ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Vigorous and Efficient Detection of Replicated Videos in Huge Databases

[1]V.V. Syam, [2]Ramakoteeswara Rao Badana, [3]Sudhir Kumar Raju Gadiraju
[1,2,3]Dept. of CSE, Al-Ameer College of Engg. and Information Technology, Visakhapatnam, AP, India

Abstarct
We present video fingerprints approach which is an efficient and accurate method for duplicate video detection in a large database. To create finger prints We consider a compact and robust frame based descriptor and the color layout descriptor, which are further encoded by Vector Quantization (VQ).We propose a new superior performance giving non metric distance measure to find the similarity between the query and a database video fingerprint for accurate duplicate detection. With existing indexing techniques Using a non metric distance measure Efficient search cannot be performed for high-dimensional data. Therefore, we develop novel search algorithms based on pre computed distances and new dataset pruning techniques yielding practical retrieval times. We perform experiments with a database of 48 000 videos, worth 1700 h of content. For individual queries with an average duration of 65 s (about 50% of the average database video length),the duplicate video is retrieved in 0.042 s, on Intel Xeon with CPU2.33 GHz, with a very high accuracy of 97.8%.

Keywords
Color Layout Descriptor (CLD), Duplicate Detection, Non Metric Distance, Vector Quantization (VQ), Video Fingerprinting

I. Introduction

Videos on commercial sites e.g., www.youtube.com and www.metacafe.com, are mainly textually tagged. These tags are of little help in monitoring the content and preventing copyright infringements. Approaches based on Content-Based Copy Detection (CBCD) and watermarking have been used to detect such infringements [1]. The watermarking approach tests for the presence of a certain watermark in a video to decide if it is copyrighted. The other approach, CBCD, finds the duplicate by comparing the fingerprint of the query video with the fingerprints of the copyrighted videos. A fingerprint is a compact signature of a video which is robust to the modifications of the individual frames and discriminative enough to distinguish between videos. The noise robustness of the watermarking schemes is not ensured in general [2], whereas the features used for fingerprinting generally ensure that the best match in the signature space remains mostly unchanged even after various noise attacks. Hence, the fingerprinting approach has been more successful. We define a "duplicate" video as the one consisting entirely of a subset of the frames in the original video—the individual frames may be further modified and their temporal order varied. The assumption that a duplicate video contains frames only from a single video has been used in various copy detection works.For some popular queries to the Yahoo video search engine, there were two or three duplicates among the top ten retrievals .The block diagram of our duplicate video detection system is shown in fig. 1. The relevant symbols are explained in Table 1. The database videos are referred to as "model" videos in the paper. Given a model video Vi, the decoded frames are sub-sampled at a factor of 5 to obtain Ti frames and a p dimensional feature is extracted per frame. Thus, a model video Vi is transformed into a Ti×p matrix Zi. We empirically observed in Section 3, that the Color Layout Descriptor (CLD) [3] achieved higher detection accuracy than other features. To summarize Zi, we perform k-means based clustering and store the cluster centroids {Xij}Fij=1 as its fingerprint. The number of clusters Fi is fixed at a certain fraction of Ti,e.g., a fingerprint size of 5× means that Fi = (5/100)Ti. Therefore, the fingerprint size varies with the video length.k-means based clustering produces compact video signatures which are comparable to those generated by sophisticated summarization techniques as discussed in [4]. In [2], we have compared different methods for key frame selection for creating the compact video signatures.
The duplicate detection task is to retrieve the best matching model video fingerprint for a given query fingerprint.
The computational cost for the retrieval of the best matching model video has two parts (fig. 1).

A . Online Cost

The query video is decoded, sub-sampled,Key frames are identified, and features are computed per key frame—these constitute the query preprocessing cost. The reported query time comprises of the time needed to obtain k-means based compact signatures, perform VQ-based encoding on them to obtain sparshistogram-based representations, compute the relevant lookup tables, and then perform two-stage search to return the best matched model video.

B. Offline Cost

Consists of the un-quantized model fingerprintgeneration, VQ design and encoding of the model signatures, and computation and storing of appropriate distance matrices.

II. Features of Candidate

We performed duplicate video detection with various frame based features: CLD, CFMT [5], LCH and EHD.The LCH feature divides the image into a certain number of blocks and the 3-D color histogram is computed per block. If the image is divided into two partitions along each direction, the total LCH feature dimension is 43×22 =256. To study the variation of detection accuracy with signature size, we have considered the LCH feature for dimensions 256 and 32 (22×23 = 32).We also considered global features computed over the entire video and not per key frame. One such global feature used is the m-dimensional histogram obtained by mapping each of the256-dimensional LCH vectors to one of m codebook vectors(this signature creation is proposed in , obtained after k means clustering of the LCH vectors. We have experimented with m = 20, 60, 256, and 8192, and L2 distance is used. The other global feature is based on a combination of the ordinal and color histograms Both the ordinal and color histograms are 72-dimensional (24 dimensions along each of the Y, Cb, and Cr channels) and the distance measure used is a linear combination of the average of the distance between the ordinal histograms and the minimum of the distance between the color histograms, among all the three channels.

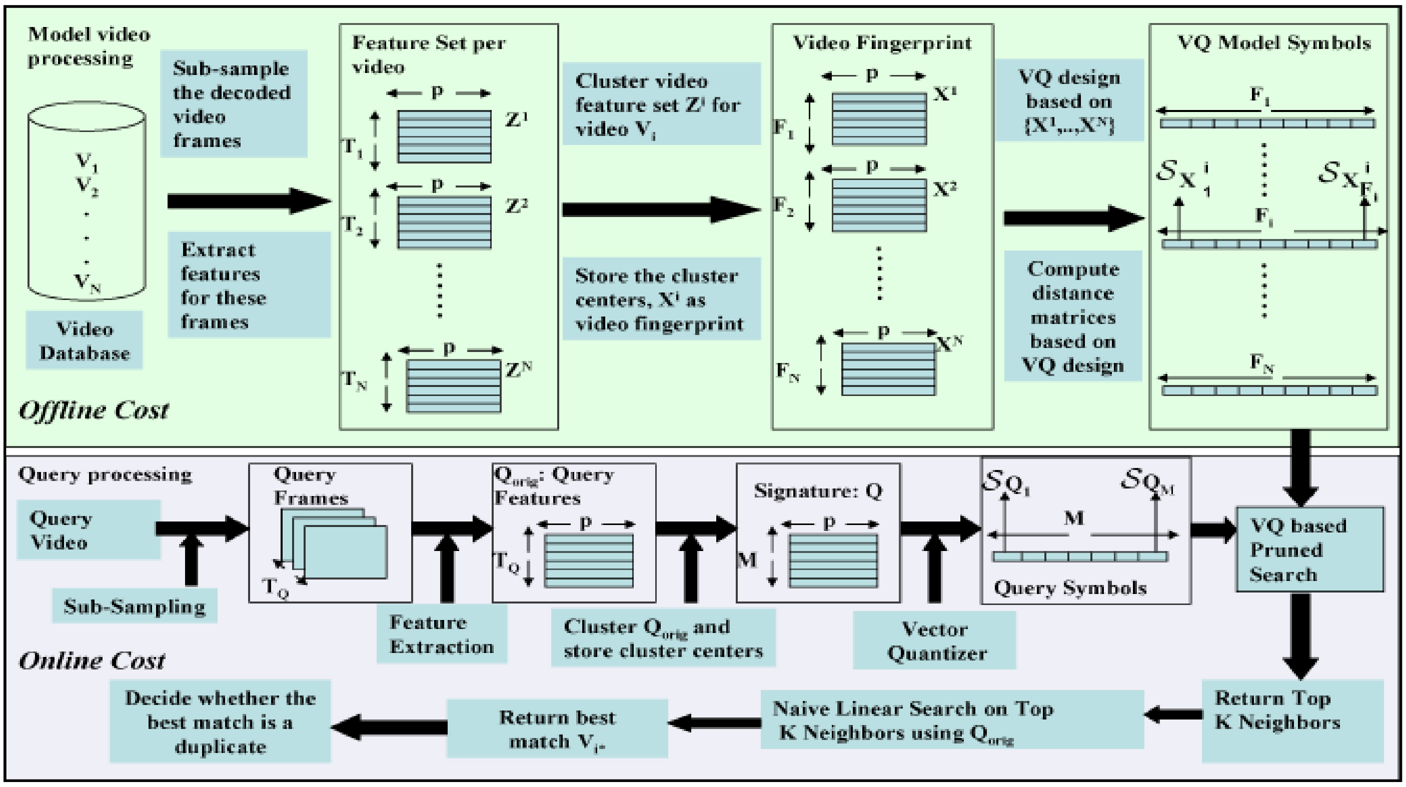
Model video processing
Video Database
Sub-sample the decoded video frames
Extract features for these frames
Feature Set per video
Cluster video feature set Zⁱ for video Vᵢ
Store the cluster centers, Xⁱ as video fingerprint
Video Fingerprint
VQ design based on {X¹,..,Xᴺ}
Compute distance matrices based on VQ design
VQ Model Symbols
Offline Cost
Query processing
Query Video
Sub-Sampling
Query Frames
Feature Extraction
Q_orig: Query Features
Cluster Q_orig and store cluster centers
Signature: Q
Vector Quantizer
Query Symbols
VQ based Pruned Search
Return Top K Neighbors
Naive Linear Search on Top K Neighbors using Q_orig
Return best match Vᵢ*
Decide whether the best match is a duplicate
Online Cost

Fig. 1: Block Diagram of the Proposed Duplicate Detection Framework. Symbols used are Explained in Table 1

Table 1: Glossary in This Paper

Notation	Definition	Notation	Definition
N	Number of database videos	Vi*	Best matched model video for a given query
p	Dimension of the feature vector computed per video frame	U	Size of the vector quantizer (VQ) codebook used to encode the model video and query video signatures
$Z^i \in R^{T_i * p}, T_i$	Z^i is the feature vector of vi	$X^i \in R^{F_i * p}, F$	Fi key frames, xi is k means signature of vi
c_i	Ith VQ code vector	$Q \cup R^{M*p}$	Query finger print
|E|	Cardinality set E	$\vec{x}_i, \vec{q}$	−→xi is VQ based signature of Vi , while −→q is VQ based query signature
$S_{x_j^i}$	X ij mapped symbol to VQ index	1	Factional query length
C(i)	I th dimension VQ based signature cluster	$D^* \in R^{N*U}$	Lookup distance matrix of shortest distance values from each model to each VQ code vector

III. Proposed Distance Measure

Our proposed distance measure to compare a model fingerprint Xi with the query signature Q is denoted by d(Xⁱ,Q) (1). This distance is the sum of the best-matching distance of each vector in Q with all the vectors in Xⁱ. In (1), kXij − Qkk1 refers to the L1 distance between Xij , the jth feature vector of Xi and Qk, the kth feature vector of Q. Note that d(·, ·) is a quasi distance.

$$d(X^i,Q) = \sum_{k=1}^{M} \{\min_{1\leq j\leq Fi} \|X^i_j - Q_K\|_1$$

What is the motivation behind this distance function? We assume that each query frame in a duplicate video is a tampered/processed version of a frame in the original model video. Therefore, the summation of the best-matching distance of each vector in Q with all the vectors in the signature for the original video (Xi) will yield a small distance. Hence, the model-to-query distance is small when the query is a (noisy) subset of the original model video. Also, this definition accounts for those cases where the duplicate consists of a reordering of scenes from the original video.A comparison of distance measures for video copy detection is presented in [7]. Our distance measure is similar to the Hausdorff distance, [7]. For our problem, the Hausdorff distance h(Xi,Q) andthe partial Hausdorff distance hP (Xi,Q) are interpreted as:

$$h(X^i,Q) = \max_{1\le k\le M}\left\{\min_{1\le j\le F_i}\|X^i_j - Q_k\|_1\right\}$$

$$h_P(X^i,Q) = \underbrace{P^{th}\text{largest}}_{1\le k\le M}\left\{\min_{1\le j\le F_i}\|X^i_j - Q_k\|_1\right\}$$

For image copy detection, the partial Hausdorff distance (3) has been shown to be more robust thanthe Hausdorff distance (2) in [18]. We compare the performance of hP (X^i,Q) (3) for varying P, with d(X^i,Q), as shown in fig. 3, using the same experimental setup as in Sec. III. It is seen that the results using d(X^i,Q) are better - the improved performance is more evident for shorter queries. Intuitively, why does our distance measure perform better than the Hausdorff distance? In (2) (or (3)),we first find the "minimum query frame-to-model video" distance for every query frame and then find the maximum (or Pth largest) among these distances. Thus, both h(X^i,Q) and hP (X^i,Q) effectively dependon a single query frame and model video frame, and errors occur when this query (or model) frameis not representative of the query (or model) video. In our distance function (1), d(X^i,Q) is computed considering all the "minimum query frame-to-model video" terms and hence, the effect of one (or more)mismatched query feature vector is compensated. Dynamic Time Warping (DTW) is commonly used to compare two sequences of arbitrary lengths. The proposed distance function has been compared to DTW in [6], where it is shown that DTW works well only when the query is a continuous portion of the model video and not a collection of disjoint parts. This is because DTW considers temporal constraints and must match every data point in both the sequences.

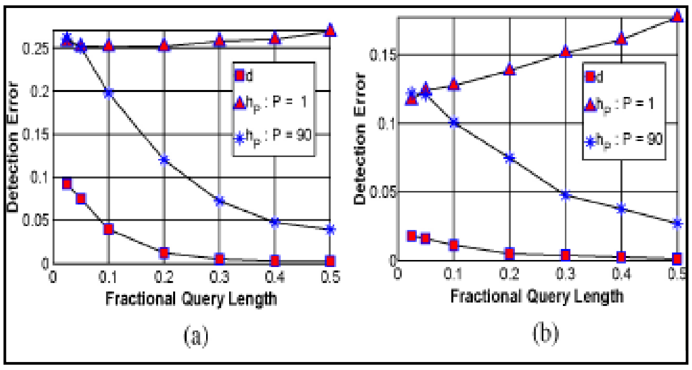

Fig. 3: Comparison of the Duplicate Video Detection Error for the Proposed Distance Measure d(·, ·) (1) and the Hausdorff Distances: Here, (hp : P =k) Refers to the Partial Hausdorff Distance (3) Where the kth Maximum is Considered

IV. Search Algorithms

We propose a two-phase approach for fast duplicate retrieval. The proposed distance measure (1) is used in our search algorithms for duplicate detection. First, we discuss a Naive Linear Search (NLS) algorithm in Section V-A. Search techniques based on the vector quantized representation of the fingerprints that achieve speedup through suitable lookup tables are discussed in Section V-B. Algorithms for further speedup based on dataset pruning are presented in SectionV-C. We sub-sample the query video along time to get a signature Qo rig having TQ vectors (see Fig. 1 and Table I). The initial coarse search (first pass) uses a smaller query signature Q, having M (M < TQ) vectors. Q consists of the cluster centroids obtained after k-means clustering on Qorig. When M = (5/100)TQ, we refer to the query fingerprint size as5×. The first pass returns the top-K NN from all the N model videos. The larger query signature (Q or ig) is used for the second pass to obtain the best matched video from these K candidates using a naive linear scan. As the query length decreases, the query key frames may differ significantly from the actual model video key frames; hence, the first pass needs to return more candidates to include the actual model video. A naive search method is to compute all the N model-to query distances and then find the best match. This set of N distances is denoted by A (4).We speedup the coarse search by removing various computation steps involved in A. To explain the speedup obtained by various algorithms, we provide the time complexity breakup in Table 2.

$$A = \{d(X^i,Q)\}_{i=1}^{N}$$

$$= \{\sum_{K=1}^{M} \min_{1\le j\le F_i}\|X^i_j - Q_k\|_1\}_{i=1}^{N}$$

A. NLS

The NLS algorithm implements the two-pass method without any pruning. In the first pass, it retrieves the top-K candidates based on the smaller query signature Q by performing a full dataset scan using an ascending priority queue L of length K. The priority queue is also used for the other coarse search algorithms in this section to keep track of the top-KNN candidates. The kth entry in L holds the model video index (Lk,1) and its distance from the query (Lk,2). A model signature is inserted into L if the size of L is less than K or its distance from the query is smaller than the largest distance in the queue. In the second pass, NLS computes the distance ofthe K candidates from the larger query signature Qorig so asto find the best matched candidate. The storage needed for allthe model signatures = $O(N\overline{F}p)$ where $\overline{F}$ denotes the average number of vectors in a model fingerprint.

B .VQ and Acceleration Techniques

From Table 2, it is observed that time T11 can be saved by pre computing the inter-vector distances. When the feature vectors are vector quantized, an inter-vector distance reduces to an inter-symbol distance, which is fixed once the VQ code vectors are fixed. Hence, we vector quantize the feature vectors and represent the signatures as histograms, whose bins are the VQ symbol indices. For a given VQ, we pre compute and store the inter-symbol distance matrix in memory. We now describe the VQ-based signature creation. Using the CLD features extracted from the database video frames, a VQ of size U is constructed using the Linde–Buzo–Gray algorithm [31]. The distance d(·, ·) (1) reduces to dVQM(·, ·)(5) for the VQ-based framework, where D is the inter-VQcode vector distance matrix (6). CSXij refers to the S Xijth code vector, i.e., the code vector to which the VQ maps

$$d_{VQM(X^i,Q)} = \sum_{k=1}^{M} \{\min_{1\le j\le F_i} \|C_{S_{x^i_j}} - C_{S_{Q_K}}\|_1\}$$

$$= \sum_{k=1}^{M} \{\min_{1\le j\le F_i} D(S_{x^i_j}, S_{Q_K})\}$$

Let $\bar{q}$ = [q1, q2, . . . , qU] denote the normalized histogrambased query signature (7) and →xi = [xi,1, xi,2, . . . , xi,U] denote the corresponding normalized model signature (8) for video
Viqk = |{j : SQj = k, 1 ≤ j ≤ M}|/M (7)
xi,k = |{j : SXij= k, 1 ≤ j ≤ Fi}|/Fi. (8)
Generally, consecutive video frames are similar; hence, many of them will get mapped to the same VQ codevector while many VQ codevectors may have no representatives (for a large enough U). Let {t1, t2, . . . , tNq } and {ni,1, ni,2, . . . , ni,Nxi}denote the

nonzero dimensions in $\bar{q}$ and $\bar{x}$, respectively, where Nq and Nxi denote the number of nonzero dimensions in $\bar{q}$ and $\bar{x}$, respectively. The distance between the VQ-based signatures −→xi and $\bar{q}$ can be expressed as

$$d_{VQ}(\vec{x_i}, \vec{q}) = \sum_{k=1}^{N_q} qtk.\{\min_{1\le j\le N_{x_i}} D(t_{k,n_{i,j}})\}$$

It can be shown that the distances in (5) and (9) are identical, apart from a constant factor
dVQM(Xi,Q) = M.dVQ($\bar{x}$,$\bar{q}$). (10)
The model-to-query distance (9) is same for different model videos if their VQ-based signatures have the same nonzero dimensions. For our database of 38 000 videos, the percentage of video pairs (among (38 0002) pairs) that have the same non zero indices is merely 3.2×10−4% [2]. A note about our VQ-based signature—since we discard the temporal information and are concerned with the relative frequency of occurrence of the various VQ symbols (one symbol per frame), the signature is similar to the "bag-of-words" model commonly used for textanalysis and computer vision applications [1].
The distance computation involves considering all possible pairs between the Nq nonzero query dimensions and the Nxinonzero model dimensions. We propose a technique where the distance computation can be discarded based on a partially computed (not all Nq.Nxi pairs are considered) distance—wecall it "partial distance-based pruning" (PDP) (Section V-B1).We then present two VQ-based techniques (VQLS-A in SectionV-B2 and VQLS-B in Section V-B3) which use different lookup tables, utilize PDP for faster search and significantly outperform NLS.

1. PDP

We present a technique, PDP, that reduces timeT3 (Table II) by computing only partially N model-to-querydistances in A. This speedup technique is generic enough to beused for both the un-quantized and the VQ-based signatures.We insert a new distance in the priority queue L if it is smallerthan the largest distance in the queue (LK,2). The logic behindPDP is that if the partially computed model-to-query distanceexceeds LK,2, the full distance computation is discarded for that model video.Let ˆd(Xi,Q, x') be the distance between Xi and the first x' vectors of Q—this is a partially computed model-to-query distance for x'< M. If ˆd(Xi,Q, x') exceeds LK,2, we discard the model video signature Xi as a potential top-KNN candidate and save time spent on computing d(Xi,Q),its total distance from the query. Though we spend additional time for comparison in each distance computation (comparing ˆd(Xi,Q, x') to LK,2), we get a substantial reduction in the search time as shown later in fig. 5(b).When PDP is used in the un-quantized feature space, we call that method as pruned linear search (PLS). The total storage space required for PLS is also O(NF¯ p), like NLS. Since we do not consider all the M vectors of Q in most of the distance computations, we have m ≤ M vectors participating, on an average, in the distance computation. Therefore, the time required to compute A, T3 (Table II) now reduces to O(mNF¯ p). The other computational costs are same as that for NLS.

$$d^{\wedge}(X^i,Q,K') = \sum_{k=1}^{k'} \{\min_{1\le j\le F_i} \|X^i_j - Q_k\|_1\}$$

Time	complexity	Operation involved
T11	O(p)	L1 distance between xij and Qk
T12=T11.Fi	O(FiP)	Finding the best matched model vector
T2=M.T12	O(MFip)	Best match of all m frames in Q
T3=$\sum_{i=1}^{N} T2$	o(MNFp)	Compute N model query distances
T4	O(NlogK)	Retrieve minimum k values from A to return top k videos
T5	O(TqKFP+K)	Finding Vi* from top K videos using query signature

2 .VQ-Based Linear Search-Method A (VQLS-A)

InVQLS-A, we pre compute the inter-VQ co devector distance matrix D (8) and store it in memory. We perform a full search on all the video signatures using dVQ($\bar{x}$,$\bar{q}$) (9) to find the top-K NN signatures—however, it directly looks upfor a distance between two VQ symbols in the matrix and hence saves time (T11 in Table II) by avoiding the L1 distance computation. This method also uses PDP for speedup. PDP in NLS implies searching along a lesser number of query frames. Here, it implies searching along a lesser number of nonzero query dimensions. Sorting of the nonzero dimensions of the query signature results in improved PDP-based speedup.
Let the average number of nonzero dimensions in the VQ-based model signatures be F, where F= (Ni=1 Nxi)/N. We need to encode Q before search which incurs a time of O(MU). Since this algorithm uses a constant time lookup of O(1), the complexity of T2 is reduced to O(NqF). The time T3 to compute all the N model-to-query distances, without PDP, is O(MU+NqNF).Using PDP, the average number of nonzero query dimensions considered reduces to Nq, where Nq < Nq. The corresponding reduced value of T3 is O(MU + NqNF). The time neededto sort the query dimensions is O(Nq logNq), which is small enough compared to $MU + N_q{'}NF'$
3) VQ-Based Linear Search-Method B (VQLS-B): This method obtains higher speedup than VQLS-A by directly looking up the distance of a query signature symbol to its nearest symbol in a

model video signature [e.g.,{min1≤j≤NxiD(tk, ni,j)} in (9)]. Thus, the computations involved in both T11 and T12 (Table 2) can be avoided, hence reducing the time to find a model-to-query distance to O(Nq).We pre compute a matrix D∗ ∊ RN×U where D∗(i, k) (15)denotes the minimum distance of a query vector, represented by symbol k after the VQ encoding, to the ith model VQLS-B differs from VQLS-A only in the faster distance computation using D∗ instead of D;

$$d_{VQ}(\vec{x_i}, \vec{q}) = \sum_{k=1}^{N_q} q_{t_k} \times \mathbb{D}^*(i, t_k)$$

$$\text{where } \mathbb{D}^*(i, t_k) = \min_{1 \le j \le N_{x_i}} \mathbb{D}(t_k, n_{i,j})$$

$$\text{for } 1 \le i \le N, \quad 1 \le k \le N_q, \text{ using (9).}$$

C. Dataset Pruning on Search Algorithm

The VQLS schemes described above consider all the N model videos to return the top-K NN videos. Further speedupis obtained by reducing the number of model videos accessed during the search. We present two dataset pruning methods forVQ-based signatures. The first method (VQ-M1) guarantees that the same top-K NN videos are returned even after pruning, as using naive linear search. The second method (VQ-M2)is an approximation of the first and achieves much higher pruning, though it is not guaranteed to return the correct top-K NN. The model-to-query distance (for the videos retained after pruning) can be computed using VQLS-A or VQLS-B(with PDP), for both VQ-M1 and VQ-M2.

1. Method VQ-M1

VQ-M1 uses a multi-pass approach for pruning. The logic is that for a given query, the model videos which are nearest to it are likely to have some or allof the nonzero dimensions, as the query signature itself, as nonzero.The pre computed information needed for VQ-M1 is listed as follows.

- We store a proximity matrix P ∊ RU×U which stores the U NN, in ascending order, for a certain VQ code vector,e.g., P(i, j) denotes the jth NN for the ith VQcodevector. For U = 8192(213), the storage cost of P = U2.13 bits (each of the U2 terms represents an integer ∊ [0, 213−1] and hence, is represented using 13bits, giving a total storage cost of 109 MB).
- We also maintain a distance matrix D ∊ RU×U which stores the NN distances, in ascending order, for each VQ code vector. Here, D(i, j) denotes the distance of the {P(i, j)}th code vector from the ith VQ code vector,i.e., D(i, j) = D(i, P(i, j)). We do not need to store D explicitly as it can be computed using D and P.
- We also store U clusters {C(i)}Ui=1, where C(i) denotes the cluster which contains those model video indices whose signatures have the ith dimension as nonzero. Thestorage cost for 8192 clusters containing 38 000 videos(the total model video dataset size for our experimentsas mentioned in Section VI-A) is found to be equal to6.3MB

  C(i) = {j : xj,i > 0, 1 ≤ j ≤ N}. (17)

We now provide a list of symbols used in VQ-M1 (Algorithm1) along with their definitions as follows.

- Sj : The set of distinct model videos considered in thejth pass.
- G: The set of nonzero query dimensions, where G ={t1, t2, . . . , tNq}.
- d∗j : The minimum of the distances of all nonzero querydimensions to their jth NN codevectors

$$d_j^* = \min_{t_K \in G} D'(t_k, j) \quad (18)$$

- Aj : The set of distinct VQ indices which are encountered on considering the first j NN for all the elements inG. Therefore, (Aj\Aj−1) denotes the set of distinct (notseen in earlier passes) VQ indices encountered in thejth pass, when we consider the jth NN of the elements in G

Algorithm 1 Algorithm for VQ-M1; here, unique(E) returns the unique (without repeats) elements in E

```
Require: N model video signatures, −→xi ∊ RU, 1 ≤ i ≤ N
Require: the query signature q, and lookup matrices P and D (along with the lookup tables needed by the distance computation method VQLS-A/B)
Ensure: Best sequence to search N videos for top-K NN and also top-K NN (model video indices)
1: Initialization: (1st pass)
2: G = {t1, t2, . . . , tNq}, the nonzero query dimensions
3: A1 = G, set of 1-NN of elements in G is G itself
4: S1 =1≤i≤NqC(ti), set of model videos having at least 1nonzero dimension from G
5: d∗1 = mintk∊G{D(tk, 1)} = 0
6: We maintain an ascending priority queue L of length K, based on the elements in S1, where dVQ(−→xi ,−→q ) is found using (9) or (14), depending on whether VQLS-A/B is being used.
7: End of 1st pass
8: for j = 2 to U do
9: d∗j = mintk∊G{D(tk, j)}, minimum distance between nonzero query dimensions to their jth NN
10: if LK,2 ≤ d∗j orjk=1|Sk| = N (all model videos have been considered) then
11: break;
12: end if
13: Bi = P(ti, j), 1 ≤ i ≤ Nq, B =set of VQ indices which are jth NN of elements in G
14: E = B\Aj−1, E = unique(E), set of VQ indices that are jth NN of elements in G and were not seen in earlier iterations
15: Sj =1≤i≤|E| C(Ei)
16: Sj = Sj \ 1≤i<jSi, set of all model videos having at least one element in E as a nonzero dimension and these videos were not seen in earlier iterations
17: Aj = Aj−1 ∪ E, set of all VQ indices which belong to one of the top j-NN for elements in G
18: Update the priority queue L based on the elements inSj
19: end for
20: return the sequences observed so far {S1, S2, . . . , SJ−1} (assuming that the search terminates at iteration j = J)and top-K NN from the priority queue L
```

2. Method VQ-M2

Based on empirical observations, we assume that the signature of the duplicate video, created from a subset of frames in the original video with noise attack son the frames, will have common nonzero dimensions with the original model video signature. Hence, the list of model videos considered for K-NN candidates corresponds to S1, the sequence of videos returned by the first iteration of VQ-M1.Thus, VQ-M2 is a single iteration process. This method introduces errors only if there is no overlap between the nonzero dimensions of the query and the original model video, i.e., if the best matched video index i∗/ ∊S1.

When the noise attacks introduce enough distortion in the feature vector space (so that nonzero query dimensions may not overlap with the nonzero dimensions of the original model signature), a simple extension is to consider P NN (P >1)across each nonzero query dimension. Thus, P should increase with the amount of distortion expected, and pruning gains decrease with increasing P. The number of videos in S1 and the storage cost for the proximity matrix P (defined for VQM1)depend on P. For P > 1, the storage cost for P is O(UP).The sequence S1, for P ≥ 1, is obtained as follows (forVQ-M1, S1 corresponds to P = 1):

S1 =1≤i≤NqC(ti), for P = 1

$S_1 = \bigcup_{1 \le i \le N_q} C(t_i)$ for p=1

$B = \bigcup_{1 \le i \le N_q, 1 \le k \le p} P(t_i, k)$ for $p \ge 1$

In fig. 4, we compare the dataset pruning obtained for different choices of P, fractional query lengths, and using different number of key frames for creating the query signatures. Using a higher fraction of query key frames (M/TQ),the pruning benefits are reduced, as more model videos are now considered due to the higher number of nonzero query dimensions. The percentage of videos retained after VQ-M2-based pruning is 3% and 7.5%, for 10% length queries, for P = 1 and P = 3, respectively. From Table V, the corresponding pruning obtained using VQ-M1 varies from21%-31% as K is varied from 10-100.

Table 3: Comparison of the Percentage of Model Videos Retained After Dataset Pruning for VQ-M1 With That Obtained Using DBH, for Different Fractional Query Lengths and K

Fractional query length	VQ-M1(k=10)	VQ-M1(k=100)	DBH(K=10)	DBH(K=100)
0.05	15.03	23.90	71.85	81.97
0.10	21.22	30.83	80.93	82.13
1.0	65.04	91.81	88.19	91.21

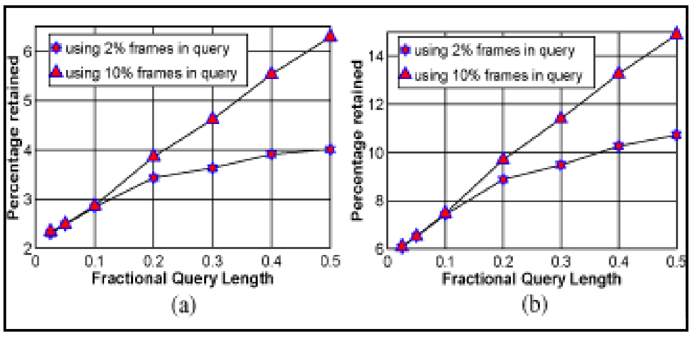

Fig. 4: Comparison of the fraction of model videos retained after VQ-M2-based pruning, for varying fractional query lengths, and using different sized query signatures. The number of cluster centers for the query is fixed at 2%and 10% of the number of query frames, after temporal sub-sampling, i.e.,M/TQ = 0.02 and 0.10 (notations as in Fig. 1) for the two cases. (a) Pruning using P = 1. (b) Pruning using P = 3.

V. Experimental Setup and Results

Section VI-A explains the dataset creation for duplicate detection using a variety of noise attacks. We also compare the duplicate detection accuracy over these attacks. Section VI-B presents the comparison of the different speedup techniques proposed for improving the coarse search.

A. Dataset Generation and Evaluation of Duplication Attacks

Two online video repositories www.metacafe.com andwww.youtube.com are crawled to obtain a database of 38 000model videos, worth about 1600 h of video content. A randomly chosen subset of 1300 videos (≈50 h of content),is used to generate the query videos. We perform various modifications on the decoded query frames for each of these1200 videos to generate 18 duplicates per video. We empirically o bserve that the CLD feature is robust to the discussed modifications. The number of duplicates for each noise classis shown in parentheses.

1. Gaussian blurring using a 3×3 and 5×5 window,
2. Resizing the image along each dimension by a factor of75% and 50%, respectively, (2).
3. Gamma correction by −20% and 20%,
4. Addition of additive white Gaussian noise (AWGN) using signal-to-noise ratio of −20 dB, 0 dB, 10 dB, 20 dB,30 dB, and 40 dB, (6).
5. JPEG compression at quality factors of 10, 30, 50, 70,and 90, (5).
6. Cropping the frames to 90% of their size .

The frame drops that are considered can be random or bursty. We simulate the frame drops by creating a query video as a fraction (2.5–50%) of the model video frames. The duplicate detection accuracy after the individual noise attacks is shown in Table 4.

Attack	Error	Attack	Error	Attack	Error	Attack	Error
Blur	0.0114	Resize	0.0111	Gamma	0.0221	AWGN	0.0113
JPEG	0.0125	Crop	0.0145	(Blur + crop)	0.0154	(Resize + crop)	0.0148
(Blur+resize)	0.0119	(AWGN + crop)	0.0156	(Gamma + crop)	0.0243	(AWGN + resize)	0.0128
MPEG-2	0.0098	MPEG-4	0.0088	WMV	0.0076	Gray-scale	0.0388
Logo (5%)	0.0140	Logo (10%)	0.1780	Logo (15%)	0.0198	Logo (20%)	0.0228
Caption (30%)	0.0155	Caption (50%)	0.0190	Caption (70%)	0.02301	Caption (90%)	0.0288

Table 4: Detection Error Obtained Using CLD Features, For Individual Noise Attacks ,Averaged Over Query Length From 2.5-50% ,And Over Varying Parameters for a given attack and they are converted toMPEG-1 format to generate the query videos.

We have also re-encoded the video using MPEG-2, MPEG-4, and Windows media video (WMV) formats. The CLD feature is robust against global attacks induced by strong AWGN and JPEG compression attacks and hence, robustness is expected against video re-encoding attacks—this is also experimentally verified. For MPEG-4 compressed videos, we experiment with varying frame rates (5 frames/s, 10 frames/s, 20 frames/s, 30 frames/s,40 frames/s, and 80 frames/s) and the average detection accuracy is 99.25%—the results remain almost constant for different frame rates.

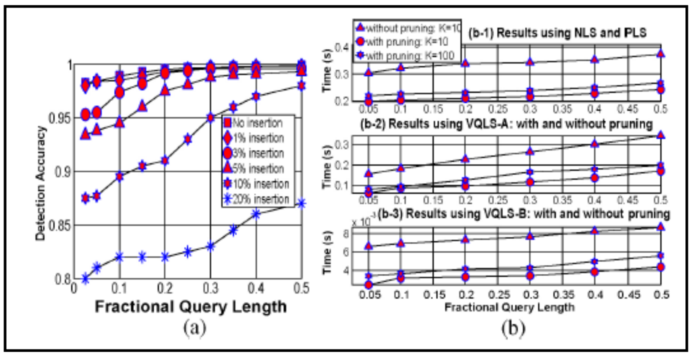

Fig. 5(a): Variation of the detection accuracy with varying levels of videoclip (from a different video) insertion—a fractional query length of 0.1 meansthat the query consists of 10% frames present in the (original query + insertedvideo clip). Model fingerprint size = 5×. (b) Runtime improvements due toPDP are shown for the PLS and VQ-based linear search schemes: (b-1) resultsusing NLS and

PLS, (b-2) results using VQLS-A—with and without pruning,and (b-3) results using VQLS-B—with and without pruning."Pruning/ nopruning" indicates whether or not PDP has been used. Here, runtime = (T3+T4)is the time needed to return the top-K model videos after the first pass.

B. Empirical Evaluation of Various Proposed Algorithms

We analyze the performance of the proposed algorithms for duplicate detection. The final detection accuracy, for a certain query length, is obtained by averaging over all the(1200 × 18) noisy queries, where the 18 duplication schemes were introduced in Section VI-A.

- Firstly, we show the speedup obtained using PDP, by comparing PLS (NLS + PDP) with NLS, and comparing VQLS-A and VQLS-B schemes, with and without PDP[Fig. 5(b)]. It is also seen that the VQ-based schemes significantly outperform NLS and PLS, that use unquantized features.
- Secondly, we show the performance improvements obtained using VQ-M1(A) and VQ-M2(A), in place of VQLS-A, and using VQ-M2(B) in place of VQLSB—these methods achieve additional speedup through dataset pruning [Fig. 6(a) and (b)].

1. Speedup Obtained Using PDP

We show the runtime needed (T3 + T4 from Table 2), with and without PDP for NLS, VQLS-A and VQLS-B schemes, in Fig. 5(b-1), (b-2)and (b-3), respectively, to return the top-K model videos. T3is reduced by using PDP. T4 = O(N logK) increases with K and thus, the effective runtime saving decreases as K increases. PDP provides significant runtime saving so that "with pruning:K = 100" takes lesser time than "without pruning: K = 10."Also, comparing (b-2) and (b-3) with (b-1) in Fig. 5, we observe that the runtime needed by VQLS-A and VQLS-B(with PDP) is much lower than that for PLS and NLS.

2. Speedup Obtained Through Dataset Pruning

We observe the runtime saving obtained through dataset pruning(using VQ-M1 and VQ-M2) using VQLS-A and VQLS-B for the model-to-query distance computation, in Fig. 6(a) and(b), respectively. PDP is employed for all the methods and "prune/ no prune" denotes whether or not we employ dataset pruning methods (VQ-M1 or VQ-M2).

- For VQLS-A, the runtime comparison for the different methods is: VQLS-A > VQ-M1(A) > VQ-M2(A). Hence, using dataset pruning results in significant speedup [Fig. 6(a)].
- For VQLS-B, the use of the lookup table D∗ reduces runtime significantly, so that the time required for the iterative pruning technique (VQ-M1) is higher than the runtime without pruning, especially for higher values of K and longer queries. Hence, for VQLS-B, for fractional query lengths exceeding 0.10, the runtime comparison for the various methods is: VQ-M1(B) > VQLS-B >VQ-M2(B) [Fig. 6(b)].

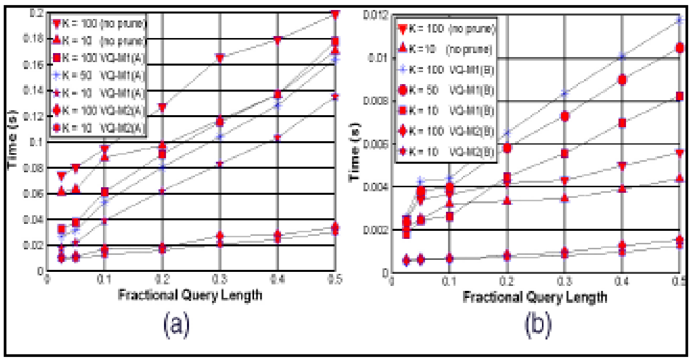

Fig. 6: Runtime improvements due to pruning in the model video space, for VQLS-A and VQLS-B, are shown. By "no prune," we mean that "pruning in model video space (VQ-M1 or VQ-M2) is absent, while PDP is used for all the methods." Significant runtime savings are obtained for VQ-M1(A) andVQ-M2(A) over VQLS-A and for VQ-M2(B) over VQLS-B. (a) Results with and without dataset pruning for VQLS-A. (b) Results with and without dataset pruning for VQLS-B.

References

[1] Joly, O. Buisson, C. Frelicot, "Statistical similarity search appliedto content-based video copy detection," in Proc. Int. Conf. Data Eng.Workshops, 2005, p. 1285.
[2] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, F. Stentiford, "Video copy detection: A comparativestudy," in Proc. Conf. Image Video Retrieval (CIVR), 2007, pp. 371–378
[3] E. Kasutani, A. Yamada,"The MPEG-7 color layout descriptor: Acompact image feature description for high-speed image/video segmentretrieval", in Proc. Int. Conf. Image Process. (ICIP), Vol. 1. 2001, pp. 674–677.
[4] P. Over, A. F. Smeaton, P. Kelly,"The TRECVID 2007 BBC rushessummarization evaluation pilot", Proc. Int. Workshop TRECVID Video Summarization (TVS), 2007, pp. 1–15.
[5] S. Derrode, F. Ghorbel,"Robust and efficient Fourier-Mellin transformapproximations for gray-level image reconstruction and completeinvariant description", Comput. Vision Image Understand., Vol. 83, No. 1, pp. 57–78, 2001
[6] A. Sarkar, V. Singh, P. Ghosh, B. S. Manjunath, A. Singh,"Duplicate video detection: Comparison of proposed distance functionwith dynamic time warping and dependence of detection accuracyon keyframe selection", Univ. California, Santa Barbara,CA, VRL Tech. Rep., May 2008
[7] A. Hampapur, R. M. Bolle,"Comparison of distance measures fovideo copy detection", in Proc. Int. Congr. Math. Educ. (ICME), Aug. 2001, pp. 737–740.
[8] D. N. Bhat, S. K. Nayar,"Ordinal measures for image correspondence", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 20, No. 4, pp. 415–423, Apr. 1998.
[9] L. Rabiner, B. H. Juang,"Fundamentals of Speech Recognition (Prentice Hall Signal Processing Series)", Englewood Cliffs, NJ: Prentice-Hall, 1993, sec. 4, pp. 200–240.

V.V. SYAM an Assistant professor in Al -Ameercollege of Engineering & Information Technology an affiliated college of JNTU, Kakinada, India over 5 years of work experience. He has done his B.Sc Degree in Computers from MIST Degree college affiliated to Andhra University.Visakhapatnam, India.He has done his Masters inM. Sc (Maths) from Andhra University Visakhapatnam and M.Tech (Cse) Nagarjuna University India.

RamakoteeswaraRao Badana an M.Tech (CSE) student of Al Ameercollege of Engineering &Information Technology, affiliated to JNTU, Kakinada , India. He has done B.Sc Degree in computers fromVidyardhi Degree college affiliated to Andhra University, Visakhapatnam, India.He has done her Masters in Master of Computer Applications (MCA) from Al-Ameer college of Engineering &TechnologyJNTU Kakinada,India.

Sudhir Kumar Raju Gadiraju an M.Tech (CSE) student of Al-Ameercollege of Engineering & Information Technology, affiliated to JNTU, Kakinada , India. He has done B.Tech in Information Technology from Swarnandhra College of Engineering to JNTU Hyderabad, India.