International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(9s) 2026 pp. 657-672
https://doi.org/10.70917/ijcisim-2026-3487
Article

Enhanced Video Shot Boundary Detection via KAZE Features Perceptual Hashing

Debjani Bhowmik[1]*, Saptarshi Chakraborty[1]

[1] Dept. of Computer Science and Engineering, ICFAI University Tripura, India, 799210.
Email: **debjani.ece@gmail.com**
[1*]ORCID: **0009-0009-9295-8622**
[1]ORCID: **0000-0002-8213-1615**

Abstract: In the era of multimedia proliferation, efficient video shot boundary detec-tion is crucial. This study introduces a robust method leveraging KAZE feature descriptors and perceptual image hashing for accurate detection. Addressing challenges in multimedia data indexing and retrieval, our approach generates unique hashes for video frames, enabling precise transition identification. Experimental results on stan-dard datasets demonstrate superior performance, with a 94.2% average accuracy in detecting abrupt transitions, showcasing the method's potential for multimedia sys-tems and digital communication applications.The proposed method is robust and discriminative, providing accurate results for short videos with less variation. The proposed method is robust and discriminative, providing accurate results for short videos. The efficiency and robustness of the technique are demonstrated through experiments on standard datasets such as TRECVid 2001 and TRECVid 2007. The RGB color histograms generated for detected cut transitions further validate the system's performance. The full source code of the proposed approach (DOI link: https://github.com/hritxx/kaze-feature-descriptor-perceptual-image-hashing) is uploaded to github, along with the associated data sets (readme files) to facilitate further research.
Keywords: NA

1. Introduction

In today's era, there is a huge growth of multimedia content like images, video, audio, and other digital content which are shared on the internet. Applications can be used on multimedia to change or manipulate their representation of data. This gives risto numerous threats to lawful contents. Problems can be created through multimedia data indexing and retrieval. It is essential to develop an efficient classification or index-ing system for multimedia data, which has a proper recovery system for the same. Using the hashing technique, a hash value is created where the input can be of vari-able length, and the output or generated hash value is of fixed size [1]. Compression and quantization techniques are generally used to decrease storage, upgrade the per-formance of the algorithm, and make the required time shorter for generating a hash value. Similarity measure and hash values are comparable. This can be determined through hash values whether visual contents of two images are similar or dissimilar.

A hashing algorithm is considered as satisfactory if the algorithm have the following outstanding properties: Discrimination and robustness. Sequence of generated hashes are similar and does not depend regardless the images are visibly identical. Sequence of hashes created by the algorithm for the same image can be fully different as applied key conditions are different [2]. If the value of the threshold is increased, robustness also increases. Increased robustness may reduce discrimination capability and vice versa. Threshold value should be balanced so that it can maintain the properties with discrimination capability and robustness [3]. Cryptographic hash functions differ from this hashing technique because they are responsive to minor changes in the digital data. Applications of perceptual image hashing are available in many areas like image indexing and image recovery, image validation, digital watermarking, and tamper detection, etc. [1]. A systematic and vigorous scheme of image hashing using the KAZE feature descriptor is proposed here, which has a discriminating property towards single and multiple sets of manipulations [3]. The method in the present work proposes using fixed key points using the KAZE feature descriptor. Key points are taken out from the image and features are created using these key points out of all fixed key points. The first key point is chosen to generate the value of the hash for the present function. Hamming distance is used to determine whether images are perceptible or not. The novel contribution of this paper is:

1. We propose an image hashing technique which can determine whether images retrieved from video are perceptible or not by applying the KAZE feature detector, which generates hashes for individual video frames.
2. The proposed technique is capable of classifying abrupt transition frames based on the Hamming distance between the hash values
3. The proposed technique is applied on a self created dataset to validate the proposed KAZE Hash-based shot boundary detection technique for short and less variation videos.

1.1 Perceptual Image Hashing

For many applications, it is necessary to detect changes made in multimedia data information in areas like image databases in the medical or artwork field, or in journalistic photography. Because of the perceptual image hashing technique and its robust algorithms, it is possible to check the confidentiality, authenticity, and

integrity of multimedia data. Perceptual image hashing is well accepted for its appli-cation in the area of medical imaging, multimedia systems, digital communication, and telemedicine. While encrypting multimedia data, a traditional cryptosystem is not a good solution because of the following reasons. The size of multimedia data is huge, and as a result, traditional cryptosystems require more time to convert into an encrypted format. Another issue is that multimedia data after decryption must be the same as the actual multimedia data. For video data or image, it is not necessary to keep the same data before and after encryption. In perceptual image hashing, multimedia data with minor distortion is also acceptable, as it is based on human perception. The watermarking field is also an application area of perceptual image hashing. A watermarking scheme is used to protect intellectual property rights, called a watermark, which holds copyright information in multimedia data. The watermark must be strong and resistant to security attacks so that it can extract and correctly identify the ownership of the host multimedia data if required [4]. For multimedia data security purposes, the technique that is introduced to authenticate data in image applications is known as perceptual image hashing or a perceptual hash function. The properties of perpetual hash functions are the following:

1. Robustness in perceptuality: When two images are perpetually similar, then they have almost similar hash values.

2. Capability of discrimination: When the visual contents of different images are dif-ferent and images are perpetually different, then these images will generate different hash values.

In last few years several techniques of perceptual image hashing was reported by var-ious researchers. Perceptual hashing has some limitations because of its properties. Minor accuracy issues are associated with perceptual hashing. A number of devel-oped image hashing algorithms are unable to handle major angle reaction as they work through blocks. Histogram based image hashing algorithms are also efficient and robust, but because of luminance, their differentiating capability is less [3]. In perpet-ual image hashing, some positives and negatives are falsely identified. It is difficult to accurately maintain the integrity of databases for videos and images. Perceptual hashing can be affected by cyberattacks. In perceptual hashing, the precise distribu-tion of images and videos is essential to prevent observation of harmful contents (false negatives) or false classification of the good contents (false positives). Several image hashing existing algorithms are developed to resist manipulation of contents and these are not robust for multiple set of alterations [2]. Extraction of features is an essential aspect for image hashing approaches that are used during authentication.

1.2 Perceptual image hashing Mathematical Definition

Perceptual hash function is of k bits and is represented by $PHF_K(.)$. It is a deter-ministic algorithm that can produce a perceptual hash [4].

$h = PHF_K(S) \epsilon \{0, 1\}^k$,where k is a string.

$d(., .) = [0, 1]$, where d is a string distance metric and threshold $\epsilon \leq 1$.
To compute similarity of two hashes threshold is used.

$$d(h_1, h_2) = \begin{cases} 0 : & h_1 \text{ and } h_2 \text{ are alike} \\ P\left[0 < d \leq \epsilon\right] \geq \partial_1 : & h_1 \text{ and } h_2 \text{ are alike} \\ P\left[d > \epsilon\right] \geq \delta_2 : & h_1 \text{ and } h_2 \text{ are unlike.} \end{cases} \quad (1)$$

Where δ_1 and δ_2 are considered probability thresholds. $1 - \delta_1$ and $1 - \delta_2$ are the probabilities of a false negative and false positive measure of similarity, respectively. When both the considered thresholds δ_1 and δ_2 100% then this particular hash function is a perfect perceptual hash function [5]. Whatever may be the suitable threshold accepted, it may differ context to context. Circumstances may appear where the benefit of decreasing the widespread presence of false positives to a lower level is considered to exceed the risk of expanding the widespread of false negatives or vice versa. Perceptual hash functions can be regulated by platform, to review the level of compatibility that is regarded as acceptable for a specific case.

1.3 Perceptual image hashing classification

Perceptual image hashing classification can be of the following four types.

1. Schemes based on statistics: Calculation of image statistics like mean, image blocks, variance, and histogram, etc., are based on extracted hash features.
2. Schemes based on Relation: In this type of approaches hash features are drawn out by using coefficient's invariant relationships in Discrete Wavelet Transform (DWT) or Discrete Cosine Transform (DCT).
3. Coarse representation Schemes: From an image, coarse information is generated, and perceptual hashes are created by using this information. Coarse information is like the spatial distribution of significant wavelet coefficients and low-frequency coefficients of the Fourier transform.
4. Low level features-based schemes: The salient image feature points are detected, and then hashes are extracted. On an image, these approaches first perform DWT or DCT transform to use the coefficients to produce the final value of the hash. These types of hash values are responsive to distortions, which can be local and global, that never cause any perceptual changes to the images.

1.4 Framework of Perceptual Image Hashing

The stages of perceptual image hashing are following

1. The transformation
2. Feature Extraction
3. Quantization Stage
4. Compression and Encryption

While transformation, first of all, spatial or frequency transformation is applied so that extracted features are based on the pixel values of the image, and also to relate to image frequency coefficients. While extracting features, the input image features are extracted by the system, and it produces Hash vectors in sequence. In the quan-tization stage, continuous hash vectors are quantized into a discrete vector of hash.

Then, discrete vectors of hashes are converted into perceptual hash string values of binary values. Binary hash string values of perceptual hash are undergone through a compression stage. The encryption technique is applied to convert the string into a final hash value [6].

1 Image Transformation stage: Dimension of the images are considered as x by y measured in bytes considered as input for the transformation stage. The image is undertaken for a spatial alteration like smoothing, color transformation, affine transformation, etc., and frequency alteration like Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT), etc. In this type of alteration, image pixel values determine the features of the extracted frames, or, in the case of frequency space, based on frequency coefficients of the image, the features of the extracted frames change.

2 Stage of Feature Extraction: In the stage of extracting features, image features are extracted by the system from the transformed image and produce the feature vector of L features where $L \ll X \times Y$.

3 Stage of Quantization: In the quantization phase, a hash vector is quantized and

4 used with *LP* byte elements. A consistent quantization is applied to each of the modules of the uninterrupted perceptual hash vector for quantization. Quantization with adaptive estimation is very much useful for image hashing.

5 Compression and encryption Stage: The compression and encryption stage is the last and final step in the perceptual image hashing system. Intermediate binary string of perceptual hash is converted into a short fixed sized perceptual hash of 1 byte through compression and encryption. It is the last and final Hash created that allow image authentication and verification at the other end i.e. receiver side. This final and last stage of the perceptual image hashing system can be guaranteed by a cryptographic hash function.

1.5 Metrics used for a perceptual image hashing:

Two types of perceptual hash functions are in use. With key value, perceptual hash functions produce a value of hash h, where x is considered as an input value of arbitrary length and k is a secret key. $h = (H(x; k))$. A perceptual hash function without a key produces a value of hash h from an input value of x and thus the hash function $H(x) = h$. Design of robust and efficient techniques for perceptual image hashing is very challenging because of several conflicting requirements [7]. If we consider P as probability and $H()$ as a perceptual hash function that generates a string of binary values of length l. If an image is considered as I, and the modified version of the image is considered as *I*alike, which is distinguishably similar to I. *I*unlike represents an image that is distinctly not similar to I. Hash values $h1$ represent the hash value of the original image I, and $h2$ represents the hash value of the perceptually non-similar image *I*unlike from I. Binary string of length consists of 0 and 1. Necessary properties of hashing functions for perceptually similar images can be summarized as follows [8].

1. Hash values: follow the equal distribution property.

1

$$P_r(H(I) = h_1) \approx \frac{1}{2^l}, \quad \in \{0, 1\}^1 \qquad (2)$$

r 1 2l 1

2. Image I and perceptually different image I_{unlike} are independent of each other.

$$P_r H(I) = \frac{h_1}{H(I_{\text{new}})} \; {}_2 \; {}_r \; {}_{\text{alike}} \; h \quad {}_1M \approx {}_1P_r\,{}_2(H(I \quad) = h\,),$$
$$\forall h\,,\, h \in \{0, 1\}^r \; (3)$$

3. Invariance property for Image I and a perceptually similar image I_{alike}.

$$P_r(H(I)) = (H(I_{\text{alike}})) \geq 1 - \theta_1, \textit{for a given } \theta_1 \approx 0 \quad (4)$$

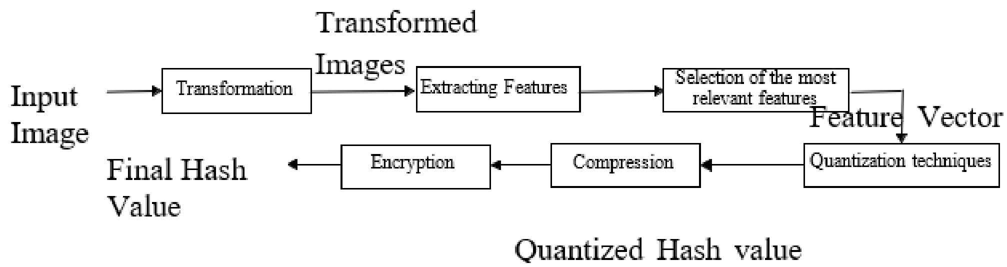
4. Distinction property of Image I and perceptually different Image

$I_u nlike.\ P_r(H(I)) \neq H(I_{\text{unlike}})) \geq 1 - \theta_2, \textit{for a given } \theta_2$

≈ 0

(5

)



For Eq. 3 in perceptual hash functions, features are extracted from images so that

they are invariant to global modifications, like enhancement and compression [4]. In Eq. 4, for any image I, it is not possible for an opponent to establish a perceptually non-identical image I, unlike such that $H(I) = H(I_{\text{unlike}})$ Perceptual Hash functions, which are published, are known to all, and this property is very difficult to achieve.

Input Image

Final Hash Value

Transformed Images

Quantized Hash value

Feature Vector

Fig. 1: Proposed approach to generate final hash value. From the Input image, relevant

features are extracted, and from the feature vector, the final Hash value is produced.

2. Related Work

In recent years, several experiments have been performed on perceptual image hashing. Some of the important, remarkable ones are discussed below. A hashing is proposed for data security in surveillance systems. Neelima et. al. proposed a robust methodology where SVD dimensions are reduced and hash values are generated with a reduced set of dimensions. The advantages of invariant features of SIFT are added with the feature points generated by SVD to get rotation, translation, and scaling [9]. Arambam et. al. suggested a Scale Invariant Feature Transform (SIFT) method where feature points are extracted from the image. The image is divided into a number of blocks, and from each block, values are extracted. The method with SIFT

gives a more robust result. But, when it is combined with Singular Value Decompo-sition (SVD), it becomes less efficient and requires more computational power [10]. In [11], a new methodology is stated through Zernike moments, which are capable of handling rotation attack for small rotation angles but not for large rotation angles. In

[12] Weber Local Binary Pattern (WLBP) is used for extracting the features where color texture and angles are combined as features. Surrounding pixels and center pixels are compared. The difference becomes less when using principal component analysis, and a histogram is generated. The method is applied to the images, and it is comparatively less robust. Gharde et. al. designed a robust perceptual hash function using the Fuzzy histogram. Hash values are derived from a standardized fuzzy his-togram. This method is robust and able to discriminate like human perception. The system can be Scale Invariant by normalizing the Fuzzy color histogram. Hash values are generated based on the normalized FUZZY color histogram [13]. A single image can be generated through the Canny operator, and after that, from each block, DCT coefficients are taken. Differentiation capability is minimum and thus not an efficient technique [6]. The Zernike moment is used for an image hashing technique. To rep-resent the features in an image, Zernike moments are robust and effective. Against content manipulations, this method is also effective and robust. This approach can handle attack upto 20º rotation [14]. DWT and local entropies are used in the image hashing method. Content manipulations are preserved, and then block entropies are changed linearly. From each of the block entropies, 2D discrete wavelet transforms are applied to generate a hash. The present image hashing technique based on DWT is systematic and vigorous toward various attacks to preserve contents. This image hashing can handle for minimal rotation [10].

A hashing method is generated to differentiate texture using grey-level co-occurrence matrix (GLCM). To eliminate discriminative limitation, cosine transform coefficients of local discrete type are used as parameters. Global features and local features are merged to produce the ultimate hash value [15]. Color vector is used to capture edge information. DWT is used to compress the color vector to generate a hash. The reported technique can work against a number of attacks, which can handle an attack of 5º angle rotation [16]. A technique based on the Laplacian pyramid is pro-posed. It works on the difference to generate multi scale parameters of an image. Fixed points are identified, and features are extracted. The technique is efficient and robust against a few attacks but cannot resist geometric attacks [17]. Tang et. al 2013, have suggested a new image hashing technique using different shape features. Shape-based features image hashing uses edge detection for capturing the angular transformation to generate a hash. The proposed technique has robustness toward various attacks except the angular rotation feature [18]. Random transform is used to produce a hash value. The overall system is efficient and vigorous in handling image retrieval and image attack with rotation.

3. Proposed video shot boundary detection:

In this section, a comprehensive description of the proposed system is given. Signif-icant locations are identified in the image that are prominent in terms of edges and texture. Key points are detected from the extracted frames. Descriptors around each key points represent those feature points. Feature points are extracted from the trans-formed image. Hash value is generated for individual video frames. Hamming distance between two consecutive KAZE hash value is computed. To compute the Hamming distance, the number of bits that differ between the two hashes is counted. It is used to find a similarity measure that can be used to identify sudden changes between two consecutive shots. Frames are classified and compared with the selected threshold to detect abrupt transitions based on the Hamming distance between two successive hashes. Fig. 2 represents the block diagram of the system.

3.1 Algorithm for Hash generation:

Extracted frames from video are converted to Grayscale image, which are considered as the input for the present algorithm. For all possible frames features are selected with 3 strongest metrics. Functions are applied to extract the local features of the selected frames. A set of collected features from the frames are represented as F. Mean value of the collected features are considered as threshold of the generated Hash of length 64 bits. Binarization is applied on the features to generate the corresponding Hash. The same is shown in Algorithm 1.

Algorithm 1 : Algorithm for Hash generation

```
Require: Input Gray Image, I
Ensure: Length of Hash Vector, hash
procedure HASH CALCULATION(I)
    for i = 1 to (met)n do
        S=(M,3);                    ▷ Feature with strongest 3 metrics
    end for
    for for i=1 to n do
        F=func(S,I);                ▷ func is a function to extract the local features
    end for
    features, F = feat1, feat2, ..., featn;
    x = ff1
    lmean=mean(x);
    for i = 1 to 64 do
        if x < lmean then
        else
            hashi = 1;
        end if
    end for
    return hash
end procedure
```

3.2 Proposed hash based abrupt transition detection system:

An effective KAZE hash algorithm has been proposed to generate the Hash. Extracted video frames are transformed to enable key points feature detection. A hash vector is created from the input gray images with feature metrics of the strongest three points.

Feature points are extracted using the KAZE feature extraction technique. Similarity measure is computed to detect an abrupt transition. Hashes generated from individual gray images are classified to detect abrupt transition based on the Hamming distance between two successive images. The below sub section gives a detailed explanation of each step.

3.2.1 Preprocessing:

Initially, input color frames are extracted from the video. Frames are resized and converted into a standard normalized image by $X \times X$ bilinear interpolation. The purpose of resizing the input frames is to generate hash values of the same length for frames of different dimensions. Then, RGB images are transformed into grayscale images to simplify processing. Grayscale images reduces computational complexity and preserves the crucial information required for feature detection.

3.2.2 KAZE Hash code Generation:

Key points from the images are found via the KAZE hash algorithm. Feature descrip-tors are calculated to detect changes across successive frames or images. KAZE enables the algorithm to discover similar characteristics in images under various sizes and ori-entations. Hamming distance is computed for successive hashes. Normalized hamming distance is utilized to compute the similarity measure by comparing consecutive vec-tors of two hashes $Hash_1$ and $Hash_2$. Difference between two consecutive hamming distance is calculated as

$$H_d = \frac{1}{L(y,2)} \sum_{i=1}^{L(y,2)} |(hash_1(i) - hash_2(i))| \quad (6)$$

$Hash_1(i)$ and $Hash_2(i)$ are the generated hashes for frame 1 and frame 2. To represent hash vector length is represented by L(y,2). Hamming distance is represented by H_d.

3.2.3 Transition Detection

Hamming distances are compared with the value of the threshold to determine if two consecutive frames are perceptually similar or dissimilar. Two consecutive frames that are perceptually dissimilar denote an abrupt transition. Hamming distance H_d is compared with the threshold value τ to identify the type of transition.

$$output = \begin{cases} gradual\ transition, & if\ H_d \leq \tau \\ abrupt\ transition, & otherwe \end{cases} \quad (7)$$

If the Hd is less than or equal to the adaptive threshold, then an abrupt transition is considered; otherwise, it is a gradual transition.

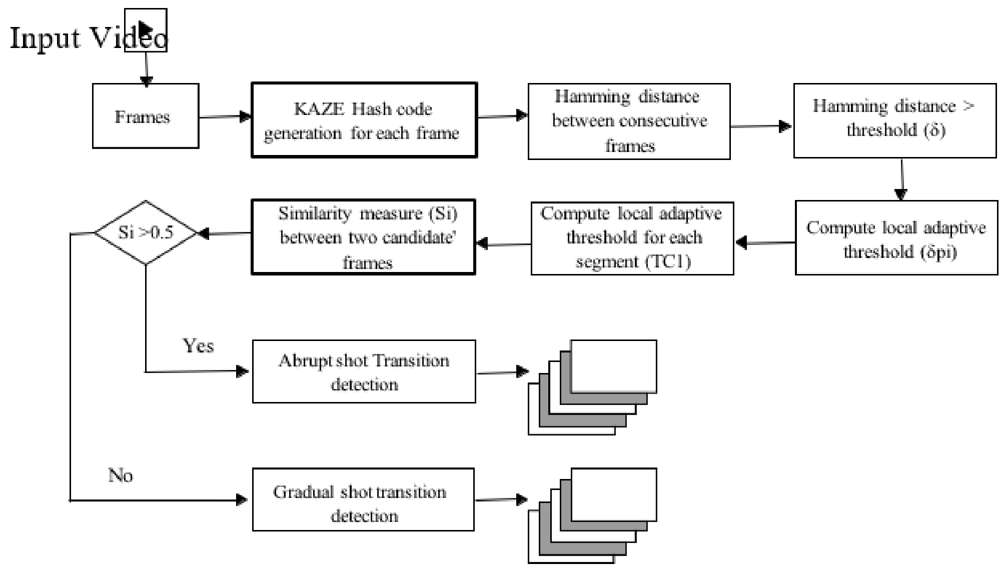


Fig. 2: Proposed Hash based video shot boundary detection system

4. Experimental Results and Discussion

This section discusses experimental results and the robustness of the suggested per-ceptual hash-based video shot boundary detection system. The standard databases TRECVid 2001 and TRECVid 2007 are considered for videos. Frames are extracted from the videos. Bilinear interpolation is applied to resize frames and to convert them into standard input images. A set of short video clips are also used for experiment purposes for validating the performance of the proposed system. The detailed source code of the proposed KAZE Hash algorithm is uploaded to GitHub, having a DOI link: https://github.com/hritxx/kaze-featur e-descriptor-perceptual-image-hashing along with the associated data sets, which will facilitate further research.

4.1 Perceptual Robustness

To show the robustness of the suggested system of video shot boundary detection, the proposed technique is applied to the extracted frames of the video. To validate the robustness of the proposed system python tools and libraries are used. An RGB color histogram is generated for the detected cut transition frames, which depicts dissimilarity between each of the consecutive frames. The results generated for the RGB color histogram from the video datasets TRECVid 2001 and TRECVid 2007 are shown in Fig. 4 and 8, respectively. Here, the x-axis denotes the similarity ranges, and the y-axis denotes the frequency count of the range. The distribution of the Hamming distance between hashes is represented. Standard deviation and variance of the normalized hamming distance also shows robustness and efficiency of the system. Standard deviation and variance for a set of detected cut transitions of the selected

Fig. 3: D5 video frames from TRECVid 2001 dataset is used to detect cut transition. KAZE Hash Algorithm is used to detect perceptually different frames and is detected as a cut transition.

video data subset TRECVid 2001 and TRECVid 2007 is shown in Fig.6 and Fig.10 respectively.

4.2 Discrimination

To ensure the discrimination capability of the proposed system, images are considered from the extracted frames of the video datasets TRECVid 2001 and TRECVid 2007. The considered images are of size 352×240 pixels. Images that are considered for this experiment are given in Fig. 3 and 7. To validate the discriminating capability, for each of the detected frames, a Hash value is generated. Similarity measure Hamming distance is computed with each other's Hash, which is shown in Fig. 5 and 9.

4.3 Performance Comparison

A few state-of-the-art techniques and associated algorithms are compared here to estimate the performance. Performance of SIFT (Scale Invariant Feature Transform), SVD (Singular Value Decomposition), LBP Histogram, Fuzzy based algorithms are compared with the proposed system. In SIFT, unique features or key points from an image are used to describe features or to detect features. SIFT is robust to rotation, scaling, and affine transformation of images. Based on the important, powerful local

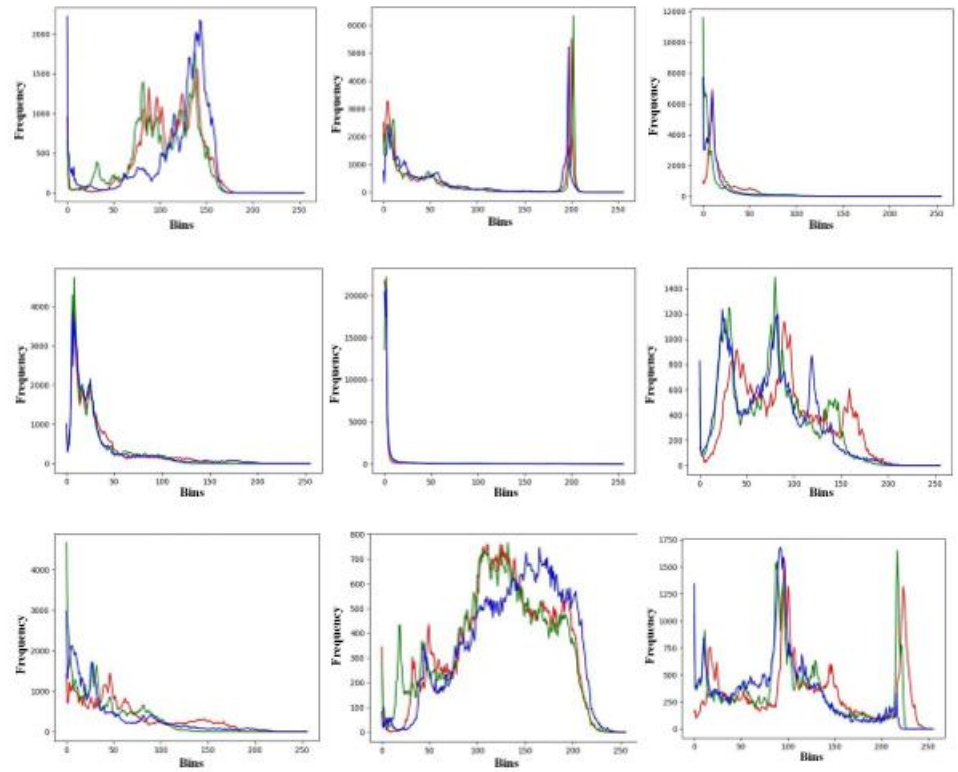


Fig. 4: RGB color histogram for detected frames. For the detected cut transition of D5 video frames of the TRECVid 2001 dataset, an RGB color histogram is generated. For perceptually different frames, a non-identical RGB color histogram is generated.

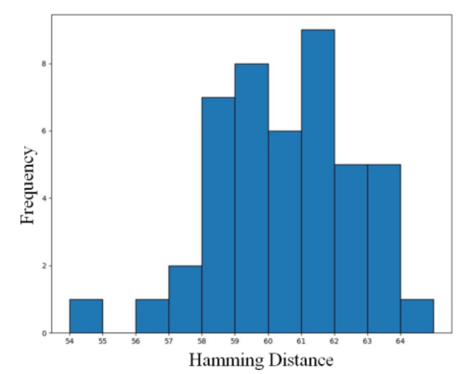
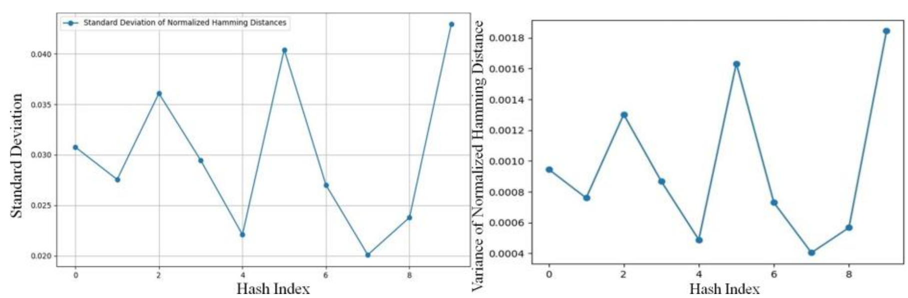


Fig. 5: Distribution of Hamming Distance between hashes generated from a set of detected cut transitions (D5 video of TRECVid 2001 dataset).



(a) standard deviation (b) Variation

Fig. 6: Standard deviation and Variance of Normalized Hamming Distance for a set of the detected cut transitions (D5 video of TRECVid 2001 dataset).

Fig. 7: *BG*37770*.mpg* video frames from TRECVid 2007 dataset is used to detect cut transition. KAZE Hash Algorithm is used to detect perceptually different frames and is detected as a cut transition.

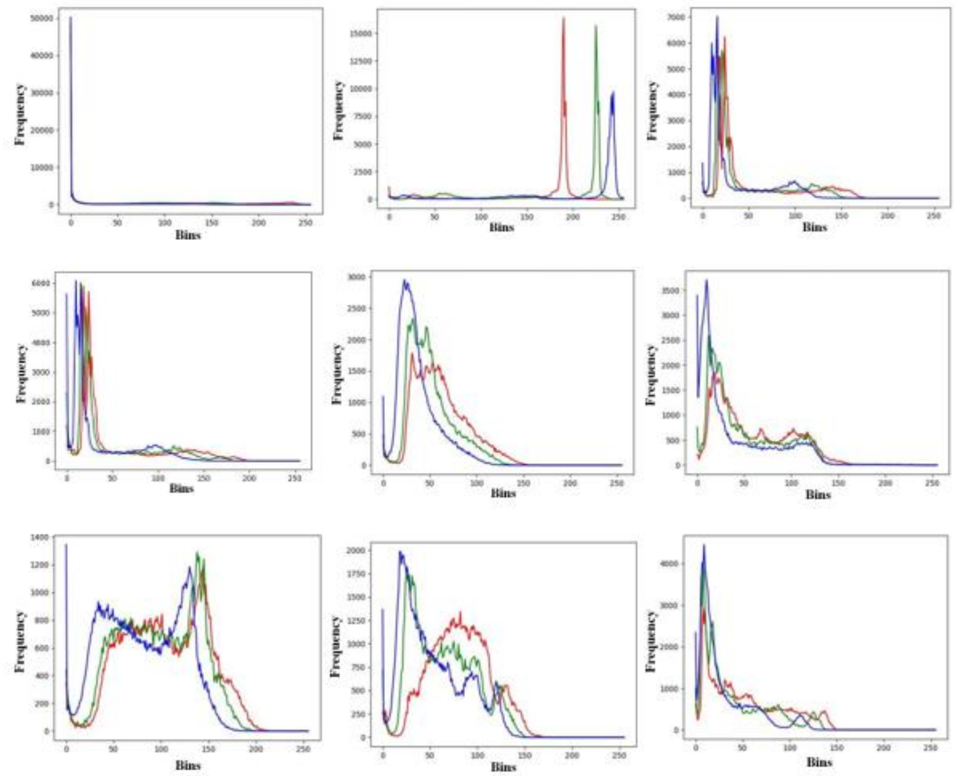


Fig. 8: RGB color histogram for a detected cut transition. For a set of detected cut transitions of *BG*37770.*mpg* video from the TRECVid 2007 dataset, a RGB color histogram is generated. For perceptually different frames, a non-identical RGB color histogram is generated.

extreme features, key points can be recognized through SIFT. SIFT is invariant to the orientation and size of the image. SVD is used to extract features, reduce noise, compress data, and reduce dimensionality in the field of machine learning. One matrix can be presented in terms of the successive multiplication of 3 special matrices. LBP histogram can be represented in matrix format,where pixels contain colour ranges from 0 to 255. Binary conditions are applied on neighborhood pixels to produce a binary pattern, which is then transformed into histograms to represent the image. A fuzzy-based algorithm follows fuzzy logic to find a solution to a problem. Fuzzy-based algorithms can generate correct results even when the data is not correct. For an unbiased comparison between the performance of the algorithms, a few of the detected abrupt transition frames of the TRECVid 2001 dataset and the TRECVid 2007 dataset, as mentioned in the Fig. 3 and 7, are used.

The result is manifested here in Fig. 11, showing the performance variation of all the algorithms, where for both the graphs the x-axis represents the state-of-the-art techniques applied on a set of cut transition frames, which can also be termed as

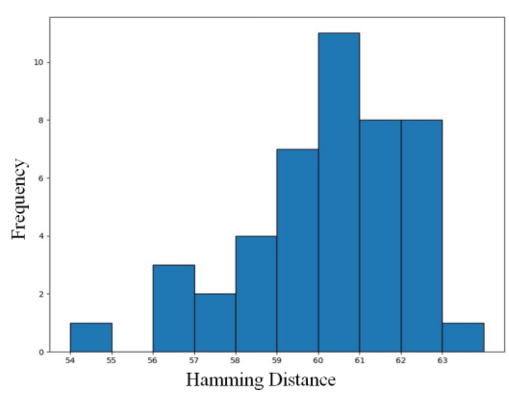
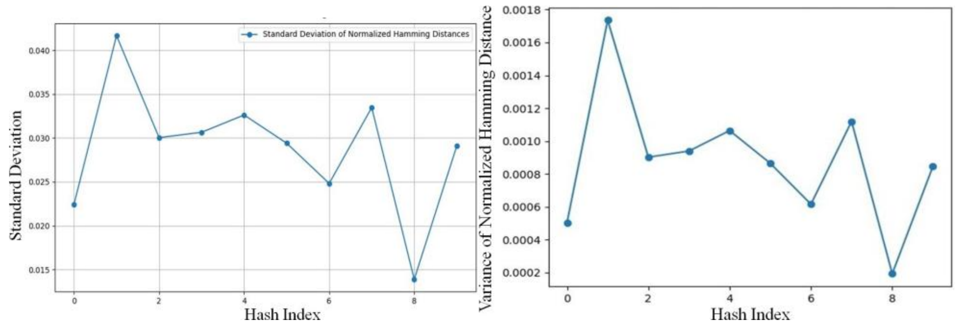


Fig. 9: Distribution of Hamming Distance between hashes generated from a set of detected cut transitions (*BG*37770.*mpg* video of TRECVid 2007 dataset).



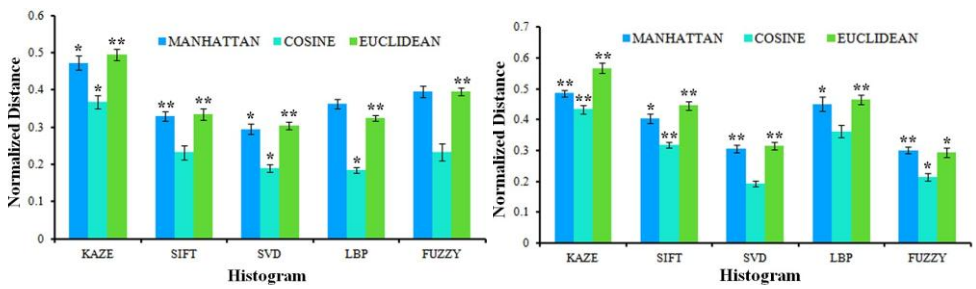
(a) Standard deviation (b) Variation

Fig. 10: Standard deviation and Variance of Normalized Hamming Distance for a set of the detected cut transitions (*BG*37770.*mpg* video of TRECVid 2007 dataset).

derived datasets, and the y-axis represents average Normalized Hamming Distance between consecutive cut transitions of the derived datasets. In both cases, our proposed technique shows better results compared to other state-of-the-art techniques.

4.4 Performance of video shot boundary detection

Being a quick-to-respond Hash-based algorithm, KAZE hash algorithm can be effective to use on short videos with less variation. Performance is computed for a dataset comprises of number of short videos. Details of the dataset is displayed in Table 1. Overall performance of the proposed system for the dataset is shown in Table 2. All the videos in the dataset with less variation shows impressive performance.



(a) Derived datasets from TRECVid 2001 (b) Derived datasets from TRECVid 2007

Fig. 11: Performance comparison of the proposed system with other state-of-the-art techniques.

Table 1: Video dataset applied for observation pur-poses

Videos	Frame number	Cut	Gradual	Total
Clip1	5359	5	35	40
Clip2	7238	11	82	93
Clip3	5980	9	26	35
Clip4	2561	2	16	18
Clip5	19410	6	127	133
Clip6	3727	5	22	27
Clip7	10055	22	109	131

Clip8	15479	9	129	138

Table 2: Overall system performance for a dataset used to validate the system's performance.

Videos	Cut			Gradual			Overall		
	Rec	**Pre**	**F1**	**Rec**	**Pre**	**F1**	**Rec**	**Pre**	**F1**
Clip1	80	100	88.88	100	90.47	94.99	90	95.23	91.93
Clip2	100	86.66	92.85	100	94.56	97.20	100	90.61	95.02
Clip3	88.88	100	94.11	100	88.23	93.74	94.44	94.11	93.92
Clip4	100	100	100	100	80.76	89.35	100	90.38	94.67
Clip5	100	88.88	94.11	100	97.67	98.82	100	93.27	96.46
Clip6	100	83.33	90.92	100	89.28	94.33	100	86.30	92.62
Clip7	86.36	100	92.68	95.41	100	97.65	90.88	100	95.16
Clip8	100	80	88.88	98.40	100	99.19	99.2	90	94.03
Average	94.40	92.35	92.80	99.22	92.62	95.65	96.81	92.48	94.22

5. Conclusion

In this paper, a robust feature point descriptor is applied to the extracted frames of a video. Standard datasets, TRECVid 2001 and TRECVid 2007, are used for the experiment. Feature descriptor KAZE is used for each frame, and the three strongest feature point metrics are extracted from the frame. Intensity values are transformed

into binary values using these frame features, and then a hash value is generated. The proposed algorithm and method generate different hash values for each of the detected cuts. To show the robustness of the proposed shot boundary detection system, RGB color histograms are created for each of the detected cuts. The distribution of Hamming distance for a few of the cut transitions is presented here. System performance is depicted after applying the Standard Deviation of the Normalized Hamming Dis-tance for each of the detected frames. After analysing the performance of the KAZE Hash-based abrupt transition detection system it became clear that, the approach is efficient to use on short videos with less variation. The approach is applied on a dataset containing a set of short videos. This technique shows robustness and good discrimination capability for detecting video shots.

Acknowledgements. Authors would like to acknowledge Dept. of CSE, ICFAI Uni-versity Tripura for extending their software and hardware support to complete the present work. Authors would also like to thank Dr. Dalton Meitei and his lab, Dept. of CSE, NIT Silchar for extending their technology support to complete the work.

Declarations

- Funding: Not Applicable
- Conflict of interest/Competing interests: The authors declare that they have no conflicts of interest.
- Ethics approval and consent to participate: Approved
- Consent for publication: Authors have no competing interest and are interested to publish this article in your esteemed journal.
- Data availability https://trecvid.nist.gov/
- Materials availability: Not Applicable

- Code availability: The full source code of the proposed approach (DOI link: https://github.com/hritxx/kaze-feature-descriptor-perceptual-image-hashing) is uploaded to github, along with the associated data sets (readme files) to facilitate further research.

- Author contribution: DB has designed experimental set up, performed all exper-iments, analyzed all observed data, written and edited the manuscript. SC has designed experimental set up, contributed in editing the manuscript. HR and AR has contributed in performing experiments. MCD has contributed in image selection and edited the manuscript.

References

1. Roy M, Thounaojam D M, Pal S, Perceptual hashing scheme using KAZE feature descriptors for combinatorial manipulations, Multimedia Tools and Applications (2022) 81:29045–29073.
2. Kr. Biswas S, Gharde1 N D, Thounaojam D M, Soni1 B, Robust perceptual image hashing using fuzzy color Histogram. Multimedia Tools Appl (2018).
3. Sajjad M, Haq IU, Lloret J, Ding W, Muhammad K (2019) Robust image hashing based efficient authentication for smart industrial environment. IEEE Trans Ind Inf 15(12):6541–6550.
4. Neelima A, Singh KM (2016) Perceptual hash function based on scale-invariant feature transform and singular value decomposition. Comput J 59(9):1275–128.
5. Arambam N, Singh KM (2016) Perceptual hash function based on scale-invariant feature transform and singular value decomposition. Comput J 59:1275–1281.
6. Singh KM, Neelima A, Tuithung T, Singh KM (2019) Robust perceptual image hashing using sift and svd. Curr Sci 117(8):1340.
7. Tang Z, Dai Y, Zhang X, Huang L, Yang F (2014) Robust image hashing via colour vector angles and discrete wavelet transform. IET Image Process 8:3:142–149.
8. Tang Z, Dai Y, Zhang X, Huang L, Yang F (2014) Robust image hashing via colour vector angles and discrete wavelet transform. IET Image Process 8:3:142–149.
9. Zhao Y, Wang S, Zhang X, Yao H (2013) Robust hashing for image authentication using Zernike moments and local features. IEEE Trans Inf ForensSecur 8:1:55–63.
10. Tang ZJ, Zhang XQ, Dai YM, Lan WW (2013) Perceptual image hashing using local entropies and DWT. Imag Sci J 61:2:241–251.
11. Huang Z, Liu S (2020) Perceptual image hashing with texture and invariant vector distance for copy detection. IEEE Transactions on Multimedia.
12. Qin C, Hu Y, Yao H, Duan X, Gao L (2019) Perceptual image hashing based on weber local binary pattern and color angle representation. IEEE Access 7:45460–45471.
13. Gharde ND, Thounaojam DM, Soni B, Biswas SK (2018) Robust per-ceptual image hashing using fuzzy color histogram. Multimed Tools Appl 77(23):30815–30840.
14. Hamid H, Ahmed F, Ahmad J (2020) Robust image hashing scheme using laplacian pyramids. ComputElectr Eng 84:106648.
15. Fridrich J, Goljan M (2000) Robust hash functions for digital watermarking. In: Proc Int Conf. information technology: coding and computing, pp 178–183.
16. Hernandez, R.A.P., Miyataki, M.N. and Kurkoski, B.M. (2011) Robust Image Hashing using Image Normalization and SVD Decomposition. Proc. IEEE Int. Symp. Circuits and Systems (MWSCAS), Seoul, August 7–10, pp. 1–4. IEEE.
17. Jie Z (2013) A novel Block-DCT and PCA based image perceptual hashing algorithm. IJCSI Int J Comput Sci 10:399–403.
18. Tang Z, Wang S, Zhang X, Wei W, Su S (2008) Robust image hashing for tamper detection using non-negative matrix factorization. J Ubiquitous Converg Technol 2:18–26.
19. Zizhuo, W. A. N. G., et al. ”Robust blind image watermarking based on interest points.” Virtual Reality & Intelligent Hardware 6.4 (2024): 308-322.
20. Lin, Chuchao, Changjun Zou, and Hangbin Xu. ”SCNet: A Dual-Branch Network for Strong Noisy Image Denoising Based on Swin Transformer and ConvNeXt.” Computer Animation and Virtual Worlds 36.3 (2025): e70030.
21. FFFN: Frame-by-frame feedback fusion network for video super-resolution. IEEE Transactions on Multimedia 25 (2022): 6821-6835.
22. Tang Z, Zhang X, Dai X, Yang J, Wu T (2013) Robust image hash function using local color features. Int J Electron Common (AEU) 67:717–722.
23. Tang Z, Yang F, Huang L, Zhang X (2014) Robust image hashing with dominant DCT Co-efficients. Optik-Int J Light Electron Opt 125:5102–5107.
24. Xiang S, Kim HJ, Huang J (2007) Histogram-based image hashing scheme robust against geometric deformations. In: Proc Int conf multimedia & security. Dallas, pp 121–128.
25. Jie, Z. (2013) A novel block DCT and PCA based image perceptual hashing algorithm. Int J. Comput. Sci., 10, 399–403.

26. Mohtavipour, S.M., Saeidi, M. & Arabsorkhi, A. A multi-stream CNN for deep violence detection in video sequences using handcrafted features. The Vis Comput 38, 2057–2072 (2022). https://doi.org/10.1007/s00371-021-02266-4.
27. Nozaripour, A., Soltanizadeh, H. Image classification via convolutional sparse coding. Vis Comput 39, 1731–1744 (2023). https://doi.org/10.1007/s00371-022-
28. 02441-1.
29. Kozat, S., Venkatesan, R., and Mihcak, M. Robust perceptual image hashing via matrix invariants. In Image Processing, 2004. ICIP '04. 2004 International Conference on (2004), vol. 5, pp. 3443 - 3446 Vol. 5.
30. Monga, V., and Evans, B. Perceptual image hashing via feature points: Perfor-mance evaluation and tradeoffs. Image Processing, IEEE Transactions on 15, 11 (Nov. 2006), 3452 -3465.
31. Tang Z, Wang S, Zhang X, Wei W, Su S. Robust image hashing for tamper detec-tion using nonnegative matrix factorization. J Ubiquitous Convergence Technol
32. 2008; 2:18–26.
33. Smith, J.R., "Color for Image Retrieval," Image Databases: Search and Retrieval of Digital Imagery, Wiley, pp.285-311, 2001.
34. Weng L, Preneel B (2009) Shape-based features for image hashing. Int Conf Multimed Expo, 1074–1077.
35. Lefebvre F, Macq B, Legat J-D (2002) Rash: Radon soft hash algorithm. In: European Signal Processing Conference. IEEE, pp 1–4.
36. Ahmed F, Mohammed YS, Abbas VU (2010) A secure and robust hash-based scheme for image authentication. Signal Process 90:1456–1470.
37. Choi YS, Park JH (2012) Image hash generation method using hierarchical histogram. Multimed Tools Appl 61:181–194.
38. K¨uc¸ ¨uktunc¸ O,G¨ud¨ukbay U, Ulusoy ¨o (2010) Fuzzy color histogram-based video segmentation. Comput Vis Image Underst 114:25–134.
39. Konstantinidis, K., Gasteratos, A., Andreadis, I., "Image Retrieval Based on Fuzzy Color Histogram Processing", Optics Communications, vol.248, Issues 4-6, pp.375-386, 2005.
40. Kucuktunc O., Zamalieva D. (2009) Fuzzy color histogram-based CBIR system. In: Proc Int fuzzy systems, symposium, pp 231–234.