If labels need to vectorize the vector in a corpus, one way refers to $\mathrm{ts}.ds_i$ design ideas that use coefficients $\mathrm{fs}.da_i$ between different words (labels) instead of word frequencies s_{a_i}, which is given by

$$v_d = \sum_{a_i \in d} \eta(a_i, d)\, s_{a_i} \tag{3}$$

where $\eta(a_i, d)$ means that the word for the vector d has a coefficient as $\mathrm{fs}.da_i$:

$$\eta(a_i, d) = \begin{cases} (1 + \log(\mathrm{fs}_{a_i})) \times \log \dfrac{N}{\mathrm{fs}_{a_i}}, & \text{if } \mathrm{fs}_{a_i} \geqslant 1; \\ 0, & \text{if } \mathrm{fs}_{a_i} = 0 \end{cases} \tag{4}$$

In order to vectorize the vector in the label vectors, JudPriNet introduces the CBOW proposed by Le and Mikolov[17], JudPriNet improved the traditional Word2Vec[18] by adding a paragraph vector. As our method, there are two parts to the proposed model: Distributed Bag of Words (DBOW) and Distributed Memory (DM)[19], where we attempt to use DBOW to predict the probability of a set of random words in a given paragraph, while DM function is to estimate the probability of a word given the context and paragraph vectors.

It should be pointed out that CBOVW in Fig. 3 and DBOW methods fs.da_i are applied to words (labels) groups in a closed corpus, that is, the text in the corpus is fixed. However, in video clips scenarios, continuous streaming of new labels needs to be performed as objects, and most of objects are moving. Thus both methods can not be applied to moving words (labels). In addition, the DBOW method claims to have achieved good results, its actual effects have been disputed in academia.

3.2 Frequent substring mining with vocabulary semantics

Vocabulary is the basic unit of natural language processing. The above describes the establishment of a variety of document space vector models, but they are all based on the correct identification and division of vocabulary. At present, the label (word) segmentation methods can be used in a large number of documents, but most of these label (word) segmentation and classification methods still require a relatively good relationship tree or the relationship tree can significantly improve the word segmentation effect[20]. However, under the current circumstances, a large number of proper words (labels), and new labels have emerged one after another, creating serious difficulties for the establishment of artificial relationships.

From the vocabulary composition point of view, the vocabulary can be considered as a fixed combination of characters with certain semantics. Therefore, it is useful to construct a relationship by considering the word (label) as a sequence of characters and mining the frequently occurring substrings from these sequences. In the following, we discuss how to find out scene transition in video clips.

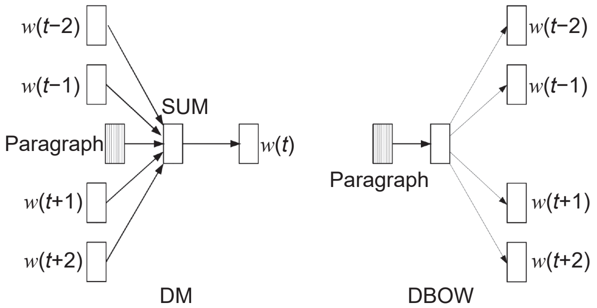


Fig. 3 CBOVW model for JudPriNet. There are differences between this model and the model in Fig. 1. CBOVW has two parts, DBOW and DM. In DM parts, there is a paragraph between $w(t-1)$ and $w(t+1)$ to sort the new labels (words), and the value of output $w(t)$ is different from that in Fig. 1, and in DBOW part, the input is changed to the paragraph, and output is $w(t-1), w(t), \ldots, w(t+n)$.

3.3 Video type inference with scene transition

Usually, video clips consist of well-designed series or disorderly and interesting scenes. Therefore, it is necessary to detect and segment the boundaries of video scenes to achieve the task of video classification, so as to extract valid video clips in disordered scenes. Assuming that the segmented video clips are all continuous frames of pictures, the detected internal objects will through the segmented video content, which helps the method in this paper reduce the semantic complexity.

Rao et al.[3] proposed a BNet method to segment the video according to their semantic content, but the aim of BNet is to cut the video with different scenes according to their manual configure, which can hardly extract the similar detect objects of elements from their segmented result. Thus in this paper, we propose a Monte Carlo sampling method to find the boundary of a video clip without impacts on the followed object detection.

4 Classification for multi-label vector

The aim of JudPriNet is to classify the labels according to their video type, and the categorization of labels is the key stage to achieve this goal. The label[21] is a mapping process that maps an unclassified text to the existing categories. The mapping can be either a one-to-one or a one-to-many mapping. However, it implies that classifiers will always classify the label from one category. Under the model of this situation, the marked label categories instead of all categories often decide the vector connection. The relationship tree of labeled categories does not cover the entire word (labels) space. Since only forward label vectors are labeled, the category boundaries identified by the classifier are only applicable to distinguishing different categories from the forward correlation data but are not applicable to discriminate whether the new sample belongs to the content of the training set, that is, all of the categories cannot be identified. The problem about identifying category boundaries based on positive samples is academically referred to as Multiple Class Classification (MCC)[22]. In order to establish the relationship between the inference video clips and multi classes, JudPriNet will process the unannotated image training set and extract the labels to create the relationship with the Coutinuious Bag-of-Words (labels) method.

We assume that n is the number of unannotated image training sets. Let S_{a_i} represent a continuous image sequence consisting of a_i frames, where f_t represents the frame at time t, and a_i is the corresponding of the frame position.

Let D_l indicate the characteristics of the study sampling. Our goal is to convert the image l (image of a set of continuous frames) to the feature subset Seq_t at time t. Through the index gear sequence, the extraction segmentation shape of the current t_S is gradually refined based on the shape of the previous frame.

The basic idea of JudPriNet is to convert the feature index sequence Seq_i into a set of frame position a_i via utilizing additional information of space appearance features in the spatial dimension and the consistency information of the neighbor frame in the time dimension. To this end, we have developed the following optimization target functions.

JudPriNet strategy processes the feature index sequence with μ establish words (labels) vector relationship strategy π. With any $l \in L$, we have

$$\begin{aligned} & p_\pi(l,\pi'(l)) = \sum_{a_i} \pi'(a_i|l) p_\pi(l,a_i) = \\ & \quad \frac{\mu}{\mathcal{E}(l)} + (1-\mu)\sum_a p_\pi(l,a) + (1-\mu)\max_a p_\pi(l,a) \geqslant \\ & \quad \frac{\mu}{\mathcal{E}(l)} + (1-\mu)\sum_a p_\pi(l,a) + \\ & \quad (1-\mu)\sigma_a \frac{\pi(a|l) - \frac{\mu}{\mathcal{E}(l)} + (1-\mu)}{1-\mu} p_\pi(l,a) \end{aligned} \tag{5}$$

$$s_\pi(l) = (1-\mu)\max_v p_\pi(l,v) + \frac{\mu}{\|\mathcal{E}(l)\|}\sum_v p_\pi(l,v) \tag{6}$$

Theorem 1 (Establish words (labels) vector relationship strategy of F) We suppose that F uses randomized distribution test and parameters $k \leqslant 1, \gamma \leqslant 1$, and $\varepsilon_0 > 0$ to represent Mechanism 1. For any set of adjacent numbers D and D' such that $|D|, |D'| \leqslant k$, any set of results $\gamma \leqslant U$, and any integer $1 \leqslant t \leqslant k$, we have

$$L = \{\mathcal{F}(D') \in Y\} \leqslant \mathrm{e}^{\varepsilon} L\{\mathcal{F}(D') \in Y\} + \delta \tag{7}$$

where $\delta = \mathrm{e}^{-\varepsilon}(k-t)$ and $\varepsilon = \varepsilon_0 + \ln\left(1 + \frac{\gamma}{t}\right)$.

We start with symbols. Let U denote the overall data record. All data records, i.e., the data records from datasets D and D', including the synthetic records produced by M and F are elements of U. Let D and D' denote two adjacent datasets, i.e., $D = D'd$ for some $d \in U$ or $D = D \cup d'$ for a certain $d' \in U$. We assume that both D and D', there are at least k records, and we have parameters $k \leqslant 1$, $\gamma > 1$, and $\varepsilon_0 > 0$. For convenience, we write $\mathcal{P}_d(y) = L\{y = M(d)\}$, and only refer to M implicitly.

Given a dataset D^*, we infer the probability that the synthetic record y is released: $L\{F(D^*) = y\}$. We observe that the records of a given synthetic record $y \in U$, D' can be divided by privacy criteria (become disjoint sets). Specifically, let $I_d(y)$ be the partition number of the record $d \in D$ for y. The partition number $I_d(y)$ is a unique non-negative integer, so that $\gamma^{-(I_d(y)+1)} < \mathcal{P}_d(y) \leqslant \gamma^{-I_d(y)}$. In other words, $I_d(y) = \lfloor -\log_\gamma \mathcal{P}_d(y) \rfloor$, if $\mathcal{P}_d(y) = 0$, then the partition number is undefined. Similarly, we define the partition (or set of partitions) with $i \leqslant 0$ as $C_i(D^*, y) = d : d \in D^*$, $I_d(y) = i$. In other words, a partition i is a record set with partition number i.

A key step is to express the probability $L\{F(D^*) = y\}$ with the following aspects: (1) the probability of generating y from a specific partition (that is, the seed is in the partition) and (2) the test. For aspect (2), the possibility of passing the distribution test depends only on the division of seeds. The detail is shown in Algorithm 1.

Cross-entropy loss for label (word) relationship reconstruction. In this paper, we study the theorem of loss function for label (word) reconstruction as follows:

Theorem 2 (Cross-entropy loss function for label (word) reconstruction) The value of $Q(\Theta, \hat{\Theta})$ during the measure the cross entropy loss step of the label

Algorithm 1 Labels (words) vector relationship search algorithm

```
Input: Matrix $A \in \mathbb{R}^{n\times d}$ whose boundary $B > 0$. In any row of $A$ in the $l_2$ paradigm, there are labels vector parameters: $\varepsilon, \delta > 0$.
The parameter $r$ represents the number of rows in the output matrix.
The dataset $\omega$ is constrained by the equation
$\omega^2 = \frac{(8B^2)}{\epsilon}\left(\sqrt{2r\ln(\frac{8}{\delta})} + 2\ln\left(\frac{8}{\delta}\right)\right)$.
Output: The sample data $Z\,\mathrm{Lap}\left(\frac{(4B^2)}{\epsilon}\right)$ and $\sigma_{\min}(A)$ represent the minimum singular value of $A$.
1: if $\sigma_{\min}(A)^2 > \omega^2 + Z + \frac{\left(4B^2\ln\left(\frac{1}{\sigma}\right)\right)}{\epsilon}$ then
2:     Sample $(r\times n)$-matrix $R$, whose entries are independent and identically distributed samples from normal Gaussian.
3:     return RA (return value $A$) and "matrix unchanged"
4: else
5:     Let $A'$ denote the result of adding $A$ to the $d\times d$ matrix $wI_{d\times d}$.
6:     Sample $a(r\times(n+d))$-matrix $R$, whose entry is identically distributed sample from ordinary Gaussian.
7:     return $\mathrm{RA}_0$ (return value $A_0$) and "matrix change"
8: end if
```

(word) reconstruction function is given by

$$Q(\Theta,\hat{\Theta}) = \sum_{i=1}^{K}[\psi_i(z;\hat{\Theta})\ln\theta_i] \tag{8}$$

where

$$\Psi_i(z;\hat{\Theta}) = \hat{\theta}_i\sum_{j=1}^{N}\frac{\Pr(Y \in z_j - \Omega_i)}{f_{z;\hat{\Theta}}(z_j)} \tag{9}$$

and if $z_j - v = \Omega_i$, it has $v = z_j - \Omega$.

Theorem 3 (Maximize value for cross-entropy loss function) The value of Θ that maximizes $Q(\Theta,\hat{\Theta})$ during the cross-entropy loss measurement phase of the reconstruction algorithm is given by

$$\theta_i = \frac{\Psi_i(z;\hat{\Theta})}{m_i N} \tag{10}$$

and

$$\Psi_i(z;\hat{\Theta}) = \hat{\theta}_i\sum_{j=1}^{N}\frac{\Pr(Y \in z_j - \Omega_i)}{f_{Z;\hat{\Theta}}(z_j)} \tag{11}$$

5 Experiment

According to the experimental results in Tables 1 and 2, the performances on Mask R-CNN+XML and JudPriNet+DeepLabV3 for labels classification accuracy are not as good as that of others, but JudPriNet+YOLOV5 outperforms others. The reason is that those two frameworks use different classifiers, and the tests show that the JudPriNet integrated with YOLOV5's classifier is with better performance. Thus, we conduct detailed analysis of these data to help understand the tasks and provide insights for the proposed solutions.

5.1 Test dataset with MOT16, YouTube-VOS, and BoundingBoxes

Since MOT datasets[15] are for evaluating few-shot methods, there are three different splits of classes,

Table 1 Results of multi-label classification.

Module	P@mAP (%)	P@AP75 (%)	Classification	Segmentation
Mask R-CNN+XML[11]	72.06	33.94	√	√
PredRNN[23]	76.72	35.96	√	√
Resnet101+ACf[24]	85.90	77.50	√	×
Bi-ConvLSTM[25]	84.60	75.30	√	×
BoWNet+Conv5[26]	85.60	75.30	√	×
JudPriNet+DeepLabV3	74.10	72.40	√	√
JudPriNet+YOLOV5	86.10	78.40	√	×

Table 2 Results of classification and few-shot with MiniImageNet datasets.

Module	AP (%)	Video content understanding	Segmentation	Label predict
BoWNet+Conv4[26]	77.2	√	×	√
BoWNet+Conv5[26]	79.6	√	×	√
LGSS (Multi-Semantics+BNet)[3]	76.2	√	×	×
JudPriNet+DeepLabV3	81.4	√	√	√
JudPriNet+YOLOV5	71.4	√	×	√

train, validation, and test with 64, 16, and 20 classes, respectively. Each class has 600 images with 84×84 resolution. YouTube-VOS[27] and BoundingBoxes[28] are video clips from the YouTube.com website. The YouTube-BoundingBoxes datasets[28] contain 10.5 million artificially labeled frames on 23 object categories, including 5 million bounding boxes. This dataset includes densely labeled segments of 380 000 YouTube video clips, each 15–20 s in length, taken from 240 000 videos. The image quality of these clips is similar to that of a mobile phone camera. The labeling and bounding box accuracy of this dataset exceeds 95%. So far, it is the largest artificially annotated video dataset containing bounding boxes for tracking objects in consecutive frames in time. YouTube-VOS datasets contain 4453 YouTube video clips featuring 94 categories covering humans, common animals, vehicles, and accessories. Each video clip lasts around 3 s to 6 s, contains multiple objects, and is manually segmented by professional annotators.

We train JudPriNet that corresponds to the training classes following the same training protocol as for image collection or video with consecutive frames.

5.2 Experiment for label prediction in objects detection

Tables 1 and 2 compare five multi-label classification methods, where JudPriNet proposed in this paper is combined with YOLO[14], and DeepLabV3[29] classification and detection frameworks. We conducted YouTube-VOS[27] dataset in Table 1 to count the results of the average accuracy of all classes for these 7 algorithms and compare them at the same time. Those 7 algorithms all use the same graphics processor NVIDIA RTX 3090 GPU card, CPU AMD Ryzen 7 3700x, the memory 64 megabytes (MB). P@mAP means precision in mean average precision, P@AP75 means average precision under 75% nonmaximal suppression rate, and AP means average precision. Tables 1 and 2 show the comparisons among JudPriNet with DeepLabV3 and YOLO and the other five algorithms. Table 2 shows that the latter two algorithms achieve unique content understanding and judgment performance. Compared with Resnet50+ACf and ConvLSTM without added content judgment, JudPriNet considers the relevance of the identification label and adjusts the label class with a low confidence rate to provide more accurate label classification information. From Table 1 among all the algorithms, JudPriNet combined with YOLO[14] has the best Mean Average Precision (mAP), and most mAPs with target recognition algorithms are also close to the optimal value (above 80), which shows that JudPriNet with content judgment is in target label prediction is more effective in classification. In Tables 1 and 2, the BoWNet+Conv4 and BoWNet+Conv5 are using BoWNet, which is a type of GCN[12] we mentioned in Section 2.2. The Local-to-Global approach to Multi-modal Movie Scene Segmentation (LGSS) contain the models of Multi-Semantics and BNet[3], the BNet we have mentioned in Section 1 as one of our motivation comparison works.

Interestingly, the accuracy of JudPriNet combined with DeepLabV3 is significantly weaker than the accuracy of the other four learning algorithms, but it can perform real-time image segmentation through the framework of DeepLabV3. For image understanding, both BoWNet and LGSS can predict the scene boundary of video, and BoWNet+Conv5 provides better accuracy precision for video boundary based on their content, which outperforms the LGSS (Multi-Semantics+BNet) approach. Since segmentation can provide more detailed information for objects and more accuracy with label, JudPriNet with DeepLabV3 has 81.4% accuracy and performs better than JudPriNet combined with YOLOV5[14].

In order to evaluate the performance of our proposed labels (words) vector relationship search algorithm, we then examine the sampling ability and problem. We assume that $B=\mathcal{H}^2[0,1]$ is label (word) vector space which can investigate the effect of parameters on the using privacy level via Monte Carlo sub-sampling. We consider the problem of estimating average performance from a random sensitive sample of functional observations using label relationship reconstruction vector $\mathcal{H}$,

$$\mathcal{H}(x,y)=\exp\left\{\frac{-|x-y|^2}{\delta}\right\} \tag{12}$$

$$X_i(t)=\mu(t)+\sum_{j=1}^{m} j^{-\frac{2}{p}}U_{ij}v_j(t), \in[0,1] \tag{13}$$

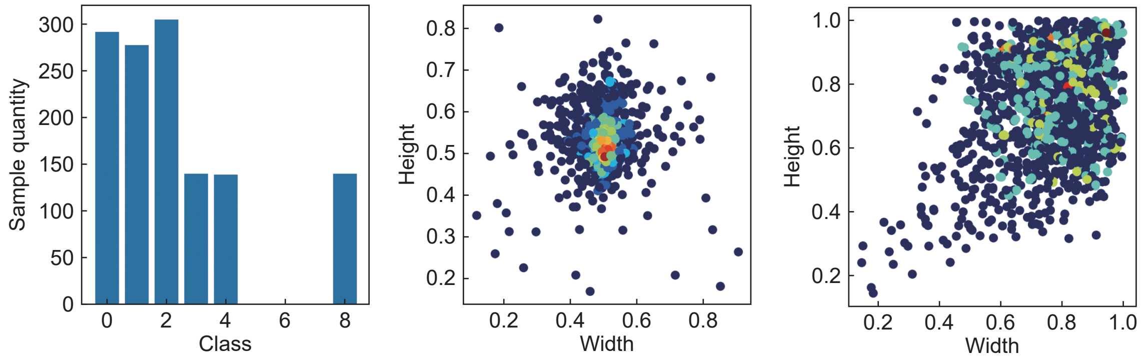
In Eqs. (12) and (13), x and y represent input datasets in the space $\mathcal{H}$, and δ means that the risk is fixed via choosing ε and δ in the definition of differential privacy. All comparisons include privacy restrictions ε and other factors that occur on a grid of equal distances between 0 and 2 with 5 points, and the label relationship reconstruction vector is the same ($\delta=0.1$). Vector $\mathcal{H}$ range parameter is $\rho=0.001$ and the parameter of $X_i(t)$ is set at $p=4$. The median function, test video size, and super parameters are set to $\mu(t)=0.1\sin(\pi t), N=25, (\varepsilon=1, \delta=0.1)$, respectively. We change the learning parameter ρ effect from 10^{-6}, 0.001 to 0.1.

5.3 Ablation study

Figures 4 and 5 show the prediction performance of JudPriNet under different categories, where the x and y axes in Figs. 4a and 5a show the category and quantity, and Figs. 4b and 5b and Figs. 4c and 5c show the projection of these labels in two dimensions in three-dimensional space. It can be seen that the vast majority of classes are concentrated in the middle part, while the distribution range of Classes 0–2 is the widest in Fig. 4 and the Classes 9 and 11 are the widest in both datasets in Fig. 5.

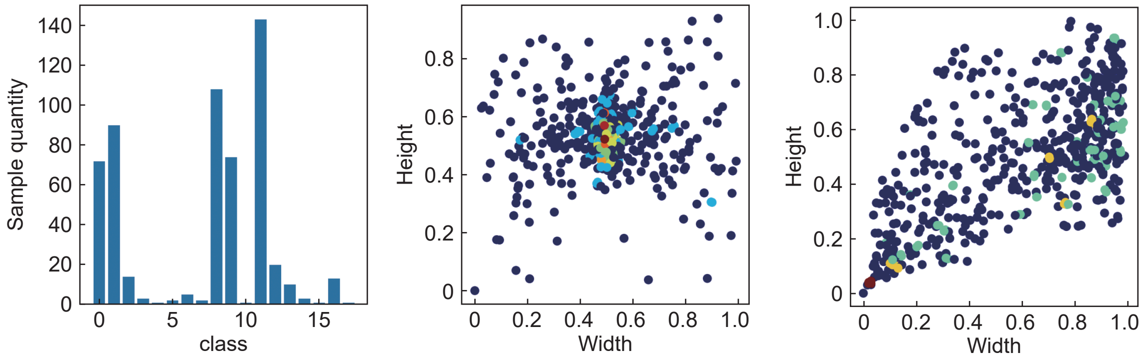
5.4 Receiver operating characteristic analysis for JudPriNet

The Receiver Operating Characteristic (ROC) curve[30] would come into play by plotting the true positive rate (sensitivity) against the false positive rate (1−specificity) at various decision thresholds. By analyzing the ROC curve, we can determine whether the JudPriNet model has the ability to make accurate predictions. An ideal model would have an ROC curve that approaches the top-left corner, indicating high sensitivity and low false positives. In contrast, a less effective model would produce a curve closer to the diagonal line, which represents random guessing.



(a) Category and quantity of classes (b) Mapping labels to cross-sections (c) Mapping labels to longitudinal sections

Fig. 4 Prediction performances with JudPriNet under MOT16[15] datasets.



(a) Category and quantity of classes (b) Mapping labels to cross-sections (c) Mapping labels to longitudinal sections

Fig. 5 Prediction performances with JudPriNet under YouTube-BoundingBox[28] datasets.

For datasets in this test, the Microsoft Research Video-to-Text (MSR-VTT) dataset[31] is designed for video captioning tasks, the images as Fig. 6 shows. It offers a large collection of video clips along with corresponding human-generated textual descriptions, making it a pivotal dataset for training and evaluating video captioning models. In addition, it also used VaTeX[32] datasets, which is a multilingual video and text dataset, which is instrumental for research in cross-modal understanding. It provides video clips with corresponding textual descriptions, offering an excellent resource for multilingual video analysis and translation tasks. In this ROC test, it also conducted MOT16[15] datasets, YouTube-VOS, and YouTube-BoundingBox[27] dataset.

Figure 7 shows the Area Under Curve (AUC) values for the mentioned datasets. The JudPriNet trained on the MOT16 dataset has an AUC of 0.89, indicating a good ability to distinguish between positive and negative instances in the context of YOLO[13] combined with JudPriNet. The JudPriNet trained on the YouTube-VOS dataset has an AUC of 0.91, suggesting a strong performance in video object segmentation. The JudPriNet model trained on the YouTube-BoundingBox dataset has an AUC of 0.87, demonstrating its effectiveness in object localization tasks. The JudPriNet model trained on the MSR-VTT dataset has an AUC of 0.91, showcasing its proficiency in generating textual descriptions for video content, which is relevant to video captioning. Lastly, the JudPriNet model trained on the VaTeX dataset excels with an AUC of 0.92, highlighting its exceptional performance in cross-modal JudPriNet understanding, which involves interpreting and connecting information between video and text.

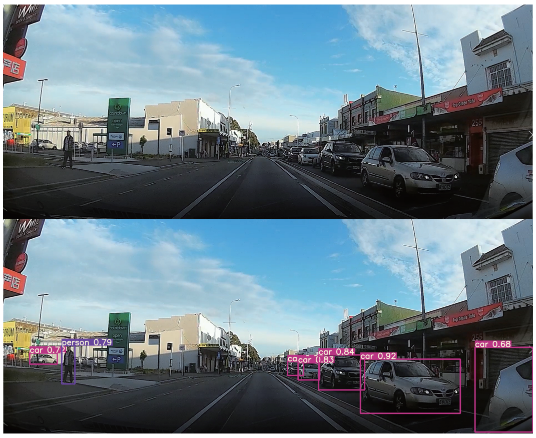


Fig. 6 Object label detecting process.

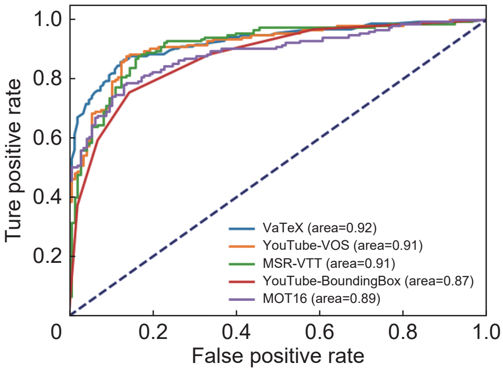


Fig. 7 Receiver operating characteristic for testing the JudPriNet.

In summary, these AUC values indicate that the JudPriNet testing on these datasets is generally well-fitted and effective in their respective tasks, with VaTeX demonstrating the highest performance in the context of ROC analysis.

6 Conclusion

We have shown that using the identification tags of the image recognition model through semantic recognition can not only provide support for video understanding but also provide very good support for later label prediction, while still maintaining the training efficiency unchanged. In order to achieve this goal, we have introduced a label relationship reconstruction algorithm, which can be used for stable mass training and enables us to optimize the judgment of video content categories through powerful data enhancement. Using simple architecture design principles, we have proposed the JudPriNet model, which obtains the most advanced performance on YouTube-VOS and YouTube-BoundingBox[27] and is comparable to competitive methods in label classification and video judgment.

Appendix

A Theorem and proof

Proof of Theorem 2 Since $X = X_1, X_2, \ldots, X_N$ are independent, so:

$$f_{X;\Theta}(V)=\prod_{i=1}^{N} f_{X_i;\Theta}(V) \tag{A1}$$

Expand:

$$\begin{aligned}Q(\Theta,\hat{\Theta}) =&E[\ln f_{X;\Theta}(X)|Z=z;\hat{\Theta}]=\\ &\sum_{j=1}^{N} E[\ln f_{X_j;\Theta}(X_j)|Z=z;\hat{\Theta}]=\\ &\sum_{j=1}^{N} E[\ln f_{X_j;\Theta}(X_j)|Z_j=z_j;\hat{\Theta}]\end{aligned} \tag{A2}$$

In Eq. (A2), only the expected value is calculated for X. This simplification results from the independence of X and Y. In addition,

$$f_{X_j|Z_j=z_j;\Theta}(v)=\frac{f_{X,Z;\hat{\Theta}}(v,z_j)}{f_{Z;\hat{\Theta}}(z_j)} \tag{A3}$$

The reasoning is

$$\begin{aligned}&E[\ln f_{X_j;\Theta}(X_j)|Z_j=z_j;\hat{\Theta}]=\\ &\int_{\Omega}\ln f_{X_i};\Theta(v)\frac{f_{X,Z;\hat{\Theta}}(v,z_j)}{f_{Z;\hat{\Theta}}(z_j)}\mathrm{d}v=\\ &\frac{\int_{\Omega}\ln f_{X_i;\Theta}(v)f_{X,Z;\hat{\Theta}}(v,z_j)\mathrm{d}v}{f_{Z;\hat{\Theta}}(z_j)}=\\ &\frac{\sum_{i=1}^{K}\int_{\Omega}\ln[f_{X_i;\Theta}(v)]\hat{\theta}_i I_{\hat{\Omega}_i}(v)f_Y(z_j-v)\mathrm{d}v}{f_{Z;\hat{\Theta}}(z_j)}\end{aligned} \tag{A4}$$

Equation (A4) comes from the assumption that f_X is a piece-wise constant. Because $I_{\hat{\theta}_i}$ is in the function Ω_i between 1.0 and 0, there are

$$\begin{aligned}&E[\ln f_{X_j;\Theta}(X_i)|Z_i;\hat{\Theta}]=\\ &\frac{\sum_{i=1}^{K}\hat{\theta}_i\int_{\Omega}\ln[f_{X_i;\Theta}(v)]f_Y(z_j-v)\mathrm{d}v}{f_{Z;\hat{\Theta}}(z_j)}=\\ &\frac{\sum_{i=1}^{K}\hat{\theta}_i\int_{\Omega}\ln[\theta_i]f_Y(z_j-v)\mathrm{d}v}{f_{Z;\hat{\Theta}}(z_j)}=\\ &\frac{\sum_{i=1}^{K}\hat{\theta}_i\ln\theta_i\int_{\Omega}f_Y(z_j-v)\mathrm{d}v}{f_{Z;\hat{\Theta}}(z_j)}=\\ &\frac{\sum_{i=1}^{K}\hat{\theta}_i\ln\theta_i\Pr(Y\in z_j-\Omega_i)}{f_{Z;\hat{\Theta}}(z_j)}\end{aligned} \tag{A5}$$

Combining Eqs. (A2) and (A5), we get the following expression about $Q(\Theta,\hat{\Theta})$:

$$\begin{aligned}Q(\Theta,\hat{\Theta})=&\sum_{j=1}^{N}\frac{\sum_{i=1}^{K}\hat{\theta}_i\ln\theta_i\Pr(Y\in z_j-\Omega_i)}{f_{Z;\hat{\Theta}}(z_j)}=\\ &\sum_{i=1}^{K}\hat{\theta}_i\ln\theta_i\sum_{j=1}^{N}\frac{\Pr(Y\in z_j-\Omega_i)}{f_{Z;\hat{\Theta}}(z_j)}=\\ &\sum_{i=1}^{K}\Psi_i(z;\hat{\Theta})\ln\theta_i\end{aligned} \tag{A6}$$

In that $\Psi_i(z;\hat{\Theta})=\theta_i\sum_{j=1}^{N}\frac{\Pr(Y\in z_j-\Omega_i)}{f_{Z;\hat{\Theta}}(z_j)}$. □

Proof of Theorem 3 Note that $\sum_{i=1}^{k}m_i\theta_i=1$. Therefore, we use the Lagrange multiplier to find the maximum value. We write the Lagrange multiplier function as follows:

$$L(\Theta,\lambda)=\sum_{i=1}^{K}\psi_i(z;\hat{\Theta})\ln\theta_i+\lambda\left(\sum_{j=1}^{K}m_j\theta_j-1.0\right) \tag{A7}$$

The Lagrangian constraint of the above function is $\frac{\partial L}{\partial\theta_i}=0$, and $\frac{\partial L}{\partial\lambda}=0$. The corresponding conditions are $\theta_i=-\frac{\psi_i(z;\hat{\Theta})}{\lambda m_j}$ and $\sum_{j=1}^{K}m_j\theta_j=1.0$. Eliminating the Lagrange multiplier lambda from these conditions, we get $\theta=\frac{\psi_i(z;\hat{\Theta})}{m_i\sum_{l=1}^{K}\psi_l(z;\hat{\Theta})}$.

At this stage, we only need to evaluate the denominator of the above θ_i value. To this end, we note:

$$\begin{aligned}\sum_{l=1}^{K}\psi_l(z;\hat{\Theta})=&\sum_{l=1}^{K}\hat{\theta}_l\sum_{j=1}^{K}\frac{\Pr(Y\in z_j-\Omega_l)}{f_{Z;\hat{\Theta}}(z_j)}=\\ &\sum_{l=1}^{K}\frac{\sum_{j=1}^{K}\hat{\theta}_l\Pr(Y\in z_j-\Omega_l)}{f_{Z;\hat{\Theta}}(z_j)}=\\ &\sum_{j=1}^{N}\frac{f_{Z;\hat{\Theta}}(z_j)}{f_{Z;\hat{\Theta}}(z_j)}=N\end{aligned} \tag{A8}$$

In order to be able to reason about Eq. (A8), we note that the density function Z can be obtained from $Z=X+Y$, and X and Y are independent of each other.

There are

$$f_{Z;\hat{\theta}}(z)=\int f_X(v)f_Y(z-v)\mathrm{d}v=$$
$$\sum_{i=1}^{K}\int_{\Omega_i}\hat{\theta}_i f_Y(z-v)\mathrm{d}v=$$
$$\sum_{i=1}^{K}\hat{\theta}_i\Pr(Y\in z-\Omega_i) \tag{A9}$$

Thus it is proved. □

References

[1] C. P. Papageorgiou, M. Oren, and T. Poggio, A general framework for object detection, in *Proc. 6th Int. Conf. Computer Vision*, Bombay, India, 1998, pp. 555–562.

[2] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, Aggregated residual transformations for deep neural networks, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, HI, USA, 2017, pp. 5987–5995.

[3] A. Rao, L. Xu, Y. Xiong, G. Xu, Q. Huang, B. Zhou, and D. Lin, A local-to-global approach to multi-modal movie scene segmentation, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Seattle, WA, USA, 2020, pp. 10143–10152.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv: 1301.3781, 2013.

[5] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, Visual categorization with bags of keypoints, presented at the ECCV: Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic, 2004.

[6] J. Y. H. Ng, F. Yang, and L. S. Davis, Exploiting local features from deep networks for image retrieval, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops* (*CVPRW*), Boston, MA, USA, 2015, pp. 53–61.

[7] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, NV, USA, 2016, pp. 5297–5307.

[8] J. Liu, W. C. Chang, Y. Wu, and Y. Yang, Deep learning for extreme multi-label text classification, in *Proc. 40th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Tokyo, Japan, 2017, pp. 115–124.

[9] J. Nam, E. L. Mencía, H. J. Kim, and J. Fürnkranz, Maximizing subset accuracy with recurrent neural networks in multi-label classification, in *Proc. 31st Int. Conf. Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 5419–5429.

[10] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, Joint embedding of words and labels for text classification, arXiv preprint arXiv: 1805.04174, 2018.

[11] W. Zhang, J. Yan, X. Wang, and H. Zha, Deep extreme multi-label learning, in *Proc. 2018 ACM on Int. Conf. Multimedia Retrieval*, Yokohama, Japan, 2018, pp. 100–107.

[12] L. Yao, C. Mao, and Y. Luo, Graph convolutional networks for text classification, *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 7370–7377, 2019.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You only look once: Unified, real-time object detection, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, NV, USA, 2016, pp. 779–788.

[14] J. H. Kim, N. Kim, Y. W. Park, and C. S. Won, Object detection and classification based on YOLO-V5 with improved maritime dataset, *J. Mar. Sci. Eng.*, vol. 10, no. 3, p. 377, 2022.

[15] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, MOT16: A benchmark for multi-object tracking, arXiv preprint arXiv: 1603.00831, 2016.

[16] T. Pedersen, S. Patwardhan, and J. Michelizzi, WordNet: Similarity: Measuring the relatedness of concepts, in *Proc. HLT-NAACL—Demonstrations '04*: *Demonstration Papers at HLT-NAACL 2004*, Boston, MA, USA, 2004, pp. 38–41.

[17] Q. Le and T. Mikolov, Distributed representations of sentences and documents, in *Proc. 31st Int. Conf. Machine Learning*, Beijing, China, 2014, pp. 1188–1196.

[18] K. W. Church, Word2Vec, *Nat. Lang. Eng.*, vol. 23, no. 1, pp. 155–162, 2017.

[19] S. Hiranandani, K. Kennedy, and C. W. Tseng, Compiling Fortran D for MIMD distributed-memory machines, *Commun. ACM*, vol. 35, no. 8, pp. 66–80, 1992.

[20] S. Goldwater, T. L. Griffiths, and M. Johnson, Contextual dependencies in unsupervised word segmentation, in *Proc. 21st Int. Conf. Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, 2006, pp. 673–680.

[21] F. Sun, J. Tang, H. Li, G. J. Qi, and T. S. Huang, Multi-label image categorization with sparse factor representation, *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1028–1037, 2014.

[22] W. Li, J. Han, and J. Pei, CMAR: Accurate and efficient classification based on multiple class-association rules, in *Proc. 2001 IEEE Int. Conf. Data Mining*, San Jose, CA, USA, 2001, pp. 369–376.

[23] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, PREDRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs, in *Proc. 31st Int. Conf. Neural Information Processing Systems* (*NIPS 2017*), Long Beach, CA, USA, 2017, pp. 879–888.

[24] H. Guo, K. Zheng, X. Fan, H. Yu, and S. Wang, Visual attention consistency under image transforms for multi-label image classification, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, CA, USA, 2019, pp. 729–739.

[25] R. Azad, M. Asadi-Aghbolaghi, M. Fathy, and S. Escalera, Bi-directional ConvLSTM U-net with densley connected convolutions, in *Proc. IEEE/CVF Int. Conf. Computer Vision Workshop* (*ICCVW*), Seoul, Republic of Korea, 2019, pp. 406–415.

[26] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, Learning representations by predicting bags of visual words, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Seattle, WA, USA, 2020, pp. 6926–6936.

[27] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, YouTube-VOS: A large-scale video object segmentation benchmark, arXiv preprint arXiv: 1809.03327, 2018.

[28] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, HI, USA, 2017, pp. 7464–7473.

[29] S. C. Yurtkulu, Y. H. Şahin, and G. Unal, Semantic segmentation with extended DeepLabv3 architecture, in *Proc. 27th Signal Processing and Communications Applications Conf.* (*SIU*), Sivas, Turkey, 2019, pp. 1–4.

[30] Z. H. Hoo, J. Candlish, and D. Teare, What is an ROC curve, *Emerg. Med. J.*, vol. 34, pp. 357–359, 2017.

[31] J. Xu, T. Mei, T. Yao, and Y. Rui, MSR-VTT: A large video description dataset for bridging video and language, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, NV, USA, 2016, pp. 5288–5296.

[32] X. Wang, J. Wu, J. Chen, L. Li, Y. F. Wang, and W. Y. Wang, VaTeX: A large-scale, high-quality multilingual dataset for video-and-language research, in *Proc. IEEE/CVF Int. Conf. Computer Vision* (*ICCV*), Seoul, Republic of Korea, 2019, pp. 4580–4590.

Bo Ma is studying at the Department of Information Technology and Software Engineering, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand. His research interest is on privacy-preserving federated machine learning methods, privacy issues of big data analytics and Internet of Things, and information theory of deep learning.

Jinsong Wu received the PhD degree from Queen’s University, Canada. He has been elected the vice-chair, Technical Activities, IEEE Environmental Engineering Initiative, a pan-IEEE effort under IEEE Technical Activities Board (TAB), since 2017. He was the founder (2011) and founding chair (2011–2017) of IEEE Technical Committee on Green Communications and Computing (TCGCC). He is also the co-founder (2014) and founding vice-chair (2015–present) of IEEE Technical Committee on Big Data (TCBD). His received both 2017 and 2019 IEEE System Journal Best Paper Awards. His co-authored paper won 2018 IEEE TCGCC Best Magazine Paper Award. He received IEEE Green Communications and Computing Technical Committee 2017 Excellent Services Award for Excellent Technical Leadership and Services in the Green Communications and Computing Community. He was the leading editor and co-author of the comprehensive book, entitled *Green Communications: Theoretical Fundamentals, Algorithms, and Applications*, published by CRC Press in September 2012. He has been IEEE senior member since 2011.

Wei Qi Yan is working at the School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand.