A Novel Tensor-information Bottleneck Method for Multi-input Single-output Applications

Liangfu Lu[a], Xiaohan Ren[a], Chenwei Cui[a], Zhiyuan Tan[*,b], Yulei Wu[c], Zhizhen Qin[d]

[a] *School of Mathematics, Tianjin University, Tianjin, 300350, China*
[b] *School of Computing, Merchiston Campus, Edinburgh Napier University, Edinburgh EH10 5DT, Scotland, U.K*
[c] *College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, U.K*
[d] *School of Computer Science, University of California, San Diego, USA*

Abstract

Ensuring timeliness and mobility for multimedia computing is a crucial task for wireless communication. Previous algorithms that utilize information channels, such as the information bottleneck method, have shown great performance and efficiency, which guarantees timeliness. However, such methods suit only in handling single variable tasks such as image processing, but are inapplicable to multivariable applications such as video processing. To address this critical shortcoming, we propose a novel tensor information channel which extends the current single-input single-output matrix information channel to a more practical multi-input single-output tensor information channel. In comparison with the classic information channel, our tensor information channel not only performs better in experiments, but also allows for a wider range of practical applications. We further build an innovative tensor-information bottleneck method upon the state-of-the-art information bottleneck method. Experiments on video shot boundary detection are conducted using benchmark data sets to demonstrate the effectiveness of our proposed approach compared with state-of-the-art methods. In specific, our approach yields a 6.2% increase compared with the information channel-based method, and when compared to other state-of-the-art methods, we achieve 0.1%-17.7% performance gains under different experimental configurations.

Keywords: Tensor information channel, Tensor-information bottleneck, Cluster, Partition.

1. Introduction

In mobile multimedia computing, solutions to many problems are influenced by the performance of clustering or partitioning methods [1, 2]. Therefore, ensuring the robustness of such methods is a key challenge in the fields of machine learning, image processing, and pattern recognition. In these fields, clustering or partitioning methods are usually required to define the “distance” or “similarity” among measured data sets, such as Pearson correlation and Euclidean distance [3, 4, 5, 6, 7, 8, 9]. One popular information theory approach of clustering is to let the clusters only capture relevant information among the data, where the relevance is explicitly determined by the various components of the data itself. Mutual information (MI), which is based on information theory, provides a general measure of dependencies among variables and serves as a key tool for clustering, feature extraction (FE), and dimensionality reduction in many areas [10, 11, 12, 13]. For example, Chen et al. [11] used minimal redundancy maximal relevance-partial mutual information clustering with least square regression to overcome the two main flaws in the structure and the weights of multi-layer feed-forward networks. Oveisi et al. [12] proposed an efficient tree-based method for FE in which at each step a new feature is created by selecting and linearly combining two features such that the MI between the new feature and the class is maximized. Bouzas et al. [13] proposed a novel algorithm for dimensionality reduction that uses as a criterion the mutual information (MI) between the transformed data and their corresponding class labels.

The main purpose of information theory is to deal with the communication or information channel between source (or input) and receiver (or output). It was initially introduced by Claude Shannon. In his paper, Shannon roughly classified communication systems into three main categories: discrete, continuous, and mixed types [14]. The information channel is generally applied to any two variables sharing information. This application is popular in many fields, such as image processing [15], computer graphics [16, 17], and visualization [18, 19, 20]. Information Bottleneck (IB) method for data compression was introduced by Tishby et al. in [21], where the key idea is to compress the observation while the output preserves most of the information of the relevant variables, i.e., the original source. Since then, various IB methods have been rapidly developed and utilized in many fields, such as neuroscience, image processing, and deep learning [22, 23, 24, 25]. The partitioning principle of it is usually divided into soft [21] and hard [26] partitions of original source. Kartik et al. [22] developed an approach based on IB that attempts to find functional relationships in a neuron population. New image segmentation algorithms based on the hard version of the information bottleneck theory are presented in [23]. The highly popular Deep Neural Networks (DNNs), a good overview of which can be found in

[27], are analyzed in [24], via the theoretical framework of the IB principle. These works mainly focus on the design of network coding scheme and quantizer using IB method. To the best of our knowledge, the input and output of the IB based methods mentioned above are always univariate, and seldom are multiple variables considered to be processed. Friedman et al. introduced a general principled framework for multivariate extensions of the IB method in [28], however, they only utilized Bayesian networks to specify the systems of clusters and what information each captures, which do not ensure a competent performance. Therefore, in this work, we propose the tensor information channel and tensor-information bottleneck method to solve the problems especially for these involving input with multiple variables.

The rest of the paper is organized as follows. In Section 2, we present some basic concepts of information theory, the structure of an information channel, and the agglomerative information bottleneck method. We construct a tensor information channel and propose the tensor-information bottleneck method in Section 3. The experimental results on video shot boundary detection are shown in Section 4, and conclusions of the paper are described in Section 5.

2. Related Work

In this section, we briefly review some basic concepts of information theory [29, 30], the structure of a two-dimensional information channel [29, 30], and the agglomerative information bottleneck method [26], which are shown in matrix-based perspectives.

2.1. Information Theory

Let X be a discrete random variable taking on values in a set $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$ and probability distribution $p(X) = \{p(x)\}$, where $p(x) = \mathtt{Pr}\{X = x\}$ and $x \in \mathcal{X}$. Likewise, let Y be a random variable and $y \in \mathcal{Y}$.

The *Shannon entropy* $H(X)$ of a random variable X is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x), \tag{1}$$

where the log is with base-2, and the entropy is expressed in bits. The convention that $0 \log 0 = 0$ should be noticed. This entropy is denoted as $H(p)$ as well, which measures the average uncertainty of a random variable X.

The *conditional entropy* $H(Y|X)$ is defined by

$$H(Y|X) = \sum_{x \in \mathcal{X}} p(x) H(Y|x), \tag{2}$$

where $H(Y|x) = -\sum_{y \in \mathcal{Y}} p(y|x) \log p(y|x)$ is the entropy of Y given x, and $p(y|x)$ is the conditional probability. $H(Y|X)$ measures the average uncertainty associated with Y if we know the outcome of X.

The *mutual information* (MI) between X and Y is defined by

$$\begin{aligned} I(X,Y) &= H(Y) - H(Y|X) \\ &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \end{aligned} \tag{3}$$

where $p(x,y) = \mathtt{Pr}\{X = x, Y = y\}$ is the joint probability, and $I(X,Y) = I(Y,X) \geq 0$. MI expresses the shared information between X and Y.

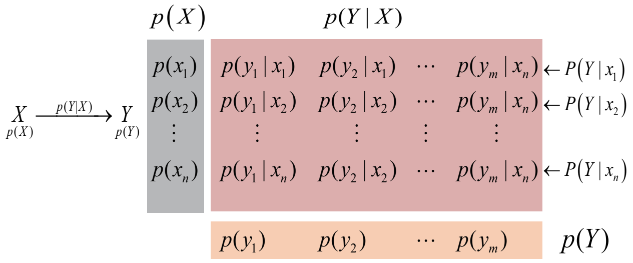
$p(X)$
$p(Y\mid X)$
$X \xrightarrow{p(Y|X)} Y$
$p(X)$
$p(Y)$
$p(x_1)$
$p(x_2)$
$\vdots$
$p(x_n)$
$p(y_1\mid x_1)$ $p(y_2\mid x_1)$ $\cdots$ $p(y_m\mid x_n)$ $\leftarrow P(Y\mid x_1)$
$p(y_1\mid x_2)$ $p(y_2\mid x_2)$ $\cdots$ $p(y_m\mid x_n)$ $\leftarrow P(Y\mid x_2)$
$\vdots$ $\vdots$ $\vdots$ $\vdots$
$p(y_1\mid x_n)$ $p(y_2\mid x_2)$ $\cdots$ $p(y_m\mid x_n)$ $\leftarrow P(Y\mid x_n)$
$p(y_1)$ $p(y_2)$ $\cdots$ $p(y_m)$
$p(Y)$

Figure 1: Main elements of an information channel.

2.2. Information Channel

Conditional entropy $H(Y|X)$ and mutual information $I(X,Y)$ can be considered as a communication channel or information channel $X \rightarrow Y$ whose output Y depends on its input X with some probability[29].

The diagram in Figure.1 shows the main elements of an information channel, which are

• Input and output variables, X and Y, with their marginal probability distributions denoted by $p(X)$ and $p(Y)$, respectively.

• Probability transition matrix $p(Y|X)$, which is composed of conditional probability $p(y|x)$. $p(Y)$ is determined by the input distribution $p(X)$: $p(y) = \sum_{x \in \mathcal{X}} p(x)p(y|x)$. Each row of $p(Y|X)$, denoted by $p(Y|x)$, is a probability distribution. All these elements are connected by Bayes' rule: $p(x,y) = p(x)p(y|x) = p(y)p(x|y)$.

2.3. Information Bottleneck Method

The *Jensen-Shannon divergence*(JS-divergence) [31] is used to measure the dissimilarity between two probability distributions, which is defined by

$$JS(\pi_1,\pi_2,\cdots,\pi_n; p_1,p_2,\cdots,p_n) = H(\sum_{i=1}^{n} \pi_i p_i) - \sum_{i=1}^{n} \pi_i H(p_i), \tag{4}$$

where p_i is a probability distribution defined over the same set with weight π_i, satisfying $\sum_{i=1}^{n} \pi_i = 1$.

The *information bottleneck method* is a technique that compresses the variable X into $\hat{X}$ with minimal loss of mutual information with respect to another variable Y. The compressed variable $\hat{X}$ can be considered as the result of merging two or more states of X, preserving as much information as possible with respect to the control variable Y [21].

Slonim and Tishby introduce the *agglomerative information bottleneck method* in [26], which assumes that a cluster $\hat{x}$ is defined by $\hat{x} = \{x_1, \ldots, x_l\}$, where for all $k \in \{1, \ldots, l\}$, $x_k \in \mathcal{X}$, and the probabilities $p(\hat{x})$ and $p(y|\hat{x})$ are defined by

$$p(\hat{x}) = \sum_{k=1}^{l} p(x_k), \tag{5}$$

$$p(y|\hat{x}) = \frac{1}{p(\hat{x})} \sum_{k=1}^{l} p(x_k) p(y|x_k), \forall y \in \mathcal{Y}. \tag{6}$$

The decrease in the mutual information $\delta I_{\hat{x}}$ from $I(X, Y)$ to $I(\hat{X}, Y)$, due to the merging of $x_1, \ldots, x_l$, is given by

$$\delta I_{\hat{x}} = p(\hat{x}) JS(\pi_1, \ldots, \pi_l; p_1, \ldots, p_l), \tag{7}$$

where $\pi_k = \frac{p(x_k)}{p(\hat{x})}$, and $p_k = p(Y|x_k)$. In the following sections, we will try to find the optimal clustering algorithm which aims to minimize $\delta I_{\hat{x}}$.

3. Tensor Information Channel and Tensor-Information Bottleneck Method

There is only one input variable in the information channel that we introduced in the previous section, where the output variable is directly determined by the input variable. However, in many cases, the output variables are determined by multiple input variables, and the traditional information channel is no longer applicable. Therefore, we construct a tensor information channel and propose the tensor-information bottleneck method in order to solve this problem.

3.1. Tensor Information Channel

Let X_i be a discrete random variable taking on values in a set $\mathcal{X}_i = \{x_{i1}, x_{i2}, \cdots, x_{id_i}\}$ with probability distribution $p(X_i) = \{p(x_i)\}$, where $i = 1, 2, \cdots, n-1$ and $x_i \in \mathcal{X}_i$. $X_1, X_2, \cdots, X_{n-1}$ are independent variables. Likewise, let Y be a discrete random variable and $y \in \mathcal{Y}$. The conditional probability $p(y|x_1 x_2 \cdots x_{n-1}) = \mathtt{Pr}\{Y = y|X_1 = x_1, X_2 = x_2, \cdots, X_{n-1} = x_{n-1}\}$.

Conditional entropy $H(Y|X_1 \cdots X_{n-1})$ and mutual information $I(X_1 X_2 \cdots X_{n-1}, Y)$ can be considered as a *Tensor information channel* $(X_1, \cdots, X_{n-1}) \to Y$, whose output Y depends probabilistically on its $n-1$ input variables. Probability transition tensor $\mathcal{T} = (p(y|x_1 x_2 \cdots x_{n-1})) \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_n}$ is a real n-th order $d_1 \times d_2 \times \cdots \times d_n$-dimensional tensor. When there is only one input variable in the information channel, the probability transition tensor degenerates into a matrix.

Without loss of generality, we consider a tensor information channel with two input variables, i.e. $(X, Y) \to Z$.

The conditional entropy $H(Z|XY)$ is defined by

$$H(Z|XY) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) H(Z|xy), \tag{8}$$

where $H(Z|xy) = -\sum_{z \in \mathcal{Z}} p(z|xy) \log p(z|xy)$ is the entropy of Z given x and y.

The mutual information between (X, Y) and Z is defined by

$$I(XY, Z) = H(Z) - H(Z|XY). \tag{9}$$

The diagram in Figure.2 shows the main elements of a third-order tensor information channel, which are:

- Input and output variables, (X, Y) and Z, with their respective probability distribution $p(X)$, $p(Y)$, and $p(Z)$.
- Probability transition tensor $\mathcal{T} = p(Z|XY) \in \mathbb{R}^{n \times m \times k}$, which is composed of conditional probability $p(z|xy)$. $p(Z)$ is determined by the input distribution $p(X)$ and $p(Y)$: $p(z) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(xy) p(z|xy)$.
- The horizontal and lateral slices of probability transition tensor $\mathcal{T}$, denoted by $T_{i::} = p(Z|x_i Y)$ and $T_{:j:} = p(Z|X y_j)$, respectively. Each row of $T_{i::}$ and $T_{:j:}$, denoted by $p(Z|x_i y)$ and $p(Z|x y_j)$, is a probability distribution. $p(Z x_i)$ is determined by the input $p(x_i)$: $p(z x_i) = \sum_{y \in \mathcal{Y}} p(x_i y) p(z|x_i y)$, and $p(Z y_j)$ is determined by the input $p(y_j)$: $p(z y_j) = \sum_{x \in \mathcal{X}} p(x y_j) p(z|x y_j)$.

3.2. Tensor-Information Bottleneck Method

The *tensor-information bottleneck method* is a technique that compresses the multiple variables $X_1, X_2, \cdots, X_{n-1}$ with minimal loss in terms of mutual information with respect to variable X_n. The compressed variable $\hat{X}_i$ can be considered as the result of merging two or more states of X_i, preserving as much information as possible about the control variable X_n.

We propose the *tensor agglomerative information bottleneck method* in this paper with the assumption that a cluster $\hat{x}_i$ is defined by $\hat{x}_i = \{x_{i1}, \ldots, x_{il}\}$ for all $k \in \{1, \ldots, l\}$ and $i \in \{1, \ldots, n-1\}$, $x_{ik} \in \mathcal{X}_i$. Without loss of generality, we set $n = 3$. Thus, the tensor information channel is $(X, Y) \to Z$. A cluster $\hat{x}$ is defined by $\hat{x} = \{x_1, \ldots, x_l\}$. Likewise, $\hat{y} = \{y_1, \ldots, y_l\}$. In the i-th horizontal slice of a third-order probability transition tensor, the probabilities $p(x_i \hat{y})$ and $p(z|x_i \hat{y})$ are defined by

$$p(x_i \hat{y}) = \sum_{k=1}^{l} p(x_i y_k), \tag{10}$$

$$p(z|x_i \hat{y}) = \frac{p(z x_i \hat{y})}{p(x_i \hat{y})} = \frac{1}{p(x_i \hat{y})} \sum_{k=1}^{l} p(x_i y_k z), \forall z \in Z. \tag{11}$$

The *tensor Jensen-Shannon divergence* (TJS-divergence) is used to measure the dissimilarity between the probability distributions of two slices, which is defined by

$$TJS(\omega_1, \ldots, \omega_n; JS_1, \ldots, JS_n) = \sum_{i=1}^{n} \omega_i JS_i, \tag{12}$$

where $JS_i = JS(\pi_{i1}, \pi_{i2}, \ldots, \pi_{il}; p_{i1}, p_{i2}, \ldots, p_{il})$ is the Jensen-Shannon divergence of the i-th slice with weight ω_i, fulfilling $\sum_{i=1}^{n} \omega_i = 1$.

The decrease in the mutual information of a third-order probability transition tensor $\delta I_{X\hat{y}}$ from $I(XY, Z)$ to $I(X\hat{Y}, Z)$ due to the merging of $y_1, \ldots, y_l$ is given by

$$\begin{aligned} \delta I_{X\hat{y}} &= p(\hat{y}) TJS(\omega_1, \ldots, \omega_n; JS_1, \ldots, JS_n) \\ &= p(\hat{y}) \sum_{i=1}^{n} \omega_i JS_i(\pi_{i1}, \ldots, \pi_{il}; p_{i1}, \ldots, p_{il}), \end{aligned} \tag{13}$$

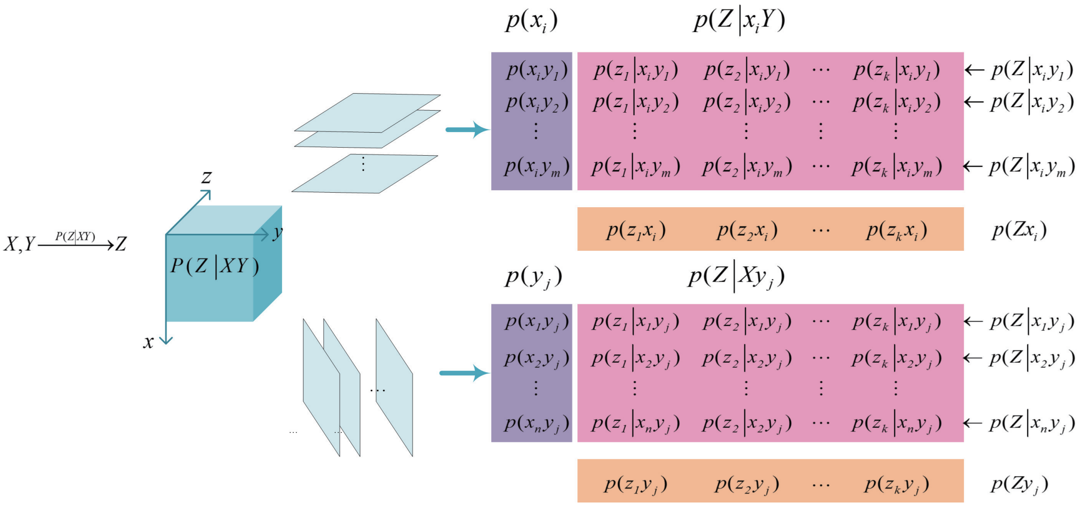
$p(x_i)$
$p(Z|x_iY)$
$p(x_iy_1)$ $p(z_1|x_iy_1)$ $p(z_2|x_iy_1)$ $\cdots$ $p(z_k|x_iy_1)$ $\leftarrow p(Z|x_iy_1)$
$p(x_iy_2)$ $p(z_1|x_iy_2)$ $p(z_2|x_iy_2)$ $\cdots$ $p(z_k|x_iy_2)$ $\leftarrow p(Z|x_iy_2)$
$p(x_iy_m)$ $p(z_1|x_iy_m)$ $p(z_2|x_iy_m)$ $\cdots$ $p(z_k|x_iy_m)$ $\leftarrow p(Z|x_iy_m)$
$p(z_1x_i)$ $p(z_2x_i)$ $\cdots$ $p(z_kx_i)$ $p(Zx_i)$
z
y
x
$X,Y \xrightarrow{P(Z|XY)} Z$
$P(Z|XY)$
$p(y_j)$
$p(Z|Xy_j)$
$p(x_1y_j)$ $p(z_1|x_1y_j)$ $p(z_2|x_1y_j)$ $\cdots$ $p(z_k|x_1y_j)$ $\leftarrow p(Z|x_1y_j)$
$p(x_2y_j)$ $p(z_1|x_2y_j)$ $p(z_2|x_2y_j)$ $\cdots$ $p(z_k|x_2y_j)$ $\leftarrow p(Z|x_2y_j)$
$p(x_ny_j)$ $p(z_1|x_ny_j)$ $p(z_2|x_2y_j)$ $\cdots$ $p(z_k|x_ny_j)$ $\leftarrow p(Z|x_ny_j)$
$p(z_1y_j)$ $p(z_2y_j)$ $\cdots$ $p(z_ky_j)$ $p(Zy_j)$

Figure 2: Main elements of a three order tensor information channel, which are connected by Bayes' rule.

where $\omega_i = \frac{p(x_i\hat{y})}{p(\hat{y})}$, $\pi_{ik} = \frac{p(x_iy_k)}{p(x_i\hat{y})}$, and $p_{ik} = p(Z|x_iy_k)$, so the equation (13) can be denoted by

$$\delta I_{X\hat{y}} = \sum_{i=1}^{n} p(x_i\hat{y}) JS_i(\pi_{i1}, \dots, \pi_{il}; p_{i1}, \dots, p_{il}). \tag{14}$$

Likewise, the decrease in the mutual information of a third-order probability transition tensor $\delta I_{\hat{x}Y}$ from $I(XY, Z)$ to $I(\hat{X}Y, Z)$ due to the merging of $x_1, \dots, x_l$ is given by

$$\delta I_{\hat{x}Y} = \sum_{j=1}^{m} p(\hat{x}y_j) JS_j(\pi_{1j}, \dots, \pi_{lj}; p_{1j}, \dots, p_{lj}), \tag{15}$$

where $p(\hat{x}y_j) = \sum_{k=1}^{l} p(x_ky_j)$, $\pi_{kj} = \frac{p(x_ky_j)}{p(\hat{x}y_j)}$, and $p_{kj} = p(Z|x_ky_j)$. To sum up, depending on the compressed variable, the optimal clustering is the one that tries to minimize the decrease in the mutual information of the variable, i.e. $\min\{\delta I_{\hat{x}Y}\}$ or $\min\{\delta I_{X\hat{y}}\}$.

In order to introduce tensor-information bottleneck method to practical applications, we extend the split-and-merge algorithm constructed from the conventional information channel to the tensor information channel. The split-and-merge algorithm, proposed by Anton et al. [23], is divided into two phases. For the first phase, a top-down strategy is applied for partitioning. In the second phase, a bottom-up strategy is used to merge similar parts. The splitting procedure is described in Algorithm 1. The target of the algorithm is to divide X into m clusters, where m is the preset number of clusters, and $m > 1$. m is a parameter which determines the maximum number of the clusters. In our experiments, we pick a sufficiently large value for m, that is 2^{15}. Please note that choosing an m that is larger than the total frame number of a video does not lead to a trivial result (i.e. a cut for each frame). Instead, most of the resulting cuts will be at the same place, while the majority of the frames will not be considered a cut. The initial number of cluster is 1. According to the probability distribution and conditional distribution of the three variables x, y, and z that affect the classification, a partition will be made where $\max(\delta I_{\hat{x}Y})$ is achieved. The process continues until the number of clusters is no smaller than m. The merging procedure is described in Algorithm 2. The purpose of this algorithm is to merge the over segmented parts and finally divide X into n clusters. Algorithm 2 has a threshold ε, and the selection of this value is thoroughly explained in Section 4. To improve the robustness of our method, we use normalized mutual information gain as the stopping criterion, which is shown as the following:

$$N\delta I = \frac{\delta I - \min \delta I}{\max \delta I - \min \delta I}. \tag{16}$$

Algorithm 1 Top-down bottleneck algorithm

```
Input:
  Initial value: $X$
  Number of clusters: $m$
Output: A partition of $X$: $\hat{X}$
  $\hat{X} \leftarrow \{X\}$
  while $|\hat{X}| < m$ do
    $\hat{X}' \leftarrow \{\}$
    for $T$ in $\hat{X}$ do
      $k \leftarrow \mathrm{argmax}(\{\delta I_{\hat{x}Y}(t)\text{for } t \text{ in } T\})$ (see (15))
      $T_1 \leftarrow T[:k]$
      $T_2 \leftarrow T[k:]$
      $\hat{X}' \leftarrow \hat{X}' \cup \{T_1, T_2\}$
    end for
    $\hat{X} \leftarrow \hat{X}'$
  end while
  return $\hat{X}$
```

4. Experiments

In this section, we testify the proposed split-and-merge algorithm which is based on the tensor information channel $(F, R) \to B$, where the random variables F, R, and B each represents the set of frames of a video, the set of regions of images,

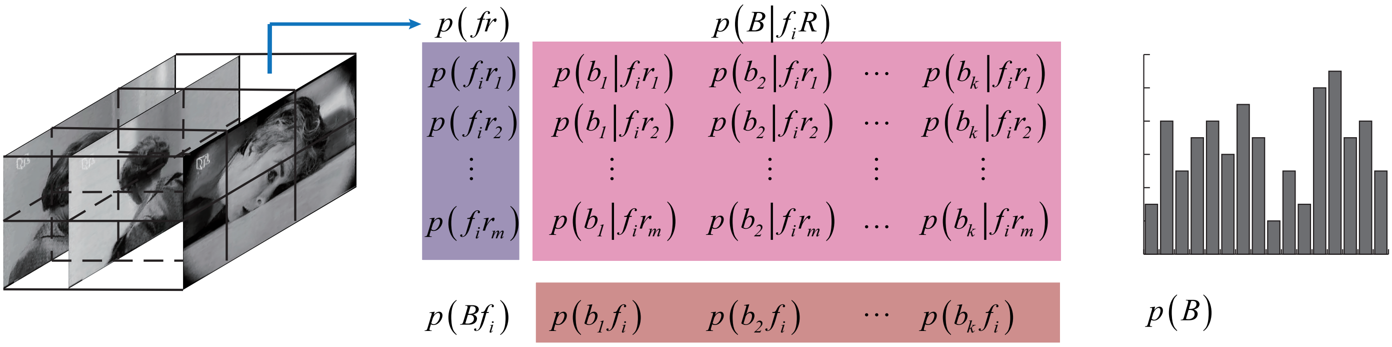
$p(fr)$
$p(B|f_iR)$
$p(f_ir_1)$
$p(f_ir_2)$
$\vdots$
$p(f_ir_m)$
$p(b_1|f_ir_1)$ $p(b_2|f_ir_1)$ $\cdots$ $p(b_k|f_ir_1)$
$p(b_1|f_ir_2)$ $p(b_2|f_ir_2)$ $\cdots$ $p(b_k|f_ir_2)$
$p(b_1|f_ir_m)$ $p(b_2|f_ir_m)$ $\cdots$ $p(b_k|f_ir_m)$
$p(Bf_i)$
$p(b_1f_i)$ $p(b_2f_i)$ $\cdots$ $p(b_kf_i)$
$p(B)$

Figure 3: Main elements of the tensor information channel belonging to the split-and-merge algorithm.

Algorithm 2 Bottom-up bottleneck algorithm

```
Input:
  An excessive partition of $X$: $\hat{X}$
  Threshold: $\varepsilon$
Output: An optimized partition of $X$: $\hat{X}$
  $D \leftarrow \{\delta I_{xY}(\hat{X}[i] \cup \hat{X}[i+1])$ for $i$ in range$(|\hat{X}|-1)\}$
  $D \leftarrow$ normalize$(D)$ (see (16))
  $\hat{X}' \leftarrow \{\hat{X}[0]\}$
  for $i$ in range$(|D|)$ do
    if $D[i] > \varepsilon$ then
      $\hat{X}' \leftarrow \hat{X}' \cup \{\hat{X}[i+1]\}$
    else
      $\hat{X}'[-1] \leftarrow \hat{X}'[-1] \cup \hat{X}'[i+1]$
    end if
  end for
  $\hat{X} \leftarrow \hat{X}'$
  return $\hat{X}$
```

and the set of intensity bins. This tensor information channel is defined by a conditional probability tensor $\mathcal{T} = (p(b|fr))$, which expresses how the pixels corresponding to each block of the video are distributed into the histogram bins.

Figure.3 shows the elements of the tensor information channel belonging to the split-and-merge algorithm. The definitions of these elements are:

• The conditional probability tensor $p(B|FR)$, which represents the transition probabilities from each block of the video to the bins of the histogram, is defined by $p(b|fr) = \frac{n(fr,b)}{n(fr)}$, where $n(fr)$ is the number of pixels of blocks formed by the intersection of frame f and region r, and $n(fr,b)$ is the number of pixels of block fr corresponding to bin b. Conditional probabilities fulfill $\sum_{b\in\mathcal{B}} p(b|fr) = 1, \forall f \in \mathcal{F}, r \in \mathcal{R}$.

• The input distribution $p(FR)$, which represents the probability of selecting each video block, is defined by $p(fr) = \frac{n(fr)}{N}$, where N is the number of pixels of the original video.

• The output distribution $p(B)$, which represents the normalized frequency of each bin b, is defined by $p(b) = \sum_{f\in\mathcal{F}} \sum_{r\in\mathcal{R}} p(fr)p(b|fr) = \frac{n(b)}{N}$, where $n(b)$ is the number of pixels corresponding to bin b.

For the first phase, a top-down strategy is applied to partition a video into quasi-homogeneous frames using binary space partition (BSP). In the second phase, a bottom-up strategy is used to merge the frames whose histograms are similar. The partitioning process is represented over the tensor information channel $(\hat{F}, R) \rightarrow B$, where $\hat{F}$ denotes the partitioned variable F according to the maximum MI gain for each partitioning step. Note that this tensor information channel varies at each partition step because the number of regions is increased. Consequently, the marginal probabilities $p(\hat{F})$ and the conditional probabilities $p(B|\hat{F}R)$ also change. For the BSP strategy, the gain of MI due to the partitioning of a block $\hat{f}r$ in two neighbor blocks f_1r and f_2r, such that

$$p(\hat{f}r) = p(f_1r) + p(f_2r) \tag{17}$$

and

$$p(b|\hat{f}r) = \frac{p(f_1r)p(b|f_1r) + p(f_2r)p(b|f_2r)}{p(\hat{f}r)} \tag{18}$$

is given by

$$\delta I_{\hat{f}R} = \sum_{r\in\mathcal{R}} p(\hat{f}r)JS(\pi_{1r}, \pi_{2r}; p(B|f_1r), p(B|f_2r)), \tag{19}$$

where $\pi_{1r} = \frac{p(f_1r)}{p(\hat{f}r)}$, $\pi_{2r} = \frac{p(f_2r)}{p(\hat{f}r)}$. The JS-divergence between two blocks can be interpreted as a measure of dissimilarity between them with respect to the intensity values. In the splitting process, the partitioning of each step is determined by the maximum MI gain $\delta I_{\hat{f}R}$ of each sub partitions.

The merging process is represented over the tensor information channel $(F, R) \rightarrow B$. From the tensor agglomerative information bottleneck method, we know that any clustering over F or R will not increase $I(FR, B)$. Analogous to the MI gain (19) obtained in the splitting phase, the loss of MI due to the clustering $\hat{f}$ of two neighbor frames f_1 and f_2 is also given by

$$\delta I_{\hat{f}R} = \sum_{r\in\mathcal{R}} p(\hat{f}r)JS(\pi_{1r}, \pi_{2r}; p(B|f_1r), p(B|f_2r)), \tag{20}$$

where $\pi_{1r} = \frac{p(f_1r)}{p(\hat{f}r)}$, $\pi_{2r} = \frac{p(f_2r)}{p(\hat{f}r)}$. In the merging phase, if two frames are very similar (i.e., the JS-divergence between them is small), the channel could be simplified by merging these two frames, without a significant loss of information. The merging of each step is determined by the minimum MI gain $\delta I_{\hat{f}R}$. Due to the inevitable redundant partitions that the BSP strategy may introduce, the merging process is indispensable since it effectively reduces such redundancy.

Theoretically, the optimal value for ε varies with the number of frames. However, it is possible to find a common value for ε which suits a variety of videos given that the length of which do not diverse a lot, ensuring the practicability of our method.

4.1. Data Set

Accurate shot boundary detection plays an important role in applications such as video extraction, video retrieval, and video summarization. Reliable shot boundary detection is a fundamental step in video segmentation applications because video shots are the elementary building blocks of a complete video sequence. We use the Video Segmentation Dataset[1] in our experiments, which contains 10 different videos. The rationale behind this selection is three-fold. First, this data set is representative of videos with all kinds of art forms, ranging from cartoon, live action, long take, and classic black-and-white movies. It examines the ability to generalize, thus ensures the usefulness and robustness of our work. Second, there already exists a comprehensive range of methodologies using this data set as a benchmark, which ensures the validity of our work. Third, it fits the other qualities that makes a good benchmark: the task is clear, it is open and accessible, and the metrics are clear. Table.1 shows the characteristics of the video data in detail, which includes significant camera parameter changes, abrupt camera movements, and so on. These 10 video sequences contain different video genres, including cartoon, action, horror, etc. The frame sizes are 327 × 288 pixels. It is challenging to find a general solution to the shot boundary detection problems since the videos vary significantly, and the correct partition is not directly related to the length of videos. Therefore, we look forward to finding a video shot boundary detection method for the various types of videos mentioned above.

The tensor information channel we proposed can be utilized in shot boundary detection. Mutual information in tensor information channel is mainly used to measure the correlation between adjacent video frames. As stated above, our proposed method is tuned to handle videos whose frame number does not vary greatly. Therefore, we ignore the label H containing 5133 frames and label F containing 236 frames since they are considered outliers in length, and, thus, the results upon which will not be representative. It should be noted that we turned both colored and black-and-white frames into grayscale images, which may result in some information loss.

To compare the different experimental results, we use Precision, Recall, and F1 score as criteria, which are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{21}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{22}$$

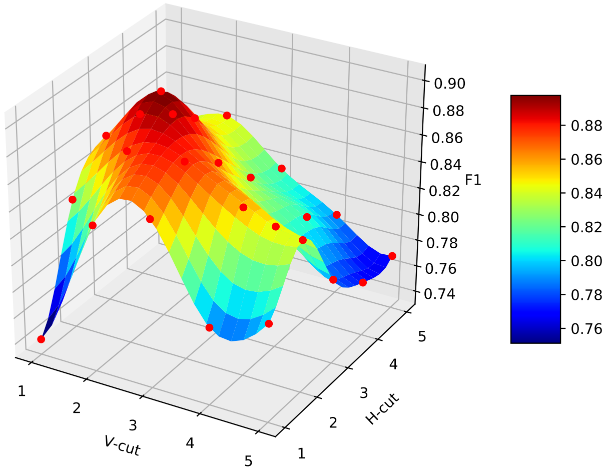

Figure 4: The value of F1 score varies with the number of regions of video images for the split-and-merge algorithm. H-cut and V-cut indicate the horizontal division and vertical division respectively.

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{23}$$

where TP, FP, FN each represents true positive, false positive, and false negative. For each video, there is a label which specifies the exact frames that should be considered a cut. For a predicted cut to be a true positive, it must strictly be one of the cuts in the label, otherwise, it is considered a false positive.

4.2. Results Analysis

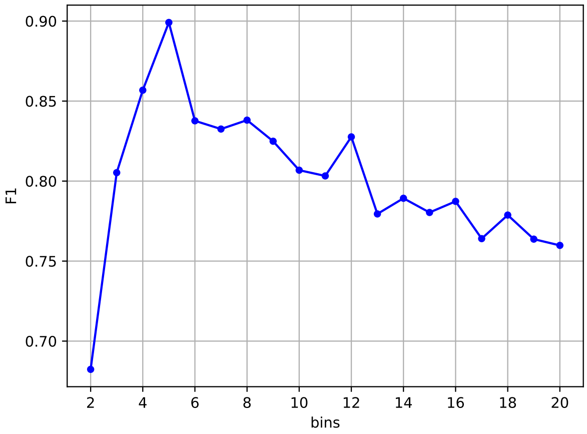

Figure 5: The value of F1 score varies with the distribution of color histograms of the split-and-merge algorithm.

In experiments, we use a data set including 8 videos that represent a variety of different video genres and compare the results of the proposed method against a feature tracking method [32], the information channel method [23], a pixel-based method with relative localization information, and a histogram-based method [33]. The feature tracking method is based on

[1] http://www.site.uottawa.ca/~laganier/videoseg/

Table 1: The Characteristics of video data.

Label	Frames	Cuts	Characteristic of video data	Genre
A	650	7	Substantial object motion.	Cartoon
B	959	8	Simulate low lighting conditions with a blue filter.	Action
C	1619	54	Black and white movie. Many close proximity cuts.	Horror
D	2632	34	High quality digitisation of a television show.	Drama
E	536	30	Low quality digitisation of a television show.	Science-Fiction
F	236	0	Commercial, no cuts, quick motion, many production effects.	Commercial
G	500	18	Commercial sequence from the MOCA Project.	Commercial
H	5133	38	Video abstract from the MOCA Project.	Comedy/Drama
I	479	4	News Sequence from the MOCA Project.	News/Documentary
J	873	87	Many computer generated features, many close proximity cuts.	Trailer/Science-Fiction/Action

Table 2: The results of tensor information channel, pixel-based method, and histogram-based method.

	Tensor information channel			**Pixel Based method with localization**			**Histogram MethodCut Det (MOCA)**		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
A	1	1	**1**	1	1	**1**	1	1	**1**
B	1	1	**1**	0.825	0.825	0.825	1	0.375	0.545
C	0.879	0.944	**0.910**	0.764	0.778	0.771	0.936	0.536	0.682
D	1	1	**1**	1	1	**1**	1	0.941	0.969
E	1	0.933	**0.965**	0.867	0.867	0.867	0.955	0.700	0.808
G	0.857	1	**0.923**	0.708	0.994	0.809	1	0.667	0.800
I	1	1	**1**	1	1	**1**	1	0.500	0.667
J	0.511	0.805	**0.625**	0.623	0.540	0.591	0.85	0.395	0.540
AVG	0.906	0.960	**0.928**	0.848	0.876	0.858	0.968	0.639	0.751
STD.DEV	0.159	0.064	**0.119**	0.136	0.152	0.134	0.050	0.219	0.164

stable feature tracking for inter frame differencing. It uses feature tracking as a metric for dissimilarity. The pixel-based method with localization evaluates the similarity of adjacent frames using metrics that are used to quantify the difference between the two adjacent frames. The histogram-based method adopts the same principle as the pixel-based one, however, it utilizes histogram values of the pixel data rather than the pixel values themselves. In the partitioning process, the input parameters include the marginal probability distributions $p(r)$, $p(b)$, the conditional probability distribution $p(b|r)$ (with initialization of $p(f) = 1$), and the number of clusters m. The calculation of marginal probability and conditional probability distribution depends on the partition of the image and the color histogram distribution of the video. We have conducted an ablation study to explore the impact of each hyperparameter. We divide the video pictures into $\{1\times1, 1\times2, \cdots, 1\times5, 2\times1, 2\times2, \cdots, 5\times5\}$, i.e. $1 \leqslant |R| \leqslant 25$, and the color histogram into $\{2, 3\cdots, 20\}$, i.e. $2 \leqslant |B| \leqslant 20$. Note that when pictures of the video are divided into 1×1, the tensor information channel degenerates to a conventional information channel. In the merging process, the input parameters $p(f)$, $p(fr)$, and $p(b|fr)$ are determined by the result of the partitioning process, while $p(b)$ stays unchanged. Figure.4 demonstrates the change in F1 score when given different video picture divisions and other hyperparameter stay fixed. Figure.5 shows the impact of the color histogram distribution on the F1 score. To ensure a fair comparison, the following evaluation of our proposed method will be conducted in three aspects: (1) local optimization, that is finding the suitable hyperparameters for each video. (2) global optimization, which means finding the suitable hyperparameters for all videos. (3) theoretical improvement, where we will compare our tensor information channel based algorithm with the conventional information channel based algorithm.

For local optimization, we find a suitable set of hyperparameters for each video. To ensure fairness, the opponent methods also follow this kind of optimization. We compare the proposed method against a pixel-based method and a histogram-based method as the proposed method utilizes only the pixels and histograms of video. Table.2 shows the results of tensor information channel, pixel-based, and histogram-based methods. On the average, the proposed method significantly outperforms both the pixel-based and histogram-based methods. The average F1 score yield by our method greatly surpasses the other approaches by 7.0% and 17.7% respectively. The lower standard deviation shows that the proposed method is more stable among different video genres. In all cases, the proposed method yields the best achievable F1 score. It is observed that for all listed approaches there is a notable performance loss on video J. It is most likely due to its abnormal cuts and frame rate which is significantly higher than other videos.

For global optimization, we find the most suitable set of hyperparameters for all of the 8 videos. Specifically, the hyperparameters we choose are: $|R| = 6 = (2 \times 3)$, $|B| = 5$, $m = 2^{15}$, and $\varepsilon = 0.10$. Table.3 shows the comparison of results between

Table 3: The results of tensor information channel and the feature tracking method.

	Feature tracking method			Tensor information channel		
	Precision	Recall	F1	Precision	Recall	F1
A	1	1	**1**	0.875	1	0.933
B	1	1	**1**	1	1	**1**
C	0.595	0.87	0.707	0.9	0.833	**0.865**
D	1	1	**1**	0.971	1	0.985
E	0.938	1	**0.968**	0.857	1	0.923
G	0.810	0.944	0.872	0.941	0.889	**0.914**
I	1	1	**1**	1	1	**1**
J	0.497	0.897	**0.637**	0.547	0.598	0.571
AVG	0.855	0.964	0.898	0.886	0.915	**0.899**
STD.DEV	0.190	0.050	0.138	0.138	0.134	**0.131**

Table 4: The results of tensor information channel and information channel.

	Tensor information channel			Information channel		
	Precision	Recall	F1	Precision	Recall	F1
A	0.875	1	**0.933**	0.779	1	0.875
B	1	1	**1**	1	1	**1**
C	0.9	0.833	**0.865**	0.75	0.889	0.814
D	0.971	1	**0.985**	0.917	0.971	0.942
E	0.857	1	**0.923**	0.833	0.833	0.833
G	0.941	0.889	**0.914**	0.929	0.722	0.813
I	1	1	**1**	1	1	**1**
J	0.547	0.598	**0.571**	0.583	0.322	0.415
AVG	0.886	0.915	**0.899**	0.849	0.842	0.837
STD.DEV	0.138	0.134	**0.131**	0.133	0.218	0.175

tensor information channel and the feature tracking method. On average, the proposed tensor information channel method outperforms the feature tracking method both in F1 score and standard deviation, showing the great robustness and stability. It is also worth noting that although adopted the less advantageous tuning strategy, our method still, by a large extent, outperforms the other two methods listed in Table.2, yielding an increment of 4.1% and 14.8% respectively compared with the pixel-based and histogram-based methods.

For theoretical improvements, we compare the proposed method with the conventional information channel method. The hyperparameters are set to be the same as the global optimized ones. In Table.4, the proposed method significantly outperforms the information channel method. The F1 score yielded by our method significantly surpasses the conventional information channel approach by 6.2%. Also, the smaller standard deviation ($0.131 < 0.175$) shows that the tensor information channel is more stable than the information channel method. In fact, the existing methodology stated in [21] lacks dimensionality. The proposed information channel and its split-and-merge algorithm only accepts matrices as input, which is best suited for image processing. However, it would be impossible for it to process three-dimensional video inputs. For it to handle video inputs, each frame must be converted to a one dimensional histogram data recording the distribution of pixel values. This process introduces a great information loss, which is in accordance with the experiments. Our generalized tensor information channel, however, can handle third-order tensors with ease. In all cases, the proposed method yields the best F1 score. The validity of the tensor information channel of multi-input and single-output is thus verified.

5. Conclusions

In this paper, we address the problem of inefficiency in the conventional information channel method, focusing on the multivariable cases. Specifically, we construct a tensor information channel and propose a novel tensor-information bottleneck method. This entails defining tensor information for the frames of a video, the regions of images, and histogram bins for a split-and-merge based algorithm. An additional advantage of this method is that we do not need to assume any priori information about the video input. We compare our proposed algorithm with other state-of-the-art methods on 8 benchmark videos from different genres. Numerical experiments demonstrate that the tensor information channel achieves improved and stable results for video shot boundary detection. However, there are a number of shortcomings in our proposed method, which will be addressed through ongoing future researches focused on higher-order tensor information channels, and evaluations on more challenging real-world applications (e.g. [34, 35, 36, 37]).

References

[1] A. Čolaković, M. Hadžialić, “Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues”, *Computer Networks*, vol. 144, pp. 17-39, 2018.

[2] H. Gao, L. Kuang, Y. Yin, B. Guo, K. Dou, “Mining consuming Behaviors with Temporal Evolution for Personalized Recommendation in Mobile Marketing Apps”, *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1233-1248, 2020.

[3] L. Sun, R.N. Liu, J.C. Xu, S.G. Zhang, “An Adaptive Density Peaks Clustering Method With Fisher Linear Discriminant”, *IEEE Access*, vol. 7, pp. 72936-72955, 2019.

[4] L.W. Wang, Y. Zhang, J.F. Feng, “On the Euclidean distance of images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1334-1339, 2005.

[5] U. Maulik, S. Bandyopadhyay, “Genetic algorithm-based clustering technique”, *Pattern Recognition*, vol. 33, no. 9, pp. 1455-1465, 2000.

[6] Z. He, S. Xie, R. Zdunek, et al., “Symmetric Nonnegative Matrix Factorization: Algorithms and Applications to Probabilistic Clustering”, *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2117-2131, 2011.

[7] X. Yang, K. Huang, R. Zhang, A. Hussain, “Learning Latent Features with Infinite Non-negative Binary Matrix Tri-factorization”, *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 5, pp. 450-463, 2018.

[8] E. Ahmed, I. Yaqoob, I.A.T. Hashem, I. Khan, A.I.A Ahmed, M.Imran, A.V. Vasilakos, “The role of big data analytics in Internet of Things”, *Computer Networks*, vol. 129, pp. 459-471, 2017.

[9] H. Gao, W. Huang, Y. Duan, “The Cloud-edge-based Dynamic Reconfiguration to Service Workflow for Mobile Ecommerce Environments: A QoS Prediction Perspective”, *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1-23, 2021.

[10] A. Martinez-Uso, F. Pla, J.M. Sotoca, P. Garcia-Sevilla, “Clustering-Based Hyperspectral Band Selection Using Information Measures”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4158-4171, 2008.

[11] C. Chen, X. Yan, “Optimization of a Multilayer Neural Network by Using Minimal Redundancy Maximal Relevance-Partial Mutual Information Clustering With Least Square Regression”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1177-1187, 2015.

[12] F. Oveisi, S. Oveisi, A. Erfanian, I. Patras, “Tree-Structured Feature Extraction Using Mutual Information”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 127-137, 2012.

[13] D. Bouzas, N. Arvanitopoulos, A. Tefas, “Graph Embedded Nonparametric Mutual Information for Supervised Dimensionality Reduction”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 951-963, 2015.

[14] C. E. Shannon, "A Mathematical Theory of Communication", *Bell Labs Technical Journal*, vol. 27, no. 4, pp. 379-423, 1948.

[15] M. Feixas, A. Bardera, J. Rigau, Q. Xu, M. Sbert, "Information theory tools for image processing", *Synthesis Lectures on Computer Graphics and Animation*, vol. 6, no. 1, pp. 1-164, 2014.

[16] F. Escolano, P. Suau, B. Bonev, "Information Theory in Computer Vision and Pattern Recognition", *Springer Publishing Company*, Incorporated, 1st edition, 2009.

[17] M. Sbert, M. Feixas, J. Rigau, et al., "Information Theory Tools for Computer Graphics", *Morgan & Claypool Publishers*, 2009.

[18] C. Wang, H.W. Shen, "Information theory in scientific visualization", *Entropy*, vol. 13, no. 1, pp. 254-273, 2011.

[19] Q. Hao, M. Sbert, L. Ma, "Gaze Information Channel in Cognitive Comprehension of Poster Reading", [Online], Available: `https://doi.org/10.3390/e21050444`, 2019.

[20] M. Chen, H. Jänicke, "An Information-theoretic Framework for Visualization", *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1206-1215, 2011.

[21] N. Tishby, F.C. Pereira, W. Bialek, "The information bottleneck method", [Online], Available: `arXiv:physics/0004057`, 2000.

[22] S. Buddha, K. So, J. Carmena, M. Gastpar, "Function identification in neuron populations via information bottleneck", *Entropy*, vol. 15, no. 5, pp. 1587-1608, 2013.

[23] B. Anton, R. Jaume, B. Imma, et al., "Image segmentation using information bottleneck method", *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1601-1612, 2009.

[24] N. Tishby, N. Zaslavsky, "Deep Learning and the Information Bottleneck Principle", [Online], Available: `http://de.arxiv.org/pdf/1503.02406`, 2015.

[25] S. Shadroo, A.M. Rahmani, A. Rezaee, "The two-phase scheduling based on deep learning in the Internet of Things", *Computer Networks*, 2020.

[26] N. Slonim, N. Tishby, "Agglomerative information bottleneck", *In Proceedings of NIPS-12 (Neural Information Processing Systems)*. MIT Press, pp. 617-623, 2000.

[27] M. Mahmud, M.S. Kaiser, A. Hussain, et al., "Applications of Deep Learning and Reinforcement Learning to Biological Data", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2053-2079, 2018.

[28] N. Friedman, O. Mosenzon, N. Slonim, et al., "Multivariate Information Bottleneck", [Online], Available: `https://arxiv.org/abs/1301.2270`, 2013.

[29] T.M. Cover, J.A. Thomas, "Elements of Information Theory", 2003.

[30] R.W. Yeung. "Information Theory and Network Coding", *Springer*, 2008.

[31] J. Burbea, C.R. Rao, "On the convexity of some divergence measures based on entropy functions", *IEEE Transactions on Information Theory*, vol. 28, no. 3, pp. 489-495, 1982.

[32] A. Whitehead, P. Bose, R. ganiere, "Feature Based Cut Detection with Automatic Threshold Selection", *International Conference on Image and Video Retrieval*, Springer, Berlin, Heidelberg, pp. 410-418, 2004.

[33] S. Pfeiffer, R. Lienhart, G. Khne, W. Effelsberg, "The MoCA Project - Movie Content Analysis Research at the University of Mannheim", *Informatik '98*, pp. 329-338, 1998.

[34] L. Zhang, Z. Liu, S. Zhang, X. Yang, H. Qiao, K. Huang, A. Hussain, "Cross-modality interactive attention network for multispectral pedestrian detection", *(Elsevier) Information Fusion*, vol. 50, pp. 20-29, 2019.

[35] F. Xiong, B. Sun, X. Yang, H. Qiao, K. Huang, A. Hussain, Z. Liu, "Guided Policy Search for Sequential Multitask Learning", *IEEE Transactions on Systems Man & Cybernetics Systems*, vol. 49, no. 1, pp. 216-226, 2019.

[36] X. Yang, S. Zhou, M. Cao, "An Approach to Alleviate the Sparsity Problem of Hybrid Collaborative Filtering Based Recommendations: The Product-Attribute Perspective from User Reviews", *Mobile Networks and Applications*, vol. 25, no. 2, pp. 376-390, 2020.

[37] H. Gao, C. Liu, Y. Li, X. Yang, "V2VR: Reliable Hybrid-Network-Oriented V2V Data Transmission and Routing Considering RSUs and Connectivity Probability", *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-14, 2020.