electronics

Article

FPGA Chip Design of Sensors for Emotion Detection Based on Consecutive Facial Images by Combining CNN and LSTM

Shing-Tai Pan * **and Han-Jui Wu**

Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan, China; m1105505@mail.nuk.edu.tw
* Correspondence: stpan@nuk.edu.tw

Abstract

This paper proposes emotion recognition methods for consecutive facial images and implements the inference of a neural network model on a field-programmable gate array (FPGA) for real-time sensing of human motion. The proposed emotion recognition methods are based on a neural network architecture called Convolutional Long Short-Term Memory Fully Connected Deep Neural Network (CLDNN), which combines convolutional neural networks (CNNs) for spatial feature extraction, long short-term memory (LSTM) for temporal modeling, and fully connected neural networks (FCNNs) for final classification. This architecture can analyze the local feature sequences obtained through convolution of data, making it suitable for processing time-series data such as consecutive facial images. The method achieves an average recognition rate of 99.51% on the RAVDESS database, 87.80% on the BAUM-1s database and 96.82% on the eNTERFACE'05 database, using 10-fold cross-validation on a personal computer (PC). The comparisons in this paper show that our methods outperform existing related works in recognition accuracy. The same model is implemented on an FPGA chip, where it achieves identical accuracy to that on a PC, confirming both its effectiveness and hardware compatibility.

Keywords: consecutive facial images; emotion recognition; convolutional neural networks (CNNs); long short-term memory (LSTM); field-programmable gate array (FPGA)

check for updates

Academic Editors: Ruiting Xu, Jiang Zhe and Leixin Ouyang

Received: 18 July 2025
Revised: 10 August 2025
Accepted: 14 August 2025
Published: 15 August 2025

Citation: Pan, S.-T.; Wu, H.-J. FPGA Chip Design of Sensors for Emotion Detection Based on Consecutive Facial Images by Combining CNN and LSTM. *Electronics* **2025**, *14*, 3250. https://doi.org/10.3390/electronics14163250

1. Introduction

In recent years, with the continuous development of artificial intelligence (AI) technology, image and speech emotion recognition have become key applications of AI. AI technologies have significant applications in many fields, such as driver monitoring, fraud detection, medical care, and education. In many application scenarios, embedded devices play an important role; so, it has become particularly important to achieve neural network model inference on embedded devices such as FPGA chips.

For data with temporal dependencies, such as consecutive images, speech, and natural language, using temporal model can enhance the performance of the model. Therefore, the model architecture used in this paper first applies CNNs to extract local features from the data. To enhance the model's learning performance on a time-series level, LSTM neural networks are then used to model the feature sequences. Finally, FCNN is employed for result classification.

Emotion recognition is a technology that utilizes various signals, such as facial images and speech, to analyze and identify emotional states. In the case of using facial images for emotion recognition, the current main approach is to extract feature values from the image using 2D convolutional neural networks, followed by prediction and classification, as demonstrated in other papers [1,2]. Emotion recognition technologies are crucial for understanding and analyzing human emotions, as it can assist us in better understanding the emotional state of the user and provide more accurate and human-like solutions for related application areas [3]. Many studies have shown that the quality of data pre-processing and feature extraction significantly affects the performance of speech emotion recognition models. The experimental results in paper [4] show that the accuracy of machine learning models is affected by using different speech features for training. Consequently, in order to obtain accurate and stable performance of artificially intelligent models on emotion recognition, this paper focuses on the emotion recognition based on consecutive facial images.

However, the accuracies of facial recognition mainly depend on the quality of the input signals, such as the contrast, brightness, and focus of images for image signals. Consequently, it may significantly reduce the reliability of emotion recognition if any of the elements in the image fails to meet the standard requirements, such as overexposure or blurring. For the above reasons, this paper proposes consecutive facial pre-processing and recognition methods, to complete more suitable emotion recognition methods based on the user's environment.

1.1. Field-Programmable Gate Array

The implementation of intelligent systems such as automatic emotion recognition technology in embedded systems faces many challenges, e.g., real-time requirements, resource constraints, and low power consumption requirements. Therefore, when implementing facial emotion recognition technology, the choice of hardware platform is crucial to the system's efficiency and performance. In this study, facial recognition is selected as the target application for implementation on the FPGA platform due to its high practicality and real-time processing requirements across various intelligent systems. Recently, FPGAs have gained attention as a hardware accelerator that can serve as an alternative to GPUs. For instance, smart access control systems can use facial recognition for identity verification and automatic door operation. In public safety surveillance, it enables real-time detection of suspicious individuals. In healthcare, facial expression analysis can assist in assessing a patient's emotional or physiological state for timely response. Additionally, facial or human recognition is essential in UAVs and robotic vision navigation for interactive decision-making, while in-vehicle systems can detect driver fatigue to enhance driving safety. These application scenarios typically demand low latency and low power consumption in embedded systems—requirements that FPGAs are well-suited to meet through hardware-level acceleration and reconfigurability. Consequently, FPGAs (Field-Programmable Gate Arrays) have become one of the ideal platforms for implementing emotion recognition due to their highly customizable features, parallel processing capabilities, and low power consumption.

The FPGA is a reconfigurable embedded device commonly used in digital logic and digital signal processing applications. FPGAs' high flexibility and programmability enable their wide application in various fields, including IC testing, embedded systems, and the IoT (Internet of Things).

The features of FPGAs include:

A. Reconfigurability: FPGAs are reconfigurable [5] and can define their digital logic circuits through programming, allowing developers to redesign the FPGA's functions according to application requirements repeatedly.

B. High parallel processing capability: FPGAs have multiple independent logic circuits and data paths that can run in parallel, enabling them to efficiently perform parallel processing for multiple tasks and hence provide high-performance computing power.

C. Low latency and high-frequency operation: Due to the fact that FPGA's logic circuits are composed of gate arrays and have high optimization capabilities, it can achieve low latency and high-frequency operation. This makes it ideal for applications requiring high-speed processing.

D. Customizability: FPGAs are highly flexible in customization and can be designed and optimized according to application requirements. This includes the design of logic circuits, data paths, memory, and interfaces.

E. Software and hardware co-design: FPGAs provide the ability to co-design software and hardware on a single chip [6]. This provides higher system integration and performance.

F. Suitable for rapid development and testing: FPGAs have a rapid development cycle. Developers can quickly develop and test them within a shorter period [7].

1.2. Experimental Protocol

This paper utilizes two deep learning frameworks, TensorFlow and Keras, to train emotion recognition models for consecutive facial images signals on a PC. The parameters of the models are transferred to an FPGA chip. The neural network model inference algorithms are used to simulate the computation of model inference in the deep learning frameworks and then obtain the final classification results. For consecutive facial emotion recognition, this paper dynamically captures 30 frames of images from a video as the consecutive image data. The facial images are extracted by using the open-source face detection model from Open Computer Vision Library (OpenCV). The CLDNN model architecture proposed in paper [8] is used to build and train the ML model. The trained model is then deployed on the FPGA chip for model inference.

2. Related Works

In this section, the CLDNN model is introduced. Moreover, some related researches of consecutive facial emotion recognition and the implementation of model inference on FPGAs are also explored in this section.

2.1. CLDNN Model Architecture

The paper [8] proposes a neural network model architecture called Convolutional Long Short-Term Memory Fully Connected Deep Neural Networks, referred to as CLDNNs, for processing time-series data. such as audio signals, consecutive images, and natural languages. The CLDNN model is composed of convolutional neural networks (CNNs), LSTM neural networks, and fully connected neural networks of DNNs. The CNNs and LSTM neural networks analyze and calculate local features and sequential patterns in the data, respectively. The fully connected layers then classify the prediction results. Since consecutive images have time-series characteristics, they are suitable for training with recurrent neural networks (RNNs), such as LSTM. According to [8], the CLDNN model outperforms those using only CNN or LSTM models in terms of the accuracies of the models.

Therefore, this paper applies and modifies the CLDNN model architecture for emotion recognition, combining CNNs, LSTM neural networks, and DNNs to build the ML model. Figure 1 shows the architecture of the CLDNN model.

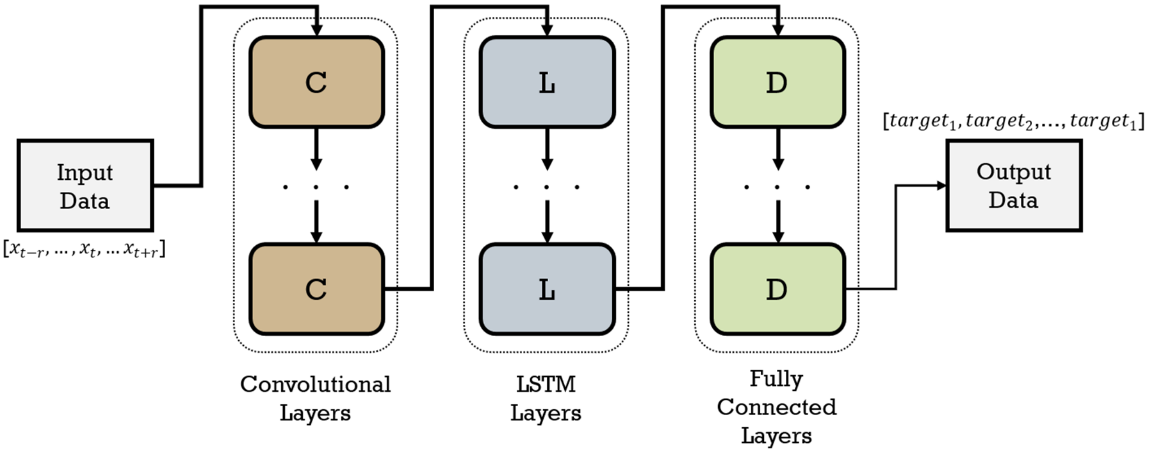

Input Data
$[x_{t-r}, \ldots, x_t, \ldots x_{t+r}]$
C
C
L
L
D
D
Convolutional Layers
LSTM Layers
Fully Connected Layers
$[target_1, target_2, \ldots, target_1]$
Output Data

Figure 1. The CLDNN model architecture, composed of multiple CNNs, LSTM neural networks, and DNNs.

2.2. Consecutive Facial Emotion Recognition (CFER)

Paper [9] proposes a method for consecutive facial emotion recognition. Firstly, the frame interval is calculated based on the average duration of all video files in the database. Frames are then extracted based on this interval to obtain a segment of consecutive image data. Next, facial landmark detection (68 points) [10] is used to identify the positions of the facial features in the images, and the Euclidean distances between these points are used as the feature value. The image feature values of the same image of consecutive image data are concatenated into a feature sequence. Finally, these feature sequences are used as inputs to train an LSTM model. This paper achieves an average accuracy of 98.9%. The paper [10] introduced one of the first large-scale facial landmark localization challenges on "in-the-wild" images, setting a benchmark for robust landmark detection under diverse real-world conditions. Their contribution involved developing robust methods that cope with variations in pose, illumination, and expression. This challenge pushed forward the state of facial landmark localization by promoting open benchmarking and advancing facial analysis technology specifically for unconstrained images.

A multimodal emotion recognition model that combines video data and audio data is proposed in [11]. The authors tested the CLDNN on various large vocabulary speech recognition tasks ranging from 200 to 2000 h of audio data. They found that CLDNN consistently outperforms the strongest individual model, the LSTM, by providing a 4–6% relative improvement in word error rate (WER). For the emotion recognition of consecutive facial image, the authors randomly extract 30 images from video data. Then, using the MTCNN (multi-task cascaded convolutional network) proposed in paper [12], they extract the facial parts from all images in each image. The images are then reshaped uniformly to (160, 160) color images as data features. Hence, the input of the facial emotion recognition model is a data sequence of shape (N, 30, 160, 160, 3), where N represents the number of video data, 30 signifies the number of images extracted from a video, (160, 160) denotes the dimensions of the images, and 3 indicates that each pixel in an image is represented by three RGB color values. In the speech emotion recognition model, the log Mel-spectrogram is extracted from every segment of the speech signals and used as input data of a machine learning (ML) model. In order to obtain the log Mel-spectrogram, the authors utilize 94 Mel-filter banks for each speech signal, with a 40 ms Hanning window and 10 ms overlapping. This result is a representation with dimensions of 94. After performing cross-validation with the ResNet-50 model, the paper found that the consecutive facial emotion recognition model achieves an accuracy of 95.49% and the speech emotion recognition model achieves an accuracy of 75.61%. The fusion model combining consecutive facial image and audio achieves an accuracy of 97.57%.

In paper [13], the authors propose a method that uses facial landmarks [10] to detect facial regions in images. The detected facial regions are then converted into grayscale images. The authors extract 32 features from the resulting images using Gabor filters. The features are then combined with the 68 positions of the facial landmarks. After processing all frames in the video, the authors obtain 2176 (32 × 68) features for each facial image. The proposed method in [13] achieves an accuracy of 96.53% on testing. This paper applies deep belief networks to fuse bimodal data (e.g., facial and speech signals) for emotion recognition. It demonstrates that deep learning techniques can effectively capture complex interactions between different modalities, similar to Ma et al. [11], but with a focus on using deep belief networks specifically. It underscores the importance of combining facial expressions with speech for enhanced emotion recognition accuracy. Paper [14] proposes a multimodal emotion recognition model that combines consecutive facial image and speech. Regarding the consecutive facial emotion recognition model, the authors use the InceptionV3 model to extract feature values from single images, which are then sent to LSTM for time-series learning. This results in an accuracy of 94% on RAVDESS database. In terms of speech emotion recognition, the authors extract Mel Frequency Cepstral Coefficients (MFCCs) from the speech as feature values and train them using the CLDNN model. The experimental results show that the speech emotion recognition model proposed in [14] achieves an accuracy of 82%. The fusion model combining consecutive facial image and audio achieves an accuracy of 96%. A brief comparative analysis of these papers [10–13] is as follows. Collectively, they illustrate the evolution of facial analysis from basic detection and landmark [10,12] to multimodal emotion recognition [11,13,14]. While early works relied on handcrafted features or shallow networks, recent approaches leverage deep learning for joint representation learning, enhancing robustness in real-world scenarios. Nevertheless, challenges in cross-modal alignment and generalizability persist. Future work could explore integrating attention mechanisms (to dynamically weigh modality importance) and self-supervised learning (to reduce annotation dependence) as promising directions.

In [15], the authors survey recent deep learning-based FER approaches, including CNN, ViT (Vision Transformer), and hybrid models. They also discuss challenges such as occlusion, pose variation, and database bias. The paper highlights the shift from handcrafted features to self-supervised learning. Wang et al. [16] propose a multi-task learning framework that combines facial expression recognition with auxiliary tasks. They use self-supervised pre-training (e.g., MoCo v3) to improve generalization on in-the-wild databases (AffectNet, RAF-DB) and achieve 92.1% accuracy on RAF-DB. The study in [17] introduces a graph-based approach to model relationships between facial regions and addresses domain adaptation (e.g., from lab-controlled to real-world data) via adversarial training. The approach is evaluated on CK+, FER2013, and SFEW, achieving results that outperform CNN baselines by approximately 5%. Chen et al. [18] propose a lightweight CNN–transformer hybrid model with channel–spatial attention. The method achieves 87.3% accuracy on LRW-FER, a low-resolution FER database. The authors in [19] employ CLIP-based vision–language models to improve the interpretability of FER. Textual explanations for predictions are generated on EmotioNet and AffectNet in this work. Zhao and Elgammal [20] capture temporal dynamics in video-based FER using a 3D-CNN combined with an LSTM. The proposed method outperforms 2D-CNNs on the DFEW (Dynamic Facial Expression in the Wild) database.

3. Facial Emotion Recognition Methods and Parameter Setting

This section will introduce the training and testing process of the consecutive facial emotion recognition method proposed in this paper. Some experiments and the experimental environment will also be introduced in this section. This paper aims to design an AI chip based on FPGAs for recognizing emotions from videos. The computation load is then an important concern for the proposed deep learning model. A simple and efficient CNN model combined with an LSTM model and fully-connected neural network was chosen for the applications in this paper. The primary contribution of this paper is that, due to the combination of LSTM, the proposed model can capture time-sequential features obtained from convolutional layers on consecutive facial emotional images captured from videos with less computational effort, without sacrificing recognition accuracy, making it suitable for implementation on an FPGA chip.

3.1. CLDNN Model

When conducting experiments using the RAVDESS database, this paper tested different model performances by employing both LSTM and Gated Recurrent Units (GRUs) as temporal neural networks. The experimental results in Table 1 demonstrate that, in the consecutive facial emotion recognition model, using LSTM to process temporal data achieves an accuracy of 99.51%, showing a 1.65% improvement compared to using GRUs. As LSTM neural networks yield higher accuracy in consecutive facial emotion recognition models, and using an LSTM model for model inference on an FPGA chip does not show a significant increase in inference time compared to using a GRU model, this paper opted to use LSTM as the modeling approach.

Table 1. The difference in model accuracy and execution time on an FPGA chip between models built using LSTM and GRU.

CNN + LSTM + DNN		**CNN + GRU + DNN**	
Execution Time on FPGA	**Accuracy**	**Execution Time on FPGA**	**Accuracy**
11.70 s	99.51%	11.67 s	97.86%

3.2. Experimental Environment for Model Training on PC

This paper performed data pre-processing on the Central Processing Unit (CPU) of a PC and trained a CLDNN model for consecutive facial emotion recognition using a Graphical Processing Unit (GPU). The trained parameters for CLDNN model were transferred to an FPGA chip for inference of the model. Table 2 shows the hardware specifications and software environment used for training the model on the PC.

Table 2. The experimental environment of the proposed emotion recognition methods.

Category	**Specification**
CPU	Intel® Core™ i7-10700 CPU 2.90 GHz Manufacturer: Intel Corporation, Santa Clara, CA, USA
GPU	NVIDIA GeForce RTX 3090 32 GB Manufacturer: NVIDIA Corporation, Santa Clara, CA, USA
IDE (Integrated Development Environment)	Jupyter notebook (Python 3.7.6)
Deep learning frameworks	TensorFlow 2.9.1, Keras 2.9.0

The experiments conducted in this paper used 10-fold cross-validation method for training and testing. In order to ensure the reliability of machine learning model, the Scikitlearn package was applied to randomly select 10% of the database as the data for testing, and the remaining 90% data were randomly divided into training data and validation data at a ratio of 9:1. Figure 2 illustrates the data proportions of the training set, validation set, and testing set.

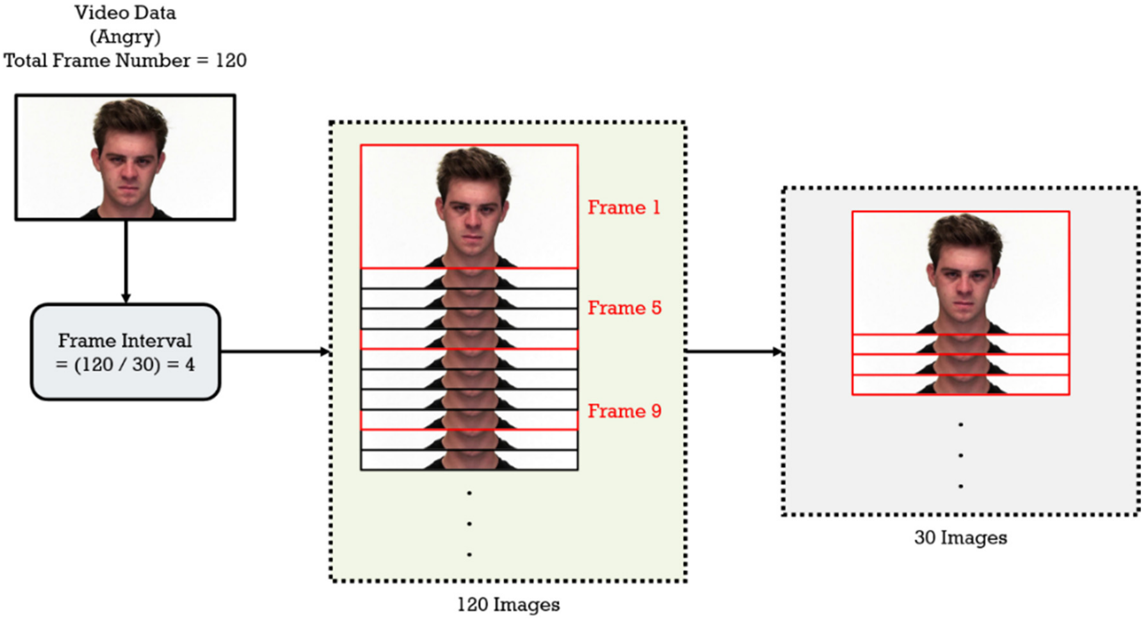


Figure 2. Frame interval based on the total number of frames in a video file, and 30 frames of video captured using the frame interval to represent a consecutive image data.

3.3. *Consecutive Facial Emotion Recognition*

In this subsection, a detailed description of the databases, data pre-processing methods, neural network model architecture, and parameter settings for proposed consecutive facial emotion recognition method will be provided. Then, the experimental results of the proposed consecutive facial emotion recognition method will be compared with other related literature.

3.3.1. Databases

This paper used the RAVDESS [21], BAUM-1s [22] and eNTERFACE'05 [23] databases for training and testing the proposed consecutive facial emotion recognition model.

RAVDESS

RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) is a database of consecutive emotional image and speech created by Ryerson University in Canada. It includes emotional expressions in both visual and auditory forms from different actors and can be used for research in the fields of emotion recognition, emotion analysis, audio processing, etc.

This database contains videos and voice data from 24 actors (12 males and 12 females) from Canada. Each actor records videos and audio in 8 different emotional states (angry, calm, disgust, fear, happy, neutral, sad, and surprised) by speaking and singing. For each emotional state, the actor performs multiple times to provide data with different emotional intensities and expressions. The RAVDESS database has a total of 2452 consecutive image and audio data, with the distribution and the amount of data for each emotional state shown in the Table 3.

Table 3. The quantity and proportion of data for each emotion in the RAVDESS database.

Label	Number of Data	Proportion
Angry	376	15.33%
Calm	376	15.33%
Disgust	192	7.83%
Fear	376	15.33%
Happy	376	15.33%
Neutral	188	7.39%
Sad	376	15.33%
Surprised	192	7.83%
Total	2452	100%

BAUM-1s

The BAUM-1s database was created by the Department of Electrical and Electronics Engineering at Bahcesehir University in Turkey. This database consists of samples of videos and voice recordings, recorded by 31 professional actors. In this study, we utilized six emotional categories, angry, disgust, fear, happy, sad, and surprised, to conduct experiments of consecutive facial and speech emotion recognition. The data number and proportion of each emotion label in the BAUM-1s database are shown in Table 4.

Table 4. The quantity and proportion of data for each emotion in the BAUM-1s database.

Label	Number of Data	Proportion
Angry	59	10.85%
Disgust	86	15.81%
Fear	38	6.99%
Happy	179	32.90%
Sad	139	25.55%
Surprised	43	7.90%
Total	544	100%

eNTERFACE’05

The eNTERFACE’05 database was recorded by 44 professional actors and consists of six different emotion categories, with different sentences expressing each category. This database includes both videos and speech samples, making it commonly used in various multimodal emotion recognition research. The data number and proportion of each emotion label in the eNTERFACE’05 database are shown in Table 5.

Table 5. The quantity and proportion of data for each emotion in the eNTERFACE’05 database.

Label	Number of Data	Proportion
Angry	211	16.71%
Disgust	211	16.71%
Fear	211	16.71%
Happy	208	16.47%
Sad	211	16.71%
Surprised	211	16.71%
Total	1263	100%

3.3.2. Pre-Processing

This paper referred to [9] and adopted a dynamic sampling method to capture 30 frames of video file to represent a consecutive image data. The frame intervals were determined by dividing the total number of frames in each video data by 30. Then, the frame intervals were used as a spacing to capture 30 frames in each video. Figure 2 shows the process of the dynamic capture of the consecutive images.

The OpenCV facial detection model was then used to locate the coordinates of the faces in the images, and any unnecessary data outside the coordinates was discarded. In order to reduce the computational load during model inference on the FPGA chip, the color images were converted into grayscale images. Moreover, all facial images were resized to a uniform size of 100 × 100 pixels. Figure 3 shows the pre-processing flowchart for the facial emotion recognition method proposed in this paper.

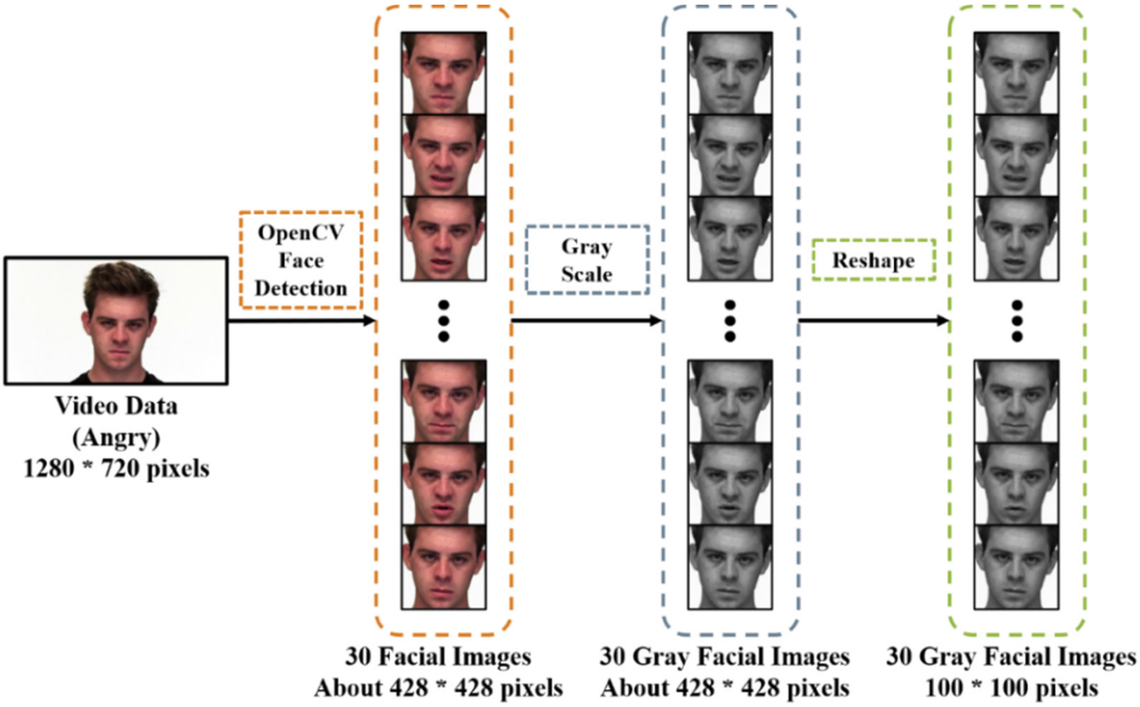

OpenCV Face Detection
Gray Scale
Reshape
Video Data (Angry) 1280 * 720 pixels
30 Facial Images About 428 * 428 pixels
30 Gray Facial Images About 428 * 428 pixels
30 Gray Facial Images 100 * 100 pixels

Figure 3. Pre-processing of the videos in the proposed facial emotion recognition method.

Facial Detection

Facial detection is necessary before training facial emotion recognition models. Nowadays, there are many open-source facial detection models available, such as facial landmark detection and the MTCNN model proposed in [11]. In this paper, the Haar cascade frontal face detection model provided by OpenCV was used, which was also used for facial detection in other related research, such as the papers [24,25]. This facial detection model can locate the starting coordinates (x, y) of the facial region in the images, and the corresponding w (width) and h (height). The facial image within the region is then extracted, while unnecessary information outside of the region is discarded. The process of facial image extraction is shown in Figure 4.

Grayscale Conversion

Paper [26] compares the performance of color and grayscale images in facial recognition. It found that, in some cases, using grayscale images performs facial recognition better than using color images. Moreover, using color images increases the computations and parameters during recognition. Overall, there are advantages and disadvantages to using color or grayscale images for facial recognition. Considering the restriction in computational requirements of performing neural network model inference on embedded devices, this paper converted facial images to grayscale as the inputs to the neural network model.

Resize

As the OpenCV face detection model was used to extract facial images, the sizes of facial images were different. Therefore, all facial images needed to be resized into a unified size of 100 × 100 pixels before being fed into the neural network model for training and testing.

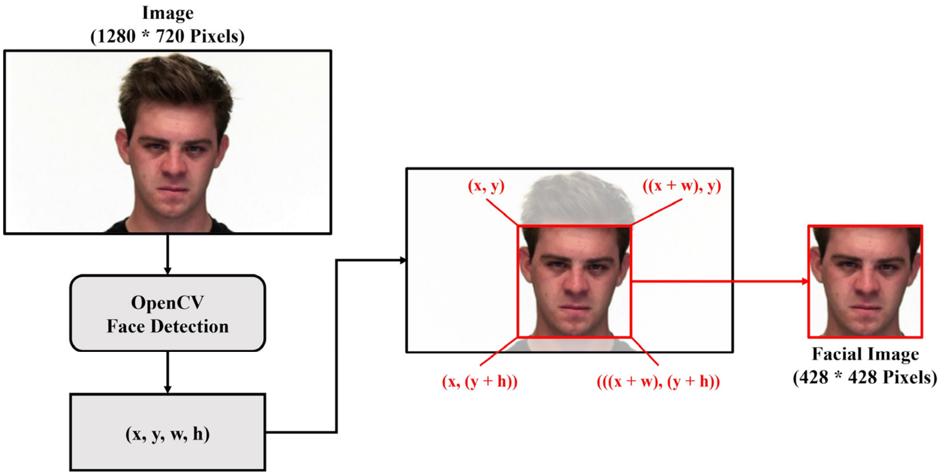


Figure 4. OpenCV face detection model was used to capture the facial part in the images.

3.3.3. Experiments

This paper proposes a consecutive facial emotion recognition method based on the CLDNN model architecture. The LFLBs (Local Feature Learning Blocks) with a CNN in the neural networks was used as the main body to extract local features of input facial images. Each of the 30 image features were concatenated into consecutive image feature sequences. Then, the LSTM layer was applied to strengthen the learning of model on the time series. Finally, a fully connected layer was used for classification to output the recognition result.

Local Feature Learning Block (LFLB)

The LFLB is a type of neural network module used in deep learning, which is primarily used to extract local feature values from input data. LFLB consists of many sub-modules, such as convolutional layers, pooling layers, and fully connected layers, etc. It can extract features from different regions of the data and share weights to reduce the number of model parameters and improve the model's generalization ability. Some research of facial emotion recognition also uses LFLBs to extract local features from image data, such as papers [27,28].

In this paper, a 2D convolutional layer, a batch normalization layer, and a max pooling layer were used as the LFLBs for extracting local features from the facial image data. The extracted local features were then mapped by the activation function ReLu (rectified linear unit) at the end of LFLB. Due to the local perception characteristics of the CNNs, it is suitable for calculating the local features of the image data and generating feature maps. The batch normalization layer can normalize the input data, scaling the values to a specific range of 0~1. It helps the model train the data faster and reduces the possibility of overfitting. The max pooling layer is mainly used to limit the sizes of the feature map. It helps preserve the most influential feature values (maximum values). This paper used the zero-padding method in the design of sub-modules of the LFLB to ensure that the size of the feature map was the same as that of the input data and to avoid data loss at the edges. Moreover, ReLU was used as the non-linear activation function of the LFLB. Figure 5 shows the schematic diagram of the LFLB applied in this paper.

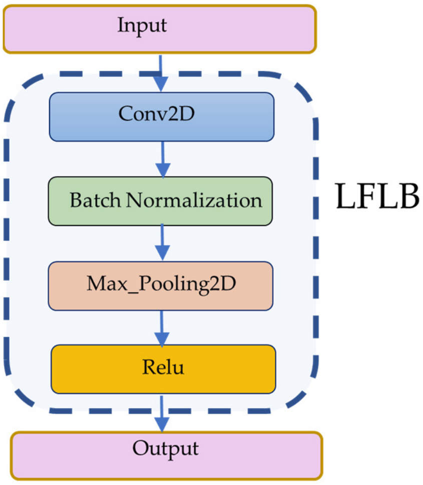


Figure 5. The LFLB used in this paper, including a 2D convolutional layer, a batch normalization layer, and a max pooling layer.

Training Process and Parameters

When conducting experiments using the RAVDESS database, we set the number of local feature learning blocks to 6 results in 64 local features per image and achieved the highest accuracy of 99.51%. However, from Table 6, it can be seen that, when increasing the number of local feature learning blocks to 7, the output of local features per image decrease to 16, leading to a sharp decline in accuracy, reaching only 32.68%. Therefore, this study set the number of local feature learning blocks in the consecutive facial emotion recognition model to 6.

Table 6. The impact of using different numbers of LFLBs on model accuracy.

Number of LFLBs	Number of Local Features	Accuracy
3	2704	28.89%
4	784	52.96%
5	256	88.58%
6	64	99.51%
7	16	32.68%

The proposed strategy in this paper for consecutive facial emotion recognition is as follows. Firstly, local features are extracted from single facial images using LFLBs, and then, the resulting 30 feature maps are concatenated into a sequence. The sequence represents the features sequences of consecutive facial images. Then, the sequences are flattened and reshaped into the shape of [time step, data length], before being fed into the LSTM layer. Next, the output data are normalized after learning on a time-series level in the LSTM layer. Finally, the SoftMax method is performed when using a fully connected layer to extract the emotion label corresponding to the maximum output value for the sequence as the predicted result. The training process of the proposed consecutive facial emotion recognition method is shown in Figure 6. It is worth noting that the design choice of using six cascading LFLBs was motivated by the need to achieve a hierarchical abstraction of spatial features while controlling model complexity for hardware deployment. Each LFLB progressively extracts higher-level features and reduces the spatial dimensions of the input data. The deeper architecture—with six layers—enables the model to learn both low-level textures and more abstract facial structures, which are critical for accurate emotion recognition. Moreover, this layered approach increases non-linearity and enhances the model's ability to generalize across variations in facial expressions. Empirical testing showed that using fewer than six LFLBs reduced the recognition accuracy, whereas deeper configurations increased hardware resource usage without yielding significant performance gains. Therefore, six LFLBs represent an optimal trade-off between model expressiveness and hardware efficiency.

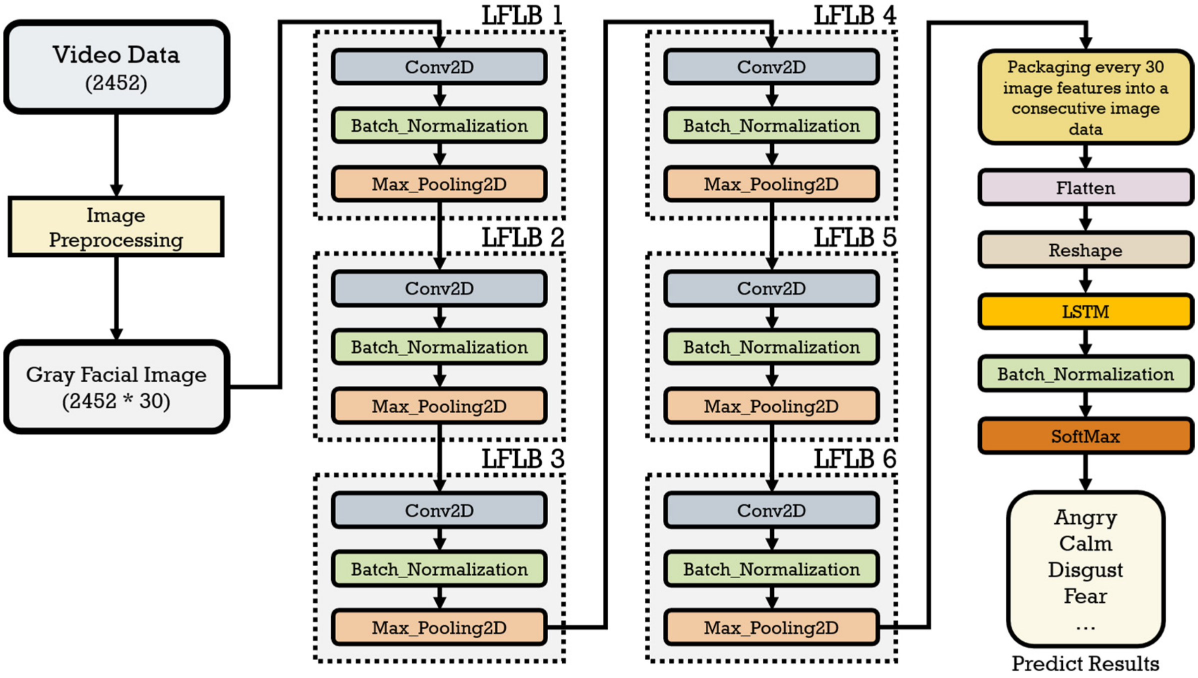


Figure 6. The flowchart of the proposed method for consecutive facial emotion recognition.

The number of memory units in LSTM layers can affect the training performance of machine learning models and needs to be adjusted according to the type of data and the length of the data sequences. In this paper, for the consecutive facial recognition CLDNN model, cross-validation was used to test the effect of different numbers of memory units in the LSTM layer for finding the best model's accuracy. The number of memory units was set from 5 to 50, with an increment of 5. For applying the 10-fold cross-validation method, the experimental results show that the highest accuracy of 99.51% is achieved when the number of memory units in the LSTM layer is 20. Therefore, this paper set the memory units of the LSTM layer in the CLDNN model for consecutive facial emotion recognition as 20. Figure 7 shows the experimental results for the relationship between the number of memory units in the LSTM layer and the accuracy of the proposed consecutive facial emotion recognition model. The proposed model's parameters are listed in Table 7. The parameter values used in the convolutional and pooling layers (e.g., filter size, kernel size, stride, and pool size) were selected based on a combination of empirical testing and reference to common practices in lightweight CNN architectures for facial image feature extraction. For instance, the use of 5 × 5 kernels in the initial LFLBs (LFLB 1–4) allows for capturing broader spatial features in the early layers, while the later use of smaller 3 × 3 kernels (in LFLB 5–6) focuses on refining local features with reduced computational cost. The stride and pooling settings (stride = 2, pool sizes = 5 or 3) were chosen to progressively reduce the spatial dimensions and control overfitting, while maintaining sufficient resolution for temporal modeling. These values were also validated experimentally to ensure a good balance between performance and efficiency on the FPGA platform. With these parameters, the proposed model achieves higher recognition rates compared to those reported in other papers for the three databases.

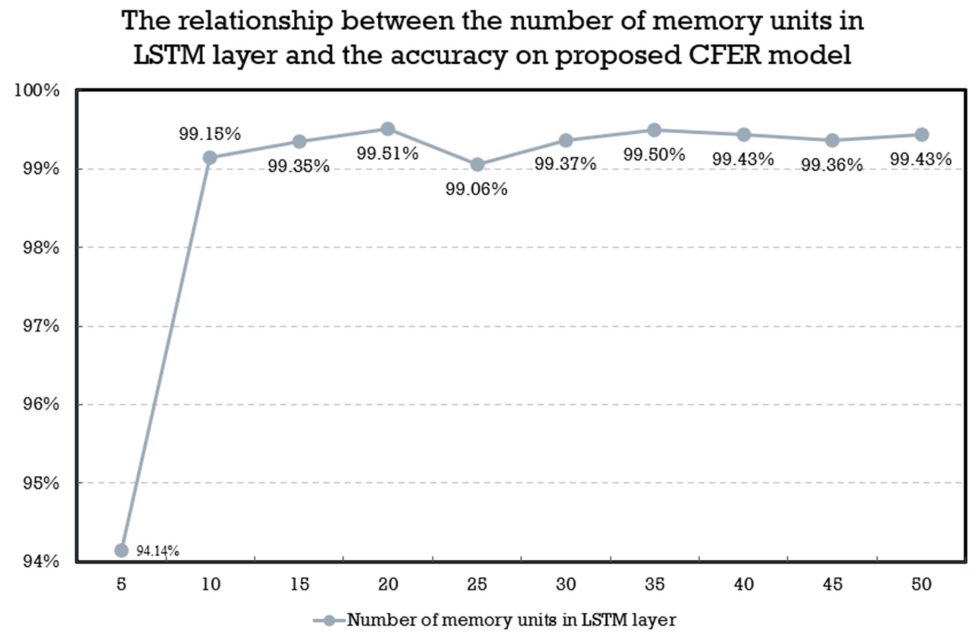


Figure 7. The relationship between the number of memory units in the LSTM layer and the accuracy of the proposed consecutive facial emotion recognition model.

Table 7. The parameters of the proposed CLDNN model for consecutive facial emotion recognition.

Model Architecture		Information

LFLB 1	Conv2d (Input)
	Batch_normalization
	Max_pooling2d
	ReLu
		Filters = 16, Kernel_size = 5, Strides = 1
		Pool_size = 5, Strides = 2

LFLB 2	Conv2d
	Batch_normalization
	Max_pooling2d
	ReLu
		Filters = 16, Kernel_size = 5, Strides = 1
		Pool_size = 5, Strides = 2

LFLB 3	Conv2d
	Batch_normalization
	Max_pooling2d
	ReLu
		Filters = 16, Kernel_size = 5, Strides = 1
		Pool_size = 5, Strides = 2

LFLB 4	Conv2d
	Batch_normalization
	Max_pooling2d
	ReLu
		Filters = 16, Kernel_size = 5, Strides = 1
		Pool_size = 5, Strides = 2

LFLB 5	Conv2d
	Batch_normalization
	Max_pooling2d
	ReLu
		Filters = 16, Kernel_size = 3, Strides = 1
		Pool_size = 3, Strides = 2

LFLB 6	Conv2d
	Batch_normalization
	Max_pooling2d
	ReLu
		Filters = 16, Kernel_size = 3, Strides = 1
		Pool_size = 3, Strides = 2

Concatenation		Packages every 30 image features into a consecutive facial image feature sequence
Flatten		
Reshape		
LSTM		Unit = 20
Batch_normalization		
Dense (Output)		Unit = 8, Activation = "softmax"

4. Experimental Results

In this section, the experimental results on the RADESS, BAUM-1s and eNTERFACE'05 databases will be shown and discussed. Then, a comparison between the results of consecutive facial emotion recognition method proposed in this paper with those of other related research will also be made in this section.

4.1. Experiments on RAVDESS Database

For the RAVDESS database, the pre-processing and emotion recognition methods in Section 3 were applied for the experiments. According to the experiments, the proposed consecutive facial emotion recognition method achieves an average accuracy of 99.51% on testing through 10-fold cross-validation experiments. The experimental results for the loss and accuracy of each fold of training, validation, and testing are shown in Table 8. The loss function in the table was used to evaluate the cost of the model during the training, validation, and testing phases. In general, a model with a lower loss value achieves higher accuracy.

Table 8. The loss and accuracy of the proposed consecutive facial emotion recognition method during training, validation, and testing.

	Training		**Validation**		**Testing**	
	Loss	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**
Fold 1	0.0308	1.0000	0.0749	1.0000	0.4998	0.9919
Fold 2	0.0366	1.0000	0.0745	1.0000	0.4517	1.0000
Fold 3	0. 0192	1.0000	0.0415	1.0000	0.1363	1.0000
Fold 4	0.0206	1.0000	0.0428	1.0000	0.2667	0.9959
Fold 5	0.0369	1.0000	0.0593	1.0000	0.2978	0.9919
Fold 6	0.0310	1.0000	0.0703	1.0000	0.4527	0.9959
Fold 7	0.0179	1.0000	0.0382	1.0000	0.1913	0.9959
Fold 8	0.0118	1.0000	0.0206	1.0000	0.0459	0.9959
Fold 9	0.0225	1.0000	0.0378	1.0000	0.1757	0.9919
Fold 10	0.0348	1.0000	0.0769	1.0000	0.2238	0.9919
Average	0.0262	1.0000	0.0536	1.0000	0.2741	0.9951
Standard Deviation	0.000145785	0	0.000398173	0	0.022792131	1.01707×10^{-5}

Additionally, the normalized confusion matrix obtained from cross-validation is shown in Figure 8. The accuracy, precision, recall, and F1-score for every emotion label can be calculated using Equations (1) to (4) with the confusion matrix in Figure 9, and the results are presented in Table 9.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

where TP, FP, TN, and FN are the number of $True\ Positive$, $False\ Positive$, $True\ Negative$, and $False\ Negative$, respectively.

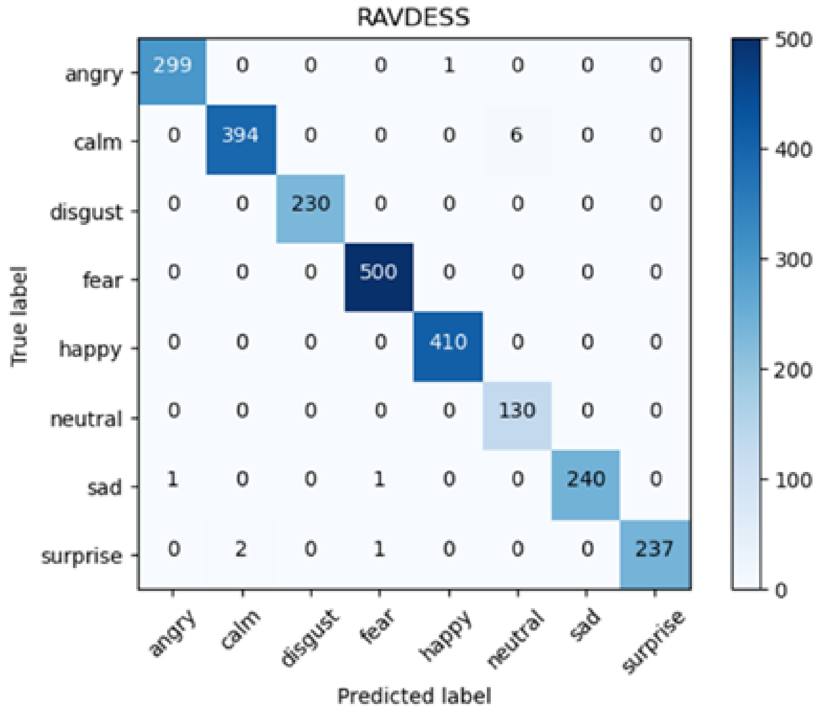
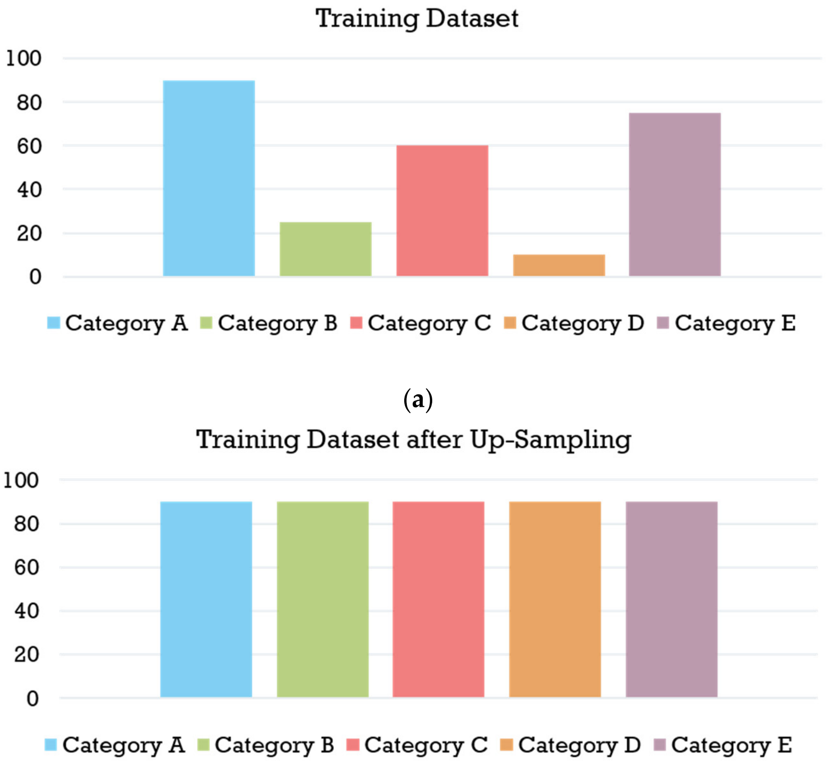


Figure 8. The confusion matrix obtained from 10-fold cross-validation of the consecutive facial emotion recognition method proposed in this paper.



(**a**)

(**b**)

Figure 9. Up-sampling for the BAUM-1s database. (**a**) Before up-sampling. (**b**) After up-sampling.

Table 9. The accuracy, precision, recall, and F1-score of each emotion calculated by the confusion matrix of the proposed consecutive facial emotion recognition method.

Label	Accuracy	Precision	Recall	F1-Score
Angry	0.9992	0.9967	0.9967	0.9967
Calm	0.9967	0.9949	0.9850	0.9899
Disgust	1.0000	1.0000	1.0000	1.0000
Fear	0.9992	0.9960	1.0000	0.9980
Happy	0.9996	0.9976	1.0000	0.9988
Neutral	0.9976	0.9559	1.0000	0.9774
Sad	0.9992	1.0000	0.9917	0.9959
Surprised	0.9988	1.0000	0.9875	0.9937
Average	0.9988	0.9926	0.9951	0.9938

This paper proposes a consecutive facial emotion recognition method and conducts experiments on the RAVDESS database. The experimental results in Table 9 show that this method performs well in terms of accuracy, precision, recall and F1-score, only performing a little poorly in predicting the emotion label of “neutral”. This is because the facial features of the emotion label “calm” are similar to those of the emotion label “neutral”. The degree of expression variation between these two emotions is small, making them more likely to be misclassified. However, our proposed method achieves an overall performance with an average value over 99% in accuracy, precision, recall and F1-score on the RAVDESS database. This verify the proposed model is precise and reliable. Based on these results, the method proposed in this paper is a feasible emotion recognition method and will have a stable performance in practical applications.

Table 10 presents a comparison of the experimental results of the proposed consecutive facial emotion recognition method with other related research on the RAVDESS database. It can be seen that the proposed method achieves an accuracy rate of 99.51%, which is higher than the methods proposed in other related research.

Table 10. Comparison of the experimental results of cross-validation for the proposed consecutive facial emotion recognition method with other related research on the RAVDESS database.

Method	Classes	Accuracy
E. Ryumina et al. [9]	8	98.90%
F. Ma et al. [11]	6	95.49%
A. Jaratrotkamjorn et al. [13]	8	96.53%
Z. Q. Chen et al. [14]	7	94%
Proposed model	8	99.51%

4.2. Experiments on BAUM-1s Database

Due to the data imbalance in the BAUM-1s database in which the emotion label “fear” accounts for only 6.99% of the data, and the emotion label “surprised” accounts for only 7.90%, the lower amount of learning data for these emotion labels during model training will affect the overall accuracy of the model. Therefore, this paper utilized up-sampling to balance the data within the training database by replicating data instances from the minority classes, thus ensuring an equal number of data samples for all emotion categories. Figure 9 shows the distribution of the BAUM-1s database before and after up-sampling.

By applying the pre-processing and emotion recognition methods in Section 3 for these experiments, the proposed consecutive facial emotion recognition method in Figure 6 achieves an average accuracy of 87.80% on the BAUM-1s database through 10-fold cross-validation experiments. The experimental results for the loss and accuracy of each fold of training, validation, and testing are shown in Table 11.

Table 11. The loss and accuracy of the proposed consecutive facial emotion recognition method during training, validation, and testing on the BAUM-1s database.

	Training		Validation		Testing	
	Loss	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**
Fold 1	0.0505	0.9137	0.9556	0.9327	0.5868	0.8600
Fold 2	0.0623	0.9951	0.2346	0.9405	0.8079	0.8600
Fold 3	0.0852	0.9764	0.5582	0.9428	0.6573	0.8800
Fold 4	0.0705	0.9553	0.4763	0.9053	0.4489	0.8600
Fold 5	0.0792	0.9202	0.8127	0.9492	0.5791	0.8400
Fold 6	0.0801	0.9015	0.6274	0.9266	0.4527	0.8600
Fold 7	0.0928	0.9589	0.3468	0.9134	0.4802	0.9200
Fold 8	0.0893	0.9668	0.7501	0.9431	0.6054	0.9000
Fold 9	0.0934	0.9015	0.4307	0.9519	0.4205	0.9400
Fold 10	0.0683	0.9907	0.6919	0.9203	0.6596	0.8600
Average	0.0772	0.9480	0.5884	0.9326	0.5698	0.8780
Standard Deviation	0.0141	0.0360	0.2225	0.0157	0.1214	0.0320

Additionally, the original and normalized confusion matrix obtained from the cross-validation experiment on the BAUM-1s database is shown in Figure 10, and the accuracy, precision, recall, and F1-score for each emotion are as shown in Table 12. As shown in Figure 10, the model tends to misclassify a significant number of angry instances (22 samples) and disgust instances (23 samples) as sad, as well as sad instances (8 samples) as angry. This is likely due to similar facial muscle activation patterns and overlapping expressions between these sentiment categories. These issues lead to the following consequences:

1. Lower recall for the angry and disgust classes due to misclassification.
2. Lower precision and accuracy for the sad class, since it receives many incorrect predictions.

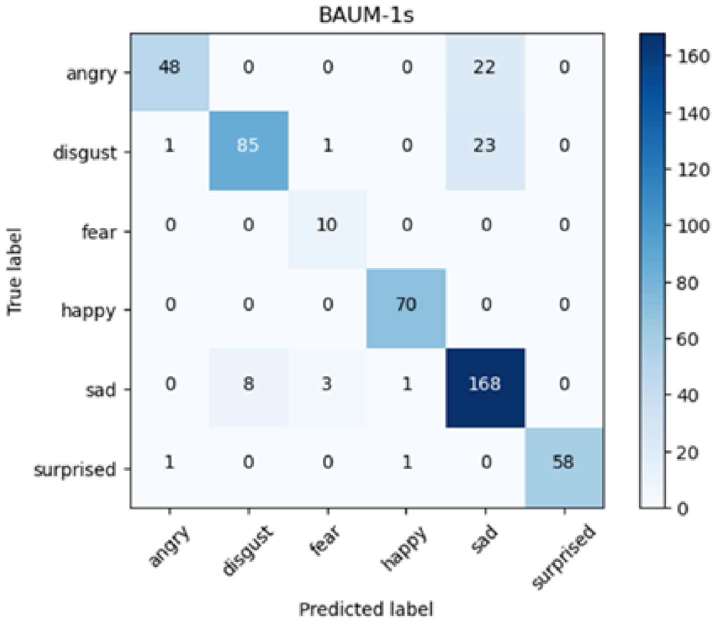


Figure 10. The confusion matrix obtained from 10-fold cross-validation of the consecutive facial emotion recognition method proposed in this paper on the BAUM-1s database.

This misclassification impacts the class-wise performance reported in Table 12, where most classes still achieve over 93% accuracy, except for sad. However, during 10-fold cross-validation, which is used to compute the average overall accuracy and loss reported in Table 11, the misclassifications in specific folds contribute to a lower average accuracy (~88%). The observed inconsistency reflects the imbalance in class-level performance affecting the macro-level average metrics.

Table 12. The accuracy, precision, recall, and F1-score of each emotion of the proposed consecutive facial emotion recognition method on the BAUM-1s database.

Label	Accuracy	Precision	Recall	F1-Score
Angry	0.9520	0.9600	0.6857	0.8000
Disgust	0.9340	0.9140	0.7727	0.8374
Fear	0.9920	0.7143	1.0000	0.8333
Happy	0.9960	0.9722	1.0000	0.9859
Sad	0.8860	0.7887	0.9333	0.8550
Surprised	0.9960	1.0000	0.9667	0.9831
Average	0.9593	0.8915	0.8931	0.8825

Moreover, Table 13 shows the comparison of the accuracy of the proposed method for consecutive facial emotion recognition and those of other related research for the BAUM-1s database. Again, according to the comparison in Table 13, the proposed strategy for consecutive facial emotion recognition outperforms the methods in the other papers.

Table 13. Comparison of the results of cross-validation for the proposed consecutive facial emotion recognition method with other related research on the BAUM-1s database.

Paper	Classes	Accuracy
F. Ma et al. [11]	6	64.05%
B. Pan et al. [27]	6	55.38%
P. Tiwari [28]	8	77.95%
Proposed model	6	87.80%

4.3. Experiments on eNTERFACE'05 Database

Similar to the experiments conducted in Sections 4.1 and 4.2 for the BAUM-1s database and RAVDESS database, respectively, the proposed consecutive facial emotion recognition method in Figure 7 achieves an average accuracy of 96.82% for the experiments on the eNTERFACE'05 database through the 10-fold cross-validation method. The experimental results for loss and accuracy of each fold of training, validation, and testing are shown in Table 14.

Table 14. The loss and accuracy of the proposed consecutive facial emotion recognition method during training, validation, and testing on the eNTERFACE'05 database.

	Training		Validation		Testing	
	Loss	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**
Fold 1	0.0437	0.9752	0.2978	0.9563	0.4727	0.9603
Fold 2	0.0389	0.9747	0.1925	0.9632	0.2063	0.9683
Fold 3	0.0471	0.9968	0.3194	0.9491	0.2167	0.9762
Fold 4	0.0423	0.9604	0.3751	0.9578	0.4335	0.9603
Fold 5	0.0312	0.9823	0.3530	0.9684	0.5808	0.9603
Fold 6	0.0430	0.9521	0.2496	0.9467	0.3345	0.9524
Fold 7	0.0488	0.9816	0.1693	0.9546	0.4768	0.9683
Fold 8	0.0495	0.9873	0.3847	0.9619	0.3880	0.9762
Fold 9	0.0456	0.9768	0.2319	0.9443	0.2920	0.9841
Fold 10	0.0345	0.9765	0.3890	0.9691	0.4589	0.9762
Average	0.0424	0.9763	0.2962	0.9571	0.3860	0.9682
Standard Deviation	3.62×10^{-5}	0.000160	0.0066	7.50×10^{-5}	0.0149	9.79×10^{-5}

Additionally, the original and normalized confusion matrix obtained from cross-validation experiment on the eNTERFACE'05 database is shown in Figure 11, and the accuracy, precision, recall, and F1-score for each emotion are as shown in Table 15. Table 16 shows the comparison of the accuracy of the proposed method for consecutive facial emotion recognition and those of other related researches for eNTERFACE'05 database. According to Table 16, it is obvious that the proposed methods achieves a much higher recognition rate than those by using the methods in the other papers. This again verifies the performance of the proposed method.

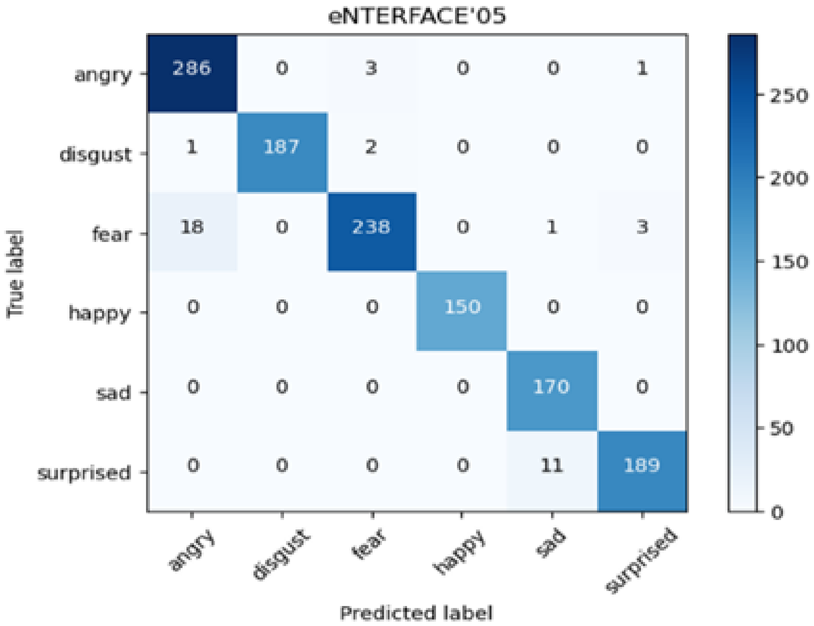


Figure 11. The confusion matrix obtained from 10-fold cross-validation of the consecutive facial emotion recognition method proposed in this paper on the eNTERFACE'05 database.

Table 15. The accuracy, precision, recall, and F1-score of each emotion of the proposed consecutive facial emotion recognition method on the eNTERFACE'05 database.

Label	Accuracy	Precision	Recall	F1-Score
Angry	0.9817	0.9377	0.9862	0.9613
Disgust	0.9976	1.0000	0.9842	0.9920
Fear	0.9786	0.9794	0.9154	0.9463
Happy	1.0000	1.0000	1.0000	1.0000
Sad	0.9905	0.9341	1.0000	0.9659
Surprised	0.9881	0.9793	0.9450	0.9618
Average	0.9894	0.9718	0.9718	0.9712

Table 16. Comparison of the results of cross-validation for the proposed consecutive facial emotion recognition method with other related research on the eNTERFACE'05 database.

Paper	Classes	Accuracy
F. Ma et al. [11]	6	80.52%
B. Pan et al. [29]	6	86.65%
P. Tiwari [30]	7	61.58%
Proposed model	6	96.82%

5. Model Inference and Experiments for FPGA Implementation

This paper implemented the proposed consecutive facial emotion recognition preprocessing methods and deep learning model introduced in Sections 3 and 4 on FPGA chip and then tested the performance of the chip using 10-fold cross-validation. Each testing database consisted of 246 consecutive image data. There are 30 facial images in each consecutive image data. The Terasic's VEEK-MT2S development board was utilized for the inference of the neural network model [31]. The VEEK-MT2S development board is a design environment that provides developers with the tools and resources required to create processing systems. This device supports the High-Level Synthesis (HLS) method for implementing deep learning models on an FPGA.

This paper applied the High-Level Synthesis (HLS) method for implementing the designed model on an FPGA chip. The designed model was first inferred using Python, and then compiled by a compiler on an ARM environment supported by Altera (Intel) to HDL (Hardware Description Language) languages Verilog/VHDL. The HDL was then ported onto the logic array of the FPGA to implement the model at the logic-circuit level. Please see Figure 12 for more details. This method makes the implementation of deep learning models more efficient and accelerates time to market. The FPGA in DE-10 Standard development board, which is the base of Terasic's VEEK-MT2S, was equipped with a dual-core ARM Cortex-A9 processor. In order to accelerate computation of the model on FPGA, the test database was divided into two sub-databases, and parallel computing was used to implement the neural network model inference. This reduced the execution time by half.

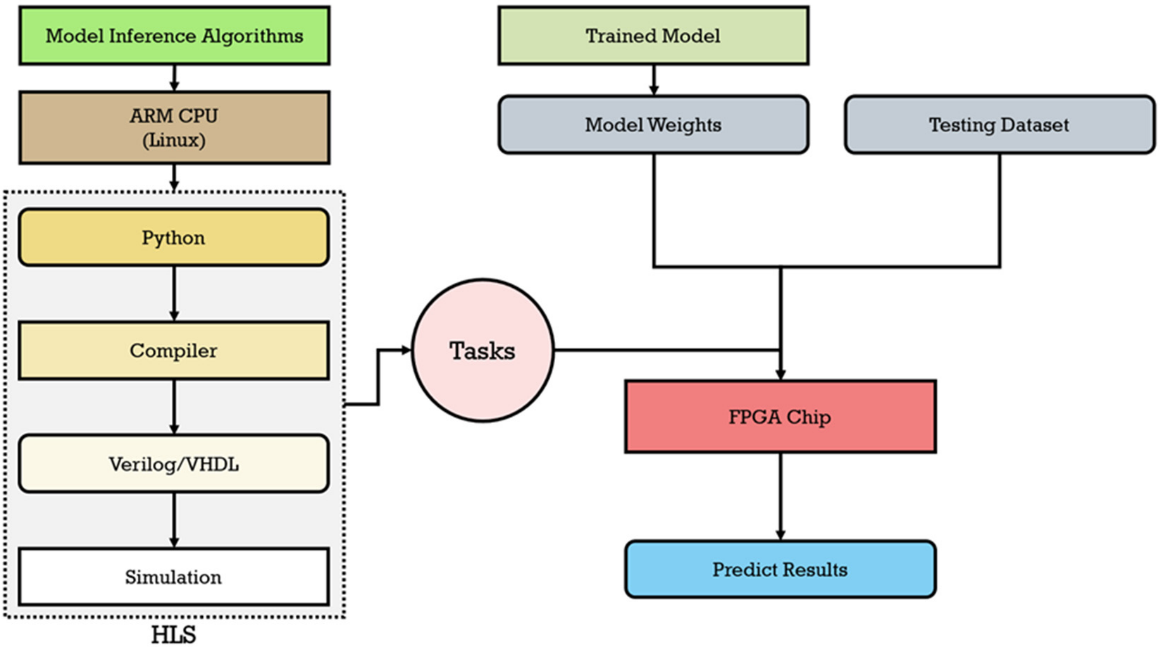


Figure 12. The HLS implementation of the proposed model on the FPGA chip.

The VEEK-MT2S development board can run the Linux Ubuntu 16.04 system using the DE10-Standard. The steps for the hardware implementation of the proposed model are as follows.

A. Download the required Linux image file and cross-compiler, which have to be suitable for the ARM architecture.
B. Connect to the DE10-Standard FPGA via JTAG and open the corresponding project file in the Quartus Prime software.
C. Use the Quartus software to download the Linux image file to the FPGA.

D. Configure the FPGA's boot settings to boot the Linux system from the SD card.

E. Add the downloaded cross-compiler to the system environment variables and compile the required applications or kernel modules, such as libraries of OpenCV and Python.

F. Copy the generated binary files (.bin) to the SD card and insert it into the FPGA. The Linux system can then be started.

5.1. Model Inference Algorithms

This paper used deep learning frameworks such as TensorFlow and Keras on a PC to train models for consecutive facial and speech emotion recognition. The model inference computation process was simulated in the Python programming language and implemented on the FPGA with a dual-core ARM Cortex-A9 processor, on which the Linux operation system is ported. This section will describe the algorithms used for the implementation of the proposed CLDNN model on this FPGA.

5.1.1. Inference of CNN

Algorithm 1 (Convolution algorithm) describes the computation process used to simulate the inference process of the model in the convolution layer. The Convolution algorithm first obtains the kernels (kernel) and biases (bias) from the parameters of trained model (weights). Then, it calculates half of the height and width of the kernel sizes (kernelsize) as the padding sizes (padheight and padwidth). And the zero-padding method (zeropadding) is used to pad with 0 to ensure that the edge pixel values of the input are involved in the convolution.

The size of input images for the first LFLB is 100 × 100 pixels. Each LFLB uses 16 filters. However, the kernel size, pooling size, and strides of the convolution and pooling layers vary across different LFLBs and are determined based on Table 7. During the convolution process, dot products (dot) are used to calculate the inner product between each input window (input[h][w]) and kernels (kernel) in the loop. Then, the biases (bias) are added to obtain the results of convolution (convolved). Finally, the results (convolved) are activated using ReLU, which is used to adjust the negative values to 0, to obtain the final the output (featuremap). In Algorithm 1, *inputs* are the images of testing database, *filters* are the tensor of convolutional filters, *kernelsize* are size of convolutional filters, *strides* are the stride of convolution, and *weights* are the parameters of convolution layers in the trained model.

Algorithm 1: Convolution

```
Input: input, filters, kernelsize, strides, weights
Output: featuremap
kernel, bias = weights
padheight = kernelsize // 2
padwidth = kernelsize // 2
paddedimage =zeropadding(image, (padheight, pad_width), 0)
for h in input do
  for w in input do
    convolved[h][w] = dot(input[h][w], kernel) + bias
featuremap = maximum(0, convolved)
return featuremap
```

5.1.2. Inference of LSTM

Algorithm 2 (LSTM algorithm) shows the computation process of the LSTM layer during testing [32]. The algorithm takes three inputs, *input*, *weights*, and *units*, and returns output sequence of *hiddenstate* as the result of the LSTM layer. In Algorithm 2, the inputs are the input sequences obtained from the preceding processes, including convolutional layers (LFLB) and flatten layers. The weights are the parameters of the LSTM layer obtained from trained models, and the units are the number of memory units in the LSTM layer, which was set to 20 according to the experiments in Section Training Process and Parameters. The parameters *W_i, W_f, W_c, W_o* are the weights of input gate, forget gate, output gate, and cell state, used for transferring input data to the hidden state. The parameters *U_i, U_f, U_c, U_o* are the weights of the four gates, used for transferring hidden state to the hidden state in time steps. The parameters *b_i, b_f, b_c, b_o* are the biases of the four gates.

Algorithm 2 mainly uses the four gates (input gate, forget gate, output gate, and cell state) to perform computations within the LSTM layer. The calculation for each element in the input sequence of LSTM is as follows. First, the inputs are calculated at the input gate (inputgate), with a sigmoid function that determines which information needs to be updated in the cell state. Next, the inputs are calculated at the forget gate (forgotstate), with a sigmoid function that determines which previously stored information in the cell state needs to be forgotten. Then, the inputs are calculated at the cell candidate gate (cellcandidategate), with a tanh function that generates a possible new candidate cell state value. The cell state (cellstate) can be updated by multiplying the previous value of cell state with the forget gate and adding the product of the input gate and cell candidate gate. Finally, the output gate (outputgate) is computed, with a sigmoid function that determines which information of cell state needs to be passed to the hidden state (hiddenstate). The hidden state (hiddenstate) is obtained by multiplying the cell state (cellstate) passing the result through a tanh function with the output gate (outputgate).

Algorithm 2 iterates over each element in the input sequence by updating the hidden state (*hiddenstate*) and cell state (*cellstate*) using the above steps. Then, Algorithm 2 returns the final hidden state (*hiddenstate*) as the output of the LSTM layer.

Algorithm 2: LSTM

```
Input: input, weights, units
Output: hiddenstate
$W, U, b = weights$
$W_i, W_f, W_c, W_o = W$
$U_i, U_f, U_c, U_o = U$
$b_i, b_f, b_c, b_o = b$
hiddenstate = cellstate = zerosarray(shape = (1, units))
for element in input do
    inputgate = sigmoid(dot(element,$W_i$) + dot(hiddenstate, $U_i$) + $b_i$)
    forgotgate = sigmoid(dot(element, $W_f$) + dot(hiddenstate, $U_f$) + $b_f$)
    cellcandidategate = tanh(dot(element, $W_c$) + dot(hiddenstate, $U_c$) + $b_c$)
    cellstate = forgotgate × cellstate + inputgate × cellcandidategate
    outputgate = sigmoid(dot(element, $W_o$) + dot(hiddenstate, $U_o$) + $b_o$)
    hiddenstate = outputgate × tanh(cellstate)
return hiddenstate
```

5.1.3. Inference of Batch Normalization

The normalization process for input data, whose size depends on the output of the pooling process in each LFLB, is described in Algorithm 3. The purpose of the normalization on input data is to maintain a smaller value range during training. This can then alleviate the situations of gradients exploding and vanishing and, hence, can improve model's stability and generalization ability.

In the training phase, this algorithm performs normalization based on the statistical information of each batch of training data. The normalization effect is optimized through adjusting the gamma and beta parameters. In the inference phase, the accumulated statistical information from the training process is used for normalization. This can maintain consistency of the model across different batches of data.

Algorithm 3: BatchNormalization

```
Input: input, weights
Output: normalized
gamma, beta, mean, variance = weights
normalized = $gamma \times (inputs - mean)/sqrt(variance + 1 \times 10^{-3.}) + beta$
return normalized
```

5.2. Experimental Results on FPGA Chip

This section reveals the results of the experiments conducted on the implemented CLDNN on the FPGA chip for emotion recognition. The accuracy values of each fold in the test and the average test time for a single consecutive image are shown in Table 17.

Table 17. Accuracies and execution time of testing the proposed consecutive facial emotion recognition model on FPGA.

	Accuracy (%)	Execution Time (sec)
Fold 1	99.19	11.19
Fold 2	100.00	12.04
Fold 3	100.00	12.15
Fold 4	99.59	11.24
Fold 5	99.19	11.20
Fold 6	99.59	12.17
Fold 7	99.59	12.06
Fold 8	99.59	11.45
Fold 9	99.19	11.92
Fold 10	99.19	11.59
Average	99.51	11.70
Variance	0.318915	0.410053
Standard Deviation	0.1017	0.1681

The experimental results show that the testing accuracies on the FPGA development board using the proposed neural network model inference algorithms are the same as those tested on a PC using deep learning frameworks such as TensorFlow and Keras for implementing the proposed model. This verifies that the implementation of the neural network model on the FPGA chip using inference algorithms proposed in this paper is efficient and performs well according to Table 17. Additionally, comparing the runtime of an AI model on an FPGA-based platform like the Terasic DE10-Standard with that on a general-purpose CPU (Intel i7-10700) is not straightforward. However, an approximation can be made based on general performance characteristics. The characteristics of the PC environment used in this paper, as described in Table 2, are as follows:

Intel Core i7-10700:

- Base Clock: 2.9 GHz (up to 4.8 GHz Turbo).
- 8 cores/16 threads.
- Peak MIPS: ~160,000–200,000 + MIPS.
- Average runtime for proposed model on a video: 0.45 s.

The characteristics of FPGA in Terasic DE10-Standard (VEEK-MT2S base) used in this paper are as follows:

- Dual-core ARM Cortex-A9 @ ~925 MHz.
- Theoretical MIPS: ~2000–2500 (combined).

Based on the MIPS characteristics, the performance of the ARM Cortex-A9 is roughly 1/80 to 1/100 of that of the i7-10700. Since the average runtime for the proposed model on a PC is 0.45 s per video, the expected runtime on the FPGA would be approximately 36 to 45 s. However, according to Table 17, the average runtime (execution time) for a video on the FPGA in the experiments presented in this paper is 11.7 s. This indicates that the proposed model has been efficiently implemented on the FPGA, achieving better performance than expected based on the PC comparison.

Moreover, Table 18 shows the average execution time and proportion of each convolutional layer, batch normalization layer, max pooling layer, LSTM layer, and fully connected layer when testing a single consecutive image data for more detailed analysis. It can be seen that, due to the first convolutional layer expanding the dimensions of the image data to 32, the input data for the second convolutional layer is much larger than that of the first layer. As a result, the execution time of the second convolutional layer is the longest at 6.9281 s (59.21%). Furthermore, since the proposed consecutive facial emotion recognition model incorporates max pooling layers, which can reduce the dimension of the feature maps, the execution time of each convolutional layer decreases sequentially. However, the average execution time for running the proposed deep learning model CLDNN to recognize emotion from a video is only 11.7 s. Additionally, the average testing time on the implemented FPGA chip for a single image data was 0.39 s per image of size 100 × 100 pixels. In the scenario of real application for emotion detection using the implemented FPGA chip, users' facial images will be captured for a fixed duration, such as 1 s, and their features will be calculated immediately within less than 0.39 s using the convolutional layers. After completing the feature extraction of consecutive 30 images (30 s in this example), these features are collected and fed into the LSTM and fully-connected layers for emotion recognition. According to Table 18, it can be observed that the computation time for the LSTM and fully-connected layers will take only a few milliseconds. This indicates that the designed chip can achieve real-time detection of users' emotions. This also verifies that the implemented AI (artificially intelligent) chip based on the FPGA is feasible and is suitable for AI edge computing application.

Table 18. Execution time and proportions of each layer in proposed consecutive facial emotion recognition model on FPGA.

Layer	Execution Time (sec)	Proportion (%)
Conv2D_1	1.3649	11.66
Batch_Normalization_1	0.0008	Less than 0.01
Max_Pooling2D_1	0.8733	7.46
Conv2D_2	6.9281	59.21
Batch_Normalization_2	0.0010	Less than 0.01
Max_Pooling2D_2	0.2260	1.93
Conv2D_3	1.5648	13.37
Batch_Normalization_3	0.0009	Less than 0.01
Max_Pooling2D_3	0.0755	0.64
Conv2D_4	0.4721	4.03
Batch_Normalization_4	0.0006	Less than 0.01
Max_Pooling2D_4	0.0386	0.32

Table 18. *Cont.*

Layer	Execution Time (sec)	Proportion (%)
Conv2D_5	0.0468	0.40
Batch_Normalization_5	0.0006	Less than 0.01
Max_Pooling2D_5	0.0248	0.21
Conv2D_6	0.0322	0.27
Batch_Normalization_6	0.0005	Less than 0.01
Max_Pooling2D_6	0.0221	0.18
LSTM	0.0071	0.06
Batch_Normalization_7	0.0000	Less than 0.01
Dense(Softmax)	0.0002	Less than 0.01
Total	11.70	100%

Moreover, the estimated logic gate usage rate for the proposed model implemented on the DE-10 FPGA is approximately 69% of the available LEs, with:

- DSP Block Usage: ~70% (for parallel multipliers).
- Memory Usage: ~50% (for weights and intermediate data).

This indicates a moderate-to-high resource utilization, leaving room for additional optimizations or smaller-scale expansions. The FPGA's parallelism is effectively leveraged for real-time emotion recognition, as confirmed by the experimental results.

6. Conclusions

This paper proposes the methods based on deep learning for consecutive facial emotion recognition. The proposed model was implemented on an embedded system with an FPGA chip without the need for a deep learning framework during the model inference process. For consecutive facial emotion recognition, this paper captured 30 frames of an image sequence to represent a consecutive image segment. The Haar cascade frontal face detection model from OpenCV was utilized to extract the facial regions from the images, followed by grayscale conversion and resizing to reduce the computational burden on the embedded device. The preprocessed images were then fed into local feature learning blocks to extract local features from individual frames. These features were then packaged into a feature sequence representing a consecutive image segment. The feature sequence was then passed through an LSTM layer for temporal sequence learning. Finally, a fully connected layer was used for classification.

For the robustness across databases of the proposed model, the proposed model achieves accuracies on different databases according to the experimental results in this paper (RAVDESS: 99.51%; eNTERFACE'05: 96.82%; BAUM-1s: 87.80%). These results suggest the model generalizes well to other balanced databases (RAVDESS, eNTERFACE'05), but class imbalance and smaller sample size in BAUM-1s reduce robustness. This demonstrates that, while the architecture is strong, robustness depends on database quality and balance.

Next, the parameters of the deep learning models for consecutive facial emotion recognition, as well as the test database, were fed into the FPGA's memory for model inference. This research implemented the neural network model inference algorithms in Python. Then, through high-level synthesis, the algorithms were automatically transformed from high-level language into circuit functionality. This allowed us to achieve model inference on the embedded device without the need of deep learning frameworks. For the model inference of consecutive facial emotion recognition, the proposed method achieved the same test accuracy as that tested on a PC using deep learning frameworks. This indicates that the neural network model inference algorithms proposed in this paper can achieve the same performance as using the deep learning frameworks. The average testing time for a single consecutive image data was 11.70 s, with an average testing time of 0.39 s per single image of a size of 100 × 100 pixels. The implemented hardware had an FPS of 2.56. The experimental results for the designed FPGA chip verify that the implemented AI (artificially intelligent) chip based on FPGA is feasible and is suitable for AI edge computing application.

Finally, according to the experimental results in Section 4, the proposed deep learning model applied on the three RADVESS, BAUM-1s and IEMOCAP databases achieves much higher recognition rates than those in other papers. This demonstrates that the proposed methods outperform the methods in the literature.

Author Contributions: Conceptualization, methodology, formal analysis, writing—review and editing, S.-T.P.; software, validation, data curation, writing—original draft preparation, H.-J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was supported by the Ministry of Science and Technology of the Republic of China under contract NSTC 114-2221-E-390-005.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Matsugu, M.; Mori, K.; Mitari, Y.; Kaneda, Y. Subject Independent Facial Expression Recognition with Robust Face Detection using a Convolutional Neural Network. *Neural Netw.* **2003**, *16*, 555–559. [CrossRef] [PubMed]
2. Pramerdorfer, C.; Kampel, M. Facial Expression Recognition using Convolutional Neural Networks: State of the Art. *arXiv* **2016**, arXiv:1612.02903. [CrossRef]
3. Ayadi, M.E.; Kamel, M.S.; Karray, F. Survey on Speech Emotion Recognition: Features, Classification Schemes, and Databases. *Pattern Recognit.* **2011**, *44*, 572–587. [CrossRef]
4. Khalil, R.A.; Jones, E.; Babar, M.I.; Jan, T.; Zafar, M.H.; Alhussain, T. Speech Emotion Recognition Using Deep Learning Techniques: A Review. *IEEE Access* **2019**, *7*, 117327–117345. [CrossRef]
5. Hauck, S.; DeHon, A. *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*; Morgan Kaufmann: San Francisco, CA, USA, 2007; ISBN 9780080556017.
6. Pellerin, D.; Thibault, S. *Practical FPGA Programming in C*; Prentice Hall Press: Upper Saddle River, NJ, USA, 2005; ISBN 9780131543188.
7. Kilts, S. *Advanced FPGA Design: Architecture, Implementation, and Optimization*; Wiley-IEEE Press: New York, NY, USA, 2007; ISBN 9780470054376.
8. Sainath, T.N.; Vinyals, O.; Senior, A.; Sak, H. Convolutional Long Short-Term Memory Fully Connected Deep Neural Networks. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015.
9. Ryumina, E.; Karpov, A. Facial Expression Recognition using Distance Importance Scores Between Facial Landmarks. In Proceedings of the 30th International Conference on Computer Graphics and Machine Vision (GraphiCon 2020), St. Petersburg, Russia, 22–25 September 2020; pp. 1–10.
10. Sagonas, C.; Tzimiropoulos, G.; Zafeiriou, S.; Pantic, M. 300 Faces in-the-Wild Challenge: The first facial landmark localization Challenge. In Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops, Sydney, NSW, Australia, 2–8 December 2013; pp. 397–403.
11. Ma, F.; Zhang, W.; Li, Y.; Huang, S.L.; Zhang, L. Learning Better Representations for Audio-Visual Emotion Recognition with Common Information. *Appl. Sci.* **2020**, *10*, 7239. [CrossRef]
12. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment using Multitask Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
13. Jaratrotkamjorn, A.; Choksuriwong, A. Bimodal Emotion Recognition using Deep Belief Network. In Proceedings of the 2019 23rd International Computer Science and Engineering Conference (ICSEC), Phuket, Thailand, 30 October–1 November 2019; pp. 103–109.
14. Chen, Z.Q.; Pan, S.T. Integration of Speech and Consecutive Facial Image for Emotion Recognition Based on Deep Learning. Master’s Thesis, National University of Kaohsiung, Kaohsiung, Taiwan, 2021.

15. Li, S.; Deng, W.; Du, J. Facial Emotion Recognition Using Deep Learning: A Survey. *IEEE Trans. Affect. Comput.* **2023**, *14*, 1234–1256.
16. Wang, K.; Peng, X.; Qiao, Y. Emotion Recognition in the Wild Using Multi-Task Learning and Self-Supervised Features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 5678–5687.
17. Zhang, Y.; Wang, C.; Ling, H. Cross-Domain Facial Emotion Recognition with Adaptive Graph Convolutional Networks. *IEEE Trans. Image Process.* **2024**, *33*, 1123–1135.
18. Chen, L.; Liu, Z.; Sun, M. Efficient Emotion Recognition from Low-Resolution Images Using Attention Mechanisms. In Proceedings of the ACM International Conference on Multimedia (ACM MM), Ottawa, ON, Canada, 29 October–3 November 2023; pp. 2345–2354.
19. Gupta, A.; Narayan, S.; Patel, V. Explaining Facial Emotion Recognition Models via Vision-Language Pretraining. *Nat. Mach. Intell.* **2024**, *6*, 45–59.
20. Zhao, R.; Elgammal, A. Dynamic Facial Emotion Recognition Using Spatio-Temporal 3D Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 9876–9885. [CrossRef]
21. Livingstone, S.R.; Russo, F.A. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A Dynamic, Multimodal Set of Facial and Vocal Expressions in North American English. *PLoS ONE* **2018**, *13*, e0196391. [CrossRef]
22. Dataset BAUM-1. Available online: https://archive.ics.uci.edu/dataset/473/baum+1. (accessed on 10 July 2023).
23. Dataset eNTERFACE'05. Available online: https://enterface.net/enterface05/emotion.html?utm_source=chatgpt.com. (accessed on 5 May 2023).
24. Adeshina, S.O.; Ibrahim, H.; Teoh, S.S.; Hoo, S.C. Custom Face Classification Model for Classroom using Haar-like and LBP Features with Their Performance Comparisons. *Electronics* **2021**, *10*, 102. [CrossRef]
25. Wu, H.; Cao, Y.; Wei, H.; Tian, Z. Face Recognition based on Haar Like and Euclidean Distance. *J. Phys. Conf. Ser.* **2021**, *1813*, 012036. [CrossRef]
26. Gutter, S.; Hung, J.; Liu, C.; Wechsler, H. *Comparative Performance Evaluation of Gray-Scale and Color Information for Face Recognition Tasks*; Springer: Heidelberg/Berlin, Germany, 2021; ISBN 9783540453444.
27. Bhattacharya, S.; Kyal, C.; Routray, A. Simplified Face Quality Assessment (SFQA). *Pattern Recognit. Lett.* **2021**, *147*, 108–114. [CrossRef]
28. Khandelwal, A.; Ramya, R.S.; Ayushi, S.; Bhumika, R.; Adhoksh, P.; Jhawar, K.; Shah, A.; Venugopal, K.R. Tropical Cyclone Tracking and Forecasting Using BiGRU [TCTFB]. *Res. Sq.* **2022**. [CrossRef]
29. Pan, B.; Hirota, K.; Jia, Z.; Zhao, L.; Jin, X.; Dai, Y. Multimodal Emotion Recognition Based on Feature Selection and Extreme Learning Machine in Video Clips. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 1903–1917. [CrossRef]
30. Tiwari, P.; Rathod, H.; Thakkar, S.; Darji, A. Multimodal Emotion Recognition Using SDA-LDA Algorithm in Video Clips. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *14*, 6585–6602. [CrossRef]
31. Available online: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=205&No=1176. (accessed on 1 June 2022).
32. Zarzycki, K.; Ławrynczuk, M. LSTM and GRU Neural Networks as Models of Dynamical Processes used in Predictive Control: A Comparison of Models Developed for Two Chemical Reactors. *Sensors* **2021**, *21*, 5625. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.