Hindawi
Wireless Power Transfer
Volume 2021, Article ID 8192582, 12 pages
https://doi.org/10.1155/2021/8192582

Research Article

Intelligent Power Grid Video Surveillance Technology Based on Efficient Compression Algorithm Using Robust Particle Swarm Optimization

Hongyang He, Yue Gao, Yong Zheng, and Yining Liu

Chongqing Fuling Electric Power Industry Co., Ltd, Chongqing, China

Correspondence should be addressed to Hongyang He; 0000003@yzpc.edu.cn

Received 19 September 2021; Revised 21 October 2021; Accepted 2 November 2021; Published 30 December 2021

Academic Editor: N. Venkateswaran

Copyright © 2021 Hongyang He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Companies that produce energy transmit it to any or all households via a power grid, which is a regulated power transmission hub that acts as a middleman. When a power grid fails, the whole area it serves is blacked out. To ensure smooth and effective functioning, a power grid monitoring system is required. Computer vision is among the most commonly utilized and active research applications in the world of video surveillance. Though a lot has been accomplished in the field of power grid surveillance, a more effective compression method is still required for large quantities of grid surveillance video data to be archived compactly and sent efficiently. Video compression has become increasingly essential with the advent of contemporary video processing algorithms. An algorithm's efficacy in a power grid monitoring system depends on the rate at which video data is sent. A novel compression technique for video inputs from power grid monitoring equipment is described in this study. Due to a lack of redundancy in visual input, traditional techniques are unable to fulfill the current demand standards for modern technology. As a result, the volume of data that needs to be saved and handled in live time grows. Encoding frames and decreasing duplication in surveillance video using texture information similarity, the proposed technique overcomes the aforementioned problems by Robust Particle Swarm Optimization (RPSO) based run-length coding approach. Our solution surpasses other current and relevant existing algorithms based on experimental findings and assessments of different surveillance video sequences utilizing varied parameters. A massive collection of surveillance films was compressed at a 50% higher rate using the suggested approach than with existing methods.

1. Introduction

As discussed by Memos et al. [1], the number of Switch-Mode Power Supply is increasing, as are incentive-based switching activities at the end-user level. A high-resolution time-resolution monitoring system will be required for future smart grids' operational stability to properly examine the state of the electricity grid. When it comes to power grid measurement applications, kilohertz frequencies are used, but the degree of aggregation and the reporting rate is not the same. Instead of using a second rate for instantaneous data like the smart meters, they utilize a day or more rate for the cumulative consumption data. As a result of the consolidation, communication lines and storage space needs have been significantly reduced. Assessing power quality (PQ) as well as disaggregating loads necessitates more data, as Gao et al. have shown [2]. Several features can be added on top of Harmonics; however, they can only provide partial information. Changes in grid operating approaches, demand-side control, and the rise of decentralized generation have led to an unknown number of combinations of interruptions. Features-based approaches may be unreliable due to the fact that data gets destroyed, particularly when exciting short-lived occurrences. Some thresholds may be adjusted by the user in commercially available equipment for PQ measurement at sample rates ranging from 10 kHz to 100 megahertz; raw data is captured when an event happens. A future smart grid, on the other hand, will have hard-to-predict threshold values. There may be further insights to be gained by examining raw data from synchronized

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

measurements at different locations—even if not all scattered sensors were able to classify events simultaneously and hence did not capture them at a high resolution was depicted by Tsakanikas et al. [3]. Deploying a continuous storage system for raw data will assist data-driven research that attempts to improve event classification and smart grid analysis algorithms; for example, when using lossless data compression, compressing and transmitting large volumes of data is considerably easier. Uncommonly, the raw data stream of a recording device has three voltage readings (from the 3 phases) and 4 current measurements (3 for the phase currents and 1 for the neutral conductor). Nominally sinusoidal voltage curves that are 120° out of phase make up a three-phase power system. A strong relationship has been established between the various modes of communication. The same applies to current lines, and leveraging this interaction allows for a particular minimization in data volume. As a result of their unique distortion, the waveforms are less connected. Conceived as a way to decrease correlation in current channels owing to phase-load distortions. It is important to note that waveforms only change at the equipment's contact state or load alterations in terms of distortions. In general, these operations are slow compared to the length of time. With only one load connection, waveform changes are rare. Because load currents are increased whenever a massive number of demands get linked to the sensing field of the grid, variations exist rapidly. Waveform compression is therefore conceivable. It is known that lossless compression methods exist for certain applications, such as music and video. However, no method has been identified that is specifically designed to take advantage of the periodicity and multichannel nature of electrical signals encountered in a stream compression methodology. An overview of lossy and nonlossy techniques is included in the book, as are the CR values from trials. Applications that focus on PQ-event compression are listed; these implementations were developed by Shidik and his colleagues [4] among themselves. There is no statistical analysis of lengthy original data. These models focused on extremely precise incident data to validate the applicability of algorithmic changes in their own unique contexts. In the majority of cases, data sources are not referenced or provided at all. We find the researchers do not have a benchmark against which to assess the feasibility of compression algorithms for grid wave information, regardless of whether they are using known techniques or new ones that have yet to be found. We have chosen to focus on the development of compression with no degradation techniques and performances for grid data at a lot of sampling to address these issues in the present contribution. When utilizing input data with a variety of ideas, we are considering new growth ideas made of natural time-series analysis. New lossless compression algorithms could be developed by using testing data and comparison parameters for the first thorough accessible standard. They can be used as a decision assistance tool by researchers dealing with data-intensive smart grid measures. The preprocessing phase entails changing the color space, after which the features may be retrieved using pseudo-component analysis. Then, utilizing Robust Particle Swarm Optimization, the encoding and decoding process may be completed. The main contribution of the research work is as follows:

(i) To design and develop a compression-based video surveillance technology based on the optimization approach

(ii) For the purpose of the authentication, run-length encoding and decoding were performed

The following is how the rest of the article is organized. In Section 2, a literature survey is being reported on strategies to reduce loss during video compression. The issue of lossless video compression mechanisms was then addressed in Section 3. Section 4 then poses the proposed mechanism over lossless video compression. The results of the suggested method and the conclusions were examined in Sections 5 and Section 6.

2. Related Works

In [1], the article looks into wireless sensor networks (WSNs) alongside the most recent research on social confidentiality and protection in WSNs. While adopting High-Efficiency Video Coding (HEVC) as a new media compression standard, a novel EAMSuS in the IoT organization is presented (HEVC). In [5], complete situational awareness is provided via real-time video analysis and active cameras. In [6], a new section of MPEG standards called Video Compression Modulation (VCM) has been suggested by the author. Video Coding for Machine Vision seeks to bridge the gap between machine vision feature coding and human vision video coding to accomplish collaborative compression and intelligent analytics. VCM's definition, formulation, and paradigm are provided first, corresponding with Digital Retina's rising compress instance. This is why they analyze video compression and features from the unique perspective of MPEG standards, which offers both academics and industry proof to accomplish the collaborative compression of the video shortly. In [7], using MapReduce, the author has developed UTOPIA Smart Video Surveillance for smart cities. From their end, we were able to incorporate smart video surveillance into our middleware platform. With the help of this article, we show that the system is scalable, efficient, dependable, and flexible. In [8], here when it comes to edge computing capabilities, the cloud object tracking and behavior identification system (CORBIS) was demonstrated. To increase distributed video surveillance systems' resiliency and intelligence, network bandwidth and reaction time between wireless cameras and cloud servers are being reduced in the Internet-of-things (IoT). In [9], an effective cryptosystem is used to create a safe IoT-based surveillance system. There are three parts to it. An automated summary technique based on histogram clustering is used to extract keyframes from the surveillance footage in the first stage. To compress the data, a discrete cosine transform is applied to it (DCT). Not to mention, a discrete fractional random transform is used to develop an efficient picture encryption approach in the suggested framework (DFRT). In [10], the author proposes a novel approach for compressing video

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

inputs from surveillance systems. There is no way to reduce visual input redundancy using outdated methods that do not meet the demands of modern technologies. Video input storage needs to increase as a result, as does the time required to process the video input in real-time. To compress video inputs from surveillance systems, a unique technique is presented in this research paper. Visual input redundancy cannot be reduced using obsolete approaches that do not match the expectations of contemporary technology. This raises the storage requirements for video input and the processing time as a result. In [11], by using compressed sensing (CS), the author suggests creating security keys from the measurement matrix elements to secure your identity. Assailants cannot reconstruct the video using these. They are designed to prevent this. A WMSN testbed is used to analyze the effectiveness of the proposed security architecture in terms of memory footprint, security processing overhead, communication overhead, energy consumption, and packet loss, for example. In [12], a new binary exponential backoff (NBEB) technique was suggested by the author to "compress" unsent data that can preserve important information but recover the electronic trend as much as feasible. Data coming in may be temporally chosen and dumped into a buffer, while fresh data can be added to the buffer as it is received. As a result of the algorithm, the incoming traffic rate can be reduced in an exponential relationship with the transmitting failure times. In [13], the author suggested the lossless compression technique to handle the problem of managing huge raw data amounts with their quasiperiodic nature. The best compression method for this sort of data is determined by comparing the many freely accessible algorithms and implementations in terms of compression ratio, calculation time, and operating principles as well as algorithms for audio archiving; there are other algorithms for general data compression (Lempel–Ziv–Markov chain algorithm (LZMA), Deflate, Prediction by partial matching (PPMd), Burrows–Wheeler algorithm (Bzip2), and GNU zip (Gzip)) that are put to the test against one other. Deal with the challenge of managing enormous raw data quantities with their quasiperiodic nature by using lossless compression. Compression ratio, computation time, and operating principles are all taken into account when comparing publicly available algorithms and implementations to decide which is the most efficient. Additionally, generic data compression techniques such as LZMA, Deflate, PPMd, Bzip2, and Gzip are also put to the test. In [14], an efficient embedded image coder based on a reversibly discrete cosine transform is proposed for lossless. ROI coding with a high compression ratio (RDCT) was suggested. To further compress the background, a hierarchical (SPIHT) partitioning technique is used to combine the proposed rearranged structure with a lost zero tree wavelet coding. Results of the coding process indicate that the new encoder outperforms many state-of-the-art methods for still photo compression. In [15–17], the focus was based on the loss of video compression. Even at lower bit rates, the novel loss-compression method improves contourlet compression performance. Along with SVD, compression efficiency is improved by standardization and prediction of broken subband coefficients (BSCs) [18]. We measure the computational complexity of our solution with a better video quality. HCD uses DWT, DCT, and genetic optimization to improve the performance of transformed coefficients, among other techniques. This method works well with MVC to get the best possible rate distortion. The simulation results are produced using MATLAB Simulink R2015 to examine PSNR, bit rate, and calculation time for various video sequences using various wavelet functions, and the performance results are evaluated [19]. To solve the optimization issue of trajectory combination while producing video synopses, a new approach has been devised. When dealing with the optimization issue of motion trajectory combination, the technique makes use of the genetic algorithm's temporal combination methods (GA) [20]. The evolutionary algorithm is utilized as an activation function within the hidden layer of the neural network to construct an optimum codebook for adaptive vector quantization, which is proposed as a modified video compression model. The context-based initial codebook is generated using a background removal technique that extracts motion items from frames. Furthermore, lossless compression of important wavelet coefficients is achieved using Differential Pulse Code Modulation (DPCM), whereas lossy compression of low energy coefficients is achieved using Learning Vector Quantization (LVQ) neural networks [21]. This paper presents a rapid text encryption method based on a genetic algorithm. It is possible to use genetic operators Crossover and Mutation to encrypt data. By splitting up the plain text characters into pairs and using a crossover operation to obtain the encrypted text from the plain text, this encryption approach uses mutations to get its encrypted message.

From the literature survey, reviewed images and videos are compressed using transform-based and fractal approaches, along with other lossless encoding algorithms, which are now the most frequently used methods for still and video compression. Each technique has its own set of pros and downsides like breaking of the wavelet signal and low compression ratio; hence, it is important to choose the right one. It is most common for video-based images to be compressed using transform-based compression (TBC). In order to achieve compression, the signal or values are altered. Using various transformations, they convert a spatial domain representation into a picture. Brushlet is an example of an adaptive transformation (Verdoja and Grangetto 2017); bandelet (Raja 2018) (Erwan et al. 2005) and directionlet (Jing, et al. 2021) give information about the picture in advance. After applying these modifications to a picture, its essential function is altered. Hence, we are motivated to develop a methodology that overcomes all the existing video compression issues.

3. Problem Statement

Rapid advances are being made in compressing technology. As a challenging and essential topic, real-time video compression has sparked a lot of studies. This corpus of information has been included in the motion video standards to a large extent. Unanswered are several significant

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

questions. According to the point of view of a compression algorithm, eliminating various redundancies from certain types of video data is a compression challenge. Thorough knowledge of the problem is needed, as well as a novel approach to solve all of the existing research gaps with irreversible video compression. Progress in other fields, such as artificial intelligence, has contributed to the breakthroughs in compression. A compression algorithm's success depends on the acceptance of a new generation of algorithms in addition to its technological excellence.

4. Proposed Work

As a result of the smart grid's usage of ICTs, the generation, distribution, and consumption of electricity are all more efficient (Information and Communication Technologies). For example, the transmission system and the medium-voltage level distribution system are monitored by Supervisory Control and Data Acquisition (SCADA) and wide-area monitoring systems (WAMS). It is important to remember that the primary objective of compression is to minimize the amount of data. That is if the compressed data retains most of its original content. Various scholars are currently involved in proposing effective techniques of data compression. Listed below are some of the most prevalent data compression techniques. With this analysis, we are focusing on compressing the PQ-event data in a video context in each successive frame to save space. To accomplish this, we must first identify the video frame object. Robust Particle Swarm Optimization is used to create a lossless video compression method. This is a diagram of the recommended technique shown in Figure 1.

4.1. Dataset. They used the UK Domestic Appliance-Level Electricity (UK-DALE) Dataset to conduct the experiments. A smart distribution system collects data on three-phase voltage, current, active and reactive power, and power factor from transformers at 54 substations as well as estimations of current and voltage at the inlets of three homes. The data is then analyzed and compared with the raw data from three homes. A 16 kHz sampling rate and a 24-bit vertical resolution were employed in the acquisition. There was a random selection of six FLAC-compressed recordings from 2014-8-08 to 2014-05-15, each having an hour of recordings. In a proprietary format, these data are recorded as four-byte floating-point numbers with timestamps at a sampling rate of 15 kHz. Voltage and current values are included in phase 2 of house 5. Every one of the four files contains 266 s. Large-scale databases hold all data transferred via a network. Raw data for three-phase voltages need 8.4 GB per day, whereas three-phase currents (including neutral) require 19.35 GB per day. To transmit the data, you need 0.8 Mbit/s and 1.8 Mbit/s in turn. This dataset was compiled in the following locations: as our institution's main power supply in Karlsruhe, Germany, we also have power outlets in our practical room and a substation transformer there. A total of seven channels consisting of four currents and three voltages are sampled at 12.8 and 25 kS/sec, respectively. There are seven channels, with four currents and three voltages sampled at 12.8 and 25 kS/sec, for a total of seven channels. Single-channel testing and dual-channel testing include measuring the current and/or voltage of a single phase in both situations, depending on which method is used. To save the data, raw 16-bit integers are stored in blocks of 60 s.

Electricity generation, transmission, and distribution in smart power systems are all affected by the analysis of the data. As a result, data exchange and memory requirements are expected to grow considerably, and data storage and bandwidth requirements for communication links in smart grids are also expected to increase. It is necessary to raise the sampling frequency to receive reliable and real-time information from the intelligent grid. There will be a greater emphasis on smart grid data compression in the future. Figure 1 illustrates the proposed compression approach. In areas of the grid with significant data volume, this approach can be used successfully.

4.1.1. Preprocessing. There are several steps to video compression, the first being preprocessing. Preprocessing is essential for a database's longevity and usefulness. For this reason, each stage in the video data processing workflow appears to be crucial. The procedure involves preprocessing, such as error detection or any other conversions that are not essential. Power grid video can cause picture frames to be split. The Bayesian motion subsampling approach may be used to create the video frame. This is a common method for removing frames from a movie. As the name implies, it is a computerized method used to enhance the frame creation process. For the most common sensitivities, the picture frame intensity range has been expanded, which results in a better image frame sensitivity value.

Let p denote the subsampled of each possible frame illustrated as

$$p^{y} = \frac{(\text{Number of the pixel frames with the } y \text{ intensity})}{\text{Total number of the pixel frames}}. \tag{1}$$

Here, $y = 0, 1, \ldots, y - 1$.

The separated pixel frames can be defined as depicted in [22]:

$$H_{i,j} = \text{base}\left((Y - 1) \sum_{Y=0}^{b_{i,j}} p^{Y} \right), \tag{2}$$

where base represents the nearest integer. This is equivalent to transforming the pixel intensity [23]:

$$\frac{\partial N}{\partial x}\left(\int_{0}^{N} pN(x)\mathrm{d}z \right) = \left(\partial N(N)\left(x^{-1}\right)(N) \right)\frac{\mathrm{d}}{\mathrm{d}N}. \tag{3}$$

Here, finally, the probability distributed uniformity function can be represented as $\partial N/\partial x$.

When it comes to histograms, the equalization procedure can soften and enhance them. However, even though the histogram produced by the equalization is perfectly flat, it will be softened. After reducing the pictures' superfluous

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

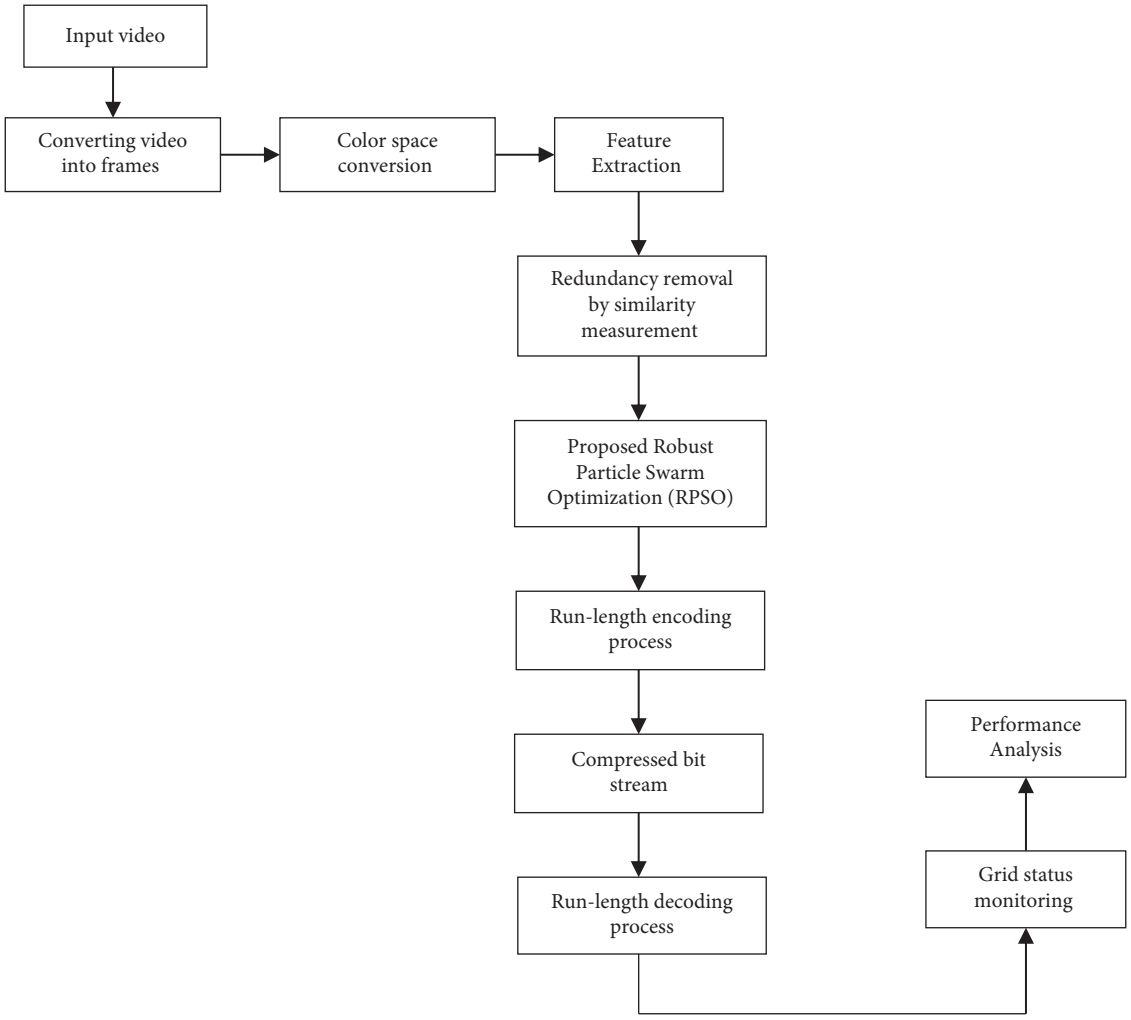


Figure 1: Schematic representation of the suggested methodology.

noise, we apply a threshold technique to improve the refined frame acquired from the context. Thereafter, binary images are created, which streamlines the image processing process. As a result of the color space conversion, we see a shading effect in the majority of pictures. The picture contains three channels in most cases (red, green, blue). In the blue channel, there is no more information, but there is a great deal of contrast. Preprocessed green channel was deleted next. For example, here is how to extract the green channel [24]:

$$\begin{aligned} I_{\text{org}} &= f(\sigma, \mu, \beta), \\ I_{\text{red}} &= f(1, \mu, \beta), \\ I_{\text{Green}} &= f(\sigma, 2, \beta), \end{aligned} \tag{4}$$

where σ denotes the Red channel, μ denotes the Green channel, and β denotes the Blue channel.

Translation of color representation from one basis to another is called color space conversion (CSC). In most cases, this occurs while converting a picture from one color space to another. The use of a single threshold value for converting the color space is thus not recommended.

$$\theta \propto \text{Threshold}(E) \approx j * \left(\frac{u}{|v^1/3|}\right)(j_{\text{best}} - j_i), \tag{5}$$

where E represents converted the color space.

The color space is transformed to grayscale by keeping the brightness information. A grayscale picture frame can be represented as a collection of grayscale images by D_2.

$$ED_2 = \text{GS}(D_1) = \{d_2(1), d_2(2), \ldots, d_2(i), \ldots, d_2(|D|)\}. \tag{6}$$

After the frame gets preprocessed, the data can undergo the step of feature extraction.

4.1.2. Feature Extraction. We implemented the pseudo-component analysis in the feature extraction module to improve the compression performance and concentrate the image's information. The method for decreasing the size and complexity of data sets involves converting huge numbers of variables into smaller ones that retain the majority of the information contained in the large set. Naturally, limiting the number of parameters in sets of data lowers the

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

information's accuracy, but the trick is to give up just a little precision for convenience. It is simpler to examine and interpret smaller data sets. Machine learning algorithms can also examine data more easily and quickly without dealing with extraneous issues. Each pseudo-redundancy component must be selected as a first stage in the process of feature extraction. In this module, the main goal is to extract the highlighted characteristics. Below are the configurations of this mechanism.

$$y_{\text{input}} = \left[V_c^T y_c + B_c, V_s^T y_s + B_s\right], \quad \beta = f_2\left(V_{\text{int}}^T f_1\left(y_{\text{input}}\right) + B_{\text{int}}\right). \tag{7}$$

Here, $[V_c^T y_c + B_c, V_s^T y_s + B_s]$ Error! Bookmark not defined denotes the overall feature level; $V_c^T \in \mathbb{Z}^{C_c \times C_{\text{int}}}, V_s^T \in \mathbb{Z}^{C_s \times C_{\text{int}}}$, and $V_{\text{int}}^T \in \mathbb{Z}^{2C_{\text{int}} \times 1}$ represent feature weights; B_c, B_s, and B_{int} depict the associated features; C_c and C_s correspond to the sizes of the input medium of the categorization and feature sections, accordingly; and C_{int} denotes the internal input. Operations $f_1(y) = \max(y, 0)$ and $f_1(y) = 1/(1 + \exp(-y))$ associate to sigmoid activation operation, accordingly. The attention map is further standardized to [0, 1]. The outcome of the feature extraction is represented as depicted in [25]:

$$y_{\text{out}} = f_3\left([\beta \times y_c, y_s]\right). \tag{8}$$

Here, f_3 consists of a sequence of the feature components.

Pseudo- and nonpseudo-component characteristics can be selected using a property calculation technique. To determine pseudo-component characteristics, Hong correlations approaches, which employ averaging techniques, and Leibovici correlations, which use mixing principles, are used. In this approach, the phase fraction values are collected from a compositional system to minimize the difference between them. Pseudo- and nonpseudo-redundancy characteristics can be retrieved, as shown as follows [26]:

$$Lo_{\text{Pseudo,nonpseudo}} = 1 - \frac{2|A \cap B|}{|A| + |B|} = 1 - \frac{\sum_j^N p_j g_j + sm}{\sum_j^N p_j + \sum_j^N g_j + sm}, \tag{9}$$

where p_j represents the pseudo features, g_j represents the nonpseudo features, and sm represents the empirical constant.

4.1.3. Optimized Compression Process. Video data may be compressed without losing any information using this method. Concerning Robust Particle Swarm Optimization (RPSO) and run-length coding (RLC), the common RLC can be optimized by using the optimization algorithm and has been employed for the compression stage. We analyze the properties of compressed data using this technique. To maximize compression-related parameters, it is advised to use this method as a population-based approach. RPSO is initialized with the sample particles and modified with the optimal answer in each cycle. The resulting answer is called fitness and is referred to as the $_{\text{best}}$. The best solution obtained by a particle in a population is considered the world's top value monitored by the particle swarm optimizer (g_{best}). By the two p_{best} and g_{best} solutions, the positions of each particle change to global optima. The individual speeds and location functions of each particle are as follows. In a dimension search space D, there is a swarm composed of particles where each particle is represented by 'i' in a vector of $X_i = x_{i1}, x_{i2}, \ldots\ldots, x_{\text{id}}$ and the particle bet solution p_{best} is denoted as $p_i = p_{i1}, p_{i2}, \ldots\ldots, p_{\text{id}}$. Then, the best solution of the subset swarm is given by $g_{\text{best}} p_g = \{p_{g1}, p_{g2}, \ldots\ldots, p_{g\,d}\}$. The i^{th} particle velocity is represented as $V_i = V_{i1}, V_{i2}, \ldots\ldots, V_{\text{id}}$. The particle velocity and location are updated based on equations (10) and (11).

The weights updates are given by [27]

$$v_{\text{id}}(n+1) = W(it) * v_{\text{id}}^n + C_1 * \text{rand} * \left(p_{\text{id}} - x_{\text{id}}^n\right) + C_2 * \text{rand} * \left(p_{g\,d} - x_{\text{id}}^n\right), \tag{10}$$

where W represents the weighed features, C represents the cross features, n represents the constant, rand represents the random number, x_{id} represents the particle velocity, and p_{id} represents the particle motion. Here, depending on the feature extracted, the details can be updated depending upon the weightage, where $V_{\text{id}} = V_{\max}$, and it reflects the number of iterations between 1 and 10, where 10 is the maximum number of iterations. The random value of 0 to 1 is represented by the rand. C_1 and C_2 normally signify a nonnegative amount of an acceleration constant; here, C_1 and $C_2 = 1.05$. The particle orientation is also modified with [28]

$$x_{\text{id}}^{n+1} = x_{\text{id}}^n + v_{\text{id}}^{n+1}. \tag{11}$$

Any swarm obtains the health or objective f and each iteration provides the best solution; then if $f(x_i) < f(p_{\text{best}})$ and $f(g_{\text{best}})$, then p_{best} and $g_{\text{best}} = x_i$. The optimal measurement is obtained to maximize the curve transformation coefficients.

Video reconstruction can be done via run-length encoding once the optimized values are acquired. Using sequential data, this is a fairly simple operation to perform on your computer. For redundant data, it is a great tool. Running symbols are replaced by shorter ones in this technique. There are two ways to express the run-length code in grayscale images: V and R, where V represents the character count and R represents the run length. For optimized run-length optimal run-length encoding (ORLE), the following steps are required:

Step 1: Coefficient optimization

Step 2: Enter the string

Step 3: Give a unique value from the very first symbol or letter

Step 4: Otherwise, leave if the character or symbol is the final one in the string

Step 4: Additional symbols can be read and counted

Step 5: Until the preceding symbol subband has a nonmatching value, move on to step 3

Step 6: This will give you a count of the number of times a certain symbol appears in a given sentence

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

The suggested methodology uses a vector that contains a variety of scales to convert subbands that are optimized minimum and maximum to achieve the best result.

$$\text{Compressed}_{\text{Fitness value}} = \frac{-40 * q\left(-3 * \sqrt{\sum S_v}\right)}{2 - \exp\left(\sum \cos\left(3\pi * S_v\right)/d_b\right)} + 10 \exp , \tag{12}$$

where q denotes the compressed reconstructed value and S_v is the compressed score value that is obtained. Finally, the best rate of compression can get obtained. The RPSO reconstructs the data by using run-length decoding after refining the transforming Algorithm 1 curvelet parameters are as follows.

Finally, after compression, the status of the grid can get sorted out and it can get monitored and the irregular grid distribution can get identified.

5. Performance Analysis

Increasingly, data is being exchanged across smart grid sectors. Many types of data are created every day. For example, meteorological data such as the amount of sun or wind, humidity, or temperature are essential for optimal performance in many industries. Two phases in the data interchange procedure are encoding and decoding (or decryption). Numerous operations take place during the encoding phase to prepare data for transmission; when the data is encoded and decoded, it will be returned to its original form. In this section, you will learn about the complete process of performing experiments for performance evaluation. It is written in MATLAB, which is a programming language. Measurement data was collected over 24 hours in 1-minute, 5-second, 10-second, and 20-second intervals to assess the proposed compression methods. Readings from multiple meters were collected for each period in a data matrix.

Table 1 illustrates the effect of truncating small singular values on the compression ratio (CR) and percentage residual root difference. It can be seen that the minimum root mean square distance is obtained when eight singular values are considered. This leads to a reduction in the signal length. Compared to other sets of data, the calculated CR values for the 5-second time interval data are closer to the Total compression ratios (TCR) values in Table 1. Data obtained at 1-minute, 10-second, and 20-second intervals have generated CRs that deviate somewhat from the TCRs. Figure 2 illustrates the relationship between the number of significant singular values and TCR. According to the plotted data, the size of the data matrix has an impact on the ratio of compression (r), the number of singularly significant values. There are two different sizes of a data matrix: 5 seconds and 1 minute. A greater number of significant single values were required to match the TCR in 5-, 10-, and 20-second datasets than in the 1-minute dataset, as can be observed in Figure 2. As an alternative, selecting a shorter time interval, such as five seconds, will offer a better approximation on the number of significant singular values, resulting in the computed CRs being closer to the TCRs.

The mean error is a colloquial phrase that refers to the average of all mistakes in a collection. In this context, an "error" refers to a measurement uncertainty or the difference between the measured value and the correct/true value. Measuring error, often known as observational error, is the more formal word for error. According to Figure 3, there is a relationship between the related mean error for different time interval data and TCRs. As shown in Figure 3, the data consisting of measurements per 1-minute interval has the lowest mean error. The MAE found for greater matrix sizes is larger when the TCRs are higher.

According to Figure 4, there is a correlation between the number of significant singular values and the rate of mistake. For the first 100 single values, the 5-second dataset has the greatest MAE, followed by 10-seconds, 20-seconds, and 1-minute time interval dataset that has the lowest MAE. There is practically no inaccuracy in any dataset after the first 100 single values. A dataset's size has a substantial impact on singular values and the correctness of reconstructed data.

A smart distribution system's data is compressed in this part to see how well the approach works. To sum up, more singular values are required to fulfill TCR as a data set grows in size, as shown by the experimental findings. Nevertheless, increasing the number of singular values will reduce the amount of data that has to be compressed. As a result, there are fewer errors when the data is rebuilt after it has been compressed. As a result, a greater amount of data must be transferred through a wider range of communication channels. By compressing information with a high number of singular values to fulfill the TCR, you will have to send more data. The TCR must be matched to the quantity of data to be compressed to maximize the connection bandwidth when transferring the compressed data. The data reconstruction error can be calculated between the reconstructed data $g(i, j, s)$ and the original data $F(i, j, s)$ using

$$P(s) = \frac{1}{3MN} \sum_{i=0}^{a-1} \sum_{j=0}^{b-1} \sum_{s=0}^{2} \|g(i, j, s) - F(i, j, k)\|. \tag{13}$$

In addition, the Mean Average Error (MSE) (calculated by averaging squared error) is another way to assess reconstruction accuracy.

The MAE is defined as [29]

$$\text{MAE} = \sum_{i=0}^{a-1} \sum_{j=0}^{b-1} \sum_{s=0}^{2} \|g(i, j, s) - f(i, j, s)\|^2. \tag{14}$$

A measure of the quality of compression and reconstruction is the signal-to-noise ratio (SNR). There are two ways to define the peak SNR [30]:

$$\text{PSNR}(\text{dB}) = 10 \log_{10} \frac{\left(\text{Max}_i\right)}{\sqrt{\text{MAE}}}, \tag{15}$$

where Max_i is the maximum possible pixel value.

MD quantifies the greatest difference between original and reconstructed values. The average difference between original and reconstructed values is denoted as SSIM. For each of the formulas [31],

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

Input: Extracted features
Output: Compressed data C_d
To compute compressed value,
For $i = 1$: size$(D_{n_\text{parameters}}, 1)$
  For $j = 1$: size$(D_{n_\text{features}}, 1)$
  Weighed updates
$v_{\text{id}}(n+1) = W(it) * v_{\text{id}}^{n} + C_1 * \text{rand} * (p_{\text{id}} - x_{\text{id}}^{n})$
$+C_2 * \text{rand} * (p_{g\,d} - x_{\text{id}}^{n})$,
End
End
Data compressed features $d_{n_\text{fea}} = [d_{n_\text{fea}} \text{rand}]$
To compute, Run length encoding
Class label = unique(target)
K = length(class label)
For $d = 1$: k
  Temp = total class mean$(I, :)$
Run length decoding
Data grouping
$\text{Compressed}_{\text{Fitness value}} =$
$(-40 * q(-3 * \sqrt{\sum S_v})/2 - \exp(\sum \cos(3\pi * S_v)/d_b))$
$+10$ exp,
End

Algorithm 1: (RPSO)

Table 1: Computed CR, and number of singular values, r of compression.

TCR	1 minute		20 seconds		10 seconds		5 seconds	
	r	CR	r	CR	r	CR	r	CR
100	6	96.68	8	103.62	9	101.89	10	96.84
80	7	82.87	10	82.90	11	83.36	12	80.70
60	10	58.01	14	59.21	15	61.13	16	60.53
50	12	48.34	17	48.76	18	50.94	19	50.97
30	19	30.53	28	29.60	31	29.58	32	30.26
10	58	10.00	83	9.99	92	9.97	97	9.98
5	116	5.00	166	4.99	183	5.01	194	4.99
4	145	4.00	207	4.00	229	4.00	242	4.00

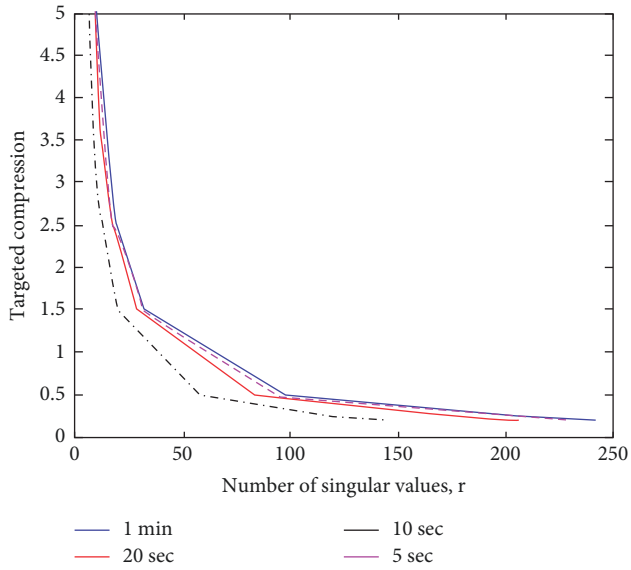


Figure 2: Plot of the number of singular values versus TCR for the different datasets.

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

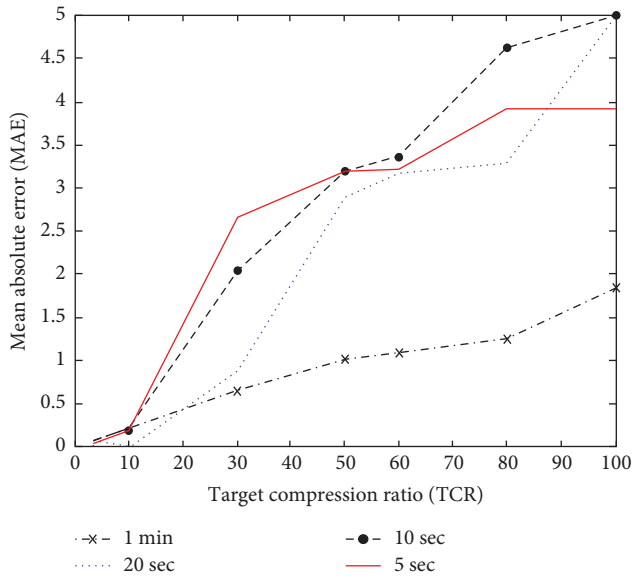


Figure 3: Plot of MAE versus TCR for different sampling rates.

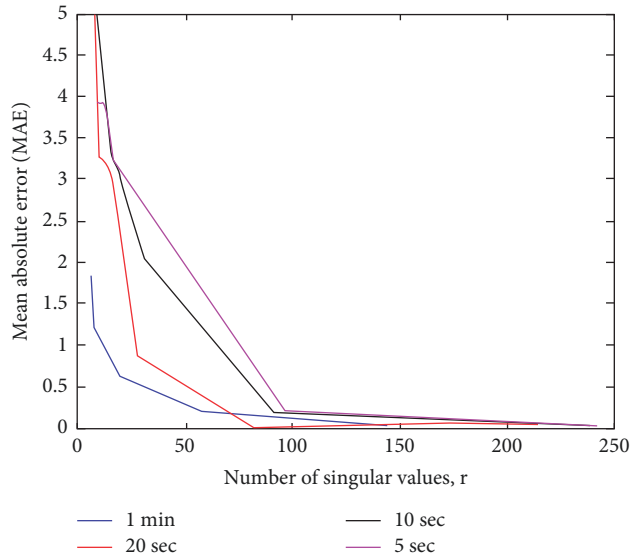


Figure 4: Plot of MAE versus the number of singular values, r for the dataset.

$$\text{SSIM} = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\sum_{s=0}^{s-1}\|I(i,j,s)\|,$$
$$\text{MD} = \max_{0\le i\le M}\|I(i,j,s)\|. \tag{16}$$

The video reconstruction error (MSE), signal-to-noise ratio (PSNR), matching distance (MD), and percent compression ratio (PCR) values are obtained as depicted in Figure 5. The satisfying results are obtained over the compression as depicted in Table 2. From Table 2 and Figure 5, the suggested methodology shows the highest performance over PSNR, MSE, and MD. As illustrated in the PSNR contours for the testing set in Figure 5, the PSNR improves as the compressed image bit rate increases. The results demonstrate a rising pattern in PSNR values, whereas MSE drops progressively as the compressed image bit rate improves. As a result, a higher compressed image bit rate means higher resolution images and fewer mistakes.

The advantages of the existing mechanism in which the high compression ratio was obtained but it takes more time for compression. Hence, it can be overcome by the proposed mechanism.

5.1. Complexity Analysis. In general, the total number of states is approximately equal to 2^N for computing nth RLE number ($F(^N)$). Notice that each state denotes a function call to 'RPSO with RLE()' which does nothing but makes another

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

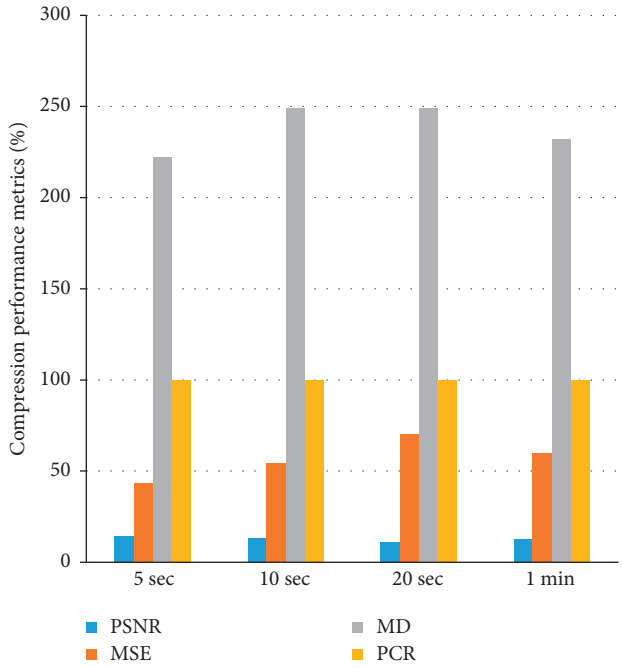


Figure 5: Image Quality metrics.

Table 2: Average Data Quality metrics.

Parameters	5 sec	10 sec	20 sec	1 min
PSNR	14.1913	13.2848	11.168	12.778
MAE	43.5359	54.1624	70.49	59.7
MD	222.1475	249.0821	249.05	232.1
PCR	100	100	100	100

Table 3: Average compression ratio.

Compression level	Number of images	Compression ratio
Discrete spatial multilayer perceptron (proposed)	100	10 : 1
Haar [17]	100	10 : 17
Cosine [17]	100	10 : 18

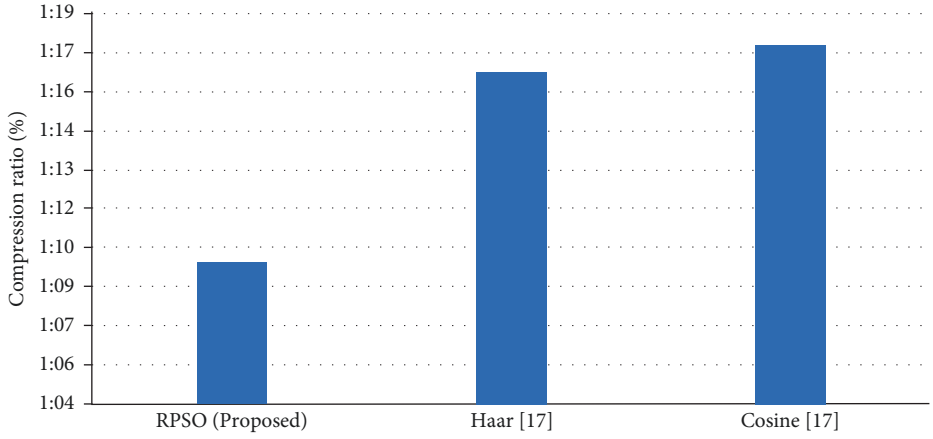


Figure 6: Compression ratio.

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

recursive call. Therefore, the total time taken to compute the nth number of the sequence is $O(2^N)$.

In digital file compression, duplication is the most important issue. If N_1 and N_2 signify the amounts of data holding units in the raw and encoded images, correspondingly, the compression ratio, CR, could be specified as $CR = N_1/N_2$ as well as the data duplication of the original image as $RD = 1 - (1/CR)$. From Table 3 and Figure 6, the proposed methodology can acquire the exact ratio of compression (10 : 1) when compared to Haar [17] (10 : 16.5) and Cosine [17] (10 : 17.2) techniques.

6. Conclusion

Data compression techniques such as RPSO compression were examined and evaluated in this article. Data from a smart distribution system was used to evaluate the algorithm with 1-minute, 10-second, 20-second, and 5-second interval datasets. The results obtained demonstrate that the amount of the data has a considerable influence on the proposed approach. Larger datasets require more significant single values to achieve low error rates. When used to the smart grid, RPSO may be used as a simple and uncomplicated compression method. The significant singular values will provide a decent approximation when the compressed data has to be rebuilt using the recommended approach. Depending on the number of singular values used, RPSO compression can lower the volume of data. However, if you have a lot of data, you should consider using the proposed compression technique, which has a faster execution time and low error rates. Also, a lot of the pointed advantages exist. There will be some disadvantages also; in the proposed work, the order of bytes is independent. Compilation needs to be done again for compression. Errors may occur while transmitting data. We have to decompress the previous data. The disadvantages can be overcome in future work.

Data Availability

The data used to verify the study's findings can be obtained from the author on request.

Conflicts of Interest

The authors state that the publishing of this paper does not include any conflicts of interest.

References

[1] V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B. B. Gupta, "An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework," *Future Generation Computer Systems*, vol. 83, pp. 619–628, 2018.

[2] Z. J. Gao and J. S. Wang, "Application of smart grid technology in the coalmine power system," *Applied Mechanics and Materials*, vol. 441, pp. 236–239, 2014.

[3] V. Tsakanikas and T. Dagiuklas, "Video surveillance systems-current status and future trends," *Computers & Electrical Engineering*, vol. 70, pp. 736–753, 2018.

[4] G. F. Shidik, E. Noersasongko, A. Nugraha, P. N. Andono, J. Jumanto, and E. J. Kusuma, "A systematic review of intelligence video surveillance: trends, techniques, frameworks, and datasets," *IEEE Access*, vol. 7, pp. 170457–170473, 2019.

[5] A. Hampapur, L. Brown, J. Connell et al., "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 38–51, 2005.

[6] L. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: a paradigm of collaborative compression and intelligent analytics," *IEEE Transactions on Image Processing*, vol. 29, pp. 8680–8695, 2020.

[7] C.-S. Yoon, H.-S. Jung, J.-W. Park, H.-G. Lee, C.-H. Yun, and Y. W. Lee, "A cloud-based UTOPIA smart video surveillance system for smart cities," *Applied Sciences*, vol. 10, no. 18, p. 6572, 2020.

[8] R. Rajavel, S. K. Ravichandran, K. Harimoorthy, P. Nagappan, and K. R. Gobichettipalayam, "IoT-based smart healthcare video surveillance system using edge computing," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2021.

[9] R. Hamza, A. Hassan, T. Huang, L. Ke, and H. Yan, "An efficient cryptosystem for video surveillance in the internet of things environment," *Complexity*, vol. 2019, Article ID 1625678, 11 pages, 2019.

[10] V. R. Prakash, "An enhanced coding algorithm for efficient video coding," *Journal of the Institute of Electronics and Computer*, vol. 1, pp. 28–38, 2019.

[11] S. A. Nandhini and S. Radha, "Efficient compressed sensing-based security approach for video surveillance application in wireless multimedia sensor networks," *Computers & Electrical Engineering*, vol. 60, pp. 175–192, 2017.

[12] T. Jiang, H. Wang, M. Daneshmand, and D. Wu, "Cognitive radio-based smart grid traffic scheduling with binary exponential backoff," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2038–2046, 2017.

[13] R. Jumar, H. Maaß, and V. Hagenmeyer, "Comparison of lossless compression schemes for high rate electrical grid time series for smart grid monitoring and analysis," *Computers & Electrical Engineering*, vol. 71, pp. 465–476, 2018.

[14] S. A. Elhannachi, N. Benamrane, and T.-A. Abdelmalik, "Adaptive medical image compression based on lossy and lossless embedded zero tree methods," *Journal of Information Processing Systems*, vol. 13, pp. 40–56, 2017.

[15] P. E. Sophia and J. Anitha, "Enhanced method of using contour let transform for medical image compression," *International Journal of Advanced Intelligence Paradigms*, vol. 14, no. 1/2, pp. 107–121, 2019.

[16] T. Kalidoss, L. Rajasekaran, K. Kanagasabai, G. Sannasi, and A. Kannan, "QoS aware trust based routing algorithm for wireless sensor networks," *Wireless Personal Communications*, vol. 110, no. 4, pp. 1637–1658, 2020.

[17] I. Yamnenko and V. Levchenko, "Video-data compression using wavelet analysis," in *Proceedings of the 2019 IEEE 20th International Conference on Computational Problems of Electrical Engineering (CPEE)*, pp. 1–4, Lviv-Slavske, Ukraine, September 2019.

[18] S. Rahimunnisha and G. Sudhavani, "Novel complexity reduction technique for multi-view video compression using HCD based genetic algorithm," *Design Engineering*, vol. 2021, no. 6, pp. 3219–3228, 2021.

[19] L. Xu, H. Liu, X. Yan, S. Liao, and X. Zhang, "Optimization method for trajectory combination in surveillance video synopsis based on genetic algorithm," *Journal of Ambient*

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press

Intelligence and Humanized Computing, vol. 6, no. 5, pp. 623–633, 2015.

[20] S. M. Darwish and A. A. J. Almajtomi, "Metaheuristic-based vector quantization approach: a new paradigm for neural network-based video compression," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7367–7396, 2021.

[21] R. B. Abduljabbar, O. K. Hamid, and N. J. Alhyani, "Features of genetic algorithm for plain text encryption," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, p. 434, 2021.

[22] B. Azam, S. Ur Rahman, M. Irfan et al., "A reliable auto-robust analysis of blood smear images for classification of microcytic hypochromic anemia using gray level matrices and gabor feature bank," *Entropy*, vol. 22, no. 9, p. 1040, 2020.

[23] D. Xiang, T. Tang, L. Zhao, and Y. Su, "Superpixel generating algorithm based on pixel intensity and location similarity for SAR image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 6, pp. 1414–1418, 2013.

[24] S. Kwon, H. Kim, and K. S. Park, "Validation of heart rate extraction using video imaging on a built-in camera system of a smartphone," in *Proceedings of the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2174–2177, San Diego, CA, USA, August 2012.

[25] T. Tuncer, S. Dogan, F. Ertam, and A. Subasi, "A novel ensemble local graph structure based feature extraction network for EEG signal analysis," *Biomedical Signal Processing and Control*, vol. 61, Article ID 102006, 2020.

[26] H. Xie, Y. Ren, W. Long, X. Yang, and X. Tang, "Principal component analysis in projection and image domains—another form of spectral imaging in photon-counting CT," *Institute of Electrical and Electronics Engineers Transactions on Biomedical Engineering*, vol. 68, pp. 1074–1083, 2020.

[27] W. Liu, Z. Wang, N. Zeng, Y. Yuan, F. E. Alsaadi, and X. Liu, "A novel randomised particle swarm optimizer," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 2, pp. 529–540, 2021.

[28] Z. Yong, Y. Li-Juan, Z. Qian, and S. Xiao-Yan, "Multi-objective optimization of building energy performance using a particle swarm optimizer with less control parameters," *Journal of Building Engineering*, vol. 32, Article ID 101505, 2020.

[29] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, pp. 79–82, 2005.

[30] Q. Huynh-Thu and M. Ghanbari, "The accuracy of PSNR in predicting video quality for different video scenes and frame rates," *Telecommunication Systems*, vol. 49, no. 1, pp. 35–48, 2012.

[31] A. K. Moorthy and A. C. Bovik, "Efficient motion weighted spatio-temporal video SSIM index," *Human Vision and Electronic Imaging*, vol. XV, Article ID 75271I, 2010.

https://doi.org/10.1155/2021/8192582 Published online by Cambridge University Press