Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Analysis of Convolutional Neural Networks for Facial Expression Recognition on GPU, TPU and CPU

Anbananthan Pillai Munanday[1], Norazlianie Sazali[2,*], Wan Sharuzi Wan Harun[3], Kumaran Kadirgama[3], Ahmad Shahir Jamaludin[2]

[1] Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
[2] Faculty of Manufacturing and Mechatronic Engineering Technology, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
[3] Faculty of Mechanical and Automotive Engineering Technology, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia

ARTICLE INFO

Article history:
Received 20 April 2023
Received in revised form 15 July 2023
Accepted 21 July 2023
Available online 5 August 2023

Keywords:
Artificial Intelligence; Artificial Neural Networks; Convolutional Neural Networks; GPU; CPU; TPU

ABSTRACT

In light of the increasing computational capacity provided by Central Processing Units (CPUs), Graphics Processing Units (GPUs), and Tensor Processing Units (TPUs), all of these were designed to speed up deep learning workloads, and the fact that this iteration of human-computer interaction is becoming more natural and social, it is clear that the field of human-computer interaction is poised for significant growth. The scientific community has found emotion recognition to be of tremendous interest and significance. Despite these advances, it is still desired that research into computational methods for identifying and recognizing emotions at the same ease as humans. This study uses Convolutional Neural Networks (CNN) for human emotion identification from facial expressions to delve deeper into this topic. The results demonstrated that training an Artificial Neural Networks (ANN) on GPUs might cut computational time by as much as 90% while accuracy could be raised up to 65%.

1. Introduction

1.1 Human Emotion Recognition

In recent years, there has been a significant rise in the number of people who think about enhancing human-computer connection. For an intelligent human-computer interface to work, the computer must relate to the user intuitively and comfortably [1]. Affective Computing, Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) can be used to help and encourage this kind of integration. Methods like these allow us to teach computers to recognize, model, and express emotions and how to respond to those emotions. It takes some commitment to programme a computer to understand human emotions, model those feelings, and convey them, as stated by Leao *et al.,* studies [2-5]. People communicate with one another using a combination of words and the nonverbal cues of body language, including gestures and facial expressions [6].

* *Corresponding author.*
E-mail address: azlianie@ump.edu.my

https://doi.org/10.37934/araset.31.3.5067

Research into the detection of emotions has gained momentum in recent decades that will categories the feeling of an individual face into one of seven classifications, namely happiness, sadness, anger, scared, surprise, disgust and neutral [7]. Data from the following sources have been examined in the long-standing study of human emotional expression: texts, the transmission of emotions, and the analysis of recorded speech and facial expressions [8]. In addition, as a significant role in interpersonal relationships, a person's facial expressions also provide information about that person's mood and intentions as they speak. The scientific community is interested in the recognition of such emotions since it opens the door to developing various applications of human-computer interaction [9].

More applications of face recognition technology appear in our daily lives [10], from paying for things to waking up a snoozing driver and customizing menu's and even sending out trade-specific advertisements. Envision a shop that arranges its inventory according to the feedback it receives from its clients. In this manner, we may attract more consumers by giving the most popular items and more prominent placement in the store or storefront. Wagner *et al.,* [11] ultimate goal is to enable the recognition of the seven emotions listed by Ekman *et al.,* [17] as the most fundamental: happiness, sadness, anger, scared, surprise, disgust and neutral as shown in Figure 1. Images of human face expressions will be utilized for the purpose, with computational vision and ANN methods employed to extract the attributes. The architecture of ANN is illustrated in Figure 2. Based on this figure, there are number of values consists to the input of a neuron, which are x_1 , x_m while single value for output, y_1. Continuous values represent as both input and output values, at the range of (0, 1). ANN neuron does the following [12]:

i. Inputs of x_1 , . . , x_m computes the weighted summation, where the weights are w_1 , . . , w_m
ii. subtracting a predefined threshold T
iii. nonlinear function results as an output, such as sigmoid function

However, this paper analyzes and compares CNN timings across with CPU, GPU, and TPU architectures as our primary focus. Besides looking into how well various ANN topology's function.

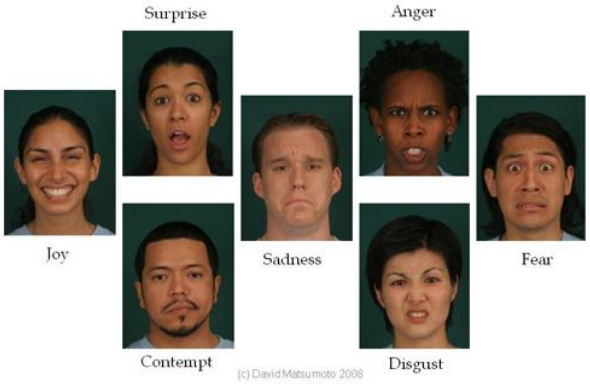

Fig. 1. Universal Facial Emotions [11]

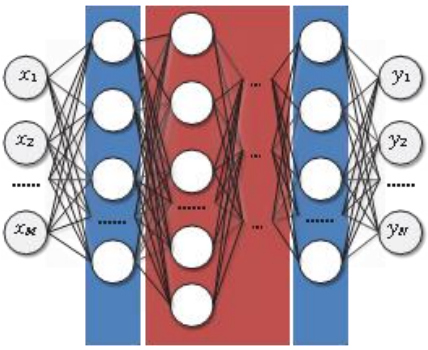

Fig. 2. CNN conceptual diagram (Left blue – Input Layer, Middle red – Hidden Layer, Right blue – Output Layer) [12]

1.2 Review of Previous Related Work

Bartlett *et al.,* [13] presents research on the problem of automatically detecting faces in a video stream and dynamically recording visual expression. The G. Littleworth *et al.,* discuss that face-to-face communication is a real-time operation with a time scale of 40 milliseconds. In real time, this function can recognize seven distinct feelings which is a big improvement over prior works: neutral, anger, disgust, fear, joy, sadness, and surprise. The CohnKanade AU-Coded Expression Database [14] was used throughout system training and evaluation, dataset for action unit shown in Figure 3. There are 210 people facial muscle movements in this database, all of adult age (18-50), 69% of them female and 31% male, 81% are of European ancestry, 13% are of African ancestry, and 6% are of some other racial or cultural background. Through experimental evaluation, it was determined that there was least discernible performance gap between the automatic detection strategy and the manual detection approach. With seven possible facial expressions to choose from, the algorithm achieved 93% identification in accuracy.

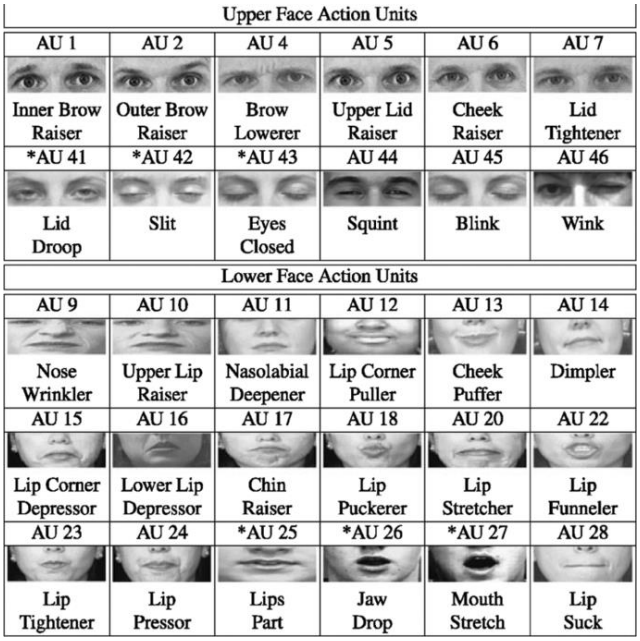

Fig. 3. Facial Action Coding System (FACS) [13]

The goal of Tang and Huang's *et al.,* [15] study is to use 3D geometry to identify the six most common human emotions expressed in facial expressions, as Figure 4. This method focuses on characteristics that are insensitive to changes in lighting or posture, which the researchers see as a significant challenge for 2D facial identification. The BU-3DFE database [16] served as the basis for this study's training and evaluation procedures. The 100 people included in this database represent a wide range of demographics and socioeconomic backgrounds. There are more women than men, and a wide range of Asian, Latinx, and other ethnicities are represented. In this study, it was discovered that the method increased the typical recognition rate by 3.5%. The overall accuracy of this method was 87.1%, with the greatest accuracy being 99.2% for the recognition of the surprise emotion on the face.

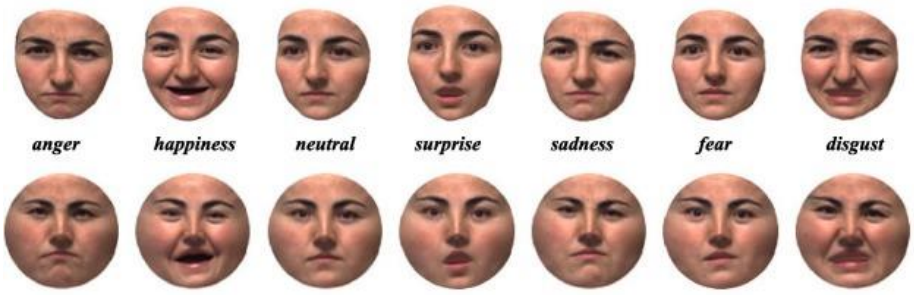

Fig. 4. 3D into 2D face mapping of one example from the BU-3DFE database with the six universal expressions and neutral [15]

Amin *et al.,* [17] created an ANN to recognize emotions from facial expressions [18] using deep learning algorithm. As stated by the researcher, an average accuracy of 60% is achieved when convolutional neural networks are used in the method of emotion recognition. The study was developed using data from the Facial Expression Recognition 2013 (FER-2013) database [19]. There are 35887 photos in this collection that Pierre Luc Carrier and Aaron Courville created. The database used to train the network consists of 48x48 grayscale pictures of human faces, each of them has been labelled with one of seven emotions. In Figure 5, demonstrates few labelled expression examples. Scale, rotation, and lighting all show significant differences throughout the dataset. The six facial expressions of Ekman [20] have been added to the samples in addition to the neutral expression [11]. The samples in this database are split between 7215 images of happiness, 436 disgusting photos, 4097 fear images, 4965 neutral images, 3995 anger images, 3171 surprising images, and 4830 sad images. The research produced positive findings, with an average categorization accuracy for the seven emotions of 61.05%. Researcher also determined that pleasure had the greatest recognition rate after analyzing the data. The methodology, however, struggles to accurately categorize the feelings of fear and sadness, since it shares similar facial traits [21].

This study mainly creates a CNN that detects facial expressions from real-time environment and determines the emotion depicted based on the different architecture's performance, which is different from the studies that have been given. The performance and training time in the CPU, GPU, and TPU architectures will be examined in addition to suggesting the creation of a CNN. This is to obtain a successful outcome as soon as feasible.

Fig. 5. Extraction of grayscale images of FER -2013 database with unique faces and emotions [17]

2. Methodology

2.1 Computing Platform Specifications

For the analysis of the proposed CNN, development environment Google Colab and the programming language Python were used. Google Colab is a free cloud service hosted by Google for Machine Learning and Deep Learning, with free GPU accelerators, pre-installed libraries, built based on Jupyter Notebook, supports bash commands and stores the notebooks in the Drive itself. The main libraries used are TensorFlow 2.0 and Keras, which are focused on deep machine learning. Keras is the most used framework in the area for its easiness. In TensorFlow 2.0, Keras has been "embedded" into TensorFlow through the module tf.keras. CNN also uses the tools: OpenCV, Scikit-Learn, NumPy, Pandas, Matplotlib. To evaluate the performance of deep learning, the multi-core computing platform have been utilised in this study as illustrated in Figure 6. In addition, specifications of each computing platform components listed in detailed on Table 1.

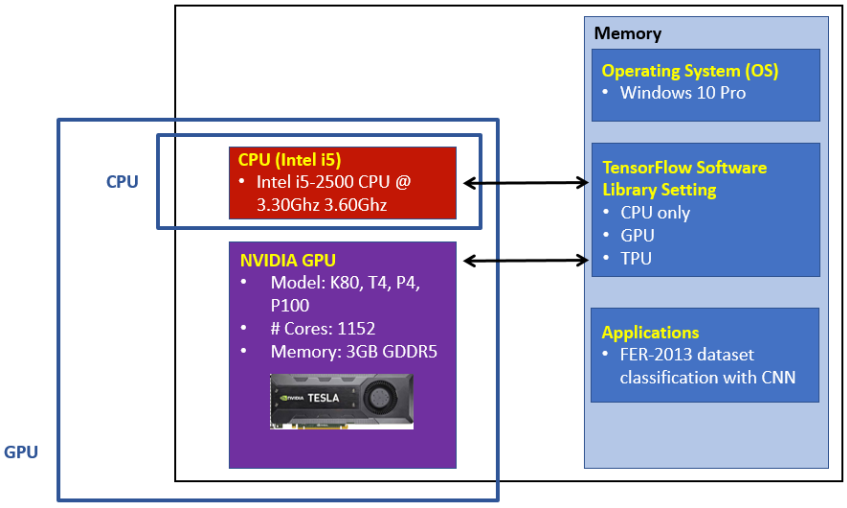

Fig. 6. A system model reference

Table 1
Computing platform specification

System	Specification
CPU	Intel® Core ™ i5-2500 CPU@3.30Ghz • RAM: 8GB
GPU	NVIDIA Tesla • Model: K80, T4, P4, P100 • Memory Build-in: 3GB • NVIDIA CUDA Cores: 1152 • Memory: 3GB GDDR5 • Speed Memory: 8Gbps
Software	OS: Windows 10 Pro (64-bits) • TensorFlow Version: tensorflow-gpu-1.0.1 • CUDA Version: cuda_8.0.61

2.2 Dataset Creation

For this research, the FER-2013 (Facial Expression Recognition 2013) database have been used. This database compiles a published open-source data collection produced by Pierre-Luc Carrier and Aaron Couville for a Kaggle contest. There are 35,887 images in this dataset, each of the 48-by-48-pixel grey-scale facial image categorized into seven distinct emotions. Each image has been automatically adjusted so that the subject's face is about in the same position and size. The objective is to classify each face into one of seven categories that are associated with different emotions and characteristics. Column headings in the file FER2013.csv are "emotions" and "pixels". A number code from 0 to 6 may be seen in the "expression" column. The grayscale values for each pixel in the image are listed in the "pixels" column.

2.3 CNN Architecture

There are a few data transformations that must be applied to the FER2013.csv file before work can begin on a CNN. The information in the csv file is in string format, thus the *tolist()* method is used to transform the data from the database to an array. It was decided to use a convolutional neural network, which has shown useful in image processing and analysis. Accordingly, its creation necessitates the following four stages.

i. **Step 1 Convolution Operator**: This process is analogous to applying filters to an image, with each filter focusing on a different and smaller section of the image. For instance, a filter that sweeps through a 3x3 section of a 48x48 image in a single hop will cause the entire image to scroll for a 48x48 image. The 3x3 filter applies a feature detector defined by the library being used to execute multiplication on each individual data point, ultimately producing a feature map.
ii. **Step 2 Pooling**: The function of this layer is to serve as a simplified representation of the information contained in the previous layer, the map of characteristics in this example. Similar to the convolution, an area unit is designed to pass over the whole output of the preceding layer, often using a 2x2 matrix as shown in Figure 7. The function of the unit is to reduce the data from that domain to a single value. For this section, data summarization method was chosen. MaxPooling is the most popular approach because it returns the largest possible unit number and sends that value to the output. This data simplification helps the Neural Network learn fewer weights and prevents overfitting.

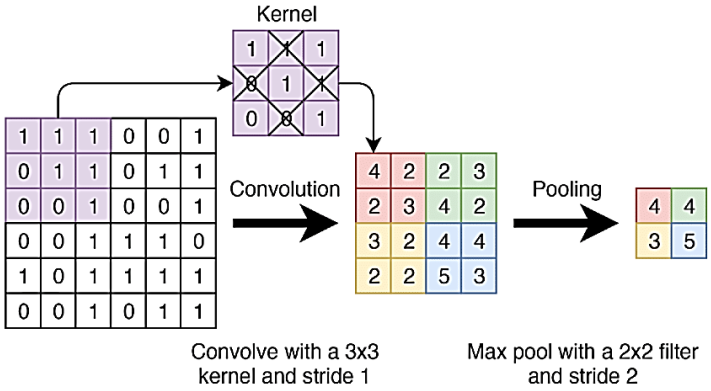

Fig. 7. Conversion of Pooling layer

iii. **Step 3 Flattening**: The image matrix produced in the Pooling phase serves as input for this step, which essentially transforms the matrix into a characteristic vector by modifying its format, as shown in Figure 8.

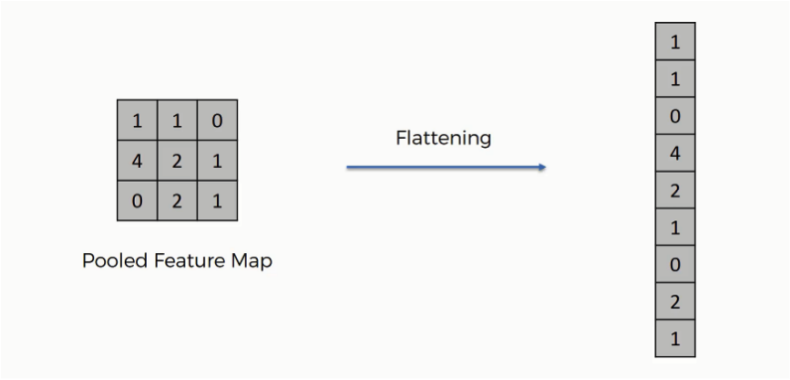

Fig. 8. Flattening conversion

iv. **Step 4 Neural Network**: Neural Networks are a computer model based on the human central nervous system, and at this point they have been fed the data from the Flattening layer and the characteristics vector for training. Here, we will be able to identify patterns in a big amount of data and assign them to a predetermined group.

2.3.1 Proposed CNN structure

The Figure 9 demonstrates the sequential model is used to create the CNN structure. There are 4 kernel size of 3x3 convolutional layers, pool size of 2x2 with two max-pooling layers, one flattening layer and two dense layers. Lastly, the SoftMax activation function is then used to categorise the seven human face emotions.

Figure 10 shows a summary of the proposed CNN architecture. Conv2D constructs a convolutional layer with 32 filters and a kernel size of 3 x 3 in block-1, padding the images with the same amount of pad before applying the relu activation. Then, 64 filters are used to generate a new convolutional layer by conv2D (conv2D 1). Batch Normalization is then implemented. A pooling layer with a 2 x 2 pool size is created by MaxPooling2D. In order to disregard the neurons selected at random during training, a Dropout is lastly set to 0.25. The layers in Block-2 are identical to those in Block-1, except the convolutional layers have 128 (for conv2D2) and 256 (for conv2D3), respectively, filters with a kernel regularize that penalises the layer's kernel by an L2 normalisation penalty of 0.01. A 1024-neuron dense layer using the relu activation function follows the flatten layer in Block-3. Finally, 50%

of the neurons selected at random are ignored using a Dropout of 0.5. The Dense layer, which is made up of seven neurons and the SoftMax activation function to categorise human emotions, is the last component of block-4.

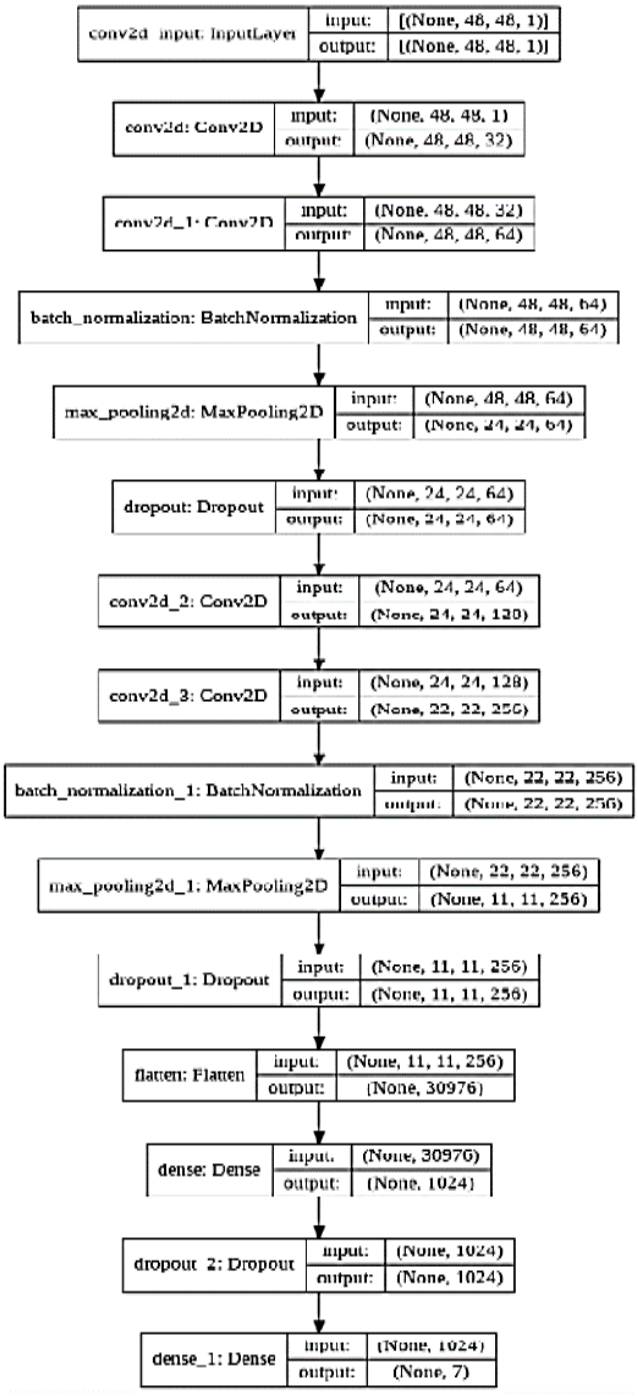

Fig. 9. Proposed structure of CNN

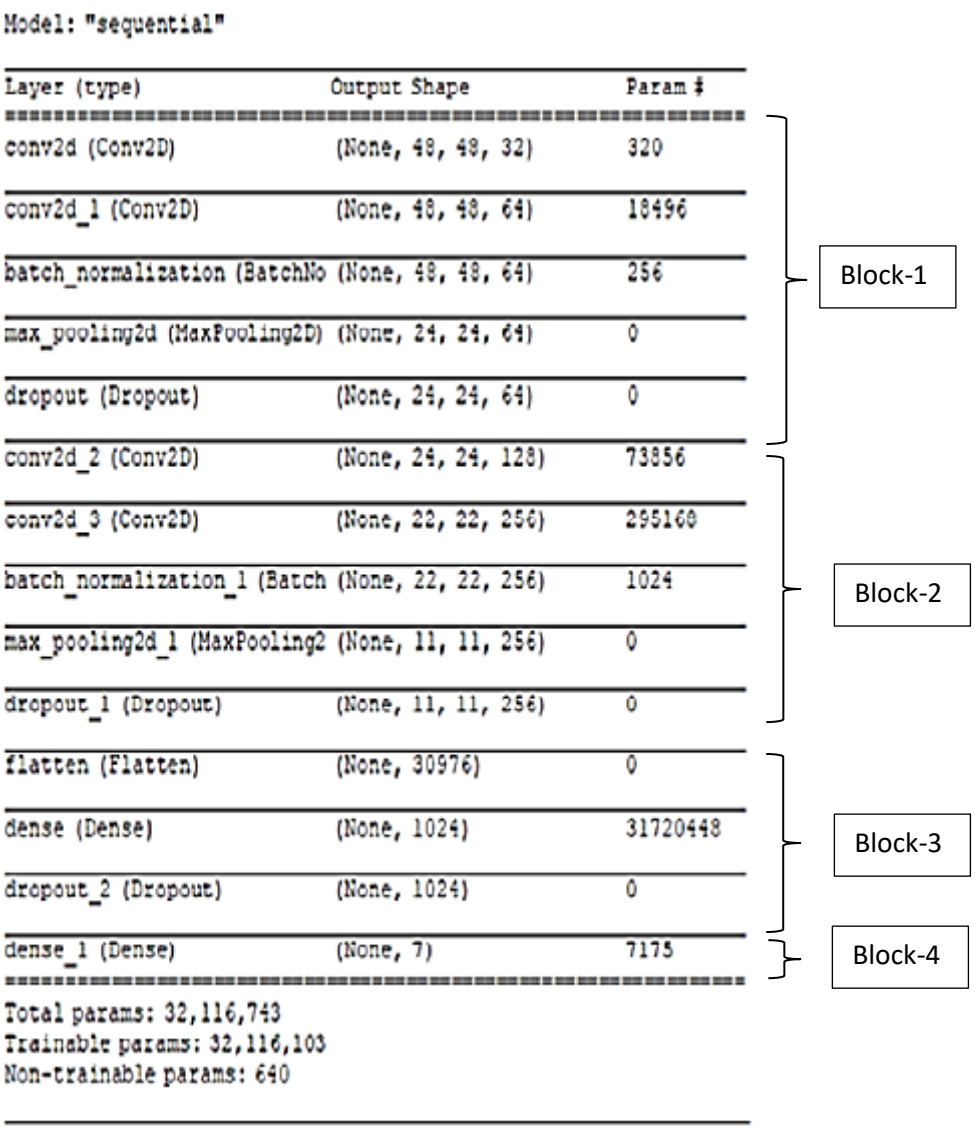

Fig. 10. Proposed structure of CNN architecture summary

2.4 System Design

Through the use of software, a camera is used to capture and recognise a person's facial expressions in real-time environment. By using the Viola Jones method and the Haarcascade frontal face dataset, among other ways, it is possible to distinguish the face region from a non-facial region in the camera and generate a rectangle frame on the face area. Captured images from real-time will be pre-processed for sharpening and enhance the image quality for feature extraction. The Trainer folder contains Trainer.xml, which contains the FER 2013 training dataset. A trained dataset is utilised to match the face in a video camera with the face in the dataset during the Face Detection process. A person's face will go through classification if it matches one in the trained dataset. Convolutional neural networks and the FER2013 database are used to do the classification on the acquired face. Based on the individual's characteristics, the facial expression reveals the likelihood of achieving the highest possible level of expression during the classification. One of seven possible facial expressions is presented in conjunction with the recognised picture of the subject.

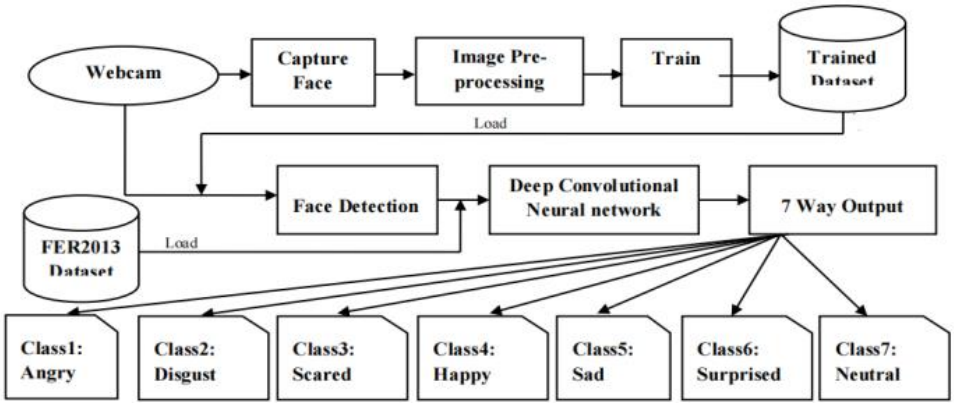

Fig. 11. Diagram of system design structure

3. Results

3.1 Performance Analysis of Convolution Neural Networks on GPU, TPU, and CPU Platforms

Using the NVIDIA Tesla K80, T4, P4, and P100 GPUs that are freely available through Google Colab, the CNN tests were run 10 times to acquire the average execution times and Speedup results for each. The highest speedup achieved by CNN in comparison to CPU shown in Figure 12. The best performance was attained by the Tesla P100. When compared to the same technique on a central processing unit (CPU), the GPU-based implementation is 10.71 times faster. As a result, we were able to cut the running duration of the leaping sequence by 90.66 percent, from 3950.44 seconds to 368.81 seconds. Similarly, the speedup achieved by the other GPUs met expectations. Algorithm speedup on Tesla K80 GPU was 3.51 times faster than on CPU. The total execution time was reduced by 71.21%, from 3950.44 seconds to 1125.37 seconds. The Tesla P4 was 7.16 times faster than the CPU.

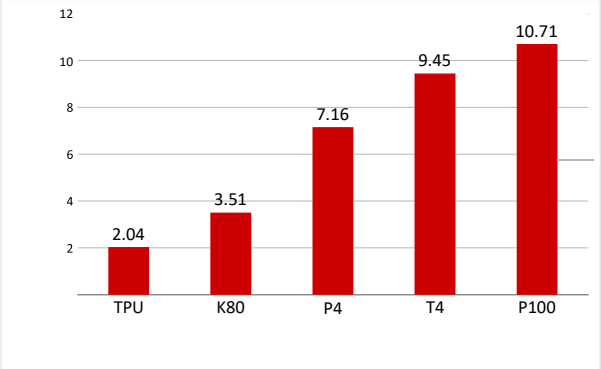

Fig. 12. Speedup architectures of TPU and GPU compared to CPU

The run-time required to complete the task has been shorten by 86.04%, or from 3950.44 seconds to 551.60 seconds. Finally, Tesla T4 demonstrated a 9.45 percent improvement over the CPU. Reducing the initialization time from 3950.44 seconds to 417.98 seconds and increasing the efficiency by 89.42%. Using a TPU instead of a CPU resulted in a 2.04x speedup, a 50.01% reduction in execution time (from 3950.44 to 1935.49 ms).

GPUs were compared to one another in order to evaluate the improvement. The execution time was lowered from 3951.44 seconds to 551.60 seconds, or an increase of 86.04%, when comparing the Tesla P4 with the Tesla K80. T4 Tesla was measured against the P4 and the K80 in these studies.

Gaining 24.23 percent, the Tesla T4 outperformed the P4 by a factor of 1.31, reducing the runtime from 551.60 to 417.98 seconds. T4 earned 2.69 times as much as K80, reducing the runtime to 417.98 seconds (from 3950.44 seconds) with an efficiency gain of 89.40%. In the end, a comparison was made between the Tesla P100 and the other GPUs, and the results were analysed. Gains of 11.77 percent were achieved, or 1.13 times faster execution compared to the T4, with a time drop from 417.98 seconds to 368.81 seconds. In all, the time spent in conference with P4 was reduced from 551.60 seconds to 368.81 seconds, a 33.14 times improvement. Gains of 3.05x and 67.23 percent were achieved in head-to-head competition with K80, with execution time reduced from 1125.37 to 368.81 seconds.

A comparison of speedup was also done between graphics processing units (GPUs) and the Google Colab TPU. The algorithm's runtime was shortened in half from 1935.49 seconds to 1125.37 seconds (a speedup of 41.86%) when it was executed on the Tesla K80 GPU rather than the TPU. As can be seen in Figure 13, a comparison of speedup was also done between graphics processing units (GPUs) and the Google Colab TPU. The algorithm's runtime was reduced in half from 1935.49 seconds to 1125.37 seconds (a speedup of 41.86%) when it was executed on the Tesla K80 GPU rather than the TPU.

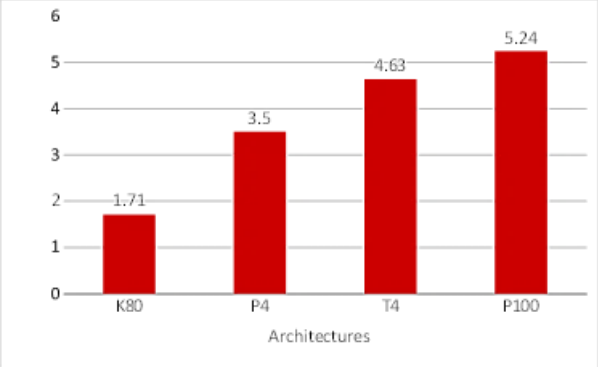

Fig. 13. GPU speedup architecture compared to TPU

Table 2 shows the comparison of runtime reduce and the speed gains when testing with different architectures of GPU, CPU and TPU.

Table 2
Comparison between GPU's, CPU and TPU

Comparison Factors	GPU'S				TPU	CPU
	Tesla K80	Tesla T4	Tesla P4	Tesla P100		
Reduce in Runtime (seconds)	1125.37 seconds	417.98 seconds	to 551.60	368.81 seconds	1935.49 ms	3950.44 seconds
Speed Gain (times, x)	3.51x	9.45x	7.16x	10.71x	2.04x	---

In the end, the Tesla P100 GPU proved to be 5.24 times more effective than the TPU, by reducing the execution time in half, from 1935.49 to 368.81 seconds. This represented an improvement of 80.95%. Table 3 describes the speedup comparison between TPU and GPU's by analysing the speed in detection.

Table 3
Speed-up comparison between all GPU's and TPU

Comparison Factors	GPU'S				TPU
	Tesla K80	Tesla T4	Tesla P4	Tesla P100	
Gain in Speed (%)	41.6%	78.41%	51.51%	80.95%	50.01%
Speed Gain (x)	1.71x	4.63x	3.50x	5.24x	2.04x

A total of 32 filters (num-features) were established for CNN's development, with 16 operators assigned to adjusting the RNA weights (batch size) throughout the course of 100 seasons of training. Over 15 seasons, a stop-metric to cease was also established (called EarlyStopping). The ELU (Exponential Linear Unit) activator was used in the convolution layer, with a 20% Dropout.

The tests were run with Google Colab's four video cards. Figure 14 shows a comparison between the accuracy levels and the execution on the CPU and TPU. Note that the RNA validation base was not used to derive the stated accuracy levels. For GPUs, the Tesla K80 had the highest accuracy, at 65.67 percent. Overall, the accuracy of the GPUs was slightly lower than that of the CPU and TPU, with the exception of the Tesla T4, which achieved an accuracy of 65.39. The Tesla P4 and P100 both attained an accuracy of 64.05, while the TPU scored 65.25 and the CPU earned 65.17.

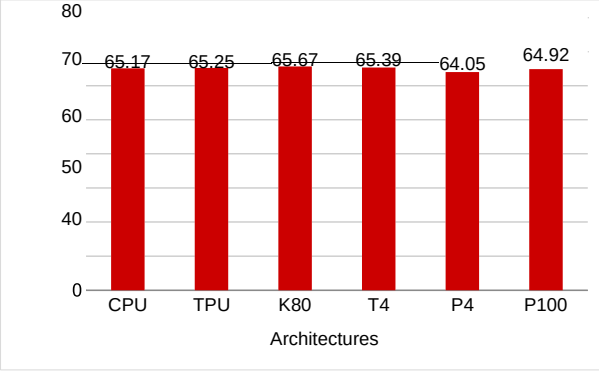

Fig. 14. Accuracy result based on each architecture

The result of the accuracy obtained in each architecture of GPU's, CPU and TPU presented in Table 4.

Table 4
Accuracy comparison between GPU's, CPU and TPU

Comparison Factors	GPU'S				TPU	CPU
	Tesla K80	Tesla T4	Tesla P4	Tesla P100		
Accuracy (Percentage)	65.67%	65.39%.	64.05%	64.05%	65.25%	65.17%

3.2 Performance Evaluation of Convolutional Neural Networks using TPUs, GPUs and CPUs with Previous Research

Convolutional neural networks (CNNs) have become a popular and effective tool for a wide range of machine learning tasks, from image processing. Training and evaluating convolutional neural networks (CNNs) can demand substantial computational resources, necessitating a considerable amount of processing capacity and duration [22]. In order to overcome this difficulty, researchers have examined different hardware platforms to expedite the training and prediction of CNNs. These

include graphics processing units (GPUs), central processing units (CPUs), and tensor processing units (TPUs).

Our objective in this study is to present a comprehensive and current assessment of CNN's efficacy employing GPUs, TPUs, and CPUs. Specifically, we will compare the accuracy, training time, inference time, and power consumption of CNNs trained and tested on each platform and discuss the relative strengths and weaknesses of each platform for different types of CNN models and tasks. Additionally, we will review and cite previous studies that have compared the performance of CNNs on TPUs, GPUs, and CPUs, and highlight any unique contributions or innovations in our methodology or experimental design. Overall, our study aims to provide insights to optimize the performance of CNNs using different hardware platforms.

Raj *et al.,* [23] compared the performance of CNNs on GPU, TPU, and CPU for FER using the FER2013 dataset. The models were evaluated based on their accuracy and the time taken for training. The researcher found that the GPU outperformed both the TPU and CPU in terms of accuracy, with an accuracy of 70.25%. The TPU had an accuracy of 68.86%, while the CPU had an accuracy of 65.63%. The training time was fastest on the TPU, with a time of 270 seconds, followed by the GPU with a time of 360 seconds, and the CPU with a time of 2700 seconds. Overall, researchers found that the GPU outperformed both the TPU and CPU in terms of accuracy, while the TPU had the fastest training time. However, it is important to note that the researcher did not evaluate other factors such as energy consumption or inference time.

The performance of CNNs for FER using the FER2013 dataset was evaluated by the researchers Sharma *et al.,* on CPU, GPU, and TPU [24]. Accuracy, training time, and inference time were used to evaluate the performance of the models. According to the results obtained, the GPU achieved the highest accuracy of 63.32%, surpassing both the TPU and CPU. The TPU exhibited an accuracy of 60.52%, whereas the CPU yielded an accuracy of 57.56%. The GPU exhibited the shortest training time of 200 seconds, followed by the TPU with 208 seconds and the CPU with 1440 seconds. In terms of inference time, the GPU demonstrated the quickest performance, with 0.02 seconds per image. The TPU followed with 0.03 seconds per image, while the CPU had the slowest performance with 0.10 seconds per image. Overall, the researcher found that the GPU outperformed both the TPU and CPU in terms of accuracy, training time, and inference time for CNNs on the FER2013 dataset.

The researcher Ravikumar *et al.,* [25] conducted a comparative analysis of CNN's performance on CPU, GPU, and TPU for FER using the FER2013 dataset, measuring accuracy, training time, and inference time. According to their findings, the GPU exhibited the highest accuracy of 70.13%, outperforming both the TPU and CPU. The TPU had an accuracy of 69.95%, while the CPU had an accuracy of 63.63%. The training time was fastest on the TPU, with a time of 14 seconds, followed by the GPU with a time of 27 seconds, and the CPU with a time of 255 seconds. The inference time was fastest on the TPU, with a time of 0.011 seconds per image, followed by the GPU with a time of 0.029 seconds per image, and the CPU with a time of 0.250 seconds per image. In summary, the study discovered that the TPU outperformed the GPU and CPU in terms of training time and inference time, while the GPU achieved the highest accuracy. Based on the comparison of different research papers, it can be concluded that the performance of CNNs on CPU, GPU, and TPU for facial expression recognition (FER) using the FER2013 dataset varies depending on the specific model architecture, dataset pre-processing, and hardware specifications.

In general, the GPU outperformed both the TPU and CPU in terms of accuracy and training time in most studies. However, the TPU had the highest performance in terms of training time and inference time in some studies. It is important to note that the choice of hardware platform for training and inference should depend on the specific needs and constraints of the application, including the size of the dataset, the complexity of the model, the available hardware resources, and

the desired performance metrics. Overall, the performance of CNNs on CPU, GPU, and TPU for FER using the FER2013 dataset has been extensively studied in the literature, and these studies can provide valuable insights for researchers and practitioners in the field of computer vision and deep learning. By doing so, they may be able to achieve faster and more accurate results, which could lead to better performance in real-world applications.

3.3 Test Performance with Analysed Platforms

In order to run the tests, an image must be loaded, and facial recognition must take place before any conclusions can be drawn about the emotions shown there. The OpenCV package was utilised for the recognition since it contains a pre-trained model for the desired characteristic, in this instance, facial features. To be more specific, we used the file named haarcascade frontalface default.xml. Haar Cascades uses the Adaboost learning algorithm that choses a few numbers of significant features from a large set-in order to provide an effective result of classifiers. After the picture and classifier have been setup, face recognition may be performed. The original image must be converted to grayscale because the utilized classifier only accepts those 48x48 grayscale images. After converting the colour image to grayscale, facial features may be identified, and the associated emotions can be predicted. Although tests of CNNs were run on all three architectures namely CPU, GPU, and TPU, the result obtained by the architecture that achieved the best precision in the training phase is illustrated in figures below. In this case, the tests were run on the Tesla K80 GPU since it able to reduce execution runtime and performed the classification with highest accuracy during classification.

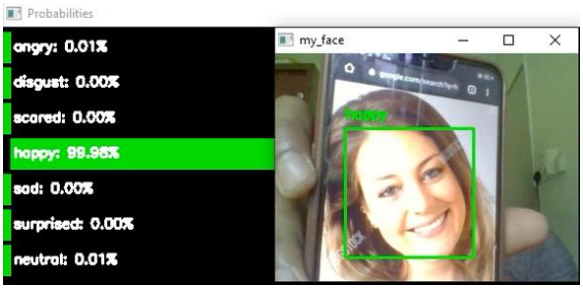

Fig. 15. Happy expression

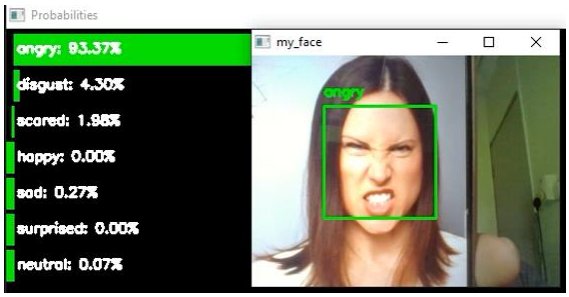

Fig. 16. Angry expression

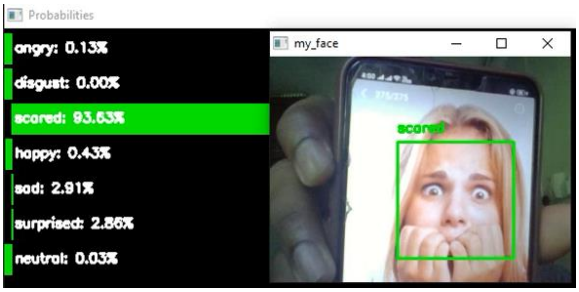

Fig. 17. Scared expression

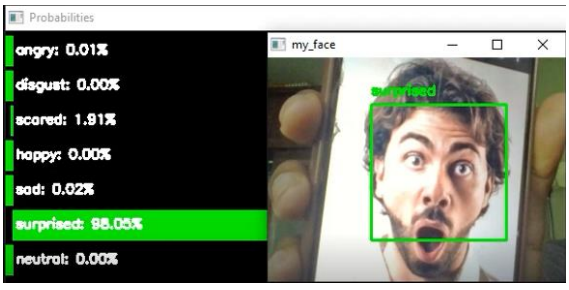

Fig. 18. Surprised expression

Fig. 19. Sad expression

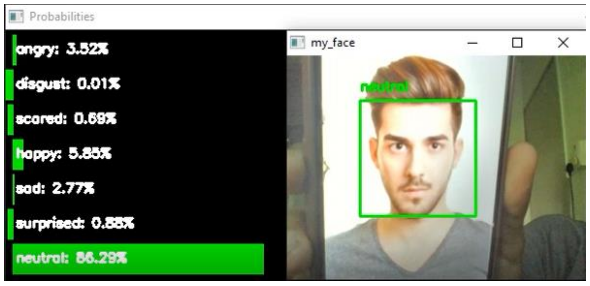

Fig. 20. Neutral expression

While CNN performed poorly in identifying both scared and surprise across all test designs, other systems performed far better. The fact that scared and surprise may be easily mistaken for one another is due to the fact that they share comparable facial traits such as brows, eyes and lips. They share characteristics such as an eyebrow raised without drawn together and open jaw as shown in Figure 21.

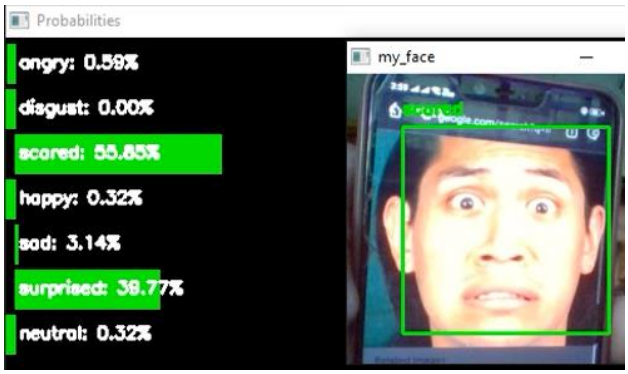

Fig. 21. Mismatch results for scared and surprised expression

The purpose of this test is to evaluate the system recognises trained facial expressions effectively. Since the system successfully recognised the user's face 99.24% of the time, it can be said that it performs well. Images are examined against the facial expression classification system after an appropriate training.

Based on the test findings for 7 various expressions, including sadness, surprise, anger, neutral, scared, disgust, and happiness, the experiment was run ten times for each expression. All tests were successfully recognised by the system. While the findings of the expressions of surprise and scared both had errors twice, and the results of the expression of anger had an error once.

Ten analyses for each expression were performed as part of the test, and the findings are displayed in Table 5 as the confusion matrix's results. The table will display which expressions are straightforward to predict, predict occasionally, and difficult to predict. According to an analysis of this table, two out of every ten attempts to predict facial expressions are made using the traits of the expression that provide the incorrect results.

Table 5
Confusion Matrix

		Number of Predictions						
		Neutral	Happy	Sad	Angry	Disgust	Surprise	Scared
Expressions	Neutral	10	0	0	0	0	0	0
	Happy	0	10	0	0	0	0	0
	Sad	0	0	10	0	0	0	0
	Angry	0	0	0	9	1	0	0
	Disgust	0	0	0	3	6	0	0
	Surprise	0	2	0	0	0	8	0
	Scared	0	0	2	0	0	0	8

4. Conclusions

Using a convolutional neural network, GPU significantly outperforms previous generations in facial emotion recognition. Its benefits over central processing units are enormous. Stream processor has been shown to operate well with convolution neural networks in experiments. This work demonstrates that GPUs are just as quick and efficient than CPUs and TPUs when it comes to deep learning. The task here involved measuring and visualizing their performance in a variety of ways to determine how well they performed. This finding proved that Deep CNNs can learn face features and perform adequately in emotion recognition. The term "deep learning" is used to describe a set of machine learning algorithms that can automatically learn a deep architecture's hierarchical representation for classification purposes. Access to data and processing power are necessities for deep learning. Not only does speed of training and scaling affect how effective a deep learning solution is, but accuracy is also important. This research provides empirical evidence that deep learning computation speeds may be significantly improved on GPU-enabled systems. The new method improved RNA accuracy by as much as 65.67%. When compared to running the programme on a traditional central processing unit (CPU), the version optimized for the Tesla P100 GPU is up to 10.71 times faster.

Acknowledgement

This research was funded by a grant from Ministry of Higher Education Malaysia and Universiti Malaysia Pahang (Grant number FRGS/1/2022/TK10/UMP/02/67) and PGRS 230344.

References

[1] Leão, Leonardo Panta, Jonas Santos Bezerra, Leonardo Nogueira Matos, and Maria Augusta Silveira Netto Nunes. "Detecção de expressões faciais: uma abordagem baseada em análise do fluxo óptico." *Revista Geintec-Gestao Inovacao E Tecnologias* 2, no. 5 (2012): 472-489. https://doi.org/10.7198/S2237-0722201200050005

[2] Sajjad, Muhammad, Fath U. Min Ullah, Mohib Ullah, Georgia Christodoulou, Faouzi Alaya Cheikh, Mohammad Hijji, Khan Muhammad, and Joel JPC Rodrigues. "A comprehensive survey on deep facial expression recognition: challenges, applications, and future guidelines." *Alexandria Engineering Journal* 68 (2023): 817-840. https://doi.org/10.1016/j.aej.2023.01.017

[3] Erbao, Peng, and Zhang Guotong. "Image processing technology research of on-line thread processing." *Energy Procedia* 17 (2012): 1408-1415. https://doi.org/10.1016/j.egypro.2012.02.260

[4] Mostafa, Kareem, and Tarek Hegazy. "Review of image-based analysis and applications in construction." *Automation in Construction* 122 (2021): 103516. https://doi.org/10.1016/j.autcon.2020.103516

[5] Dores, Artemisa R., Fernando Barbosa, Cristina Queirós, Irene P. Carvalho, and Mark D. Griffiths. "Recognizing emotions through facial expressions: A largescale experimental study." *International journal of environmental research and public health* 17, no. 20 (2020): 7420. https://doi.org/10.3390/ijerph17207420

[6] Zhi, Ruicong, Mengyi Liu, and Dezheng Zhang. "Facial Representation for Automatic Facial Action Unit Analysis System." In *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pp. 1368-1372. IEEE, 2019. https://doi.org/10.1109/ITAIC.2019.8785870

[7] Jack, Rachael E., Oliver GB Garrod, Hui Yu, Roberto Caldara, and Philippe G. Schyns. "Facial expressions of emotion are not culturally universal." *Proceedings of the National Academy of Sciences* 109, no. 19 (2012): 7241-7244. https://doi.org/10.1073/pnas.1200155109

[8] Arora, Himanshu, Manish Kumar, Tawsai Rasool, and Pooja Panchal. "Facial and Emotional Identification using Artificial Intelligence." In *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1025-1030. IEEE, 2022. https://doi.org/10.1109/ICOEI53556.2022.9776862

[9] Abdullah, Muhammad, Mobeen Ahmad, and Dongil Han. "Hierarchical attention approach in multimodal emotion recognition for human robot interaction." In *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pp. 1-4. IEEE, 2021. https://doi.org/10.1109/ITC-CSCC52171.2021.9501446

[10] Topitzer, Maya, Yueming Kou, Robert Kasumba, and Philip Kreniske. "How Differing Audiences Were Associated with User Emotional Expression on a Well-Being App." *Human Behavior and Emerging Technologies* 2022 (2022). https://doi.org/10.1155/2022/4453980

[11] Wagner, Hugh L. "The spontaneous facial expression of differential positive and negative emotions." *Motivation and Emotion* 14 (1990): 27-43. https://doi.org/10.1007/BF00995547

[12] Mo, Young Jong, Joongheon Kim, Jong-Kook Kim, Aziz Mohaisen, and Woojoo Lee. "Performance of deep learning computation with TensorFlow software library in GPU-capable multi-core computing platforms." In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 240-242. IEEE, 2017. https://doi.org/10.1109/ICUFN.2017.7993784

[13] Bartlett, Marian Stewart, Gwen Littlewort, Ian Fasel, and Javier R. Movellan. "Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction." In *2003 Conference on computer vision and pattern recognition workshop*, vol. 5, pp. 53-53. IEEE, 2003. https://doi.org/10.1109/CVPRW.2003.10057

[14] Lucey, Patrick, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression." In *2010 ieee computer society conference on computer vision and pattern recognition-workshops*, pp. 94-101. IEEE, 2010. https://doi.org/10.1109/CVPRW.2010.5543262

[15] Tang, Hao, and Thomas S. Huang. "3D facial expression recognition based on properties of line segments connecting facial feature points." In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1-6. IEEE, 2008. https://doi.org/10.1109/AFGR.2008.4813304

[16] Song, Kai-Tai, and Yi-Wen Chen. "A design for integrated face and facial expression recognition." In *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4306-4311. IEEE, 2011. https://doi.org/10.1109/IECON.2011.6120016

[17] Ekman, Paul. "Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique." (1994): 268. https://doi.org/10.1037/0033-2909.115.2.268

[18] Rahim, Robbi, Ansari Saleh Ahmar, and Rahmat Hidayat, eds. "Proceedings of the Joint Workshop KO2PI and the 1st International Conference on Advance & Scientific Innovation." (2018).

[19] Zahara, Lutfiah, Purnawarman Musa, Eri Prasetyo Wibowo, Irwan Karim, and Saiful Bahri Musa. "The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based Raspberry Pi." In *2020 Fifth international conference on informatics and computing (ICIC)*, pp. 1-9. IEEE, 2020. https://doi.org/10.1109/ICIC50835.2020.9288560

[20] Bennett, Casey C., and Selma Šabanović. "Deriving minimal features for human-like facial expressions in robotic faces." *International Journal of Social Robotics* 6, no. 3 (2014): 367-381. https://doi.org/10.1007/s12369-014-0237-z

[21] Wegrzyn, Martin, Maria Vogt, Berna Kireclioglu, Julia Schneider, and Johanna Kissler. "Mapping the emotional face. How individual face parts contribute to successful emotion recognition." *PloS one* 12, no. 5 (2017): e0177239. https://doi.org/10.1371/journal.pone.0177239

[22] Alzubaidi, Laith, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions." *Journal of big Data* 8 (2021): 1-74. https://doi.org/10.1186/s40537-021-00444-8

[23] Raj, P., and Ch Sekhar. "Comparative Study on CPU GPU and TPU." *Int. Journal of Computer Science and Information Technology for Education* 5 (2020): 31-38. https://doi.org/10.21742/IJCSITE.2020.5.1.04

[24] Sharma, Vijeta, Gaurav Kumar Gupta, and Manjari Gupta. "Performance Benchmarking of GPU and TPU on Google Colaboratory for Convolutional Neural Network." In *Applications of Artificial Intelligence in Engineering: Proceedings of First Global Conference on Artificial Intelligence and Applications (GCAIA 2020)*, pp. 639-646. Springer Singapore, 2021. https://doi.org/10.1007/978-981-33-4604-8_49

[25] Ravikumar, Aswathy, Harini Sriraman, P. Maruthi Sai Saketh, Saddikuti Lokesh, and Abhiram Karanam. "Effect of neural network structure in accelerating performance and accuracy of a convolutional neural network with GPU/TPU for image analytics." *PeerJ Computer Science* 8 (2022): e909. https://doi.org/10.7717/peerj-cs.909