This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

Effective and Efficient Content Redundancy Detection of Web Videos

Yixin Chen, Dongsheng Li, Yu Hua, Wenbo He

***Abstract*—Currently, an unprecedentedly vast amount of videos are hosted on the Internet and shared by users across the world. Within these videos, a considerable portion is duplicate or near-duplicate. Consequently, building an effective yet efficient content-based redundancy detection system is of importance, as this research would be beneficial to a variety of applications. Despite the progress in this field, designing a practical detection system for web videos continues to be difficult, because of the contradictions between the accuracy and speed requirements.**

In this paper, we propose a novel near-duplicate video detection system, CompoundEyes, whose design philosophy deviates from the conventional feature-centered paradigm. Instead, the focus of our system has been shifted from the design of an advanced feature representation to the design of system architecture. This design methodology not only ensures a decent detection accuracy by the collaboration of the classifiers but also substantially accelerates the detection speed due to the low dimensionality of the feature representations and the exploitation of the parallelism among the components. Experiments have been conducted to demonstrate that the CompoundEyes is both accurate and fast.

***Index Terms*—Near-duplicate Detection, Web Videos, Instance-based Learning, Multiple Instance Learning.**

I. Introduction

The rapid progress in multimedia technologies and online video hosting services (e.g., YouTube) prompts the expansion of web videos. Due to the astronomical volume, videos have consumed a significant amount of Internet resources. In 2014, 78% of all U.S. Internet traffic was web videos; this share increases to 84% in 2018 [1].

Meanwhile, duplicated video content is pervasive in the ever-increasing web videos. In a web video dataset, around 27% of videos are duplicate or near-duplicate [2]. Accurate and fast detection of the redundancy in web videos is of importance. For instance, redundancy-aware content distribution networks are amenable to the optimization of bandwidth and storage provision. Moreover, pirate videos and false/polluted tags [3] can be identified by comparing the visual content of videos.

Due to the semantic gap, detecting NDVs (Near-Duplicate Videos) based on the associated keywords, tags, or description of videos is fast but less accurate. In contrast, content-based NDVD (Near-Duplicate Video Detection) systems [2], [4], [5] are more accurate but suffer from efficiency issues, because these systems apply sophisticated and high-dimensional feature representations for satisfactory detection accuracy. Every minute, 300 hours of videos are uploaded to YouTube [6]. If the detection speed is an order of magnitude slower than this growth rate, the NDVD (Near-Duplicate Video Detection) system will be impractical for large-scale applications. However, balancing the demands of accuracy and speed is challenging:

- **High Complexity**: The complexity of videos is higher than that of other types of big data. The feature representations developed hitherto are not capable of perfectly handling numerous types of content modifications in web videos. Thus the accuracy can be affected if certain information is ignored in the detection.
- **Contradictory Demands**: In order to increase the accuracy, the feature representations are designed to be complicated and high-dimensional [4], [7], thus more aspects of video content are included. However, this design yields high computation cost [4], which makes it not suitable for online redundancy detection.

To overcome these challenges, academic communities attempt to fuse the different features of video (e.g., temporal, spatial relations) into advanced features [4], [7] and conduct the extraction and representation of the features offline. Intuitively, since these representations contain more information, they are helpful for revealing subtle redundancy in NDVs. However, in this paper, by resorting to the information theory, we prove that fusing different features into an advanced feature does not necessarily increase its informativeness.

Therefore, we can enhance the discriminative ability of the detection/retrieval system by making use of multiple features rather than designing a complex composite feature representation. This methodology is analogous to the structure of the compound eyes of insects. The compound eyes are composed of numerous small optical systems, each of which is simple and weak; but the composition of these systems can achieve comprehensible eyesight and high visual acuity.

Following this analogy, the NDVD system we proposed is named CompoundEyes [8]. In this system, every video is represented as a bag of feature vectors of different types. Each type is generated by an independent feature extraction and representation component. As the compound eyes of insects, the satisfactory detection accuracy is achieved by the whole system rather than a particular feature extractor. This design seamlessly integrates multiple instance learning and the principles of the systems approach, and the algorithms utilized by the components are simple, fast, and adapted for parallelism exploitation. In this manner, the conflicting demands of detection accuracy and speed are mitigated.

The contributions of our paper can be summarized as follows:

- **A Shift of Detection Methedology**: In the design of an NDVD system, we shift the focus from advanced composite feature representations to the architecture of the system. First, by defining the informativeness of feature representation, we prove that fusing multiple features

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

does not increase the amount of information. Second, based on this proof, we employ the systems approach to structure the system and apply the multiple instance learning method for the fusion of the information of features.

- **Efficiency Improvement**: The paradigm shift also enhances the speed efficiency. First, the dimensionality of representations is reduced. Second, the system is parallelized to accelerate the detection speed further.
- **Implementation**: Our implementation of CompoundEyes exhibits satisfactory performance. The system maintains decent detection accuracy while substantially expedites the detection process.

The rest of the paper is organized as follows. Related work is reviewed in section II. Background knowledge concerning the feature-centered detection paradigm and the proofs that buttress the design of CompoundEyes are discussed in section III. In section IV, the architecture of CompoundEyes is proposed. This system is evaluated in section V. Section VI concludes the paper. A preliminary version of this paper has been published [8]. Here we prove the theoretical foundations for the paradigm shift, implement two types of conventional feature-based NDVD systems, and conduct more experiments to demonstrate the advantages of our system.

II. Related Work

In this section, we briefly survey the approaches and techniques that have been applied in multimedia duplicate detection/retrieval. These approaches include how to represent a multimedia item (i.e., an image or a video clip) as a processable data type (e.g., vectors), enhance the speed efficiency, and fuse features.

Despite the difference between images and videos, their representations are generally transferable, since a video consists of frames (i.e., images). There are two steps in representing an image: extracting visual features and describing the features with data types. Based on the granularity, the features can be categorized into global and local features, so are the representations.

Global features capture the global properties of an image [9], [10]. In contrast, local features are localized, salient regions in an image [11]. Global features can only be described with global representations (e.g., fingerprints, signatures), while local interest region can be described by using descriptors such as SIFT [12] or PCA-SIFT [13], or summarized into a global representation by applying the BoWs (Bag-of-Words) method [14].

Compared with images, videos have an additional temporal dimension. Taking the average of the global representations of all frames [2], [15] and sequence matching techniques [16]–[18] are on the two ends of the spectrum.

In the literature, filtering and indexing are two commonly applied approaches to accelerate the processing speed. Zhao et al. [19] filter candidate near-duplicate images by comparing their BoWs representations, before local interest regions matching. In [2], Wu et al. build a hierarchical system with color histogram representation and local interest region representation based approaches. Indexing structures are used to expedite the retrieval of near-duplicate images or videos. Hash table is one of the most popular indexing structures. Other examples include LIP-IS [20], [21], LSH (Locality Sensitive Hashing) [15], [22], or inverted indexing [7].

In order to overcome the limitations of the global and local feature representations, academic communities have investigated various strategies to fuse visual features. Shang et al. [7] utilize Conditional Entropy (CE) and Local Binary Pattern (LBP) to capture the spatial information within frames and preserve the temporal information by applying the w-shingling method. In [4], [23], by making use of manifold information, Song et al. translate key-frames into binary hash codes. The affinity relations of videos in HSV and LBP spaces are preserved in the training of the hash functions. A similar approach called kernelized multiple feature hashing (KMFH) is proposed by Zou et al. [24]. Alternative fusion strategies include multiple instance learning [25] and ensemble fusion [9], which are similar to the idea of CompoundEyes.

III. Preliminaries

There are various definitions of NDV in the literature. In this paper, we adopt the least subjective [26] definition proposed by Wu et al. [2], in which NDVs are videos with similar visual content but have undergone various modifications such as illumination changes or caption insertion. Therefore NDVD is based on visual content rather than semantics.

A. Two-stage NDVD/NDVR

Near-duplicate Video Detection (NDVD) and Near-duplicate Video Retrieval (NDVR) are different in their objectives, but the underlying techniques are transferable. In detection, the goal is to determine whether a pair of videos are similar; in retrieval, the aim is to locate the videos that are near-duplicate to the query video and position them correctly. The typical process of content-based NDVD/NDVR systems is comprised of two stages: feature extraction and representation, neighborhood construction.

1) Feature extraction and representation: A video feature is a summary of information in the visual content. Stability and distinguishability are its two valued traits. Where a feature is extracted may span globally across the whole video (e.g., color distribution), or be localized to a region (e.g., interest regions).

Extracting features from a video is conducted on a frame-by-frame basis. For instance, to calculate the color distribution of a video, the color distribution of each frame is computed first, then the average of them is taken as the color distribution of the video.

Extracted features are described by representations. Among numerous representations, histograms are widely adopted, to represent both global features (e.g., color distribution), or local features (e.g., SIFT, and BoWs).

2) Neighborhood construction: Owing to speed efficiency concerns, global representations (e.g., signatures) rather than a sequence of pattern symbols are generally preferable in NDVD/NDVR systems. When the first stage ends, videos are summarized as a point in a multi-dimensional feature space.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

If the feature is both stable and distinguishable, NDVs are adjacent whereas different videos are distant in this space. Therefore, we can identify the near-duplicate videos to a video by constructing its neighborhood in the feature space. In other words, this neighborhood is a decision boundary. The videos that reside within the boundary are regarded as duplicated videos to the given video, and others are non-duplicate.

Constructing neighborhood is critical for detection speed, especially when the dataset is large. As mentioned in Section II, to accelerate this construction, retrieval assistance schemes are introduced, such as hash tables, inverted indexing file, or LSH (Locality Sensitive Hashing) [27].

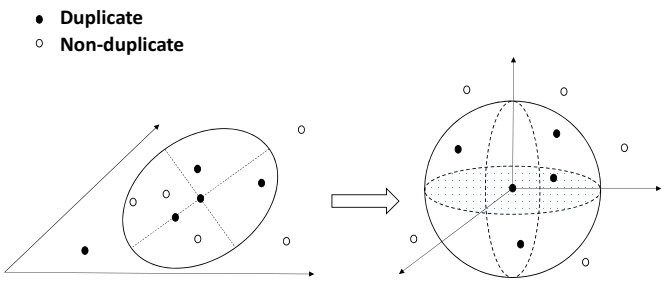


Fig. 1: Transform the Low-dimensional Feature Space into a High-dimensional Space

B. Feature-centered detection paradigm

Conventionally, feature extraction and representation in the first stage are the core of the design of NDVD systems, which have been profoundly studied. In this part, we commence our discussion about this feature-centered detection paradigm with a theoretical model, upon which the drawbacks of this paradigm are investigated, to introduce and justify the design philosophy of CompoundEyes.

1) Mathematical Model: First, we define four relevant concepts in NDVD systems as follows:

Definition 1. The neighborhood of a video $v \in V$ is $N(v) = \{v' \in V | v' \in duplicate(v)\}$.

Definition 1 is independent of feature representations.

Definition 2. The representation of a video $v \in V$ under feature $f \in F$ is defined as $X_f(v) \in R^n$ (i.e., Euclidean space). Defining the feature space as Euclidean space is not mandatory.

Definition 3. A hypersphere neighborhood of a video $v \in V$ under feature $f \in F$ is defined as $S_f(v,\tau) = \{v' | v' \in V | X_f(v') - X_f(v)| \leq \tau\}$, where $|.|$ is a distance measurement in the feature space, and τ is a parameter of S_f.

Definition 4. The error set of $S_f(v,\tau)$ is defined as $E_f(v,\tau) = \{v' \in V | v' \in N(v),\, v' \notin S_f(v,\tau)\} \cup \{v' \in V | v' \notin N(v),\, v' \in S_f(v,\tau)\}$.

With these definitions, after establishing the feature f, the task of NDVD/NDVR in this paradigm is as simple as testing whether $v' \in S_f(v,\tau)$, $v, v' \in V$. Detection/retrieval accuracy is measured by the volume of $E_f(v,\tau)$. The smaller E_f is, the better f is to embody videos.

As shown in the left part of Fig. 1, the hypersphere neighborhood in a low-dimensional space under a simple feature f_1 may not be a satisfactory approximation of $N(v)$, as $|E_{f_1}| = 4$. To increase the distinguishability, a higher-dimensional feature representation $X_f, f \in F$ is created by combining feature representations $X_{f_1}, X_{f_2}, \ldots, X_{f_n}$, $f_1, f_2, \ldots, f_n \in F$ [4], [7], [23]. The hypersphere neighborhood in this feature space is more accurate as shown in the right part of Fig. 1, since $|E_f| = 0$. However, this paradigm may encounter problems of dimensionality and informativeness.

2) Dimensionality: The first potential issue of the feature-centered paradigm is the high dimensionality of representations. Typically, there are two manners of dimensionality growth: more features being integrated, or the vocabulary of visual words expanding:

- The LBP-based spatiotemporal feature [7] is an example of feature fusion. First, each frame is represented by a binary vector of 16 dimensions; thus there are $2^{16} = 65536$ distinctive vectors (or patterns). Then the video representation is constructed by counting the frames that belong to each pattern. In this way, the dimensionality of representations is 65536.
- In BoWs methods, the dimensionality of representations is the number of visual words, or $O(\sqrt{n})$ according to a rule of thumb, where n is the number of interest regions extracted from all videos. Suppose there are 10^7 videos in a dataset, each of them has 10^2 frames and the average number of extracted regions in a frame is 10^3, the dimensionality of this representation is $10^{\frac{7+2+3}{2}} = 10^6$.

Either the combinatorial explosion or sublinear growth could lead to the high-dimensionality of representations, which imposes heavy processing cost. On the other hand, the accuracy could also be negatively affected. When dimensionality increases, the maximum distance between two random representations becomes indiscernible compared to the minimum distance, as

$$\lim_{d \to \infty} E(\frac{dist_{max}(d) - dist_{min}(d)}{dist_{min}(d)}) = 0. \quad (1)$$

Thus the neighborhood becomes less meaningful. Besides, when irrelevant or noisy dimensions are incorporated into representations, the accuracy of neighboring video retrieval drops.

3) Informativeness: The second potential issue concerns the reduction of informativeness. Informativeness, or the amount of information in representations, is critical to detection accuracy. We assume that features are represented as one-dimensional histograms because histogram is adopted to describe both global features (e.g., color distribution) and local features (e.g. SIFT, PCA-SIFT, BoW). The informativeness of a representation is defined as entropy:

Definition 5. Suppose $f_1, f_2, \ldots, f_k, \ldots \in F$ are visual features. The informativeness of a video representation $X(v) \in \{X_{f_1}(v), X_{f_2}(v), \ldots, X_{f_k}(v), \ldots\}$ is $H_v(X) = -\sum_i p_v(x_i) \log \frac{p_v(x_i)}{w_i}$.

The feature representations $X(v)$s are normalized beforehand, which is a widely adopted technique to boost the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

performance of learning algorithms. The term $p_v(x_i)$ is defined as $p_v(x_i) = g_v(x_i)w_i$, where $g_v : range(X) \rightarrow [0, 1]$ is the underlying probability density function of the one-dimensional histogram $X(v)$, and w_i is the width of the i-th bin of $X(v)$. In other words, the range of $X(v)$ is divided into consecutive, non-overlapping bins, and the value of $X(v)$ stays the same within a bin.

The formal definitions of these variables are: $w_i = u_i - l_i$, $x_i \in [l_i, u_i]$, $u_{i-1} = l_i$, $\cup_{i=1}^{n}[l_i, u_i] = range(X)$, where n is the dimensionality (i.e., number of bins) of $X(v)$.

The following properties can be proved with Definition 5.

Property 1. *$H_v(X) = 0$, if $n = 1$; $H_v(X) \rightarrow 0$, if $n \rightarrow \infty$ and g_v is discrete.*

Proof. With Definition 5, the proof of the first part is straightforward.

For the second part, $H_v(X)$ can be calculated as:

$$\begin{aligned} H_v(X) &= -\sum_i p_v(x_i) \log \frac{p_v(x_i)}{w_i} \\ &= \sum_i w_i(-g_v(x_i) \log g_v(x_i)) \end{aligned} \tag{2}$$

Two properties of the term $-g_v(x_i) \log g_v(x_i)$ in Equation 2 are relevant to the limit of $H_v(X)$, boundedness and continuity.

First, since the range of g_v is $[0, 1]$, and $\lim_{g_v(x)\rightarrow 0} g_v(x) log g_v(x) = 0$, which can be proved by applying L'Hopital's rule, $| - g_v(x) log g_v(x)| < \infty$. Thus $-g_v(x_i) \log g_v(x_i)$ is bounded.

Second, the continuity of $-g_v(x) log g_v(x)$ depends on the continuity of g_v. Consequently, we branch the proof of this part into two cases:

Case 1. *According to Equation 2, $H_v(X)$ is the Riemann sum of the function $-g_v(x) log g_v(x)$. Since this function is bounded, $H_v(X)$ is Riemann integrable when g_v is continuous or piece-wise continuous. The limit of $H_v(X)$ when n approaches infinity is not necessarily zero.*

Case 2. *If g_v is discrete, there are at most countable bins over which $g_v(x_i)$ is defined when n approaches infinity.*

For a bin in which $g_v(x_i)$ is defined, $w_i(-g_v(x_i) \log g_v(x_i))$ is an infinitesimal when n goes to infinity (i.e., $w_i \rightarrow 0$).

With the induction technique, we can prove that the summation of countable infinitesimals is still an infinitesimal. We define a statement that $P(n)$ is an infinitesimal for all $n \in \mathbb{N}$. $P(n) = \sum_{i=1}^{n} \epsilon_i$, where $\epsilon_i, i = 1 \ldots n$ is an infinitesimal.

***Base Case:** When $n = 1$, an infinitesimal ϵ_1 itself is an infinitesimal;*

***Inductive Step:** Suppose $P(n)$ holds until $n = k - 1, k > 2, k \in \mathbb{N}$, then $P(k) = P(k-1)+\epsilon_k$, where both $P(k-1)$ and ϵ_k are infinitesimals. By the definition of infinitesimal, $P(k)$ is also an infinitesimal, so $P(n)$ is valid for all $n \in \mathbb{N}$.*

To summarize, if g_v is discrete, $H_v(X) \rightarrow 0$ as $n \rightarrow \infty$.

□

Due to the losses in sampling and digital computations, the underlying probability density functions of the feature representations in numerous real-world applications are discrete rather than continuous. According to Property 1, increasing the dimensionality of this type of representations does not necessarily make it more informative. On the contrary, as the number of bins (i.e., dimensions) rises, the representations become sparse, and their informativeness gets closer to zero. Experiments with BoWs representations [28] buttress this corollary. Essentially, Property 1 reveals the curse of dimensionality as Equation 1 does, from another perspective.

Property 2. $H(X_{f_1}(v), X_{f_2}(v), \ldots, X_{f_k}(v)) \geq H(X_{f_i}(v)), i = 1, \ldots, k, \ldots$.

Property 3. $H(X_{f_1}(v), X_{f_2}(v), \ldots, X_{f_k}(v)) \leq H(X_{f_1}(v)) + H(X_{f_2}(v)) + \ldots + H(X_{f_k}(v))$;

Proof. By Definition 5, $P\{X_f(v) = x_i\} = g_v(x_i), f \in \{f_1, f_2, \ldots, f_k \ldots\}, x_i \in range(X_f(v))$. Therefore, a feature representation $X_f(v)$ can also be viewed as a random variable. In this way, the term $H(X_{f_1}(v), X_{f_2}(v), \ldots, X_{f_k}(v))$ is mathematically sound, and the definition of informativeness remains the same.

According to the chain rule,

$$\begin{aligned} &H(X_{f_1}(v), X_{f_2}(v), \ldots, X_{f_k}(v)) = \\ &\sum_{i=1}^{k} H(X_{f_i}(v)|X_{f_1}(v), \ldots, X_{f_{i-1}}(v)). \end{aligned} \tag{3}$$

Equation 3 can also be written as,

$$\begin{aligned} &\sum_{i=1}^{k} H(X_{f_i}(v)|X_{f_1}(v), \ldots, X_{f_{i-1}}(v)) = \\ &\sum_{i=1}^{k-1} H(X_{f_i}(v)|X_{f_1}(v), \ldots, X_{f_{i-1}}(v)) + H(X_{f_k}(v)). \end{aligned} \tag{4}$$

Due to the non-negativity of entropy and Equation 4, the following inequality holds,

$$H(X_{f_1}(v), X_{f_2}(v), \ldots, X_{f_k}(v)) \geq H(X_{f_k}(v)). \tag{5}$$

Based on Inequality 5 and the symmetry of joint entropy, Property 2 can be proved.

To prove Property 3, we use the monotonicity property that conditioning reduces entropy,

$$H(X_{f_i}(v)|X_{f_1}(v), \ldots, X_{f_{i-1}}(v)) \leq H(X_{f_i}(v)). \tag{6}$$

By plugging Inequality 6 into Equation 3, Property 3 can be proved. □

In Property 2 and 3, the joint distribution of random variables $X_{f_1}, X_{f_2}, \ldots, X_{f_k}$ models the combination of these feature representations. From Property 2, constructing a sophisticated representation via feature fusion does increase its informativeness compared with every single feature representation. However, according to Property 3, the informativeness of this composite representation is upper bounded by the sum of the informativeness of component representations. Therefore, building a sophisticated classifier, and feeding it

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

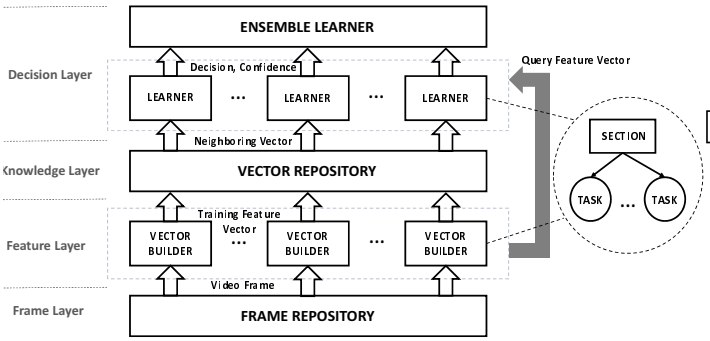


Fig. 2: The Architecture and Parallel Organization of CompoundEyes

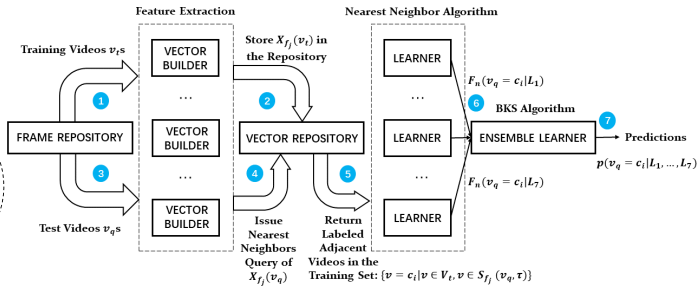


Fig. 3: The Data Flow of CompoundEyes

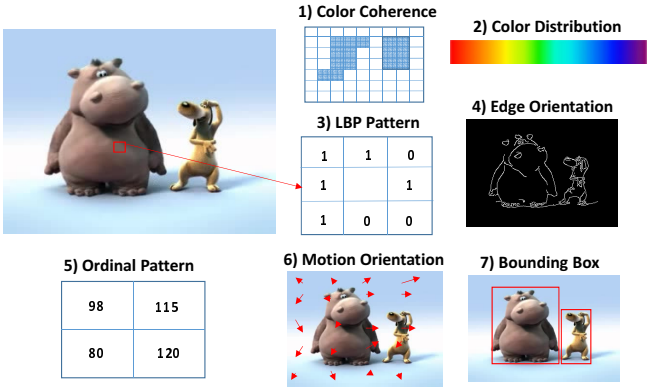


Fig. 4: The Seven Features in CompoundEyes

with multiple feature representations could achieve higher accuracy than feeding a simple classifier (i.e., the hypersphere neighborhood) with a composite representation.

IV. System Design

According to Property 1 and 3, shifting the focus from building an advanced feature representation to an advanced classifier is beneficial to detection accuracy. In order to satisfy the speed requirement, our system is designed according to the principles of the systems approach. In this approach, components are simple, efficient, and independent of each other. Parallelism generated from this autonomy is also exploited to increase speed further.

A. Architecture

An abstraction layer model can illustrate the architecture of CompoundEyes. In this model, frames are sampled at the Frame layer; the features of these frames are extracted and represented at the Feature layer. From these representations, patterns of NDVs generate at the Knowledge layer, which finally emerge at the Decision layer and are used to make predictions about videos being duplicated or not.

The system is divided into three subsystems: Feature Vector Builder, Vector Repository, and Ensemble Learner. These subsystems are located on the Feature, Knowledge and Decision layers, as shown in Fig. 2.

In related systems, most of the computational overhead is originated from Feature Vector Builder. The subsystem is intrinsically complicated due to the complexity of the visual content of multimedia objects. By following the principles of systems approach, we divide the Feature Vector Builder subsystem into various Vector Builders, each of which uses a unique feature extraction and representation algorithm. For each Vector Builder, there is a weak Learner that uses its representations to make predictions. The Ensemble Learner collects these predictions, to make final predictions. This design also conforms to the ideas of multiple instance learning.

The division of the functionalities of the system ensures the exploitation of the hidden parallelism. The parallel organization of CompoundEyes is hierarchical, as depicted inside the dashed rectangles and circle of Fig. 2. The first level is the function parallelism among components, i.e., Vector Builders and weak Learners. They compete for computation resources to perform their computations. The second level is the data parallelism within the computations of Vector Builders. Upon the allocation of computation resources, one or more parallel tasks are spawned, among which the computations of the Vector Builder are divided.

B. Data Flow

The data flow of CompoundEyes is presented in Fig. 3.

1) Feature Layer: In the Feature layer, we utilize seven feature extraction algorithms: color coherence, color distribution, LBP (Local Binary Pattern), edge orientation, ordinal pattern, motion orientation, and bounding boxes of objects, as explained in Fig. 4. All of these algorithms are simple and efficient. Furthermore, the diversity of these features enhances the accuracy of the final prediction [29].

The Vector Builders work on a frame-by-frame basis. Suppose the j-th Vector Builder deals with feature $f_j, j = 1, \ldots, 7$, it first extracts f_j from a key-frame of video v and represents it as a histogram $X^i_{f_j}(v), i = 1, \ldots, |v|$, where $|v|$ is the number of key-frames in v. Then the representation of v is calculated as $X_{f_j}(v) = \frac{1}{|v|} \sum_{i=1}^{|v|} X^i_{f_j}(v)$. The summation can be computed in parallel, which is referred to as the frame-level data parallelism. This parallelism is exploited by distributing the computations of $X^i_{f_j}(v), i = 1, \ldots, |v|$ onto the tasks obtained by this Vector Builder, as shown in Fig. 2 and Fig. 3.

2) Knowledge layer: The neighborhoods constructed by Vector Repository are hypersphere neighborhoods defined in Definition 3. After converting videos into bags of representations $\{X_{f_j}(v) | v \in V, j = 1, \ldots, 7\}$, the representations of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

the videos in the training set V_t are stored and indexed in Vector Repository along with their ground-truth labels. Vector Repository consists of seven subspaces, each of which only stores a type of feature representations $X_{f_j}, j = 1, \ldots, 7$. With Vector Repository, CompoundEyes acquires the capabilities of both NDVD and NDVR systems.

When the representations of a query video v_q from the test set V_q, $\{X_{f_j}(v_q)|j = 1, \ldots, 7\}$, are issued to Vector Repository, its neighborhoods, $\{S_{f_j}(v_q, \tau)|j = 1, \ldots, 7\}$, are computed and returned to Learners in the Decision layer respectively. Video Repository makes this neighboring video retrieval procedure more efficient.

We implement the Vector Repository as an LSH [30], [31] structure. The reason is twofold. First, LSH is more accurate in neighboring video retrieval. Second, the temporal cost of retrieval is $O(1)$. Besides, the LSH structure is combined with Cuckoo Hashing [32]. As a result, the problems of unbalanced load among hash tables and of local similar sets are mitigated, which further enhances its retrieval performance.

3) Decision layer: We model the NDVD task as a classification problem. A handful of videos are designated as seed videos (i.e., reference videos). Compared with one of the reference videos, a video $v \in V$ can be labeled as n possible classes $c_i, i = 1, \ldots, n$. For example, when $n = 2$, the classes are duplicate and non-duplicate. In CompoundEyes $n = 7$, because the dataset which we adopt divides videos into seven categories: Exactly Duplicate, Similar, Different Version, Major Change, Long Version, Dissimilar, and Do not Exist. Dissimilar and Do not Exist are treated as the same.

As shown in Fig. 2, Learners (or classifiers) in the Decision layer are organized hierarchically. The prediction about a video being duplicate is made upon the hypotheses of the seven weak Learners.

The weak Learners are denoted as $L_j, j = 1, \ldots, 7$. The seven Learners corresponds to the seven features that we utilize. The videos from both the training set V_t and test set V_q are summarized as bags of representations $\{X_{f_j}(v)|v \in V_t \cup V_q, j = 1, \ldots, 7\}$ in the Feature layer. $\{X_{f_j}(v)|v \in V_t, j = 1, \ldots, 7\}$ are stored in Vector Repository along with their ground-truth labels $\{v = c_i|v \in V_t, i = 1, \ldots, 7\}$, while $\{X_{f_j}(v)|v \in V_q, j = 1, \ldots, 7\}$ are directed to Learners $L_j, j = 1, \ldots, 7$, respectively, as shown in Fig. 2 and Fig. 3. In L_j, the probabilities $p(v_q = c_i|L_j), i = 1, \ldots, 7$ are approximated with frequencies,

$$Fn(v_q = c_i|L_j) = \frac{|\{v = c_i|v \in V_t, v \in S_{f_j}(v_q, \tau)\}|}{|\{v|v \in V_t, v \in S_{f_j}(v_q, \tau)\}|}$$
$$i = 1, \ldots, 7.$$

The computation of $S_{f_j}(v_q, \tau)$ is performed by Vector Repository, as mentioned above.

These frequencies are taken as input to Ensemble Learner, which calculates the posterior probabilities $p(v_q = c_i|L_1, \ldots, L_7), i = 1, \ldots, 7$, utilizing the BKS (Behavior-Knowledge Space) method [33] as follows,

$$p(v_q = c_i|L_1, \ldots, L_7) \cong \hat{p}(v_q = c_i|L_1, \ldots, L_7),$$

$$\hat{p}(v_q = c_i|L_1, \ldots, L_7) = \frac{Fn(v_q = c_i|L_1, \ldots, L_7)}{\sum_j Fn(v_q = c_j|L_1, \ldots, L_7)}.$$

To make the estimation of $Fn(v_q = c_i|L_1, \ldots, L_7), i = 1, \ldots, 7$ easier, we assume $L_j, j = 1, \ldots, 7$ are conditionally independent, which is sensible because of the diversity of features. With the approximation $p(v_q = c_i|L_j) \cong Fn(v_q = c_i|L_j), i = 1, \ldots, 7$, we have,

$$\begin{aligned} p(v_q &= c_i|L_1, \ldots, L_7) \propto p(L_1, \ldots, L_7|v_q = c_i) \\ &= \prod_{j=1}^{7} p(L_j|v_q = c_i) \propto \prod_{i=1}^{7} p(v_q = c_i|L_j) \\ &\cong \prod_{j=1}^{7} Fn(v_q = c_i|L_j), i = 1, \ldots, 7. \end{aligned}$$

Therefore, with appropriate normalization, the probabilities are estimated as

$$p(v_q = c_i|L_1, \ldots, L_7) = \frac{\prod_{j=1}^{7} Fn(v_q = c_i|L_j)}{\sum_{k=1}^{7} \prod_{j=1}^{7} Fn(v_q = c_k|L_j)}$$
$$i = 1, \ldots, 7.$$

The class with the largest posterior probability would be the final prediction of the class of v_q.

Applying the Nearest Neighbor algorithm on the weak Learners and the BKS method on Ensemble Learner yields satisfactory performance. The reasons are as follows: first, Vector Repository computes $S_{f_j}(v_q, \tau)$ efficiently, the cost of which is $O(1)$; second, the Nearest Neighbor algorithm is non-parametric, which is helpful to reduce the training cost to $O(1)$, fulfilling the in-situ requirement; third, the Nearest Neighbor algorithm is sensitive to the variations of feature types [29], thus making it suitable in the scenario of multiple features; fourth, the BKS method is sufficiently accurate to be applied on Ensemble Learner [33].

C. Advantages

The advantages of CompoundEyes can be illustrated from the following aspects:

a) Accuracy: The accuracy improvement is primarily achieved via the collective efforts of Learners. First, the coverage of features is broader. Not only are the spatial and temporal information used in learning, but also the color, edge orientation, texture, and object sizes information. Second, the diversity of representations enhances the accuracy of learning.

b) Detection Speed: Two factors contribute to the improvement of detection speed. The first one is the compactness of representations, which shortens the temporal cost of extracting feature vectors and of neighboring vectorial representation retrieval. The second one is the exploitation of the function parallelism among Vector Builders and weak Learners, and the frame-level data parallelism within Vector Builders.

c) In-situ Updating: CompoundEyes has the capacity of continually updating its classifiers with new knowledge (i.e., videos and corresponding ground-truth labels), because the training cost is $O(1)$, and the changes in classifiers do not affect the construction of representations in the Feature Layer.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

d) Modularity: The components in CompoundEyes are independent, so they can be changed without affecting others. For example, a new Vector Builder that extracts a new feature can be admitted if necessary, so is the case with weak Learners implementing other classification algorithms, and Vector Repository utilizing alternative indexing schemes. Therefore, the system could be easily upgraded.

V. Evaluation

A. Experimental Setup

We implement CompoundEyes in C++, C, and Matlab. Specifically, Vector Builders are coded in C++, with the assistance of OpenCV libraries. Weak Learners and Vector Repository are implemented in C, and Ensemble Learner is programmed in Matlab. The parallel parts of CompoundEyes are implemented by using OpenMP libraries.

Experiments about CompoundEyes are conducted on a 64-core Intel Xeon E5-4640 machine (2.4GHz, 12.5GB memory) with the Ubuntu system. The cores are distributed equally into 4 NUMA nodes. This multi-core machine is favorable for the parallel computations of CompoundEyes, and boosts the speed substantially.

CompoundEyes is evaluated against other NDVD/NDVR systems that adopt the CC_WEB_VIDEO dataset. The source code of these systems is not available, except MFH [4]. However, the memory demand for the matrix computations in this system is too large to be satisfied by our machines. Therefore, in the comparisons of accuracy and response time, we adopt the values reported in the papers. Concerning the preprocessing (i.e., feature extraction and representation, representation storage, and training) time, the comparisons are conducted theoretically. The reason is twofold. First, the temporal cost of the preprocessing of other systems is not provided in those papers. Second, the parallelization of the preprocessing of CompoundEyes substantially increases its detection speed, which makes the comparisons unfair.

BoWs feature representation is widely used by the vision community. Recently, the feature representations generated by deep neural networks have shown promising results in challenging tasks such as automatic image annotation. We implement two NDVD systems based on these two classical features and compare these systems with CompounEyes regarding accuracy. The two systems are coded in Python, with the assistance of OpenCV and TensorFlow libraries. The BoWs-based system is deployed on a 4-core Intel i3-3220 machine (3.3GHz, 16GB memory), and the deep neural network-based system is deployed on a 4-core Intel i5-4460 machine (3.2GHz, 12GB memory), whose GPU is GeForce GT-720.

B. Dataset description

We evaluate CompoundEyes on the CC_WEB_VIDEO dataset, which contains 12790 web videos. There are four reasons for this selection.

- First, this dataset was constructed from real online videos. The videos are from YouTube, Google Video, and Yahoo! Video.
- Second, various formats and content modifications are included.
- Third, it has been widely adopted, which facilitates us to compare the performance.
- Fourth, ground-truth labels are provided. The videos are labeled manually by the researchers, which is laborious and makes the dataset precious for NDVD/NDVR research.

The CC_WEB_VIDEO dataset is comprised of 24 independent groups. Each group corresponds to a search keyword and contains the videos that are returned by video search engines after entering the search keyword. In each group, a video is designated as the seed video, and others are compared with it and labeled accordingly. The similarity relations between them, such as "Exactly Duplicate," are attached to the corresponding videos as their classes/categories/labels. As mentioned in section IV-B3, there are seven classes. The researchers who built the dataset defines these classes. They want to define more fine-grained categories for the similarity relations between videos, rather than just simply "Similar" and "Dissimilar."

C. NDVD/NDVR systems in the literature

To evaluate the performance of CompoundEyes, we compare it with existing state-of-the-art NDVD/NDVR systems that have been evaluated on the CC_WEB_VIDEO dataset or extended datasets. They are described as follows:

Hierarchical detection system (HIER): Wu et al. [2] propose a hierarchical NDVD system, which uses a global signature-based method to filter out duplicates with minor changes first, leaving more sophisticated changes to the local feature-based method.

Video Cuboid based detection system (VC): Zhou et al. [22] introduce the Video Cuboid signature, an n-gram based representation, to integrate the temporal and spatial information. Further optimizations include the use of the EMD distance, incremental signature construction, and an LSH based matching scheme.

Spatial-temporal feature based detection system (ST): Shang et al. [7] explore alternative approaches to combine the temporal and spatial information into signatures. Two approaches are proposed: Conditional Entropy (ST-CE) and Local Binary Pattern (ST-LBP). The retrieval process is accelerated by applying a fast intersection kernel and inverted files.

Multiple feature hashing based detection system (MFH): Song et al. [4] provide another combination of global and local features. A series of hash functions are learned from the feature representations. The neighboring video searching is conducted in the Hamming space of the hash codes.

In these systems, VC provides us with the results of accuracy, while others are more concerned with mean average precision and average response time. Hence, we will compare CompoundEyes with VC regarding accuracy, and with others regarding mean average precision and average response time.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

D. NDVD Systems based on classical visual features

These NDVD systems are designed with the feature-centered paradigm. The feature extraction and representation algorithms are advanced, through which the videos are represented as discriminative, high-dimensional vectors in Euclidean space. The distances between the representations, along with corresponding ground-truth labels of videos in the training set are fed into a one-vs-the-rest SVM (Support Vector Machine) classifier. Because we compare CompoundEyes with these NDVD systems only concerning the accuracy, efficiency-boosting techniques such as LSH are not involved in the design.

Bag-of-Words (BoWs): As aforementioned, in the BoWs approaches, local features of an image are extracted first, then summarized into a global representation (i.e., a histogram of the frequencies of the occurrence of visual words). Converting a video into a BoWs representation consists of two phases: the construction of visual word vocabulary, and the interpretation of videos based on this vocabulary.

In the first phase, we extract two types of local features from frames and build vocabularies accordingly. The first type is SIFT (Scale-Invariant Feature Transform), and the second one is SURF (Speeded Up Robust Features). Both of them are effective for a variety of computer vision applications. We randomly select 10% of the local features of all the keyframes of the videos in the training set and perform K-Means clustering on them. By rule of thumb, the number of clustering centers is set to be the squared root of the number of local features.

Converting the videos in the test set into BoWs representations commences when the vocabulary is available. Since the dimensionality of the BoWs representation equals to the size of the visual word vocabulary, the computation of the representation of a video can be conducted by merely adding the representations of each frame of the video.

Deep convolutional neural network (CNN): With extensive training sets, deep convolutional neural networks are capable of outperforming humans in visual recognition tasks. The trained networks with good generalizability can be used as a base network in transfer learning. The feature representations generated by these networks are more effective in vision tasks than descriptions such as color histograms or BoWs representations, even on a different image set.

By detaching the last softmax layer, a standard deep convolutional neural network converts a frame into a high-dimensional feature vector. The feature representation of a video can be computed by averaging the feature vectors of its keyframes. Generally speaking, the feature representations of the videos generated in this way are effective if the pre-trained neural network performs well in annotating the frames of the videos. From preliminary experiments, we discovered that regarding annotation performance, VGGNet [34] is superior to Inception-v3 [35] on annotating the frames of the CC_WEB_VIDEO dataset. Therefore, we use the 16-layer and 19-layer VGGNets implemented in TensorFlow for the NDVD task. The dimensionality of the feature representations of both networks is 1000.

TABLE I: The Comparisons of Performance with other NDVD/NDVR Systems in the Literature

SYSTEM	VC	HIER	ST-CE	ST-LBP	MFH	**Ours**
AC(%)	80	N/A	N/A	N/A	N/A	**89.2**
MAP(%)	N/A	95.20	95.30	95.00	95.40	**99.75**
RT (ms)	N/A	9600	3.7	3.6	N/A	**0.2051**
PMU	N/A	$O(k)$	$O(n)$	$O(n)$	$O(k^3n^3)$	$\boldsymbol{O(k)}$
TC	N/A	$O(kn^2)$	$O(kn)$	$O(kn)$	$O(k^3n^3)$	$\boldsymbol{O(kn)}$

TABLE II: The Comparisons of Accuracy with Classical Feature-based NDVD Systems

SYSTEM	BoWs-SIFT	BoWs-SURF	CNN-16	CNN-19	**Ours**
AC (%)	79.27	78.66	76.93	78.80	**80.91**
MAP (%)	98.26	98.18	92.53	97.66	**99.21**

E. Experimental results

In this subsection, experiments are conducted to evaluate the performance of CompoundEyes. Datasets of various sizes are constructed by randomly selecting videos from the CC_WEB_VIDEO dataset. Unless stated otherwise, in each one of them, 50% are used as the training set and the other 50% as the test set.

1) Accuracy:

a) Evaluation metrics:

- **Accuracy**: The portion of correct predictions in total results.
- **Mean Average Precision**: The Mean Average Precision (MAP) is computed by averaging the Average Precision (AP) of each group g, as $MAP = \frac{1}{24}\sum_{g=1}^{24} AP_g$, $AP_g = \frac{1}{n}\sum_{i=1}^{n}\frac{i}{r_i}$, where n is the number of correct predictions , r_i is the rank of i-th correct prediction.

b) Results: CompoundEyes shows improvements on detection accuracy. It achieves a higher Accuracy (AC) than the VC system, 89.28% vs. 80%, and outperforms other NDVD/NDVR systems in Mean Average Precision (MAP), as shown in TABLE I.

A subset of the CC_WEB_VIDEO dataset with 10% randomly selected video clips is constructed to evaluate CompoundEyes against the NDVD systems based on the BoWs and CNN feature representations. The construction of the BoWs visual word vocabularies will fail due to the shortage of memory if more portions of videos or local features are involved in the computations of vocabulary construction. Besides, the temporal cost of the computations of the CNN feature representations for the videos is high, especially when these computations are conducted on outdated machines.

TABLE II shows the comparisons of Accuracy and Mean Average Precision between CompoundEyes with the two classical feature-centered NDVD systems. Depending on what type of local features are extracted, and the number of layers in the convolutional neural network, the two systems can be further divided into four systems (i.e., BoWs-SIFT, BoWs-SURF, CNN-16, CNN-19).

From TABLE II, we observe that CompoundEyes is more effective than other NDVD systems concerning both Accuracy and Mean Average Precision, despite the simplicity and low

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

dimensionality of the features that it applies, and the low cost of training. To further investigate the reason behind these counter-intuitive comparison results, we decompose the comparisons of average Accuracy and Mean Average Precision for all the videos in the subset into the comparisons over the 24 groups, as shown in Fig. 5(a) and Fig. 5(b).

To investigate what factors impact the accuracy in different video groups, we pinpoint the incorrect predictions made by CompoundEyes and other models. The following observations are obtained:

- In group 4, the seed video is about two cats playing. Both of the SIFT and SURF models achieve perfect accuracy, but CNN-16 and CNN-19 make plenty of false predictions. The themes of the videos that they incorrectly predict as "Similar" are cat related indeed, but different from the seed video in content. Where CompoundEyes fails are the videos that are different in length, luminescence, or the addition of certain close-up shots.
- In group 14, the seed video has a lot of redundant frames, which downgrade the performance of all models. However, the accuracy of benchmark models is better than that of CompoundEyes, since their mistakes are insignificant (mistaking "Similar" videos for "Exactly Similar"). Whereas CompoundEyes incorrectly classifies three "Similar" videos as "Dissimilar." All the three misclassified videos are edited. A fraction of frames are added, deleted, or changed, but the editions do not affect the theme of the videos.
- In group 20, the seed video does not have a specific topic and contains plenty of close-up shots. CompoundEyes achieves perfect performance, while benchmark models are not capable of handling these videos. Their mistakes do not have a characteristic, and the topics of these videos are diverse, including piano playing, manga, drama, dancing.

From these observations, it follows that one of the significant factors influencing the accuracy of prediction is the abstraction level of features. The seven features utilized in CompoundEyes are essentially low-level visual features; thus the similarity detection is vulnerable to changes in length, luminescence, or the addition/deletion of frames. In contrast, the features extracted by deep neural networks such as CNN-16 are more about high-level semantic meanings of frames. Consequently, distinct videos that share a similar topic are less discernible to the classifiers that apply these features. Without further fine-tuning, high-level semantic features may not be a better option than low-level visual features in the NDVD/NDVR tasks.

The comparisons in Fig. 5(b) are slightly different than the ones in Fig. 5(a). Both the CNN-19 and CompoundEyes achieve 100% Average Precision on almost the 24 video groups, whereas other systems fail on certain groups. In conclusion, from the comparisons of Accuracy and Mean Average Precision, CompoundEyes built on simple visual features surpasses or is on par with the sophisticated 19-layer VGGNet.

2) Detection Speed:

a) The Definition of Temporal Cost: The detection speed of CompoundEyes is measured by the temporal cost, which is the sum of the preprocessing time and response (i.e., retrieval and classification) time:

$$Temporal\,Cost = Preprocessing\,Time + Response\,Time.$$

b) Analysis of Preprocessing Time Cost: In the literature, preprocessing is performed offline thus its temporal cost is not measured. The overhead of preprocessing can be estimated from the fact that in HIER, ST-CE or ST-LBP, extracting features on a dataset of 132647 videos is practically impossible [4].

Suppose the number of videos is n, and the average number of keyframes in a video is k. The peak memory usage and worst case time complexity of the preprocessing of the systems are estimated in TABLE I.

According to the fifth and sixth rows of TABLE I, CompoundEyes has advantages in both the peak memory usage and time complexity. It neither involves the computations and pairwise comparisons of SIFT descriptors as HIER, nor the computations of certain global variables, such as the entropy of ordinal relations in ST-CE, the correlation between LBP patterns in ST-LBP, and the transformation and bias matrices in MFH. The computations of these variables are both spatially and temporally exhaustive. In contrast, the two major operations of CompoundEyes in preprocessing, constructing feature representations and inserting them into Vector Repository, are spatially and temporally efficient. The average temporal cost of preprocessing of CompoundEyes is 1.4537s.

c) Experimental Results of Response Time Cost: The advantage of CompoundEyes in detection speed can also be manifested from response time, as shown in TABLE I. The average response time of CompoundEyes only accounts for 5.70% of ST-LBP's.

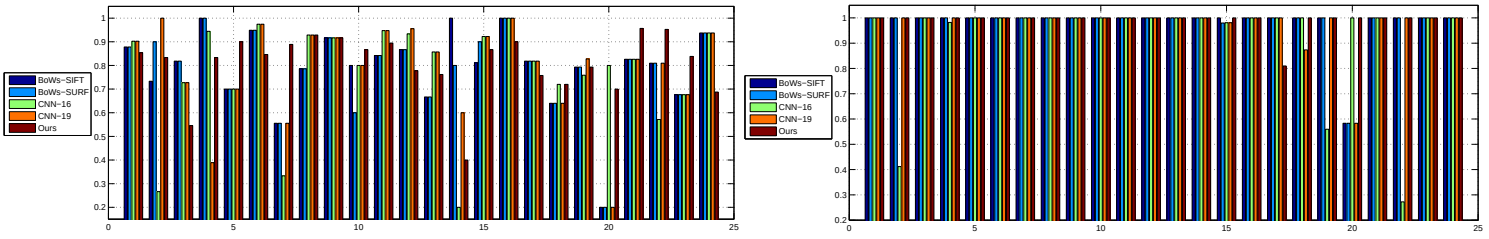
Implementing the central part of CompoundEyes in C++ instead of Matlab may contribute to the reduction of response time. However, such a substantial reduction could not be explained merely by the efficiency of C++. In CompoundEyes, the dimensionalities of representations are 16, 32, and 64, all of which are much lower 65536 of ST-CE and ST-LBP [7]. This reduction in dimensionality is the main reason for the improvement in response time.

3) Parallel Speedup: Experiments in this subsection are also performed on a 10% subset of CC_WEB_VIDEO. The temporal costs of the sequential and parallel version of CompoundEyes are compared to evaluate the parallel speedup.

The average temporal cost of each Vector Builder is estimated in Fig. 6(a) first, and used as a reference for workload distribution. On the horizontal axis are the abbreviations of the features extracted, which are color histogram (HSV), color coherence (CC), ordinal pattern (SP), edge orientation (EO), bounding boxes of objects (BB), local binary pattern (LBP), and motion orientation (OPT_FLOW).

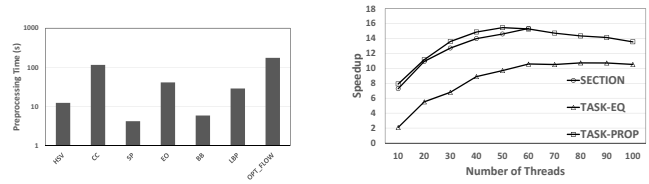
a) Thread Allocation Strategies: Both the parallel sections and tasks in Fig. 2 are OpenMP abstractions of threads. With different thread allocation strategies, the overall parallel speedup would be different. Therefore, we design and compare

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data



(a) Accuracy (b) Average Precision

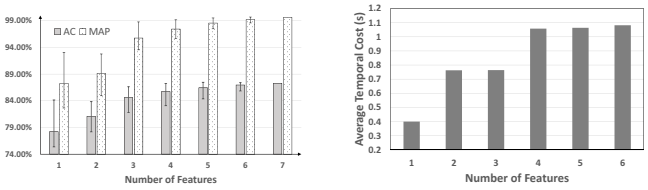
Fig. 5: Comparisions with Classical Feature-based NDVD Systems on the 24 Groups of Videos



(a) The Sequential Preprocessing Time of All the Vector Builders.

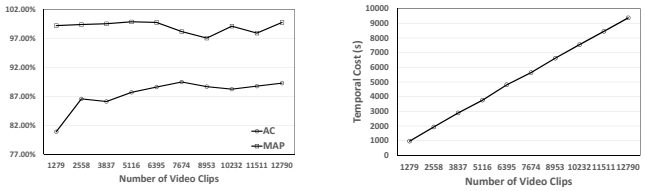
(b) The Speedup of CompoundEyes Under 3 Thread Allocation Strategies

Fig. 6: Sequential and Parallel Versions Comparison



(a) On AC and MAP (b) On Temporal Cost

Fig. 7: The Effect of Feature Information Fusion



(a) On AC and MAP (b) On Temporal Cost

Fig. 8: The Effect of the Size of Dataset

three allocation strategies as follows, to sensibly provision the computing resources:

- **SECTION**: What varies in this strategy is the number of parallel sections competed by Vector Builders, from 1 to 7. Once a parallel section is assigned to a Vector Builder, a number of parallel tasks will be allocated for computing. This number is proportional to the Vector Builder's sequential running time.
- **TASK-EQ**: In this strategy, every Vector Builder acquires a parallel section. What varies is the number of tasks spawned by a section, which is the same for all Vector Builders.
- **TASK-PROP**: In this strategy, every Vector Builder obtains a parallel section, and the number of tasks allocated to a Vector Builder is proportional to its sequential running time.

b) Results: As expected, from Fig. 6(b), TASK-PROP achieves the best speedup, because it efficiently utilizes allocated threads. Moreover, we notice that when the number of threads exceeds 60, speedup ceases to rise. Since the value equals the number of cores in the machine, this phenomenon is a hint of resource contention.

We also notice that even the best allocation strategy does not deliver linear speedup. The reason is that in CompoundEyes, videos are processed sequentially, which limits the throughput of the system.

4) Feature Information Fusion: In this part, we assess the impact of feature information fusion, mainly on the detection accuracy. The experiments are conducted on a 10% subset. For the sake of fairness, the number of parallel sections is equal to the number of features to be combined, and the number of tasks that a section can spawn is equal for all Vector Builders.

As shown in Fig. 7(a), on average, the fusion increases the detection accuracy, both regarding Accuracy and Mean Average Precision. This growth shrinks when measured by the best accuracy of fusion. For example, the accuracy difference between the optimal combination of three features and four is negligible. Therefore, it is of importance to select the features to fuse.

For the optimal combinations except all-included, corresponding average temporal costs are shown in Fig. 7(b). They are helpful when choosing the number of features. For instance, fusing three features is better than four, because it costs less time but achieves comparable detection accuracy.

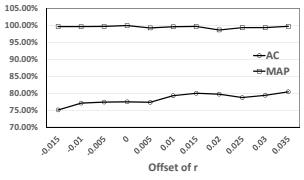
5) Relevant Parameters:

a) The Size of the Dataset: The first relevant parameter is the size of the dataset. According to Fig. 8(a), Accuracy is above 80% when the size is 1279. It increases as the size of dataset grows. Therefore, CompoundEyes is accurate when sufficient knowledge has been learned, and its discriminative capability develops as knowledge accumulates.

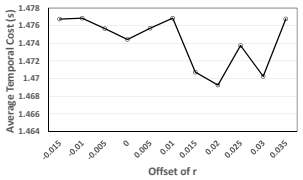
Fig. 8(b) affirms that the total temporal cost increases linearly rather than exponentially with the growth of dataset. This linearity confirms that Vector Repository is capable of maintaining decent performance even if the size of dataset becomes large.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

Fig. 9: The Effect of the Size of the Training Set on AC and MAP



(a) On AC and MAP



(b) On Temporal Cost

Fig. 10: The Effect of r

b) The Size of the Training Set: Because a system well-tuned on the training set could behave poorly on the test set, it is necessary to evaluate the detection accuracy of CompoundEyes with training sets of different sizes.

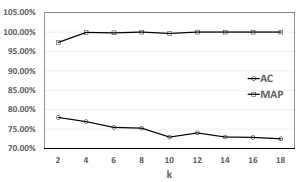
The effect of the size of the training set on Accuracy and Mean Average Precision is recorded in Fig. 9. In these experiments, the CC_WEB_VIDEO dataset is divided into a training set and a test set. The ratio of the size of the training set to the size of the test set varies along the horizontal axis. The value of Mean Average Precision stays stable, and the value of Accuracy increases as the ratio increases. Both of them peak around $5:5$. Afterward, the classifiers are over-trained.

c) Vector Repository-related Parameters: Two Vector Repository-related parameters, r and k, are of importance. Parameter r has the same meaning as τ that appears in Definition 3. Parameter k is the number of hash tables. Generally speaking, a larger value of k increases the detection accuracy, at the expense of longer response time.

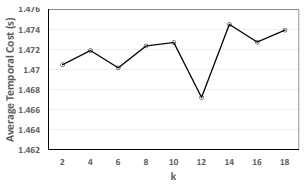
Because the value of r is different for each type of feature representations, we set them by experience first, then change them with the same offset. The effect of r on Accuracy and Mean Average Precision is shown in Fig. 10(a), and the effect on average temporal cost is shown in Fig. 10(b).

Since k is same for all subspaces in Vector Repository, we vary its value directly. From Fig. 11(a), we observe that Accuracy and Mean Average Precision exhibit different trends, the former one goes down while the latter one goes up and stays around 100%. This difference in trends is because as k increases, the recall of neighboring feature representation retrieval grows, but the precision goes down. These changes reflect on Accuracy but not Mean Average Precision, for the number of correct results and their ranks are barely affected.

The effect of k on average temporal cost is shown in Fig. 11(b), from which we know that 12 is the optimal value for the detection speed.



(a) On AC and MAP



(b) On Temporal Cost

Fig. 11: The Effect of k

VI. Conclusion

In this paper, we proposed and developed CompoundEyes, an effective and efficient NDVD system. The design of this system follows a novel detection paradigm, where a bag of simpler feature representations has replaced the sophisticated feature representation. With this functionality decomposition, the structure of the system can be designed by the principles of the systems approach, thereby the lower complexity of each component and the parallelism among them can be exploited to reduce the temporal overhead for NDVD tasks. Meanwhile, the accuracy of the detection remains decent because of the effective fusion of the information in features. The experiment and analysis results corroborate that concerning the detection accuracy, CompoundEyes not only surpasses other contemporary feature fusion NDVD/NDVR systems but also is on par with the feature-centered systems based on BoWs and CNN features. In the meantime, CompoundEyes outperforms other systems in the peak memory usage and time complexity. In conclusion, CompoundEyes is sufficiently effective and efficient for large-scale NDVD tasks of web videos.

References

[1] M. LOPES. Videos may make up 84 percent of internet traffic by 2018: Cisco. [Online]. Available: http://www.reuters.com/article/2014/06/10/us-internet-consumers-cisco-systems-idUSKBN0EL15E20140610

[2] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007.

[3] J. S. Pedro, S. Siersdorfer, and M. Sanderson, "Content redundancy in youtube and its application to video tagging," *ACM Transactions on Information Systems (TOIS)*, 2011.

[4] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011.

[5] M. Hefeeda, T. ElGamal, K. Calagari, and A. Abdelsadek, "Cloud-based multimedia content protection system," 2013.

[6] YouTube. Statistics. [Online]. Available: http://www.youtube.com/yt/press/statistics.html

[7] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua, "Real-time large scale near-duplicate web video retrieval," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010.

[8] Y. Chen, W. He, Y. Hua, and W. Wang, "Compoundeyes: Near-duplicate detection in large scale online video systems in the cloud," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, 2016.

[9] A. Amir, M. Berg, S.-F. Chang, W. Hsu, G. Iyengar, C.-Y. Lin, M. Naphade, A. Natsev, C. Neti, H. Nock *et al.*, "Ibm research trecvid-2003 video retrieval system," *NIST TRECVID-2003*, 2003.

[10] B. Thomee, M. J. Huiskes, E. Bakker, and M. S. Lew, "Large scale image copy detection evaluation," in *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, 2008.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2019.2913674, IEEE Transactions on Big Data

[11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International journal of computer vision*, 2005.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, 2004.

[13] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004.

[14] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003.

[15] O. Chum, J. Philbin, M. Isard, and A. Zisserman, "Scalable near identical image and shot detection," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007.

[16] X. Zhou, L. Chen, and X. Zhou, "Structure tensor series-based large scale near-duplicate video retrieval," *IEEE Transactions on multimedia*, 2012.

[17] C.-Y. Chiu, T.-H. Tsai, Y.-C. Liou, G.-W. Han, and H.-S. Chang, "Near-duplicate subsequence matching between the continuous stream and large video dataset," *IEEE Transactions on Multimedia*, 2014.

[18] C.-L. Chou, H.-T. Chen, and S.-Y. Lee, "Pattern-based near-duplicate video retrieval and localization on web-scale videos," *IEEE Transactions on Multimedia*, 2015.

[19] W.-L. Zhao and C.-W. Ngo, "Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection," *IEEE Transactions on Image Processing*, 2009.

[20] W.-L. Zhao, C.-W. Ngo, H.-K. Tan, and X. Wu, "Near-duplicate keyframe identification with interest point matching and pattern learning," *IEEE Transactions on Multimedia*, 2007.

[21] X. Zhou, X. Zhou, L. Chen, A. Bouguettaya, N. Xiao, and J. A. Taylor, "An efficient near-duplicate video shot detection method using shot-based interest points," *IEEE Transactions on Multimedia*, 2009.

[22] X. Zhou and L. Chen, "Monitoring near duplicates over video streams," in *Proceedings of the 18th ACM international conference on multimedia*, 2010.

[23] J. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo, "Effective multiple feature hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Multimedia*, 2013.

[24] F. Zou, Y. Chen, J. Song, K. Zhou, Y. Yang, and N. Sebe, "Compact image fingerprint via multiple kernel hashing," *IEEE Transactions on Multimedia*, 2015.

[25] Y. Chen, J. Bi, and J. Z. Wang, "Miles: Multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[26] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang, "Near-duplicate video retrieval: Current research and future trends," *ACM Computing Surveys (CSUR)*, 2013.

[27] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, ser. STOC '98, New York, NY, USA, 1998. [Online]. Available: http://doi.acm.org/10.1145/276698.276876

[28] R. Ji, H. Yao, W. Liu, X. Sun, and Q. Tian, "Task-dependent visual-codebook compression," *IEEE Transactions on Image Processing*, 2012.

[29] C. Domeniconi and B. Yan, "Nearest neighbor ensemble," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004.

[30] Y. Hua, B. Xiao, and X. Liu, "Nest: Locality-aware approximate query service for cloud computing," in *INFOCOM, 2013 Proceedings IEEE*, 2013.

[31] Z. Nie, Y. Hua, D. Feng, Q. Li, and Y. Sun, "Efficient storage support for real-time near-duplicate video retrieval," in *Algorithms and Architectures for Parallel Processing*, 2014.

[32] R. Pagh and F. F. Rodler, *Algorithms — ESA 2001: 9th Annual European Symposium Århus, Denmark, August 28–31, 2001 Proceedings*, Berlin, Heidelberg, 2001, ch. Cuckoo Hashing. [Online]. Available: http://dx.doi.org/10.1007/3-540-44676-1_10

[33] S. Tulyakov, S. Jaeger, V. Govindaraju, and D. Doermann, "Review of classifier combination methods," in *Machine Learning in Document Analysis and Recognition*, 2008.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Yixin Chen Yixin Chen received the B.E. degree in computer science from National University of Defense Technology, China, in 2009. He is currently a Ph.D. candidate in computer science from McGill University, Canada. His research interests include cloud computing, distributed computing, and multimedia big data systems.

Dongsheng Li Dongsheng Li received the B.Sc. degree (with honors) and Ph.D. degree (with honors) in computer science from College of Computer Science, National University of Defense Technology, Changsha, China, in 1999 and 2005, respectively. He was awarded the prize of National Excellent Doctoral Dissertation of PR China by Ministry of Education of China in 2008. He is now a full Professor at National Lab for Parallel and Distributed Processing, National University of Defense Technology, China. His research interests include distributed computing, Cloud computing, computer network and large-scale data management.

Yu Hua Yu Hua received the B.E. and Ph.D. degrees in computer science from Wuhan University, China, in 2001 and 2005, respectively. He is currently a professor at Huazhong University of Science and Technology, China. His research interests include computer architecture, cloud computing, and network storage. He has more than 100 papers to his credit in major journals and international conferences including the IEEE Transactions on Computers, the IEEE Transactions on Parallel and Distributed Systems, Proceedings of the IEEE, USENIX ATC, USENIX FAST, INFOCOM, SC, ICDCS, ICPP, MSST, and MASCOTS. He has been on the organizing and program committees of multiple international conferences, including USENIX ATC, INFOCOM, ICDCS, ICPP, ICNP, MSST, RTSS, and IWQoS. He is a senior member of the ACM, IEEE and CCF, a member of USENIX.

Wenbo He Wenbo He received the PhD degree from the Department of Computer Science at University of Illinois at Urbana-Champaign in 2008. She is currently an Associate Professor in the Department of Computing and Software at McMaster University, and Adjunct Professor in the School of Computer Science at McGill University. Her research focuses on Multimedia Big Data, Cloud Computing, Information Security and Privacy, and Pervasive Computing, etc.