IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 22, 2020, accepted October 8, 2020, date of publication October 28, 2020, date of current version November 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034319

Real-Time Streaming Image Based PP2LFA-CRNN Model for Facial Sentiment Analysis

CHANG-MIN KIM[1], KI-HWAN KIM[2], YOUNG SIL LEE[3], KYUNGYONG CHUNG[4], AND ROY C. PARK[5]

[1]Division of Computer Information Engineering, Sangji University, Wonju 26339, South Korea
[2]Department of Ubiquitous IT, Dongseo University, Busan 47011, South Korea
[3]Division of Computer Engineering, Dongseo University, Busan 47011, South Korea
[4]Division of Computer Science and Engineering, Kyonggi University, Suwon 16227, South Korea
[5]Department of Information Communication Software Engineering, Sangji University, Wonju 26339, South Korea

Corresponding author: Roy C. Park (roypark1984@gmail.com)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1A6A1A03040583.

ABSTRACT In modern society, the real-time emotion adaptive driving system for providing safety to drivers and emotion-based services are being researched. However, in the service process have problem of personal information might get leaked. Therefore, a robust personal information protection method is required for face recognition services based on real-time images. In this study, we propose a real-time streaming image based PingPong256 (PP2) algorithm, line-segment feature analysis (LFA), convolutional recurrent neural network (CRNN) model for facial sentiment analysis. The proposed method applied the PP2 algorithm to images for encryption and decryption for the security of the real-time images collected by image devices. For transmitting images to a server, LFA, as a dimensionality reduction algorithm, is used to extract facial information. PP2 encrypts and decrypts an image through a linear feedback shift register with a different length and sets a random value other than 0 so that inferring the initial value of encryption becomes difficult, and then executes the random operations approximately 1,000 times. The LFA analyzes the line segments of an image, assigns a different unique number depending on its type, and cumulatively adds them to generate a Line-Segment map (LS-map) with a size of 16 × 16. The LS-map is used as an input of the CRNN model designed in this study, and the facial expressions are classified. Performance evaluation compares the accuracy of face recognition by using the proposed method with the loss rate for other models. Performance evaluation renders excellence to the accuracy of face recognition and loss rate comparison.

INDEX TERMS Image security, facial expression analysis, PingPong256 security method, line-segment feature analysis, PP2LFA-CRNN, smart-driving service.

I. INTRODUCTION

The continuous development of automobiles has augmented the concept of a car from a simple transportation means to a convenient space for a driver. Based on self-driving functions, automobiles have developed in providing more convenience to drivers through multiple sensors and cameras [1]–[3]. With the evolution of technologies, a real-time emotion adaptive driving (R.E.A.D) system based on emotions of the drivers is considered as a new futuristic automobile service [4], [5]. This system is the futuristic driving concept as suggested by Kia Motors. Various technologies such as artificial intelligence (AI), camera, and automobile control technology, have been designed to enable an automobile to learn the biometric information and the emotional state of a driver. By analyzing the condition of a driver in real time, through the learned information, a car can actively provide an optimized indoor environment (music, light, air conditioning, scent, etc.) to a driver. In addition, the system offers a safety service for preventing accidents that can be caused due to the health of a driver. In essence, the R.E.A.D system is an integrated service for the convenience and safety of a driver. Toyota Boshoku proposed an indoor space model for self-driving. The service is aimed at establishing an environment for stimulating the five senses (seat, music, light, etc.) by detecting the changing condition and drowsiness of a driver or a passenger

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

using a camera to provide a safe and convenient driving experience [6]. BMW considers the futuristic connected technology that connects augmented reality, voice recognition, and home network together [7]. Automobile makers pay a great deal of attention to the development of next-generation driving service technologies based on the conditions and emotions of drivers. Although a variety of smart services make it possible to enjoy a convenient life, there is a high risk of privacy violation because the data collected in a smart home is based on the data exchange between humans and objects and includes information related to one's private life, such as personal information and personal image information. In particular, security threats to cameras, webcams, and other devices used in the establishment of a smart environment are increasing considerably [8], [9].

Most cameras that are used in smart driving support ultra-high-definition shooting. Although clearer images and videos can be collected through such shooting to provide convenient services, in the case where a transfer is made without encrypting the original copy information, the process may be exposed to diverse security threats, which may lead to critical problems. An actual case of this would be when Germany conducted an experiment in 2013 and successfully reproduced the fingerprint of the Minister of National Defense at the time. This was done using the fake fingerprint extracted from a couple of high-definition videos to unlock the iPhone 5s fingerprint sensor. In 2017, a fingerprint was discovered from the V-sign of the person included in an image shot within a 3 m distance, and this event served as an opportunity to emphasize the need to enhance the security of images and videos [10]–[12]. In addition, when a public institution uses an image file that includes the personal information of users for work purposes, personal information is stored in the server of that institution. In this case, there is a risk that the personal information of users may be exposed to other people who work at the institution. If the server of the institution is hacked from the outside, there is a risk that the personal information of the users may be exposed to many and unspecified individuals. Therefore, it is necessary to develop a more fundamental security measure applicable to the processing of image files, including the personal information of users [13]–[15]. With regard to this matter, Tekeoglu and Tosun [16], through the study ''Investigating Security and Privacy of a Cloud-Based Wireless IP Camera: NetCam'' dealt with the issue of personal information leakage through IP camera hacking. Xu *et al.* [17] reported cases where material damages were caused because personal information and control authority were seized due to network protocol vulnerabilities.

In this study, we propose a real-time streaming image based PingPong256 algorithm, line-segment feature analysis, convolutional recurrent neural network (PP2LFA-CRNN) model for facial sentiment analysis. To actualize our proposal, a smart driving facial expression recognition service, which used the Pingpong256 (PP2) algorithm and line-segment feature analysis (LFA) algorithm to encrypt/decrypt the videos collected from smart driving cameras on a real-time basis and analyzed and managed the facial expressions within the videos collected on the central management server, respectively, was studied. The PP2 algorithm that encrypted/decrypted real-time streaming videos used two linear-feedback-shift registers (LFSR) of mutually different lengths to guarantee the safety of long-term operation. The two LFSRs consisted of a total of 512-bit memory and made it difficult to analogize the initial values through random operations. This structure was used to encrypt and decrypt the videos. The LFA algorithm removed unnecessary data by decrypting and pre-processing the videos delivered after they were encrypted through the PP2 algorithm. The removed line segments of the data were analyzed, a unique number was assigned to each line segment, and the line segment-related cumulative aggregate data were calculated. This data was called the Line-Segment map (LS-map). The LS-map was learned through the convolutional recurrent neural network (CRNN) model, and the facial expressions were classified.

This study is structured as follows. The face recognition-based service in a smart driving system and security threat in real-time image transmission cases are described in Section 2. The real-time streaming image based PP2LFA-CRNN model for facial sentiment analysis is described in Section 3. The user expression recognition of performance evaluation was conducted, and a comparison is made through the preexisting CRNN model and AlexNet model in Section 4. Finally, the conclusion is specified in Section 5.

II. RELATED RESEARCH

A. FACE RECOGNITION BASED SERVICE IN SMART DRIVING SYSTEM

Recent advancement of automobile technologies has developed the systems for monitoring the conditions of drivers and preventing accidents. By acquiring information on health conditions of drivers such as fatigue, attention, and drinking, it is possible to prevent car accidents. With the development of automated driving systems, bio-signal recognition technologies have been used to acquire information for driving takeover [18]. As a bio-signal recognition security technology, face recognition has attracted a lot of attention. The combination of face analysis and AI have been actively researched for accident prevention and self-driving. Yandex detected the fatigue and distraction of a driver with the use of machine learning on the front glass of a car [19]. In collaboration with Genesis Lab, Hyundai Mobis tried to recognize the emotions of drivers by analyzing the emotions and voices of drivers using image analysis and AI technologies, and this culminated in designing a system that could detect drowsy driving, drinking under influence (DUI), driving incompetence, and could stop on the shoulder when driving autonomously. In infrared camera-based face recognition and pupil tracking systems that could detect the biometric information of a driver, including eyes, pupils, nose, mouth, and ears as well as recognize eye tracking, the function of detecting the carelessness of drivers was reviewed.

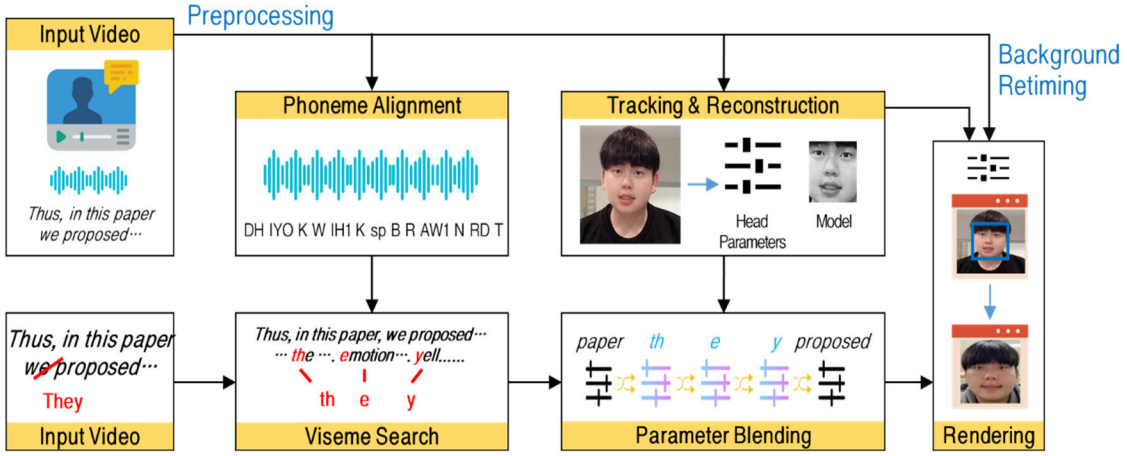


FIGURE 1. Components of image-hacking technology in Deepfake.

A variety of enterprises made efforts to reduce accidents by using face analysis technology [20]. IS Technology Co., Ltd. has released the driver status monitoring device and the cloud platform for the safety of drivers on the basis of AI and deep learning-based image recognition technology and cloud platform [21]. Driven state monitoring (DSM) recognized the conditions of a driver in real time and provided, almost perfectly, a warning for carelessness or drowsy driving. In addition, it monitored the fatigue and distraction of a driver for safe driving and activated the warning and intervention for correcting dangerous driving to provide the driver with an audio or display warning. It also detected the distraction levels of drivers and called the attention of the driver to any abnormality to induce safe driving. In addition, a danger prediction algorithm was applied to provide a safe driving service. In interaction with a variety of advanced drive assistance systems (ADAS) products, a complex safe driving service was also offered. The NUVIS center provided a service in real-time for 24h. When an event occurred, it was sent to an expert team for analysis and to a driver or a manager. Through the NUVIS center, a manager for drivers was able to respond to the warning of the fatigue and distraction of each driver to reduce accidents.

B. SECURITY THREATS IN REAL TIME IMAGE TRANSMISSION: CASE STUDIES

As smart driving services expand, diverse user behavior-based services using cameras are increasing. The performance of cameras is gradually being enhanced to provide accurate services suitable for each situation, and most of the recently released smartphone cameras support ultra-high definition. We can achieve clearer images and videos through such enhancements. However, a transfer made without encrypting the original copy may lead to diverse problems. Figure 1 shows the components of the image-hacking technology, Deepfake. Deepfake is a compound word comprising ''deep learning'' (one of the AI technologies) and ''fake''. It is an AI-based composite technology that uses deep learning and facial mapping or face swapping technologies to synthesize the face and body parts of a particular person with a totally different image to create or change fake video contents [22]. Although Deepfake can be utilized for positive purposes such as multilingual video dubbing (David Beckham's Deepfake video was manufactured in April 2019 for malaria eradication), there is a high risk that it can be misused for socially/politically negative purposes such as composite pornography, stock price manipulation, and election manipulation. When it first appeared, the difference between the original copy and the counterfeit copy was clear. However, as AI technology progressed, Deepfake also made rapid progress. Now, the difference between the original copy and the counterfeit copy is no longer clear, and a small number of original images can be used to manufacture a Deepfake video. Therefore, it is essential to develop a method to protect all images and videos sent/received through the internet from unauthorized personnel, and the most commonly used method is encryption [23]. Figure 2 shows an example of security threats in public sector organizations. As Figure 2 shows that the involved image files, including personal information of users, are continuously stored on the server run by institutions such as financial companies, public offices, and telecommunication companies. There is a risk that personal information of users may be exposed to the personnel who work at the involved institution. In the case, where the server of the involved institution gets hacked, there is a risk that the personal information of the user may be exposed to many unspecified individuals. Therefore, it is necessary to develop a more fundamental security measure applicable to the processing of image files, including personal information of users [24].

III. REAL-TIME STREAMING IMAGE BASED PP2LFA-CRNN MODEL FOR FACIAL SENTIMENT ANALYSIS

Recently, to support a safe and convenient driving environment for users, an optimized service was provided in a smart

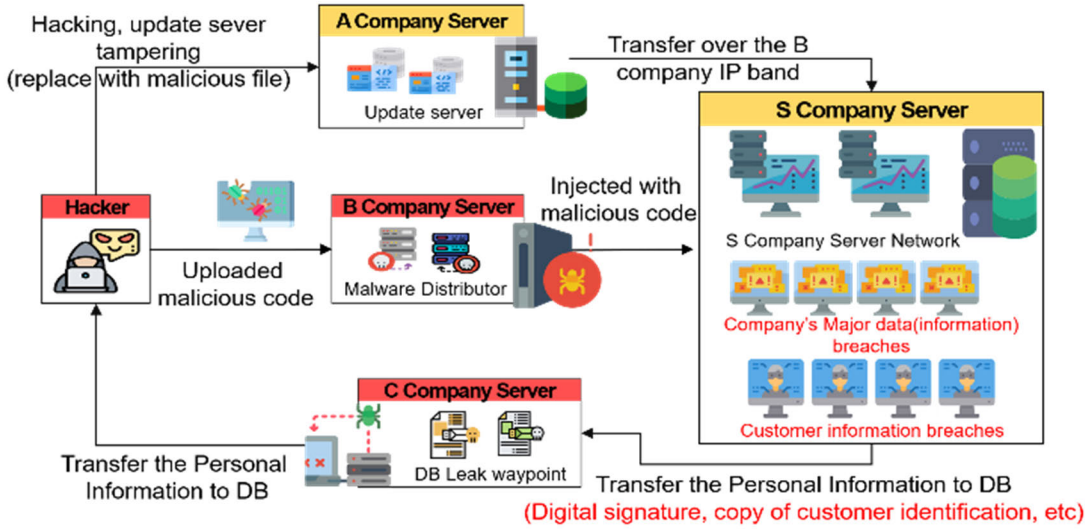


FIGURE 2. Example of security threat in public sector organizations.

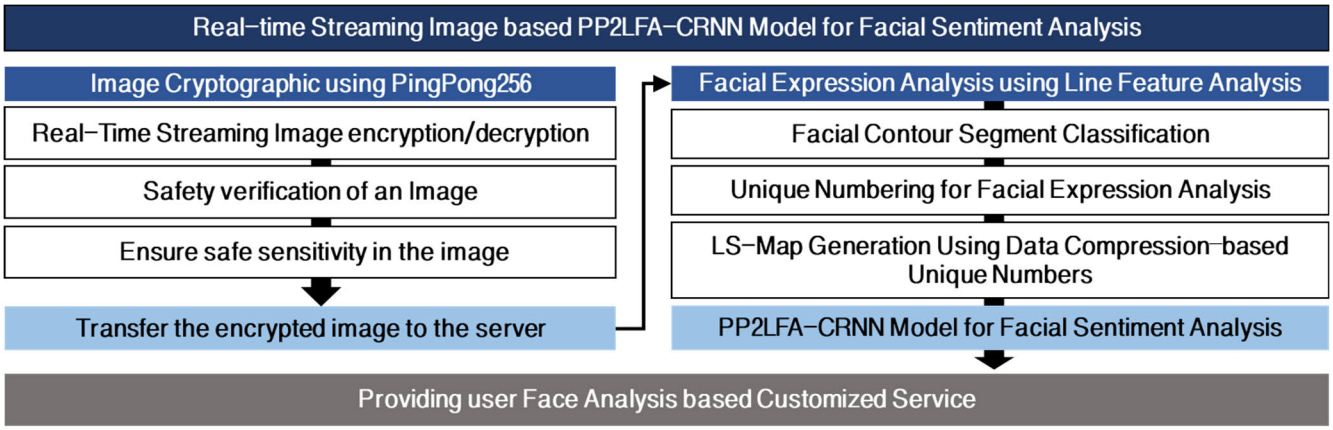


FIGURE 3. Process of a real-time streaming image-based PP2LFA-CRNN model for facial sentiment analysis.

driving environment. It used a variety of video devices that recognized the movement of the user and utilized computer vision technology to track and extract objects to provide customized services. Smart driving services provide convenience to our lives. However, since the data collected by smart driving was based on the exchange of data between people and things, it was very sensitive to safety because it contained important factors such as personal information and addition, the data collected by the smart driving system in various sensing devices were concentrated on the server and required data processing and synchronization. Smart-powered systems could pose a serious security threat if attackers use sophisticatedly crafted data. In smart driving, the standard of autonomous judgment is the driver. By adjusting the intensity of the driving assist according to the facial expression of the driver, it can provide driving convenience or inhibit the behavior of stealing control through the focused expression of the driver. Thus, we proposed a real-time streaming image-based PP2LFA-CRNN model for facial sentiment analysis. Figure 3 shows the process of a real-time streaming image-based PP2LFA-CRNN model.

As shown in Figure 3, to enhance the security of the real-time video, data was collected by a smart driving camera, the PP2 was used for encryption/decryption, and a smart driving facial expression recognition service using the LFA algorithm for analyzing and managing facial expressions within the video data on the central management server was studied. In this study, a series of processes using these two algorithms are referred to as PP2LFA.

A. REAL-TIME STREAMING IMAGE CRYPTOGRAPHIC USING PP2 ALGORITHM

In this study, the PP2 was previously applied by the research team to encrypt the streaming videos collected on a real-time basis in a smart driving environment [25], [26]. The PP2 algorithm was designed for video device security streaming

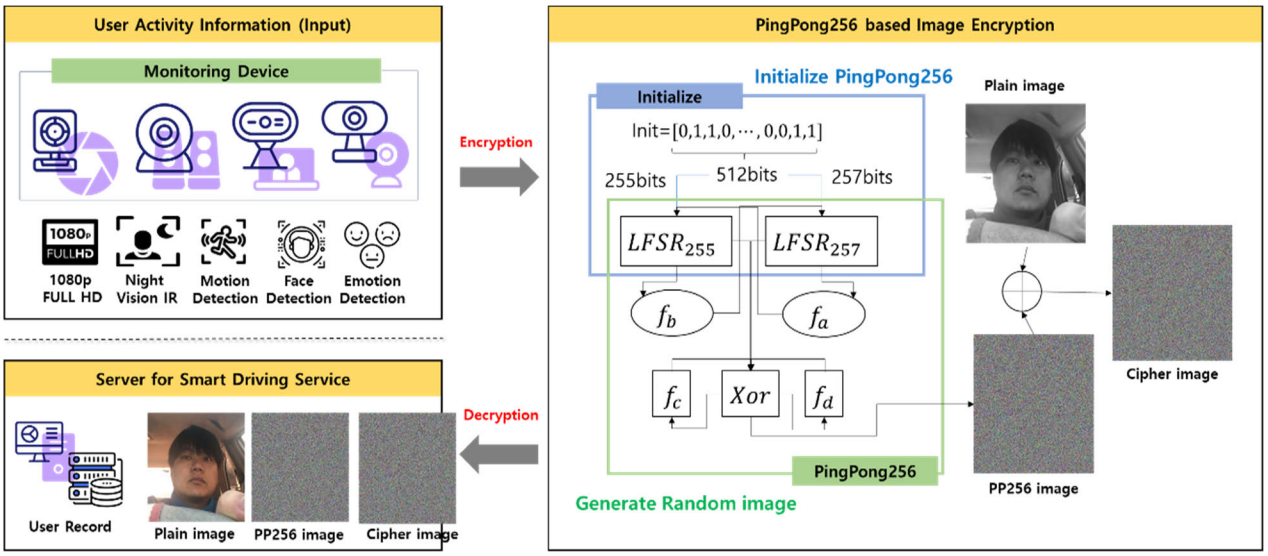


FIGURE 4. PP2-based image encryption/decryption process.

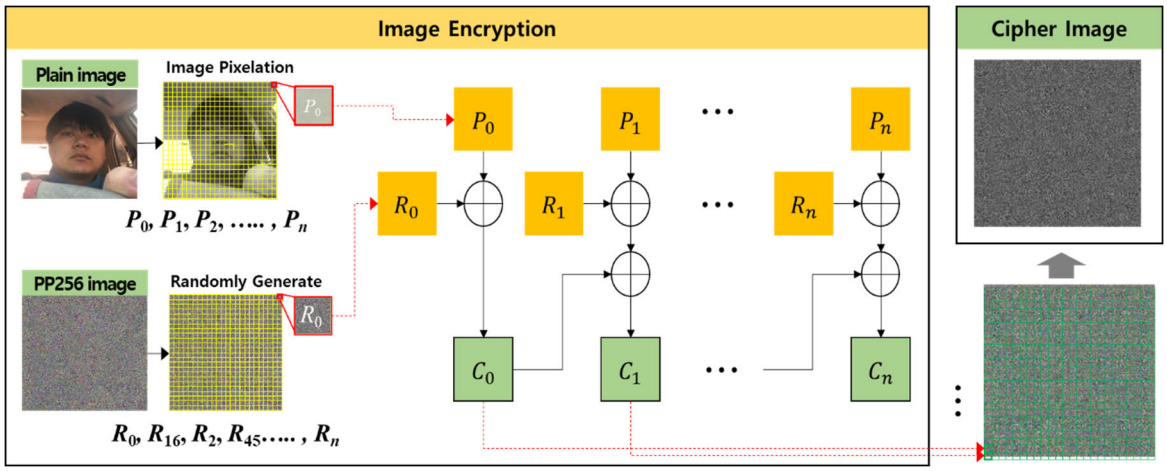


FIGURE 5. PP2-based image encryption process.

encryption that used LFSRs of different lengths and a variable clock structures. As streaming encryption had an unlimited output length, it was impossible to analogize the initial value according to the calculation results, and since it used the results omitted as many times as the random runs, it outputted irregular calculation results and cycles. For the security of all streaming videos shot with devices in a smart driving environment, the following processes were conducted: image encryption/decryption, image safety verification, and sensitivity analysis.

Figure 4 shows the PP2-based image encryption/decryption process. Images are encrypted before they are transferred to the server and decrypted when they are transferred from the server. As shown in Figure 4, PP2 uses LFSRs of different lengths, and since it has a variable clock (f_a, f_b) and memory (f_c, f_d), it is difficult to save all the cycles, and it is almost impossible to reverse engineer [27]. The two LFSRs used by PP2 consisted of 512-bit memory. In the initialization stage, the 512-bit memory was set to a random value that was not 0, and then approximately 1,000 random operations were run to make it difficult to analogize the initial value. Figure 5 shows the PP2-based image encryption process. Initially, a random array sharing the same size as the original image is generated and converted into an image through the PP2 algorithm. Then, as shown in Figure 5, a pixel unit exclusive-OR (XOR) calculation is run on the original image, the random array is converted into an image, and the final encrypted image is generated. When the random array generated through the PP2 algorithm is converted into an image, it shows characteristics similar to those of noise. During the initial calculation, the XOR calculation is run on the original pixel of the image (P_0) and the random number pixel of the image (R_0), and the result (C_0) is the output and serves as the initial pixels of the encrypted image.

Then, all the pixel unit calculations are outputted by running an XOR calculation on the original image (P_i), the random-number image (R_i), and the previously encrypted calculation result (C_{i-1}). An image is encrypted with the uses of the random image converted from PP2 on the basis of an original image and the XOR operator. Accordingly, image encryption can be represented by Equation (1). An encrypted image is defined as C_i, a plain image as P_i, and a random image as $random_{\mathrm{i}}$. At this time, $C_{\mathrm{i}-1}$ is 0.

$$C_i = P_i \oplus random_i \oplus \mathrm{C_{i-1}} \tag{1}$$

Algorithm 1 PP2 Function

Input: *LFSRinit* $= [x_0, x_1, \ldots, x_{512}]$, *len* = *bit length of image*
def *PingPong256(len)*
$LFSR_{255} = [x_0, x_1, \ldots, x_{254}]$
$LFSR_{257} = [x_{255}, x_{256}, \ldots, x_{512}]$
$A_i = b_i = c_i = d_i = 0$
$f_a = f_b = 1$
// Clock control
for *i* ***in*** *range(len):*
for *clk_0* ***in*** f_a*:*
$a_i = LFSR_{255}$
for *clk_1* ***in*** *fb:*
$b_i = LFSR_{257}$
$f_a = b_i$
$f_b = a_i$
// XOR operation
$c_i = (a_i b_i)|(a_i \oplus b_i) \oplus c_{i-1}$
$d_i = b_i|(a_i \oplus b_i) \oplus d_{i-1}$
$random_i = a_i \oplus b_i \oplus c_i \oplus d_i$
Output: *random*

Algorithm 1 shows the pseudocode of PP2 to generate random numbers. For the generation of a random number, the variable clock variables f_{a} and f_{b} in Equation (2) are calculated to be approximately 83 of 255-bit LFSR, 193$^{\mathrm{rd}}$ memory, 85 of 257-bit LFSR, and 198$^{\mathrm{th}}$ memory in the PP2 based image encryption area, as shown in Figure 4. In this way, f_{a} and f_{b} are determined to be the outputs ranging from 1 to 4. A 257-bit LFSR gives the result after f_{a} repetition to a_{i} as an output, and 255-bit LFSR gives the result after f_{b} repetition to b_i as an output.

$$\begin{aligned} f_{\mathrm{a}} &= 2 * LFSR_{255}\,[83] + LFSR_{255}\,[193] + 1 \\ f_{\mathrm{b}} &= 2 * LFSR_{257}\,[85] + LFSR_{257}\,[197] + 1 \\ &for\ (c = 0;\ c < f_b;\ i++)\ a_i = LFSR_{255} \\ &for\ (c = 0;\ c < f_a;\ i++)\ b_i = LFSR_{257} \end{aligned} \tag{2}$$

f_c and f_d are the memories used for the complexity of random outputs. f_c can be represented by c_{i} and f_{d} by d_{i}. The memory function calculation utilizes the previous outputs c_{i-1} and d_{i-1}. The initial value of all is 0, and the final random number z_i is operated at the time when the memory operation is complete. These random numbers are calculated using Equation (3).

$$\begin{aligned} c_i &= a_i b_i \oplus (a_i \oplus b_i) \oplus c_{i-1} \\ \mathrm{d_i} &= b_i \oplus (a_i \oplus b_i) \oplus d_{i-1} \\ \mathrm{z_i} &= a_i \oplus b_i \oplus c_{i-1} \oplus d_{i-1} \end{aligned} \tag{3}$$

Algorithm 2 Image Encryption

Input: *Plainimage*
def *Image Encrypt(Plainimage):*
pp2 = PingPong256(Plainimage)
Cipherimage[0] = pp2[0]^*Plainimage[0]*
for *i* ***in*** *range (1, len(Plainimage)):*
Cipherimage[i] = pp2[i]^*Plainimage*[*i*]^
Cipherimage[*i-1*]
Output: *Cipherimage*

Algorithm 2 shows the process of image encryption. In Algorithm 2, the input is a plain image (or original image), and the output is a ciphered image (or encrypted image). PP2 works with a 512-bit initial value such that its key space is 2^{512}. In terms of key sensitivity, a key sensitively reacts to the initial value, and the output is changed by the variable clock. If a random number image whose randomness is guaranteed by PP2 is encrypted for an original image and each pixel in the encryption procedure, it is possible to obtain an encryption image. A server can decrypt the transferred encryption image. Algorithm 3 shows the process of image decryption. The input is a ciphered image, and the output is a plain image.

Algorithm 3 Image Decryption

Input: *Cipherimage*
def *Image Decrypt(Cipherimage):*
pp2 = PingPong256(Cipherimage)
Plainimage[0] = *pp2*[0]^*Cipherimage*[0]
for *i* ***in*** *range (1, len(Cipherimage)):*
Plainimage[*i*] = *pp2*[*i*]^*Cipherimage*[*i-1*]^
Plainimage[*i-1*]
Output: *Plainimage*

Figure 6 shows the PP2 based image decryption process. The image decryption utilizes the same initial value as PP2 to recover its original image. Equation (4) is used for image decryption. With the use of an encryption image, XOR operation with a random image is repeated. In this way, it is possible to decrypt a plaintext image fast.

$$P_i = C_i \oplus random_i \oplus \mathrm{C_{i-1}} \tag{4}$$

The second process is image safety verification. From the perspective of an attacker, to seize an encrypted image transferred through the network and restore the original image, the attacker must know the random array converted into an image, and the attacker also must know the encryption algorithm as well as the initial value. PP2 uses a pseudo-random

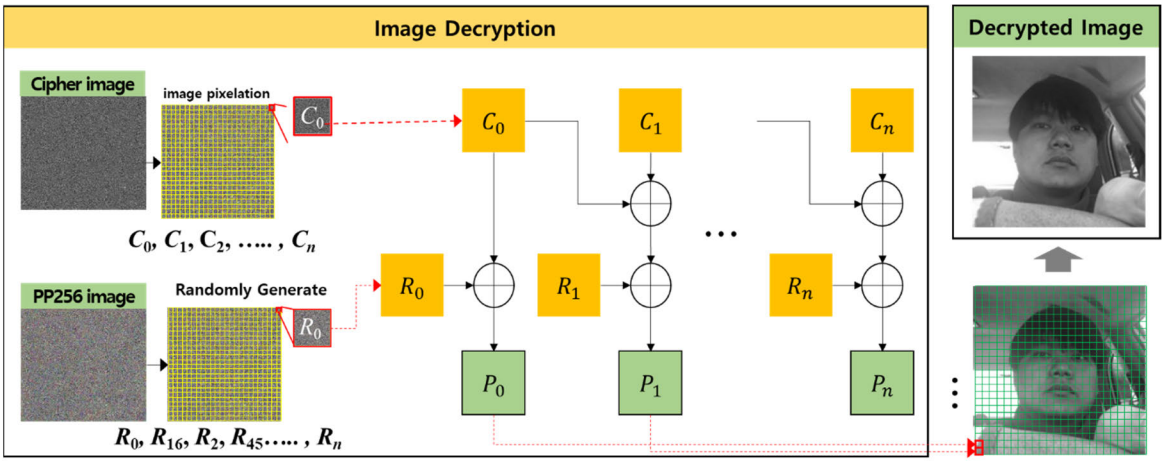


FIGURE 6. PP2-based image decryption process.

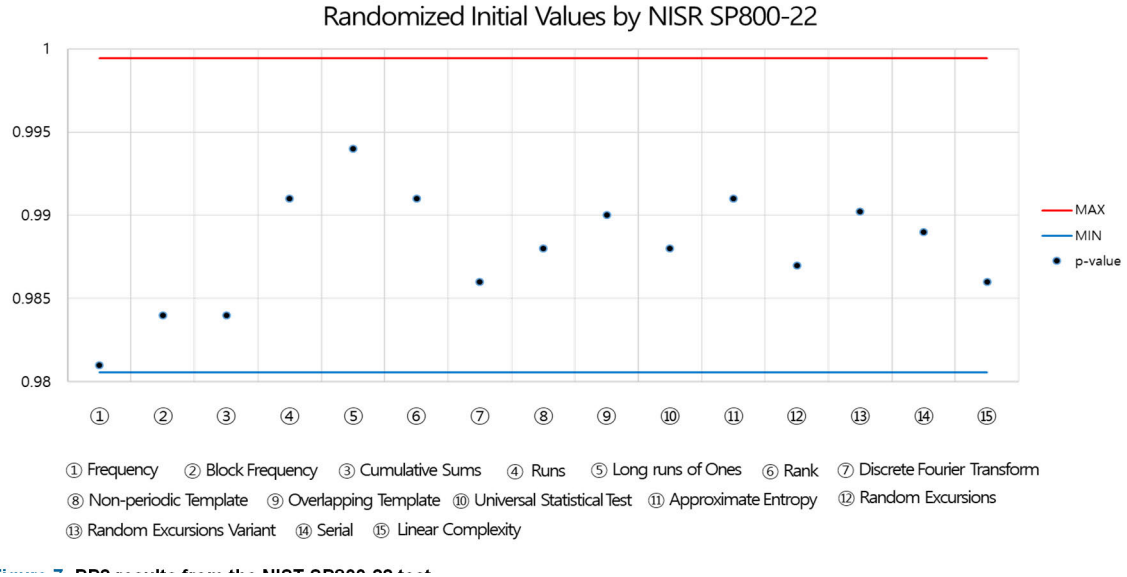


FIGURE 7. PP2 results from the NIST SP800-22 test.

number generator (PRNG) that uses a random initial number to generate random numbers. This means that even when an attacker finds the encryption algorithm type, it is impossible to analogize the initial value through the random output array. In general, the PRNG verification method recommended by the National Institute of Science and Technology (NIST) to certify that the results generated through a random algorithm are statistically random numbers is NIST Special Publication (SP) 800-22.

NIST SP800-22 utilizes 16 mathematical and statistical approaches to certify that the random PRNG calculation results, generate random numbers with high probability, and to satisfy this, all the verification results (p-values) must exceed 0.01 [28]. Therefore, in this study, to evaluate the performance of PP2 through NIST SP800-22, tests were conducted by setting a key and generating the data on a random basis. Figure 7 shows the PP2 results from the NIST SP800-22 test. As shown in Figure 7, the experimental results confirms that the p-value exceeds 0.01 in all the 16 tests.

NIST SP800-22 calculates p-value from 16 statistical viewpoints, and p-value can be seen as a method of expressing random number properties in each test technique. To prove the randomness of the random number generator from a statistical point of view using SP800-22, generate 1,000,000,000 bits with the random number generator and treat 1,000,000 bits as one sample so that the sample size (m) is 1,000 and the significance level (α) is 0.01. The average ratio ($\hat{p}$) is expressed from 0.99 to $\hat{p} = 1 - \alpha$. In this case, the number of times each test is satisfied for all samples should be recorded to satisfy the range of Equation (5).

$$\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}} = 0.99 \pm 0.0094392 \tag{5}$$

To verify PingPong256, if the sample per time is 1,000,000 bits and the significance level is set to 0.01, the range that satisfies the range of Equation (5) is a value between 0.9805607 and 0.9994392 at the minimum. The value of 16 items can be calculated by summing up the number of passes passed when testing 1000 samples with 16 items and dividing by 1000. As a result of the verification, it can be seen that it falls within the normal range in all tests as shown in Figure 7. As far as an image encrypted by PP2 is concerned, it is impossible to visually determine whether the encrypted results are genuinely random. Therefore, differential cryptanalysis was used to confirm whether it was possible to decrypt the results. Differential cryptanalysis refers to the process in which the output value changes based on the changes in the input value. The procedures are as follows. When an attacker randomly selects the original image, the attacker can randomly modify and encrypt a part of the original image and observe the calculated result changes to find the encryption key. In general, where the input value is an image, to verify it with differential cryptanalysis, the following methods can be used: the number of pixel change rate (NPCR) and unified average changed intensity (UACI) [29], [30]. NPCR and UACI use hypothesis tests with significance levels of 5%, 1%, and 0.1%, and where an encrypted image is input and passes all the significance levels, it is logically verified as random. Equation (6) shows the NPCR. When the image size is $M \times N$, $D(i, j)$ is a function that outputs 1 when the pixels sharing the same position in different images are different and outputs 0 when they are the same.

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i, j) \times 100\% \tag{6}$$

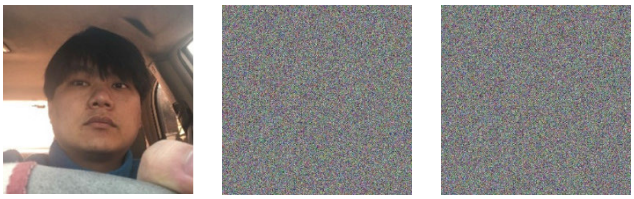
Equation (7) shows UACI. When the image size is $M \times N$, $D(i, j)$ is the absolute value calculated by converting the pixels sharing the same position in different images into whole numbers and subtracting them.

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|c_1(i, j) - c_2(i, j)|}{255} \times 100\% \tag{7}$$

Figures 8 and 9 are encrypted and decrypted using a color image and a grayscale image, respectively. Figure 8 shows the original color image, which is a random image generated using the PP2 algorithm, and an encrypted image. Figure 9 shows the result of the decrypted image using the encrypted gray-scale image with a random image generated using the PP2 algorithm.

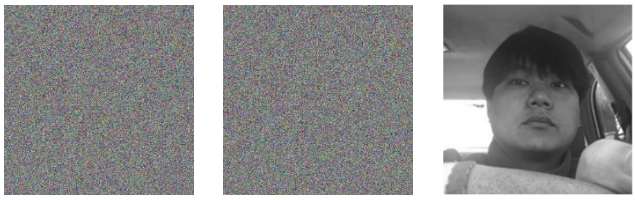
Table 1 lists the results of the NPCR/UACI randomness test for the color crypto image. The results in Table 1 prove that the color image passed the test of the hypothesis and that the original color image and the encrypted color image are different at significance levels of 5%, 1%, and 0.1%.

Table 2 lists the results of the NPCR/UACI randomness test for gray-scale crypto. The results in Table 2 prove that the gray image passed the test of the hypothesis and that the original monochrome image and the encrypted monochrome image are different at significance levels of 5%, 1%, and 0.1%. Thus, at significance levels of 5%, 1%, and 0.1% with NPCR and UACI, by checking whether the result of PP2 is a random number based on NIST SP800-22 and also by checking the outline from which the original copy can be inferred (if parts of the image are found), we find that both the color and monochrome images differ.



(a) Original (b) Random (PP2) (c) Encrypted

FIGURE 8. Result of color image encryption.



(a) Encrypted (b) Random (PP2) (c) original

FIGURE 9. Result of grayscale image decryption.

TABLE 1. Results of NPCR/UACI randomness test for Color crypto images.

	Level		5%	1%	0.1%
NPCR	Color	Value (%)	*99.5893*	*99.5810*	*99.5717*
	Red	***99.6078***	PASS	PASS	PASS
	Green	***99.5941***	PASS	PASS	PASS
	Blue	***99.6185***	PASS	PASS	PASS
	Level		5%	1%	0.1%
UACI	Color	Value (%)	*33.373 ~ 33.554*	*33.345 ~ 33.583*	*33.312 ~ 33.616*
	Red	***33.3769***	PASS	PASS	PASS
	Green	***33.4610***	PASS	PASS	PASS
	Blue	***33.5220***	PASS	PASS	PASS

TABLE 2. Results of NPCR/UACI randomness test for Gray crypto images.

	Level		5%	1%	0.1%
NPCR	Color	Value (%)	*99.5893*	*99.5810*	*99.5717*
	Gray	***99.5964***	PASS	PASS	PASS
	Level		5%	1%	0.1%
UACI	Color	Value (%)	*33.373 ~ 33.554*	*33.345 ~ 33.583*	*33.312 ~ 33.616*
	Gray	***33.4911***	PASS	PASS	PASS

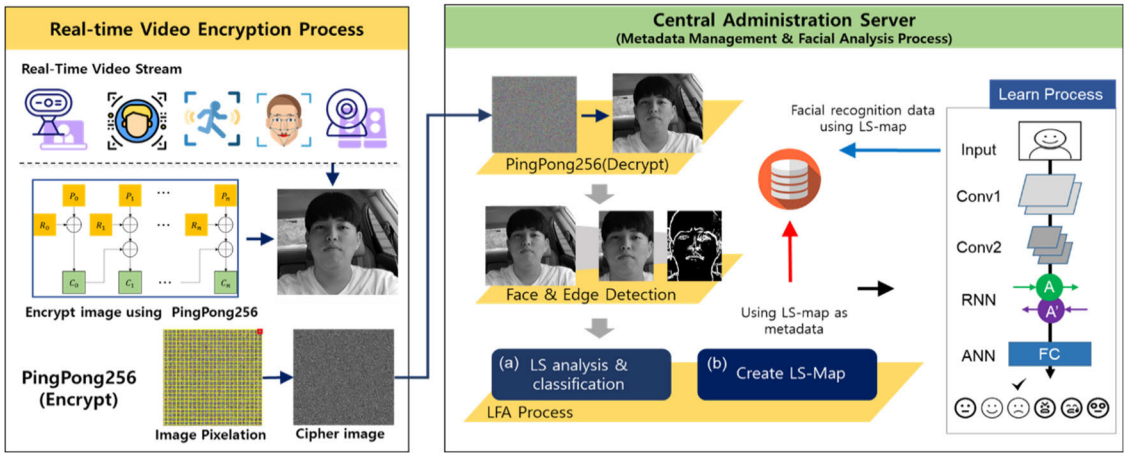


FIGURE 10. Process of real-time streaming image-based data compression.

B. FACIAL EXPRESSION ANALYSIS USING LFA ALGORITHM

This section describes the line-segment analysis algorithm for analyzing and managing images collected and delivered from video devices and stored on a central management server. An image fundamentally consists of points, lines, and planes (hereinafter referred to as line segments). Objects of diverse shapes can be expressed, and a three-dimensional (3D) effect can be added through such line segments. The algorithm used the line segment information and compressed the data, as shown in Figure 10, for memory capacity reduction. Figure 10 shows the process of the real-time streaming image-based data compression.

The LFA algorithm converted and collected line segments (the fundamental elements of an image) and generated a two-dimensional (2D) array with a size of 16 × 16. This 2D array was known as the LFA-feature map, and this LFA-feature map was learned by the CRNN model to classify facial expressions. It was impossible to restore a compressed LFA-feature map, but the image contained strong features. Such LFA-feature maps were used to classify the facial expressions, and the feature maps were utilized and managed, instead of the original image, as metadata. Then, facial expression analysis using the LFA algorithm was conducted as a two-step process. The first step was the facial contour segment classification and unique numbering for facial expression analysis. The second step was the LS-Map generation using data compression-based unique numbers.

- *Step 1 (Facial Contour Segment Classification and Unique Numbering for Facial Expression Analysis):* First, the face contour segments were classified and a unique number was assigned for facial expression analysis. The encrypted data delivered to the central server were decrypted as specified in Section 3.A. The calculated image was converted into contour data through pre-processing, the line segment information contained in the contour data was classified into 16 different types, and a unique number was assigned to each type. LFA makes use of the line-segment information on the facial regions in an analyzed image. To extract a face from a real-time image, we apply the Haar based cascade classifier using multiple AdaBoost [31].

The classifier can effectively detect facial regions in real-time images and can robustly detect such features as rotated faces, glasses, and mustaches. The LFA algorithm is a line segment-based data compression method, and the contour detection process was conducted in advance to obtain the line segment information of an image. This study detected a face from a real-time image with the use of the classifier and extracted the facial contour with the use of the Canny edge detection technique. Based on the experimental results, the optimal technique was selected. Of the various contour extraction techniques, Canny algorithm showed the best result.

TABLE 3. Type and unique number of each line segment according to the scan area.

Scan Area	Unique Number	Line Type	Scan Area	Unique Number	Line Type
0000	0	Non-Activity	1000	8	Point
0001	1	Point	1001	9	Vertical
0010	2	Point	1010	10	Diagonal
0011	3	Horizontal	1011	11	Curve
0100	4	Point	1100	12	Horizontal
0101	5	Diagonal	1101	13	Curve
0110	6	Vertical	1110	14	Curve
0111	7	Curve	1111	15	Activity (Side)

The extracted contours had a value of 0 or 1, and comprised information such as the eyes, nose, and lips (the fundamental elements of a face). Such information was classified into 16 line segment types. Table 3 lists the type and the unique number of each line segment according to the scan area. To classify these types, a filter f, having a parameter from {1, 2, 4, and 8} was used. A contour image was divided into 16 equal parts, and filter f was used to calculate each part of

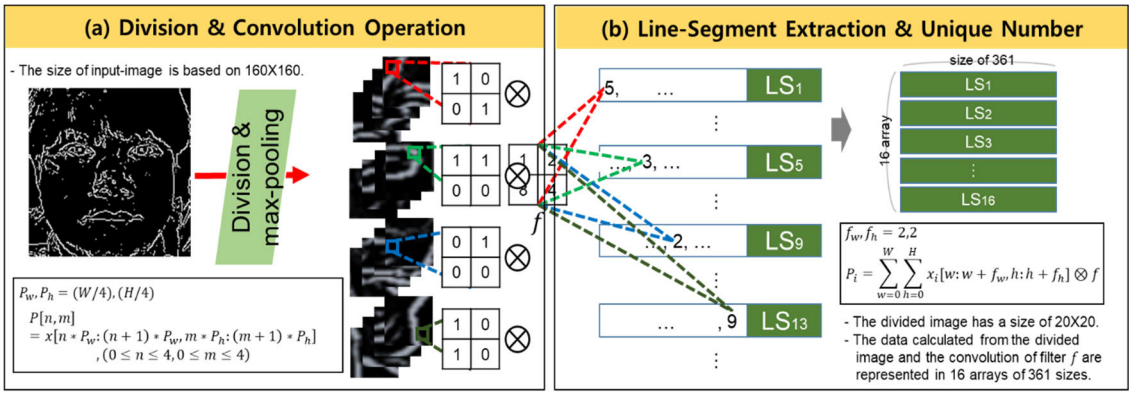


FIGURE 11. Process for extracting line segments and assigning unique numbers using filter f. (a) Extract line segment and (b) Specify unique numbers by line segment.

the divided image. The filter f has the size of 2 x 2, and has the filter coefficient of 2^n (1, 2, 4, and 8. The response values calculated through scanning are classified into sixteen types of line segments.

As far as this calculation process was concerned, as shown in Figure 11, f traveled the entire area of each divided image (hereinafter referred to as the fragmented image) and replaced every value of 1 with the corresponding parameter value of f, for all overlapping image areas. Figure 11 shows the process where the line segments are classified from the contour data, and a corresponding unique number is assigned to each line segment. As shown in Figure 11(a), contour data sized 160 × 160 are divided into 16 equal parts to generate fragmented images at 40 × 40, and the data are reduced to 20 × 20 through the max-pooling process. In the equation shown in Figure 11(a), P represents a segmented image, and n and m represent the number of segmentations by row and column, respectively. W and H are the sizes of an input image, and P_w and P_h are the sizes of the segmented images. Through this process, the internally expressed noise could be removed, and the line segment-related directivity could be minimized. As shown in Figure 11(b), the line segments expressed within a fragmented image at 20 × 20 are analyzed through a filter f. For example, as shown in Figure 11(a), where $\{\{1, 0\}, \{0, 1\}\}$ exist within a fragmented image, values of $\{\{1, 0\}, \{0, 4\}\}$ are obtained when the convolution between the fragmented image and the filter f is calculated, and a value of 5 is obtained by adding the values. (These values are called 'unique number'.)

In the equation shown in Figure 11(b), x_i represents the image calculated in Figure 11(a), W and H represent the size of x_i and f_w, respectively, and f_h represents the size of f. P_i is the convolution result of the input x and the filter f, and P is the LS-map. Through this process, the filter f compared overall pixels of each fragmented image and calculated arrays ranging from LS1 to LS16. The LS array has a size of $361 (= (20 - 2 + 1) \times (20 - 2 + 1))$, and the internal parameters had a unique number assigned to each line segment type. Algorithm 4 shows the LS extraction and creates a unique number.

Algorithm 4 LS Extraction & Create Unique Number

```
Input: [x_1, x_2, ..., x_n]
def Create unique number
    MASK = [[1,2], [4,8]]
    Y = List()
    for x_i in [x_1, x_2, ..., x_n]:
        // 361 means the size of the input image and
        the size of the data
        // computed with filter f: (20 − 2 + 1)×
        (20 − 2 + 1) = 361
        numList = List()
        for p_i in x_i:
            uni_num = List((361, 1))
            // W, H means the size of p_i; m_w, m_h is size
            of MASK.
            for w from 0 to W:
                for h from 0 to H:
                    uni_num[(W × w) + h] = sum(p_i
                    [w : w+m_w, h : h+m_h] × MASK)
                    numList.append(nui_num)
        Y.append(numList)
Output: Y[Y_1, Y_2, ..., Y_n]
```

• *Step 2 (LS-Map Generation Using Data Compression-Based Unique Numbers):* Second, an LS-Map was created using a unique number based on data compression. The unique numbers assigned to different line segment types (calculated as explained in Section 3.2.1) were converted into cumulative aggregate data to generate a 2D feature map. This feature map is referred to as an LS-map. Algorithm 5 shows the creation process of an LS-map using the unique numbers as an index. Figure 12 shows the process where a unique number is used to generate the LS-map. The cumulative aggregate data of the line segment were calculated based on the different unique numbers assigned to the line segment types. Through this process, the distribution rate of the same line segment could be measured. As shown in Figure 12(a), the LS array data having the unique numbers of the line segments as the

Algorithm 5 Create an LS-Map Using the Unique Numbers as an Index

// The input consists of 16 elements, each of which is
// an array type consisting of a unique number.
Input: $[x_1 = [u_1, u_2, \ldots, u_{361}], x_2, \ldots, x_n]$
def *Cumulative aggregation algorithm*
  map_list = List()
  for x_i ***in*** $[x_1, x_2, \ldots, x_n]$:
    LS-map = List((16, 16))\{0,\}
    // DIV is the number of data divisions, consisting of $(4 \times 4 =)16$.
    for *w* ***from*** *0* ***to*** *DIV:*
    // x_i is data that list the unique numbers for the line segments.
      for *h* ***in*** x_i:
        LS-map$[w, h] += 1$
    map_list.append(LS-map)
Output: *map_list*

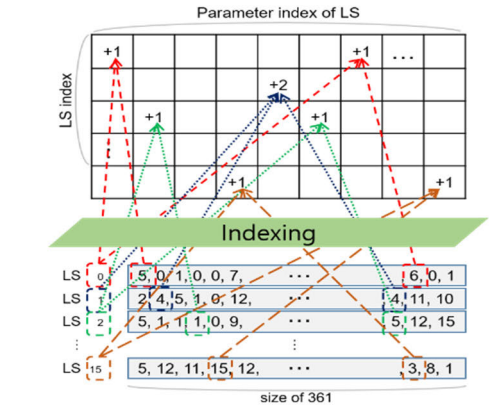
parameters are utilized as the index values of an LS-map sized at 16 × 16. The number of arrays is recognized as the X-coordinate, and the parameter of the LS array is utilized as the Y-coordinate to increase the involved position value of the LS-map by 1. For example, as shown in Figure 12(a), where the LS_0 array consists of parameters from $\{5, 0, 1, 0, 0, 7, \ldots\}$, X is set as 0, and Y is set as $\{5, 0, 1, 0, 0, 7, \ldots\}$. Through this process, the positions of the LS-map at coordinates $\{0, 5\}$, $\{0, 0\}$, $\{0, 1\}$, $\{0, 0\}$, $\{0, 0\}$, and $\{0, 7\}$ are increased by 1 each time a call is made. The data calculated through this process are shown in Figure 12(b).

Specifically, the LFA algorithm calculated the contour to compress the original image in advance, examined the types of line segments that made up the contour, assigned a unique number to each line segment type, and thereby converted the visual data into a series of patterns.

In addition, the image data can be completely compressed by using the distribution rate of the same unique numbers to digitize and express the image data. As far as the data calculated through such a process was concerned, it was impossible to completely identify or restore the original image information. In this study, the LFA algorithm was used to completely compress the facial expression recognition-based smart driving camera videos. This process made it impossible to obtain the desired information, even when data leakage was caused by problems such as hacking. In addition, it made it possible to manage high-resolution images at a low capacity.

C. PP2LFA-CRNN Model for Facial Sentiment Analysis

This section describes the PP2LFA-CRNN model for the facial sentiment analysis process. The LS-map generated through the PP2LFA algorithm was the aggregated data collected by classifying face contours into 16 line segment types, and the map itself could be considered as a major feature. The max-pooling process could not be used within the model; the stride was set to 1, and the padding was set to 1. No



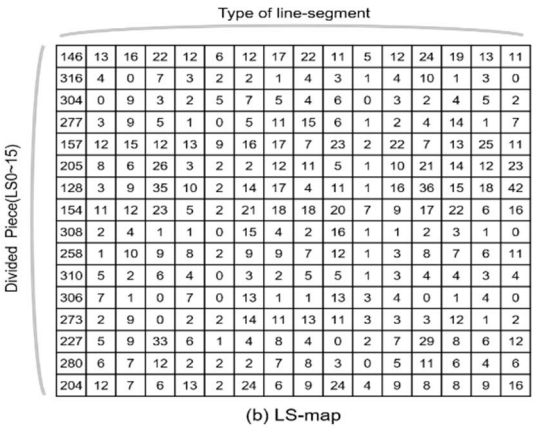


FIGURE 12. LS-map using unique numbers as index.

separate reduction or space adjustment was required to learn the LS-map, and there was the possibility that the main features might be removed or distorted during the adjustment. Figure 13 shows the proposed CRNN model for learning the LS-map. It consists of the following: one convolutional layer, a general resource unit (GRU), reshape density, dropout, and sigmoid functions. As far as the convolution layer was concerned, the filter size was set to 16 × 16 to express the input LS-map diversely. Through this setting, one LS-map mutually constructed different maps from the weighted values of diverse filters, and generated a total of 64 feature maps. The activation function used in this process was a rectified linear unit (ReLU), and the unnecessary data were removed through batch normalization. The feature map calculated through the convolutional layer was reconstructed into one-dimensional (1D) data through reshaping, and such data were used as an input to the recurrent neural network (RNN) layer. The RNN consisted of two layers, and each layer used the GRU cell to circulate. One layer consisted of 64 nodes, and the other layer consisted of 32 nodes. The data calculated through the RNN process was run through the density (ReLU) function and finally arrived at the sigmoid function. A dropout was arranged between the density and sigmoid functions to reduce the node calculation. The CRNN designed for LFA

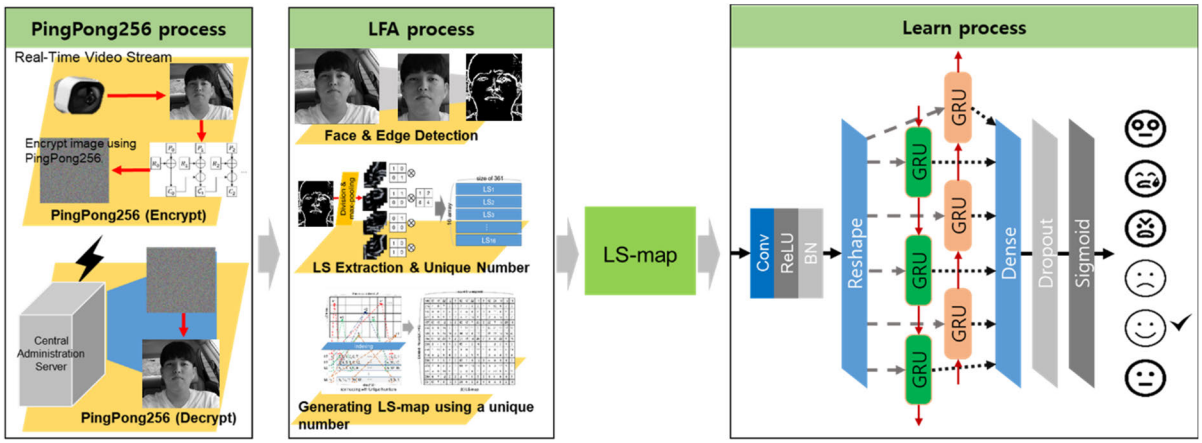


FIGURE 13. Proposed CRNN model architecture for LS-map learning.

learning was referred to as PP2LFA-CRNN. To evaluate the performance of the PP2LFA-CRNN model proposed in this study, its performance was compared with that of the AlexNet model and that of the selected CRNN model based on images taken from the Extended Cohn-Kanade Dataset (CK+) and Japanese Female Facial Expression (JAFFE) database.

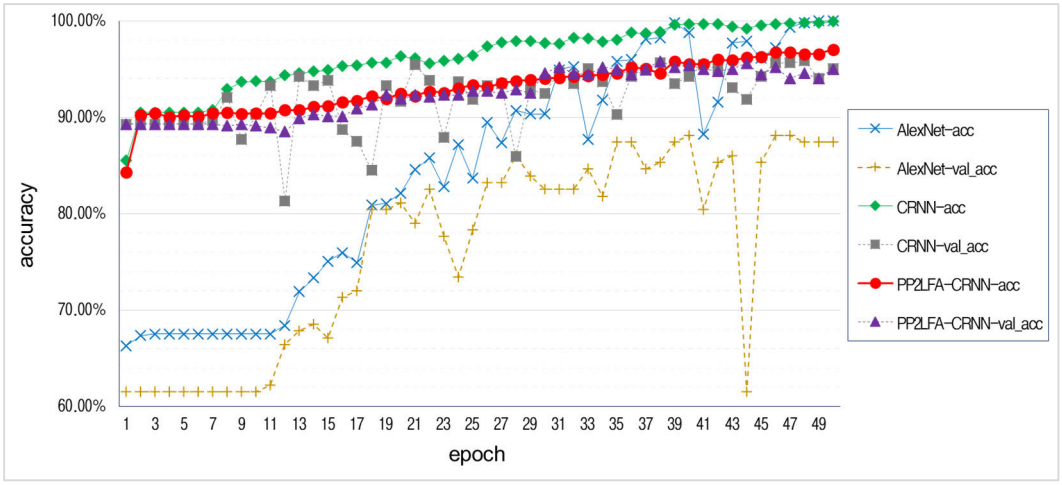
IV. PERFORMANCE EVALUATION

Simulation was conducted in the following environment: a 64-bit Intel® Core CPU i7-6700 with 16 GB RAM and a VGA NVIDIA GeForce GTX 1060 with 6 GB of memory. As far as the test was concerned, to examine the accuracy of the PP2LFA-CRNN face recognition model proposed in this study, the performance evaluation was conducted using the CK+ [32] and JAFFE databases [33], and it was compared with a CRNN [34] and AlexNet [35].

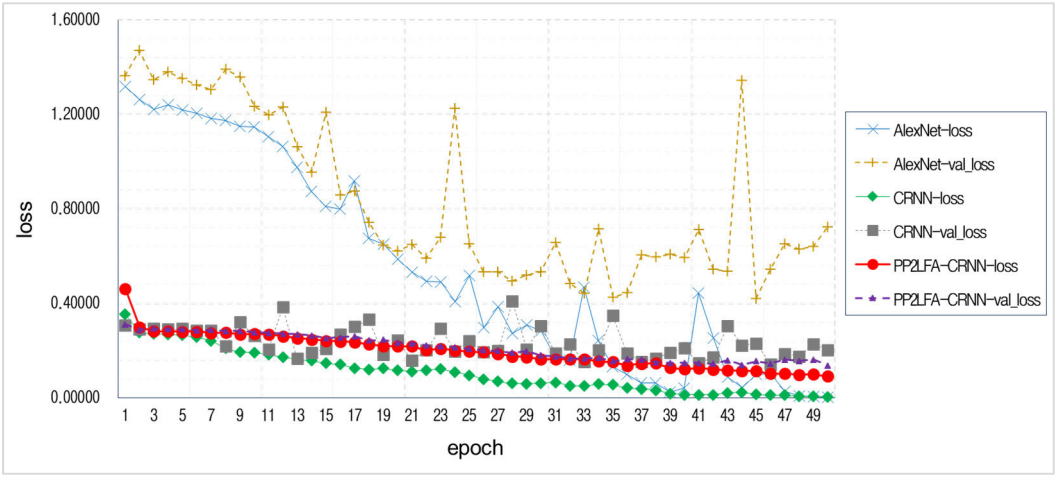
Each database was divided into seven facial expression classes, and each classified image was sized 160 × 160. In this study, the test was conducted using 645 data, classifying 20% from each database as the test data and utilizing 10% of the remaining 80% as the verification data. To classify the data and to prevent them from leaning towards a particular class, a specific percentage per class was designated. Figure 14 shows the learning rate and the loss measurement results obtained using the CK+ database. The data calculated through pre-processing (face detection and contour extraction) and LFA conversion were used as the input for PP2LFA-CRNN model.

For the CRNN and AlexNet performance comparison, data calculated through the face detection process were used [36], [37]. As shown in Figure 14, the preexisting CRNN model shows the highest learning rate, and the remaining models are ranked with the AlexNet model as the second and the PP2LFA-CRNN model as the third. However, the AlexNet and CRNN models show a large gap between the learning rate and the verification data along with a temporary decrease in the learning rate due to problems such as overfitting. The loss data shows similar patterns. When epoch = 45, the learning and loss rates of AlexNet model fluctuate severely. The CRNN model shows a smaller gap between the learning rate and the verification data, compared to the AlexNet model, but the gap increases as the epochs progress. In addition, we confirm that the data measured after epoch = 9 shows rapid changes in the graph due to problems such as overfitting. On the other hand, the PP2LFA-CRNN model shows only a small gap between the learning rate and the verification data, along with a stabilized graph. In addition, the learning rate of PP2LFA-CRNN gradually increases as the epochs progress. Similarly, the loss of the LFA-CRNN model gradually decreases as the epochs progress. Figure 15 shows the accuracy and the loss measurement graph calculated based on the test data. The PP2LFA-CRNN model shows the highest accuracy at 96.8% and the lowest loss at 0.132. The remaining models are ranked as follows: the CRNN model is the second (94.2%), and the AlexNet model is the third (91.3%). The CRNN model shows low performance results, compared to the PP2LFA-CRNN model because of the verification data gap, and it is determined that the AlexNet model is also unable to demonstrate good accuracy because of the verification data gap. In addition, it was expected that temporary overfitting would occur during learning, which would decrease the accuracy. This test can be summarized as follows. As far as CK+ database learning is concerned, the learning rate and loss of the PP2LFA-CRNN model showed no rapid changes and gradually increased as the epochs progressed. In the performance test, the PP2LFA-CRNN model shows the highest accuracy at 96.8%.

Then, the JAFFE database accuracy and loss test were conducted. For the JAFFE database, the contour achieved through pre-processing was converted into LFA data. The data used for the AlexNet model and the CRNN model for detecting only the face area in pre-processing was used as the input. Figure 16 shows the results obtained using the JAFFE database as the test data. Even with the JAFFE database, the PP2LFA-CRNN model, proposed in this study, shows the highest accuracy at 82.2%. The remaining models are ranked



(a) Learning rate using CK+ database



(b) Loss in the learning process using CK+ database

FIGURE 14. (a) Learning rate and (b) loss in the learning process using the CK+ database.

as follows: the CRNN model is the second, and the AlexNet model is the third. In addition, the PP2LFA-CRNN model shows the lowest loss at 0.515.

Finally, to determine the accuracy and reliability of the proposed algorithm, the accuracy per class was measured with each database.

Figure 17 shows the confusion matrices for the proposed method using each database. It shows the percentage of the number of samples actually determined to be true out of the samples predicted to be true for each class. Figure 17(a) shows the confusion matrices with the CK+ database and the results are as follows: anger, disgust, fear, happiness, neutral, sadness, and surprise were 92.3%, 91.2%, 93.5%, 92.9%, 99.2%, 82.4%, and 100.0%, respectively. Based on these results, it is confirmed that each class shows a high accuracy.

Although the results are not as high as those of the CK+ database, the overall accuracy is 77.8%. This study's test can be summarized as follows. The model proposed in this study maintained comparatively stable learning and loss rates compared to other models, and using the test dataset, the model proposed in this study showed the highest accuracy at 96.8% and the lowest loss at 0.13. Figure 17(b) shows the confusion matrices with the JAFFE database. In the test conducted using the JAFFE database, the proposed model shows the highest accuracy at 82.2%.

Therefore, accurate and confident face recognition was possible using the proposed PP2LFA-CRNN algorithm. In addition, based on real-time images collected in a smart driving environment through a security technique, the accuracy of sentiment analysis was experimented. For the experiment, the system shown in Figure 18 was established. This system was expanded from the system established in a related work [27], [31], [36], [37]. The real-time images collected for sentiment analysis were captured in the unit of frame and then analyzed. The system functions include image analysis, analysis visualization, recognition options, binarization, sentiment presentation with a broken line graph, and object

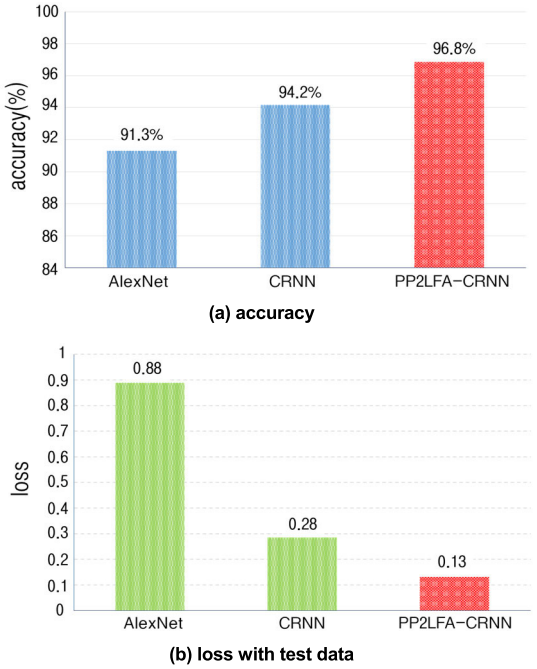


FIGURE 15. (a) Accuracy and (b) loss with the test data (CK+ database).

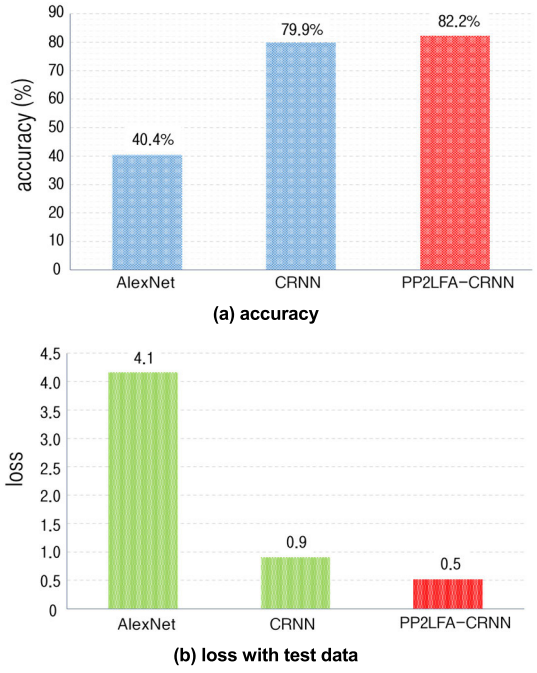


FIGURE 16. (a) Accuracy and (b) loss with the test data (CK+ for training and the JAFFE database for testing).

feature analysis [38]. In the experiment, the application of the security technique to the real-time image sent by a webcam was added. Once the security recording started in the system, it applied the security technique proposed in this study. Once the recording ended, server file upload application program interface (API) was executed to upload an image file to the server [39], [40]. When uploading was complete, analysis

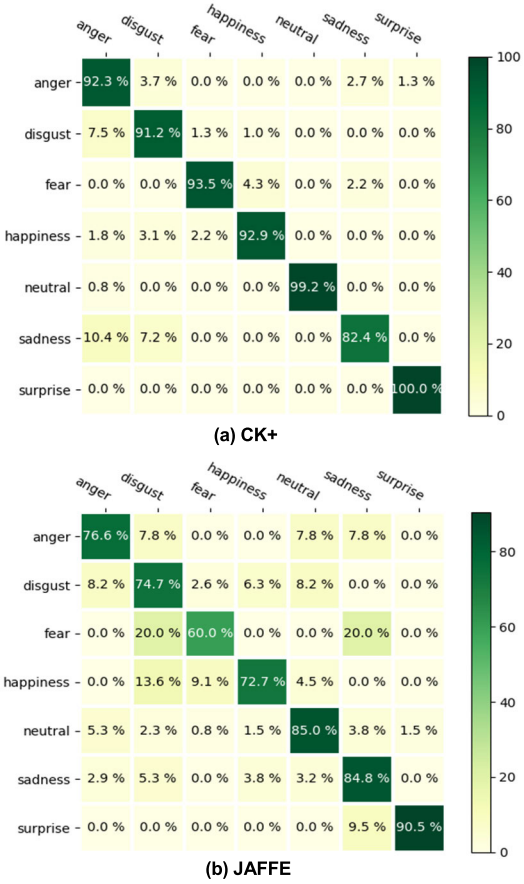


FIGURE 17. Confusion matrices for the proposed method using each database. (a) CK+ and (b) JAFFE.

was conducted through the module. The system selected a proper region of the video that was captured in the unit of frame by using the recognition option, and then recognized it through binarization. Then, it could find feature points of the image with the use of the LFA technique. Sometimes, the face in the image could be rotated. Therefore, with the use of the rotation option, it was possible to rotate the face by selecting a proper number ranging from -25 to $+25$. Object features such as mustache, beard, and glasses, could be analyzed with the use of the method proposed in the previous study [27], [31], [36], [37]. With the use of these functions, the images to which the security technique was applied were compared with general images in terms of sentiment analysis. Figure 18 shows the confusion matrices for the image sent in real-time by a webcam.

Figure 18(a) shows the accuracy of each sentiment class for a general image. As shown in the figure, "anger", "disgust", "fear", "happiness", "neutral", "sadness", and "surprise" are 90.4%, 89.2%, 90.0%, 90.4%, 96.1%, 74.2%, and 99.2%, respectively. Therefore, each class had high accuracy. Figure 18(b) presents the accuracy of each class with the application of the security technique. As shown in the figure, "anger", "disgust", "fear", "happiness", "neutral", "sadness", and "surprise" are 90.1%, 89.5%, 88.0%, 91.0%,

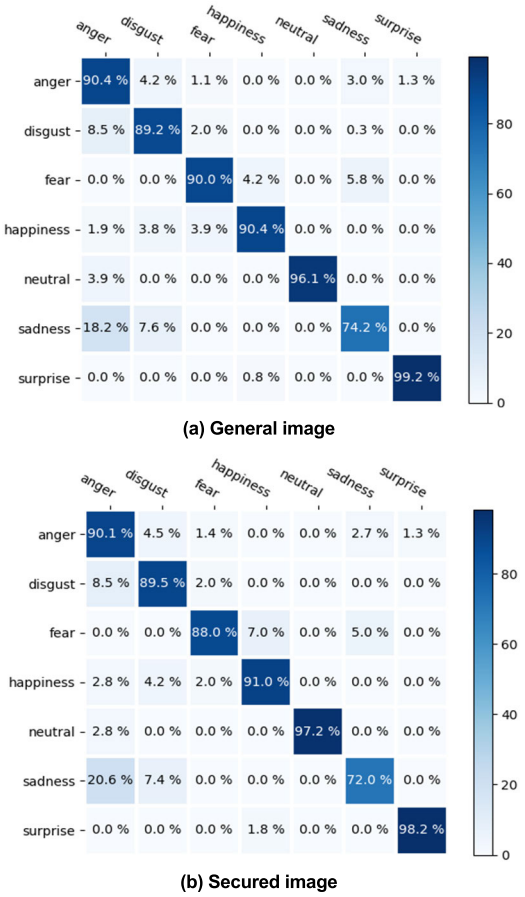


FIGURE 18. Confusion-matrices of the proposed method using real-time video. (a) General image and (b) Secured image.

97.2%, 72.0%, and 98.2%, respectively. Although the result rates were somewhat lower than those of the general image, the overall accuracy was 89.4%; thus, there was no significant difference. However, even when security technology was applied to an image collected in real time, a high accuracy rate was evaluated. Therefore, it can be seen that the image analysis applied to the security method proposed to protect personal information was suitable for a smart driving control system.

V. CONCLUSION

In this study, we proposed a real-time streaming image-based PP2LFA-CRNN model for facial emotion analysis. This study was conducted to solve security problems that might occur in driver convenience services using video for smart cars. First, to enhance the security of real-time stream videos, we implemented an encoding-decrypting process on videos using the PP2 algorithm. The PP2 algorithm generated random numbers using two variable clock functions and memory, which referred to the conditions of 255-bit and 257-bit LFSRs. The initialization process of PP2 was normally operated after 512-bit and was inputted and operated 1,000 times. We used the NIST SP800-22 statistic verification to verify the random number generation ability of PP2. In addition, regarding the NPCR of each color in the safety verification, red, green, blue, and gray were 99.6078%, 99.5941%, 99.6185%, and 99.5964%, respectively. In the UACI evaluation, red, green, blue, and gray were 33.3769%, 33.4610%, 33.522%, and 33.4911%, respectively. Thus, we identified that it was possible to encrypt and decrypt when random numbers were generated with the use of PP2 and combined with real-time video. Next, we performed the LFA process for analysis and management of images collected from the video device. LFA is an algorithm that extracts the face and the contour of the resident in the transmitted video and aggregate types of segments. To classify segments consisting of a face image of a driver into 16 types, we divided the types of segments through a filter that had a series of parameters and granted serial numbers to each type to perform cumulative aggregate on the segments. In the image through the LFA process, a 16 $\times$ 16 LS-map consisting of cumulative aggregate data on the segments of the original image was generated. This LS-map could not identify the face of the user or the person who made this facial expression, but it could only gain information about the facial expression when data was leaked. Facial expressions were classified using the PP2LFA-CRNN with the LS-map designed in this study. PP2LFA-CRNN was compared with AlexNet and CRNN models to evaluate the performance. In the performance evaluation, the learning rate and loss rate maintained relatively stable conditions compared to the comparison models. In the experiment using the test dataset, the learning rate was 96.8%, which was higher than other algorithms (CRNN: 94.2% and AlexNet: 91.3%). The loss rate was the lowest in the technique suggested in the study (CRNN: 0.28, AlexNet: 0.88, and PP2LFA-CRNN: 0.13). This seemed to occur because the strong characteristics of LFA were clearly divided. LFA was data generated as types of segments that were aggregated, and could definitely classify the number of segments that consisted of object (class). In addition, the comparison with other objects (class) could have functioned easily owing to the difference in numbers. These characteristics seemed to have deduced high results.

In the future, a class that can be used inside the nerve network model is expected to be constructed with the converted LFA algorithm. It is also expected that the LFA-CRNN model can be utilized in miniaturization systems such as mobile edge computing systems through an improved process.

REFERENCES

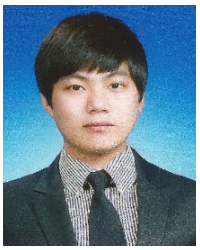
[1] K. Bylykbashi, E. Qafzezi, M. Ikeda, K. Matsuo, and L. Barolli, "Fuzzy-based driver monitoring system (FDMS): Implementation of two intelligent FDMSs and a testbed for safe driving in VANETs," *Future Gener. Comput. Syst.*, vol. 105, pp. 665–674, Apr. 2020.

[2] Y.-C. Tsai, P.-W. Lai, P.-W. Huang, T.-M. Lin, and B.-F. Wu, "Vision-based instant measurement system for driver fatigue monitoring," *IEEE Access*, vol. 8, pp. 67342–67353, 2020.

[3] J. Jang, Y. Jo, M. Shin, and J. Paik, "Camera orientation estimation using motion-based vanishing point detection for advanced driver-assistance systems," *IEEE Trans. Intell. Transp. Syst.*, early access, May 14, 2020, doi: 10.1109/TITS.2020.2990983.

[4] J. Cordero, J. Aguilar, K. Aguilar, D. Chávez, and E. Puerto, "Recognition of the Driving Style in Vehicle Drivers," *Sensors*, vol. 20, no. 9, pp. 2597–2623, Mar. 2020.

[5] *R. E. A. D. System*. Accessed: Jun. 20, 2020. [Online]. Available: http://pr.kia.com/en/future/future/emotive-driving-ces.do
[6] *Automobile Interior spaces of Toyota*. Accessed: Jun. 20, 2020. [Online]. Available: https://www.toyota-boshoku.com/
[7] *BMW Vision iNext*. Accessed: Jun. 20, 2020. [Online]. Available: http://www.bmw.com
[8] M. Shuai, N. Yu, H. Wang, and L. Xiong, "Anonymous authentication scheme for smart home environment with provable security," *Comput. Secur.*, vol. 86, pp. 132–146, Sep. 2019.
[9] D. Mocrii, Y. Chen, and P. Musilek, "IoT-based smart homes: A review of system architecture, software, communications, privacy and security," *Internet Things*, vols. 1–2, pp. 81–98, Sep. 2018.
[10] N. Nedjah, R. S. Wyant, L. M. Mourelle, and B. B. Gupta, "Efficient fingerprint matching on smart cards for high security and privacy in smart systems," *Inf. Sci.*, vol. 479, pp. 622–639, Apr. 2019.
[11] J. J. Engelsma, S. S. Arora, A. K. Jain, and N. G. Paulter, "Universal 3D wearable fingerprint targets: Advancing fingerprint reader evaluations," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 6, pp. 1564–1578, Jun. 2018.
[12] L. Y. Mano, B. S. Faiçal, L. H. V. Nakamura, P. H. Gomes, G. L. Libralon, R. I. Meneguete, G. P. R. Filho, G. T. Giancristofaro, G. Pessin, B. Krishnamachari, and J. Ueyama, "Exploiting IoT technologies for enhancing health smart homes through patient identification and emotion recognition," *Comput. Commun.*, vols. 89–90, pp. 178–190, Sep. 2016.
[13] N. Anciaux, P. Bonnet, L. Bouganim, B. Nguyen, P. Pucheral, I. Sandu Popa, and G. Scerri, "Personal data management systems: The security and functionality standpoint," *Inf. Syst.*, vol. 80, pp. 13–35, Feb. 2019.
[14] S. Mamonov and R. Benbunan-Fich, "The impact of information security threat awareness on privacy-protective behaviors," *Comput. Hum. Behav.*, vol. 83, pp. 32–44, Jun. 2018.
[15] H. N. Chua, S. F. Wong, Y. C. Low, and Y. Chang, "Impact of employees' demographic characteristics on the awareness and compliance of information security policy in organizations," *Telematics Informat.*, vol. 35, no. 6, pp. 1770–1780, Sep. 2018.
[16] A. Tekeoglu and A. S. Tosun, "Investigating security and privacy of a cloud-based wireless IP camera: NetCam," in *Proc. 24th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2015, pp. 1–6, doi: 10.1109/ICCCN.2015.7288421.
[17] B. Xu, D. Huang, and B. Mi, "Smart city-based e-commerce security technology with improvement of SET network protocol," *Comput. Commun.*, vol. 154, pp. 66–74, Mar. 2020.
[18] R. Muthalagu, A. Bolimera, and V. Kalaichelvi, "Lane detection technique based on perspective transformation and histogram analysis for self-driving cars," *Comput. Electr. Eng.*, vol. 85, pp. 1–16, Jul. 2020.
[19] A. Karkouch, H. Mousannif, and H. Al Moatassime, "CADS: A connected assistant for driving safe," *Procedia Comput. Sci.*, vol. 127, pp. 353–359, Jan. 2018.
[20] *Leading Future Car Technology*. Accessed: Jun. 20, 2020. [Online]. Available: https://en.mobis.co.kr/
[21] *NUVIS System*. Accessed: Jun. 20, 2020. [Online]. Available: http://www.istechnology.co.kr/
[22] J. Kietzmann, L. W. Lee, I. P. McCarthy, and T. C. Kietzmann, "Deepfakes: Trick or treat," *Bus. Horizons*, vol. 63, no. 2, pp. 135–146, Mar. 2020.
[23] H. R. Hasan and K. Salah, "Combating deepfake videos using blockchain and smart contracts," *IEEE Access*, vol. 7, pp. 41596–41606, 2019.
[24] M. Alsmadi, "Facial recognition under expression variations," *Int. Arab J. Inf. Technol.*, vol. 13, pp. 133–141, Jan. 2016.
[25] K. Kim and H. Lee, "Proposal of multi-channel operation technique using PingPong256," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./ 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1551–1554, doi: 10.1109/TrustCom/BigDataSE.2018.00222.
[26] K. H. Kim, T. Y. Kim, S. Lee, and W. T. Jang, "Ping pong stream cipher hardware implementation," in *Proc. IJCC*, Honolulu, HI, USA, vol. 11, 2018, pp. 133–138.
[27] K. H. Kim, H. J. Lee, and Y. S. Lee, "Proposed image encryption method using PingPong256," *J. Korea Soc. Comput. Inf.*, vol. 25, no. 1, pp. 71–77, Jan. 2020.
[28] R. Hamza, "A novel pseudo random sequence generator for image-cryptographic applications," *J. Inf. Secur. Appl.*, vol. 35, pp. 119–127, Aug. 2017.
[29] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber J., Multidisciplinary J. Sci. Technol., J. Select. Areas Telecommu.*, vol. 1, pp. 31–38, Apr. 2011.
[30] Y. Liu, J. Wang, J. Fan, and L. Gong, "Image encryption algorithm based on chaotic system and dynamic S-boxes composed of DNA sequences," *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4363–4382, Apr. 2016.
[31] D. H. Shin, K. Chung, and R. C. Park, "Detection of emotion using multi-block deep learning in a self-management interview app," *Appli. Sci.*, vol. 9, no. 22, pp. 4830–4845, Nov. 2019.
[32] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Workshops*, Jun. 2010, pp. 94–101.
[33] M. J. Lyons, S. Akamatsu, J. Gyoba, J. Budynek, and M. Kamachi, "The Japanese female facial expression (JAFFE) database," in *Proc. 3rd Int. Conf. Autom. Face Gesture Recognit.*, Apr. 1998, pp. 14–16.
[34] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.
[35] X. Han, Y. Zhong, L. Cao, and L. Zhang, "Pre-trained AlexNet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification," *Remote Sens.*, vol. 9, no. 8, pp. 848–869, Aug. 2017.
[36] J.-W. Baek and K. Chung, "Context deep neural network model for predicting depression risk using multiple regression," *IEEE Access*, vol. 8, pp. 18171–18181, 2020.
[37] J.-S. Kang, J.-W. Baek, and K. Chung, "PrefixSpan based pattern mining using time sliding weight from streaming data," *IEEE Access*, vol. 8, pp. 124833–124844, 2020.
[38] S. Park, M. Ji, and J. Chun, "2D human pose estimation based on object detection using RGB-D information," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 2, pp. 800–816, Dec. 2018.
[39] H. Yoo and K. Chung, "Deep learning-based evolutionary recommendation model for heterogeneous big data integration," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 9, pp. 3730–3744, Sep. 2020.
[40] J. C. Kim and K. Chung, "Prediction model of user physical activity using data characteristics-based long short-term memory recurrent neural networks," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 4, pp. 2060–2077, Apr. 2019.



CHANG-MIN KIM received the B.S. and M.S. degrees from the Department of Computer Information Engineering, Sangji University, Wonju, South Korea, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Department of Information Communication Software Engineering, Sangji University. His research interests include computer vision, database, artificial networks, programming language, data mining artificial intelligence, machine learning, and deep learning.

KI-HWAN KIM received the B.S. and M.S. degrees from the Department of Computer Engineering, Dongseo University, South Korea, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree. His research interests include cryptography, networks, side channel attack, programming language, machine learning, and deep learning.

YOUNG SIL LEE received the B.S. and master's degrees from Dongseo University, and the Ph.D. degree from the Graduate School, Dongseo University, in 2015. She is currently an Assistant Professor with the Computer Engineering Department, International College, Dongseo University, South Korea. Her research interests include cryptography, information security, sensor networks, body area networks, healthcare, artificial intelligence, and cloud computing.

KYUNGYONG CHUNG received the B.S., M.S., and Ph.D. degrees from the Department of Computer Information Engineering, Inha University, South Korea, in 2000, 2002, and 2005, respectively. He has worked with the Software Technology Leading Department, Korea IT Industry Promotion Agency (KIPA). From 2006 to 2016, he was a Professor with the School of Computer Information Engineering, Sangji University, South Korea. Since 2017, he has been a Professor with the Division of Computer Science and Engineering, Kyonggi University, Suwon, South Korea. His research interests include data mining, artificial intelligent, healthcare, biomedical and health informatics, knowledge systems, HCI, and recommendation systems. He was named a 2017 Highly Cited Researcher by Clarivate Analytics.

ROY C. PARK received the B.S. degree from the Department of Industry Engineering, and the M.S. and Ph.D. degrees from the Department of Computer Information Engineering, Sangji University, South Korea, in 2010 and 2015, respectively. From 2015 to 2018, he was a Professor with the Division of Computing Engineering, Dongseo University, South Korea. Since 2019, he has been a Professor with the Department of Information Communication Software Engineering, Sangji University, Wonju, South Korea. His research interests include WLAN systems, heterogeneous networks, ubiquitous network service, human-inspired artificial intelligent and computing, health informatics, knowledge systems, peer-to-peer, and cloud networks.