A Novel Approach to Key Frame Detection Using Histogram and Dynamic Clustering

Neha Katre[1*], Meera Narvekar[2], Chirag Jagad[3], Chirag Jain[4], Ishika Chokshi[5]

[1,2,3,4,5]Dwarkadas J. Sanghvi College of Engineering, Mumbai, 400057, Maharashtra, India.
Email: neha.mendjoge@djsce.ac.in
*Corresponding Author

Received: 14.04.2024 Revised: 11.05.2024 Accepted: 26.05.2024

ABSTRACT
Video analysis is the ability to automatically analyse a video for its temporal and spatial events. Videos consist of frames. Thus, to analyse a video, each and every frame of the video needs to be analysed. High-performance computing is needed for this. Keyframes can be extracted from the videos and used to lessen the computational load. Keyframes are the exemplary frames that contain the significant data needed for analysis. This study suggests a method for identifying the keyframes using the histogram of the frames and dynamic clustering. An average of 97**%** decrease is obtained in the number of frames required for further analysis.

Keywords: key frame detection, object detection, principal component analysis, YOLO

1. INTRODUCTION

Video data has grown to be of the utmost importance due to advancements in network architecture, abundant storage, and the ubiquitous accessibility of digital cameras. As the principal method of information delivery, vibrant digital video has gradually takenthe role of dry printed material. If abnormalities are discovered in surveillance videos, the concerned authorities would be able to identify unexpected events without the need for human intervention. The practice of automatically reviewing videos to find any events occurring over the duration of a continuous stream of video frames is called video analytics. A crucial component of video analytics is the capability to automatically detect spatial and temporal events in videos and produce significant insights. Detecting anomalies from videos in real-time can greatly increase the computation time as all the video frames need to be passed to the object detection model to identify the anomaly. Thus, there is a need to identify those frames that can represent the video's salient content and information. Such frames are known as keyframes. Keyframes are considered as a set of frames that can describe the features of the entire video and portray the prominent content and information of the same with a minimal number of frames. Instead of analysing contents from all the frames of the video, only the key frame images are analysed. Since the video is compressed into a lesser number of frames, the time and resources required for processing these frames are reduced.

2. LITERATURE REVIEW

There are different types of techniques and algorithms available to extract key-frames from videos. These techniques are majorly divided into four categories namely – Clustering based, Motion-analysis based, Shot-Boundary based, and Deep Learning based[17].

2.1 Clustering Based Methods

In [1], clustering based on density peak is used by the authors as an approach for video key frame extraction that employs the HSV histogram to turn high dimensional abstract video image content into a quantifiable two-dimensional input matrix thus minimizing the computational complexity. This low-dimensional data is clustered using the density peak clustering algorithm, to find the cluster centers. These results are combined to get a different number of key frames thereby overcoming the traditional technique's drawback of extracting a fixed number of key frames. This technique can effectively integrate the properties of video footage while extracting crucial frames. The extracted key frames can better reflect the primary information of a video, have minimal redundancy, are noise resistant, and can create clusters of any shape without the need to artificially build up the beginning parameters. A fuzzy C-means clustering-based improved key-frame extraction approach has been put forth in [2]. The shots are divided into many sub-shots by first utilizing the color feature information contained in the video frames. Next,

the video sequence clustering technique is applied to obtain the center value of different classes and the membership degree of each frame in relation to the classes. Based on the significant differences between classes and the comparatively uniform content in sub-shots, the maximum entropy frame value is extracted as the keyframe for each class. Additionally, the maximum image entropy value correlates to the maximum amount of information according to information theory [3]. This method selects the frame with the largest entropy as the key-frame from each class based on the following properties: the differences between two classes are greater, the content within each class is relatively consistent after clustering, and the information theory fact that the larger the image entropy, the more information the image contains. The approach surpasses the limitations of the classic key-frame method extraction approaches in which the keyframe numbers are fixed. In [4], a unique resilient key frame extraction and foreground isolation approach for variable frame rate films utilising k-means clustering and mean squared error method is proposed and implemented. While removing the noise made during the recording, foreground items have been isolated in the video. This technique significantly lessens the flickering of the frames brought on by a fluctuating frame rate in a recorded video. On Apache's Hadoop architecture, the k-means clustering is carried out to speed up the computation's results. The method's outcomes have been contrasted with comparable outcomes attained using well-known methodologies like the Gaussian Mixture Model, and it has been determined that the method's outcomes are superior. The video frames are subtracted from the background once the background has been modelled in order to separate the foreground elements. A bilateral filter is applied to the frame to further reduce the noise, thus improving the clarity of the foreground items. The foreground mask is acquired, denoised, and then color quantization using k-means clustering is carried out which eliminates the background noise. The frames that produce flicker due to the frame rate show as fully black frames after clustering of the foreground masks. A mean squared error comparison of the foreground mask frames and the black frames is performed after sorting all of these frames according to how many black pixels are present in them. A key frame is one that is distinctive.

2.2 Motion Analysis Based Methods

Motion Analysis-based key frame extraction algorithms use motion features to detect important frames so that the original video is compressed without losing important actions. Motion features describe the visual changes in the video with temporal differences. It can be calculated using motion estimation techniques such as optical flow. Optical flow estimates the motion of image intensities, which may be ascribed to the motion of objects in the scene. The motion-based method is used to extract key frames from the video. The optical flow between frames in a shot is calculated and frames that are at local minima of the motion of the shot are considered keyframes. The algorithm involves two steps: Horn and Schunck's algorithm is used to calculate optical flow. In [5], the sum of the magnitudes of the components of optical flow at each pixel as a motion metric M(t) for frame t is computed as shown in Equation 1:

$$M(t) = \sum_{i=1}^{k} \sum_{j=1}^{l} |O_x(i,j,t)| + |O_y(i,j,t)| \quad (1)$$

The second step involves identifying local minimas. The graph between M(t) vs t is plotted and it identifies two local maxima m1 and m2 such that the value. The values of m1 and m2 are such that m2 varies by at least N% from the M(t) value for m1. The key frame is the local minimum of M(t) between the local maxima. An important advantage of this algorithm is that it does not assume a fixed number of keyframes per shot. Instead, it selects the number of keyframes appropriate to the composition of the shot.

2.3 Shot Boundary Based Methods

Shot Boundary-based key frame extraction algorithms use visual dissimilarity to identify key frames. Such dissimilarities can be abrupt or gradual and it occurs due to the transitions in video. Shot boundary detection is a key step of this method and various techniques such as pixel differences, statistical differences, histogram comparisons, edge differences, etc. can be used to calculate the same. This technique works very poorly for unstructured videos. It produces comprehensive results for structured video with a very steady shot change rate. Key frames are extracted utilizing the histogram-based method by applying the threshold and Difference of the Histogram Method. The threshold is determined by the difference between the histograms. First, every frame from the intended video is taken out and put into a directory. Every frame is transformed into the matching grayscale picture. The full video is now repeated. The total of all the histogram's elements is returned at each stage, which involves calculating the Histogram Difference between grayscale photos of two successive frames. The mean and standard deviation are determined after the complete iteration. Finally, the mean and standard deviation values are used to calculate the threshold. We now compare this threshold value with the sum value subtracted from

the computed histogram difference at each iteration.If the sum value of difference histogram for two consecutive frames is greater than threshold value then the latter frame is considered as key frame. In this way, after an entire iteration, a set of key frames from the entire video is obtained based on the threshold value [6]. In [8], a pixel-based method is suggested. This method considers the change of pixels between successive frames to identify the key frames. Either pixel difference between two successive frames is calculated or the percentage of pixels which are varied is considered. In [7], pair-wise comparison approach is implemented wherein the pixels whose value is changed by a predefined threshold are counted. If significant number of pixels change, greater than threshold T, then a segment boundary is declared. This method is extremely sensitive to camera motion. Another method used for key frame extraction is the Edge Difference-based keyframe extraction method. It depends on the contents of the frames and how those contents change. It considers edge difference since the edge is content-dependent. In this method, edge pixels of one frame are mapped to nearby edge pixels of the next frame to detect the changes in the content of the frames [9]. In [10], Canny Edge detection is used to calculate the difference of edge pixels in consecutive frames. The entire video is iterated and, in every iteration, the current frame and its consecutive frame are converted into its corresponding grayscale image and their edge difference is calculated using the Canny edge detector. This difference value is stored and this process is followed for each iteration. At the end of the entire procedure, differences between all the consecutive frames are added up and are used to calculate the mean and standard deviation. The threshold is computed using mean and standard deviation values. The differences which exceed the threshold value are considered keyframes.

Statistically based approaches use statistical differences to identify keyframes that contain the relevant contextual information. The suggested technique in [11] reduces the processing power and time needed for evaluating the massive amount of data in aerial surveillance imagery by using the statistical difference method to find keyframes.

Using this method, each frame is divided into smaller sections, and for each subsequent frame, statistical features such as the mean and standard deviation of each pixel within these regions are computed. Utilizing the mean and standard deviation, the adaptive threshold is computed. Keyframes are defined as frames with statistical changes larger than the adaptive threshold. This method divides frames into small sections for statistical calculations, which makes it useful for identifying frames with minute content changes. Hence this method is noise tolerant. But in contrast, since all the frames are partitioned and analyzed, this process can be slow due to intricate statistical computation.

2.4 Deep Learning Based Methods

In [12], the pertinent key frame from the video is obtained using a combination of deep learning and histogram approaches. When the images are transformed into HSV model space, which expresses information about a particular color, gray shade, and brightness, they assert that this produces improved histograms. Consequently, the video frames that are gathered are converted to HSV model space. After normalization, the histograms are contrasted with those of the next frames. When comparing histograms, the Bhattacharyya distance is employed. A list of keyframes is created by comparing the histogram's value with the threshold. This list is then compared to the real list of frames in order to remove any superfluous keyframes. They use a convolutional neural network as the following stage. This article [13], which is based on the Visual Geometry Group (VGG), suggests an image saliency extraction model with the use of deep prior information. To create a trained model, a large-scale data set is used for server training, and various features are then integrated. A saliency extraction algorithm based on multi-feature fusion and deep prior information is proposed by combining the saliency extraction technique with the picture saliency extraction model with the help of deep prior knowledge. in [14], a convolutional neural network (CNN) based key-frame extraction (KFE) engine with GPU acceleration that can quickly and accurately extract key-frames with high quality faces. The experimental results demonstrate that our CNN-based KFE engine may significantly decrease the overall processing time for face identification in videos while simultaneously increasing the back-end face recognition accuracy. In [15], an approach based on combination of CNN and RNN is suggested. To enhance the performance, a template-based approach is used. A supervised learning approach to key frame detection is suggested in [16]. The method leverages motion information from video frames containing the detected object region to identify false negatives.

Standard performance criteria for key frame extraction techniques are not established since there is no literature that provides the formal definition of "key frame". The attributes that key frames must have vary depending on the application. The main requirement is to reconstruct the video with the least number of key frames. But it is also important to choose frames that include crucial data concerning anomalies. In order to detect abnormalities in real time, the key frame extraction approach should also provide key frames as rapidly as feasible.

3. METHODOLOGY

This section describes the working of our approach to detect key frames. In our approach, key frame extraction is obtained using Histogram, Principal Component Analysis, and Dynamic Clustering. The system architecture is shown in Figure 1.

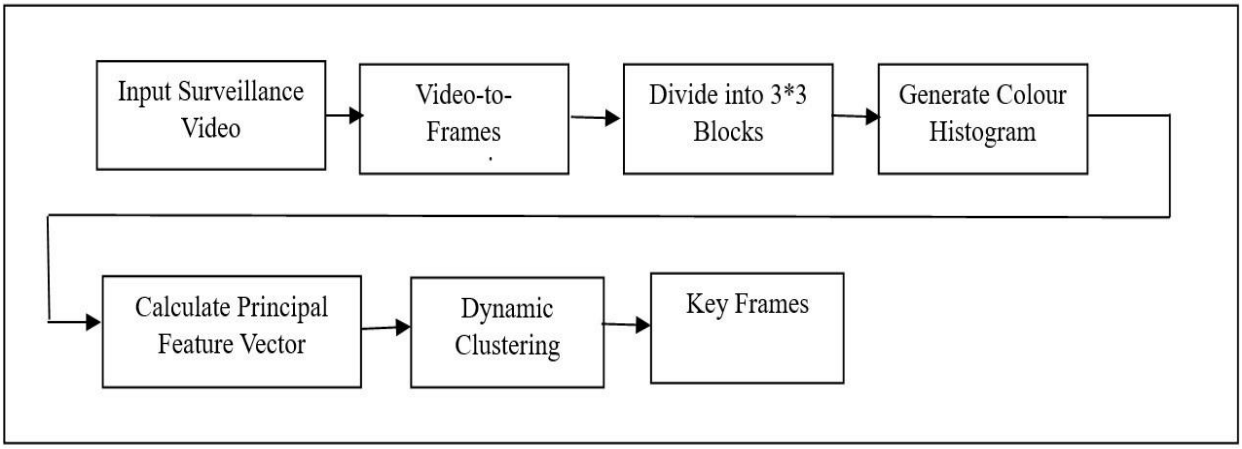

Fig. 1: System Architecture

In the proposed approach, frames are extracted from the input video. Each frame is then divided into 3 * 3 = 9 blocks. It is essentially split the image into smaller, nonoverlapping regions. This allows to analyze the color information in a more localized and granular way. This also overcomes the drawback of histogram-based key frame extraction technique of losing location information. Then, calculate the color histogram of all 9 blocks. The color histogram of an image is a representation of the frequency of occurrence of different colors in the image. A color histogram can be calculated for each channel of an image and can be used as a feature vector. For each block, the color histogram is computed separately for each color channel using 6 bins. This results in a 1944-dimensional feature vector (9 blocks * 6 bins of red channel * 6 bins of blue channel * 6 bins of green channel). After processing the video using the method described above, a matrix of M x N dimensions is obtained, where M is the dimensionality of the feature vector for each frame (here, 1944), and N is the number of frames in the video. The size of the matrix depends on the number of frames in the video, and the length of the video. If the video is long, the matrix can become quite large and require significant computational resources. Hence, Principal Component Analysis (PCA) is used to reduce the dimension of the matrix. The basic idea behind PCA is to identify a new set of variables, called principal components, that capture most of the variability in the original data. The principal components are linear combinations of the original variables and are ordered in such a way that the first principal component explains the largest amount of variation in the data, the second explains the second largest amount, and so on. Once the principal components have been calculated, only the top k components are retained, where k is much smaller than the original number of features. By doing so, the dimensionality of the dataset is reduced and only the most important information is retained. After selecting the principal components the matrix is reduced to P x N dimensions, where P is the dimensionality of the principal components, and N is the number of frames in the video. Further, Structural Similarity Index Measure (SSIM) is used to compare video frames and create clusters of consecutive frames. The SSIM measures the structural similarity between two images, taking into account factors such as luminance, contrast, and structure. It is based on the idea that the human visual system is more sensitive to structural information in images rather than absolute pixel values. The SSIM index is a value between -1 and 1, where 1 indicates perfect similarity between the two images, and -1 indicates complete dissimilarity. The formula for computing the SSIM index is given in Equation 2:

$$SSIM(x,y) = (2 * \mu x \mu y + c1) * (2 * \sigma xy + c2) / (\mu x^2 + \mu y^2 + c1) * (\sigma x^2 + \sigma y^2 + c2) \quad (2)$$

where x and y are the two images being compared, μx and μy are the means of the two images, $\sigma x$and $\sigma y$are the standard deviations of the two images, $\sigma xy$is the covariance between the two images, and c1 and c2 are constants used to stabilize the division.

The similarity threshold is calculated using adaptive thresholding. Firstly, the frames are converted into grayscale to simplify the processing. Next, the absolute difference between the grayscale frames is calculated using histograms. Once the absolute difference is calculated, a sum of all the differences is performed to obtain a total difference value. Then the mean and standard deviation of the total difference value to obtain an estimate of the variation in pixel values between the frames is calculated. Finally, this information is used to calculate the similarity threshold using a formula:

$T_s = \mu * k1 + \sigma * k2$ (3)

where T_s is the similarity threshold, μ is the mean, σ is the standard deviation, and k1 and k2 are the constants. By dynamically adjusting the similarity threshold based on the level of variation between the frames, more accurate and consistent results are achieved.

Using SSIM, it can be accurately determined whether a new frame is similar to the previous cluster formed, enabling to dynamically adjust the clustering process based on the level of similarity between frames. This technique is called Dynamic Clustering which allows to group similar frames in a video into clusters and extract representative key frames from each cluster. Firstly, the first frame of the video is assigned to the first cluster. Next, the similarity of the first frame is checked with the similarity of the next consecutive frame using SSIM. If the similarity is less than the threshold, the frame is assigned to a new cluster, and a new centroid is computed for the cluster. However, if the similarity is greater than or equal to the threshold, the frame is assigned to the previous cluster and the centroid is updated. This process is repeated for all the remaining frames in the video, allowing to create a set of clusters that group similar frames. In the next step, a reduction in the number of clusters is done by considering only those clusters that have a sufficient number of frames. Sparse clusters with fewer frames will be removed, resulting in a more concise and meaningful set of clusters. In the clustering process, a threshold for the number of frames required to form a dense cluster needs to be set. This threshold is set based on the desired level of granularity in the clustering process. For instance, a higher threshold would result in fewer but more densely populated clusters, while a lower threshold would result in more but sparsely populated clusters. To determine whether a cluster is dense, adaptive thresholding is used. This technique follows the same steps as discussed earlier for calculating the similarity threshold but with a different final formula. Specifically, using the grayscale frames, calculate the absolute difference using histograms, sum all the differences,calculate the mean and standard deviation, and use them to compute the adaptive threshold using the formula given in Equation 4:

$Td = \mu + \sigma$ (4)

where Td is the dense cluster threshold, μ is the mean, and σ is the standard deviation.

Once the adaptive threshold is established, it can be used to determine whether a cluster is dense or not. If the number of frames in a cluster exceeds the adaptive threshold, the cluster is considered dense. The dense clusters are used to extract representative key frames by selecting the last frame of the cluster. These key frames are representative of the entire cluster and can be used to summarize the video.

4. RESULTS AND DISCUSSION

The testing for the developed system is carried out on 10 videos pertaining to surveillance videos containing instances of attacks with weapons like knives and guns, and instances of vehicles and buildings catching fire. These videos are CCTV footage obtained from the internet. While testing, two aspects are checked, viz., the number of keyframes obtained and whether the obtained key frame contains an anomalous event. The Python code begins by taking the video input and breaking it down into frames. Each frame is then passed through the steps discussed in Section 3. Table 1 given below shows the experimental results obtained for 10 videos. For a video that originally had 3993 frames, our system detects only 11 frames as keyframes. It can also be seen from Table 1 that the original time of video 1 was 86.39 seconds and the optimized time is now 1.018 seconds.

From Table 1, it is observed that for the 10 videos tested an average reduction in the number of frames to be processed is approximately 97%.

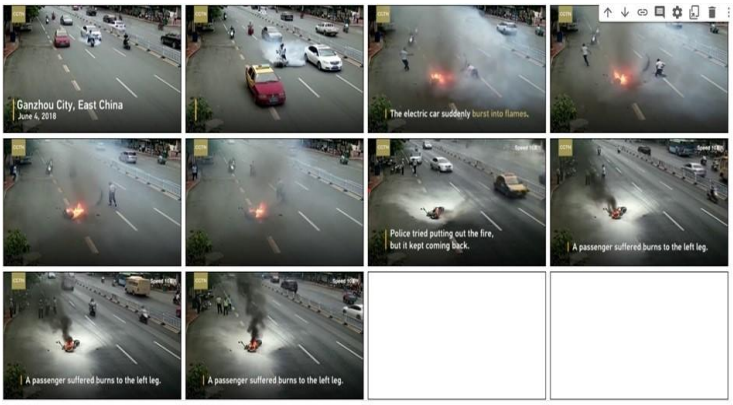

Fig. 2: Key Frames for Test Video 2

It can be seen from Table 1, for test video 2, there were total 1168 frames. The system detected only 10 frames as the key frames as shown in Figure 2. The time of the original test video 2 was 31.41 seconds which is now reduced to 1.03 seconds and the percentage decrease obtained is 96.72%. Figure 3 shows for test video 8, which originally consisted of 1171 frames as given in Table 1, the system detects only 16 frames as keyframes. The time of the original test video 8 was 28.96 seconds which is now reduced to 0.74 seconds and the percentage decrease obtained is 97.43%.

Table 1: Experimental Results

Test video	Total Frames	Key Frames	Original Time	Optimized Time	Percentage Decrease
1.	3993	11	86.39	1.018	98.82
2.	1168	10	31.41	1.03	96.72
3.	803	11	23.966	0.75	96.86
4.	470	11	15.98	0.359	97.7
5.	770	13	19.25	0.3864	98.17
6.	573	13	21.16	0.66	96.84
7.	1206	23	28.86	1.26	95.62
8.	1171	16	28.96	0.74	97.43
9.	5642	16	1888.2	8.15	99.56
10.	1196	10	31.62	0.9	97.13

Fig. 3: Key frames for Test video 8

To determine the efficiency of the approach discussed, the measure of fidelity is employed for evaluation. Fidelity ensures the precision of keyframes [18]. A high fidelity rating signifies an accurate and efficient reproduction of the original video, achieved through a reduction in the number of frames. The values for fidelity measure obtained for the proposed method areshown in Table 2. As seen in Table 2, for all the test videos, a high fidelity value is obtained.

Table 2: Experimental Results for Fidelity

Test Video	Fidelity Measure
1	103692.37
2	42885.98
3	94162.73
4	81871.58
5	154727.21
6	22784.42
7	26232.277
8	70534.220
9	30007.41
10	62601.04

5. CONCLUSION

Keyframes form an essential part when it comes to video analysis and video summarization. In this paper, a hybrid approach using a histogram and dynamic clustering-based approach is presented for key frame

extraction. An average of 97% decrease is obtained in the number of frames required for further analysis. To test the efficiency of the approach, a fidelity measure was used. For all the test videos, a high-fidelity measure was obtained.

Declarations

The authors declare that they have no conflicts of interest regarding the research presented in this manuscript. This work was not supported by any external funding or sponsorship. The authors declare that they have no financial or personal relationships with individuals or organizations that could inappropriately influence or bias the content of this manuscript.

REFERENCES

[1] Zhao, H., Wang, T., Zeng, X.: A clustering algorithm for key frame extraction based on density peak. Journal of Computer and Communications **6**(12), 118–128 (2018)
[2] Pan, R., Tian, Y., Wang, Z.: Key-frame extraction based on clustering. Paper presented at the IEEE International Conference on Progress in Informatics and Computing (2010)
[3] Pan, R., Tian, Y., Wang, Z.: Key-Frame Extraction Algorithm Based on Entropy. IEEE International Conference on E-Product E-Service and E-Entertainment (2010). https://doi.org/10.1109/ICEEE.2010.5660916
[4] Nasreen, A., Roy, K., Shobha, G.: "Key frame extraction and foreground modelling using k-means clustering. Paper presented at the 7th International Conference on Computational Intelligence, Communication Systems and Networks (2015)
[5] W, W.: Key frame selection by motion analysis. IEEE international conference on acoustics, speech, and signal processing conference proceedings (1996)
[6] Agarwal, A.K., Agarwal, K., Choudhary, A. J.and Bhattacharya, Tangudu, S., Makhija, B. N.and Rajitha: Automatic traffic accident detection system using ResNet and SVM. Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) (2020)
[7] Zhanga, H., Kankanhalli, A., Smoliar, S.W.: Automatic partitioning of full-motion video," multimedia systems. Multimedia Systems 1, 10–28 (1993) https://doi. org/10.1007/BF01210504
[8] Boreczky, J.S., Rowe, L.A.: Comparison of video shot boundary detection techniques. In: Sethi, I.K., C., J.R. (eds.) Storage and Retrieval for Still Image and Video Databases IV vol. 2670, pp. 170–179. SPIE, ??? (1996). https://doi.org/ 10.1117/12.234794 . https://doi.org/10.1117/12.234794
[9] Nandini, H.M., Chethan, H.K., Rashmi, B.S.: Shot based keyframe extraction using edge-lbp approach. Journal of King Saud University - Computer and Information Sciences **34**(7), 4537–4545 (2022) https://doi.org/10.1016/j.jksuci.2020. 10.031
[10] Khurana, K., Chandak, M.: Akey frame extraction methodology for video annotation. International Journal of Computer Engineering and Technology **4**(2), 221–228 (2013)
[11] Essa, A., Sidike, P., Asari, V.: A modular approach for key-frame selection in wide area surveillance video analysis. National Aerospace and Electronics Conference (NAECON) (2015)
[12] Gawande, U., Hajari, K., Golhar, Y.: Deep learning approach to key frame detection in human action videos. In: Sadollah, A., Sinha, T.S. (eds.) Recent Trends in Computational Intelligence. IntechOpen, Rijeka (2020). Chap. 7. https: //doi.org/10.5772/intechopen.91188 . https://doi.org/10.5772/intechopen.91188
[13] Zhong, Q., Zhang, Y., Zhang, J., Shi, K., Yu, Y., Liu, C.: Key frame extraction algorithm of motion video based on priori. IEEE Access 8, 174424–174436 (2020) https://doi.org/10.1109/ACCESS.2020.3025774
[14] Qi, X., Liu, C., Schuckers, S.: Cnn based key frame extraction for face in video recognition. In: 2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA), pp. 1–8 (2018). https://doi.org/10.1109/ISBA.2018. 8311477
[15] Savran Kızıltepe, R., Gan, J.Q., Escobar, J.J.: A novel keyframe extraction method for video classification using deep neural networks. Neural Comput Applications 35, 24513–24524 (2023) https://doi.org/10.1007/ s00521-021-06322-x
[16] Sinulingga, H.R., Kong, S.G.: Key-frame extraction for reducing human effort in object detection training for video surveillance. Electronics (2023)
[17] Sadiq, B., Muhammad, B., Ab-dullahi, M., Onuh, G., Ali, A., Babatunde, A.-g.:Keyframe extraction techniques: A review. ELEKTRIKA- Journal of Electrical Engineering (2020)
[18] Habib, S., Hussain, A., Islam, M., Khan, S., Albattah, W.: Towards efficient detection and crowd management for law enforcing agencies. In: 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), pp. 62–68 (2021). https://doi.org/10.1109/CAIDA51941.2021.9425076