ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 14 (2024) No. 6, pp. 43-60
DOI:10.33168/JSMS.2024.0604

Micro-Expression Recognition with Pre-trained Neural Network Models

Ho Jia Jun, Khoh Wee How, Pang Ying Han, Yap Hui Yen

Faculty of Information Science & Technology (FIST), Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, 75450, Melaka, Malaysia

Abstract. Micro-expression (ME) is a form of reflexive behaviour and indirect communication in which an individual expresses their emotions via facial muscle movements. Due to the nature of ME that only appears for a fraction of a second, and even a trained individual will have a hard time detecting and recognizing it. In order to avoid human error and achieve better results, an automatic ME recognition system is introduced. In this work, a transfer learning approach is utilized to recognize the static facial micro-expression images. Six pre-trained convolutional neural network (CNN) models, including the AlexNet, SqueezeNet, GoogleNet, EfficientNet-b0, ResNet 50, and MobileNet-v2 are employed and evaluated on the improved version of the Chinese Academy of Sciences Micro-expression (CASME II) database. The pre-trained CNN models are compared to each other by their accuracies. We conducted the experiments under three different settings, in Setting 1 the parameter of entire learnable layers are unaltered, setting 2 is by freezing the first 20% of the learnable layers and Setting 3 the first 50% of the learnable layers are frozen. AlexNet obtained higher accuracy of 99.84% in Setting 3. The freezing learnable layer approach is able to improve a pre-trained model's accuracy and accelerate training time by not altering the parameters of the frozen layers. This research not only benefits the psychology field, but it also benefits the marketing field, security purposes, and other applicable fields.

Keywords: Micro-Expression, convolutional neural network, pre-trained, transfer learning, AlexNet, SqueezeNet, GoogleNet, ResNet50, EfficientNet-b0, MobileNet-v2

1. Introduction

A micro-expression (ME) is an unintentional, momentary facial expression. It is a non-verbal communication that occurs through the facial muscle's movement underneath the facial skin. ME was first discovered and spotted by researchers while trying to search for non-verbal cues in a recorded interview between a doctor and a patient (CASME Database, n.d.). It tends to appear unconsciously while a person is trying to mask their genuine feelings. Without conscious awareness, it shows a person's genuine feelings. Unlike the macro-expression, MEs are hard to notice by human eyes and it is often missed due to that it only appears in a fraction of seconds with subtle facial movement. The Facial Action Coding (FACS) System was commonly adopted to study and taxonomize the ME by examining facial muscles movements that appear on the face (Chernykh & Prikhodko, 2017). The FACS can train a person to recognize and classify ME manually. However, considering humans make mistakes, accuracy is considerably low. There are still some concerns regards to the accuracy of recognizing MEs with the naked eyes. On the other side, a highly precise categorization is essential for security and mental health diagnosis in order to avert any negative incidence. Therefore, an automatic computer-based system is highly in demand as it has fewer errors and provides better accuracy than trained individuals.

Before the spontaneous database was publicly available, the earlier ME recognition studies were conducted using the posed ME database (Fan et al., 2016) (Gan et al., 2017). Posed databases are databases which contain a series of voluntary expressions that obtained from participants, which is not the genuine feeling of the participants. Spontaneous databases are databases which contain a series of involuntary expressions. It appears without the participant's consciousness and reveals their true feeling. The expressions in the posed database are not genuine and it is irrelevant due to not being elicited spontaneously and lacking the characteristics of a genuine expression. One of the challenges of ME recognition that has yet to be solved is the lack of samples. The research of ME in the field of computer vision is still relatively new, which causes the lack of samples compared to other domains such as palmprint recognition, which has been researched longer and has sufficient samples. In this study, a transfer learning strategy is employed for ME recognition to get over the issue of not having enough samples. The major objective of this study is to categorize emotions such as disgust, fear, happiness, repression, sadness, and surprise using a variety of pre-trained CNN models. The experiments were conducted using six pre-trained CNN models without any preprocessing on the ME samples. The models' performance was assessed using the CASME II (CASME Database, n.d.) micro-expression database. Then, the performance of the pre-trained CNN models will be compared to each other and also compare them with other authors' models.

The rest of the paper is organized as follows. Section 3 describes the feature extraction method and preprocessing method of the dataset, and the models used in this research. Section 4 describes the experimental setup and the dataset used in this research. Section 5 describes the settings used and also discussed the model's performance in this work. Section 6 concludes the findings of this work and suggests some ideas for the future work.

2. Literature Review

Madupu et al. (Madupu et al., 2020) introduced an automated facial emotion classification system based on the Convolution Neural Network (CNN) and the extracted features of the Speeded Up Robust Features (SURF). The remaining noise from the photos was removed using the high Boost filtering approach. The characteristics from the image were then extracted using SURF features extraction in this study. The extracted images that included different expressions were fed into CNN for training. In order to train and test the classifiers, Back Propagation Neural Network (BPNN) and CNN were utilized. The performances were reported to achieve 91% and 88% accuracies for both CNN and BPNN, respectively. Jain et al. (D. K. Jain et al., 2019) suggested a Deep Convolution

Neural Networks (DNNs) on the facial emotion recognition. In the work, two databases, named Japanese Female Facial Expression (JAFFE) and Extended Cohn-Kanade (CK+) were adopted to train the model. The proposed DNN managed to achieve as high as 95% accuracy, and it outperformed the other six different models that proposed by Lopes et al. (Lopes et al., 2017), Khorrami et al. (Khorrami et al., 2016), Jain et al. (N. Jain et al., 2018), Krestinskaya & James (Krestinskaya & James, 2017), Chernykh et al. (Chernykh & Prikhodko, 2017), and Zhang et al. (Zhang et al., 2019).

Nasri et al.'s (Nasri et al., 2020) proposed a facial emotion recognition system by adopting Xception CNN paired with a K-fold cross-validation technique for the static expression images. The Xception CNN model was trained in two different methods, one from scratch and the other b acial emotion recognition system was developed by combining the Xception CNN with the K-fold cross-validation method for images of static expressions using the fine-tuning method. Then, Empathic, AffectNet and CK+ databases were used to test the proposed model. The fine-tuning technique achieved 98.2% accuracy on the CK+ database. The authors also experimented on more complex datasets where the AffectNet and Empathic datasets were combined, the proposed model could also achieve a promising result of 91.2% accuracy. Lasri et al. (Lasri et al., 2019) proposed a study to identify students' moods based on their facial expressions. In this work, the model was trained, validated, and tested using data from the FER 2013 facial expression database. The database consists of 32298 gray-scaled facial images which contain seven different facial expressions such as angry, disgust, fear, happy, sad, surprise and neutral. In addition, all the images were cropped and normalized into a 48×48 pixels resolution. For categorizing those seven emotion classes, a CNN model comprising four convolutional layers, four pooling layers to extract features, two fully connected layers, and a SoftMax layer was proposed. The proposed model achieved a 70% accuracy. Pranav et al. (Pranav et al., 2020) employed a 2-dimensional (2D) CNN to identify the facial emotion. A self-collected facial emotion database which contains five emotions including angry, happy, neutral, sad, and surprise were used. The proposed 2D-CNN model achieved an accuracy of 78.04% on its self-collected database. With the use of the Hybrid Convolution-Recurrent Neural Network (CNN-RNN) technology, Jain et al. (N. Jain et al., 2018) developed recognizing facial emotions. MMI Facial Expression Database and JAFFE were employed. Firstly, a CNN model was used to extract the features of the facial images. RNN was then employed to classify facial expressions. In this work, a few CNN architectures were proposed. The first architecture includes ReLU layer, and it reached an overall accuracy of 94.46% while the second architecture involves 150 hidden units and achieved 94.21% accuracy. The third architecture which is the hybrid of CNN-RNN with six hidden layers. It slightly surpassed the formers architectures which obtained the accuracy of 94.91%. The proposed model outperforms the other four models proposed by Khorrami et al. (Khorrami et al., 2016), Zhang et al. (Zhang et al., 2019), Fan et al. (Fan et al., 2016) and Chernykh et al. (Chernykh & Prikhodko, 2017).

A Venturi Architecture for CNN was proposed by Verma et al. (Verma et al., 2019). The Venturi architecture contains 6 hidden layers and one output layer to classify seven facial emotions. Due to the structure of the hidden layers, which resembles a Venturi tube, this architecture received its name Both the training and testing of the models employed the Karolinska Directed Emotional Faces (KDEF) dataset. Proposed Venturi Architecture CNN was benchmarked with the Rectangular and Modified Triangular models in which the Rectangular architecture contains six hidden layers, and it was named as the number of nodes in each hidden layer is equal, which looks like a rectangle. On the other hand, the Modified triangular architecture proposed by Haque et al. (Haque et al., 2018). It is a CNN architecture that contains 7 hidden layers which were built up by ReLU. It contains 256 nodes in the first hidden layer and 512 nodes in the second layer. From the third to the seventh layer, there were fewer triangle-shaped nodes, giving the entire design the appearance of a modified triangle. As for the performance, the best result was obtained by Venturi Architecture, whose accuracy was 86.78%, while the Rectangular model and Modified Triangular model earned an accuracy of 79.61%

and 82.70%, respectively. Gan et al. (Gan et al., 2017) utilized the Latent Regression Bayesian Networks (LRBN) to explicit the model spatial patterns embedded in posed and spontaneous expressions, respectively. In this work, SPOS database and NVIE database were used to train the LRBN model, separately. The stochastic approximation procedure (SAP) framework was used to learn the LRBN during the training process. All the samples from the databases were classified in binary classes, either a posed expression or spontaneous expression. As a result reported, it obtained higher accuracy of 98.94% on NVIE database than SPOS database with accuracy of 76.07%.

Zhi et al. (Zhi et al., 2019) also a study using a 3D CNN and transfer learning to recognize facial micro-expression. In this work, the proposed 3D-CNN models were pre-trained by using Oulu-CASIA database in a supervised learning condition and evaluated by using CASME II and SMIC. The images in the databases were pre-processed by using a 3D spline interpolation to normalize the length of the input facial image sequences. Data augmentation was also carried out by flipping every image horizontally to increase the image by seven times. The methods were compared with Local Binary Patterns with three orthogonal planes (LBP-TOP) paired with extreme learning machine (ELMtg) and LBP-TOP paired with Nearest Neighbor in CASME II and SMIC databases, respectively. By incorporating the transfer learning technique, the 3D-CNN model achieved 97.6% accuracy and achieved 97.4% accuracy in the SMIC database in a five-folded cross-validation. Wang et al. (Wang et al., 2018) adopted Transferring Long-term Convolutional Neural Network (TLCNN) model to recognize micro-expression (ME) with a tiny sample size. The transferring could be done in a two-step method. The knowledge is first transferred from expression data. In order for ME to obtain the temporal sequence knowledge, a single frame of micro-expression video clips is transmitted and fed into the Long Short-Term Memory (LSTM). The pre-trained model was developed implementing the Radboud Faces Database, MMI Facial Expression Database, Taiwanese Facial Expression Image Database, and Karolinska Directed Emotional Faces. The proposed TLCNN model was evaluated against 3D-CNN, Directional Mean Optical Flow Feature (MDMO), LBP-TOP, STLBP-IP, Spatiotemporal Completed Local Quantization Patterns (STCLQP), and Facial Dynamics Map (FDM). All the models were evaluated by using SMIC, CASME and CASME II databases. Among the techniques, the TLCNN model outperformed others by achieving a mean accuracy of 71.2% in a video clip of 32 frames and 69.1% in a video clip of 64 frames. Sun et al. (Sun et al., 2020) adopted Knowledge Distillation in recognizing a dynamic Micro-Expression. The author proposed a residual network with a multi-task, multi-label network. The deep pre-trained teacher neural network was composed of the two final fully connected layers. The deep teacher neural network was pre-trained using the FERA2017 dataset, and its distilled knowledge was then transferred to and utilized to direct the training of the shallow student neural network. To get the final results, SVM was applied as the classifier. SAMM, SMIC2, CASME, and CASME II databases were employed to assess the model. On the SMIC2, CASME, CASME II, and SAMM databases, respectively, average accuracy was found to be 76.1%, 81.8%, 72.6%, and 86.7%.

3. Methodology

3.1. Convolutional Neural Network (CNN) Model

The convolutional neural network (CNN), which can evaluate input like pictures or numerical data, is an example of a neural network that is used to evaluate input. CNN does not require much preprocessing. The architecture design of CNN is inspired by biological neural networks. The three primary types of layers that comprise CNN are dense layer, pooling layer, and fully connected layer (FC). The CNN model's structure is shown in Figure 1. The following sections provide an overview of each layer:

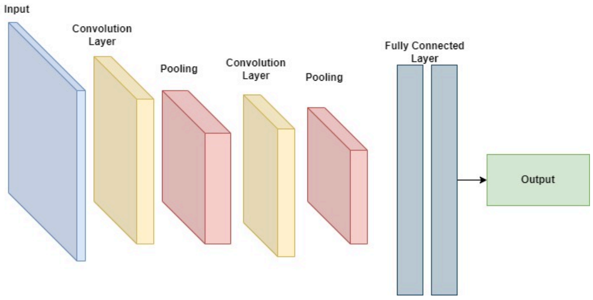


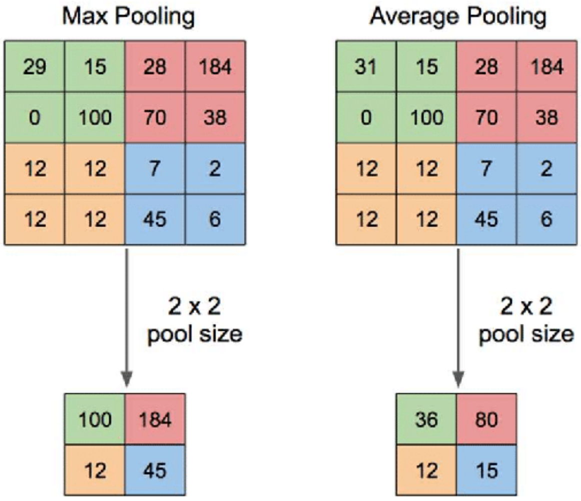
Fig. 1: Structural components of the CNN model

3.1.1. Convolution Layer

A convolutional layer, which is where most of the processing is done, and it is a crucial part of CNN. Input data, a filter, and a feature map are among the things it requires. The input data is processed using convolutional techniques to extract important characteristics and capture spatial correlations. In the convolution procedure, kernel is slid over the input data. The outcome of convolving the filter with a corresponding local area of the input is then denoted by every segment of the feature map, which is formed by applying the filter to specific local sections of the input.

3.1.2. Pooling Layer

A pooling layer is typically included in the construction of CNN used for deep learning tasks. Its objective is to maintain the most important features while shrinking the spatial dimensions of the input tensor. There are two types of pooling layers: Average pooling and Max pooling. The types of pooling layers are illustrated in Figure 2.



Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

100	184
12	45

Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

36	80
12	15

Fig. 2: Illustration of Average pooling and Max-pooling (Yani et al., 2019)

3.1.3. Fully Connected Layer (FC Layer)

Occasionally, a thick layer in CNN is used to refer to a FC layer. It usually appears at the end of a neural network architecture. Each node in the FC layer's output layer has a direct connection to a node in the pooling layer above. Then, the output will be sent to a corresponding layer for image classification.

3.2. Transfer Learning

A previously trained convolutional neural network (CNN) will be utilized as the starting point on the new job is employed in the machine learning methodology known as transfer learning. By using data from a previously trained model for a new job, this technique avoids the requirement of training a CNN model from ground up. It cuts down on the time and resources required for training a model from the ground up, which involves a large amount of data in order to reach maximum performance. Up to date, several popular CNN models, for instance, AlexNet, SqueezeNet, GoogleNet, ResNet, etc are available and have been pre-trained with the large-scale databases, e.g., ImageNet, and the trained parameters could be transferred to other target domains without retraining the model from scratch, to achieve computational efficiency. The concept of transfer learning is shown in Figure 3. A huge number of input photos from a source domain, such as ImageNet, are used to train Network A completely from scratch in the beginning. After a network has been thoroughly trained, the parameters (knowledge) from Network A may be entirely or partially transferred to Network B (also known as a pre-trained CNN model) to address our target domain problem without the use of massive input photos or expensive computing resources. Usually, the last fully connected and classification layers are to be replaced to suit the number of classes of our problem domain.

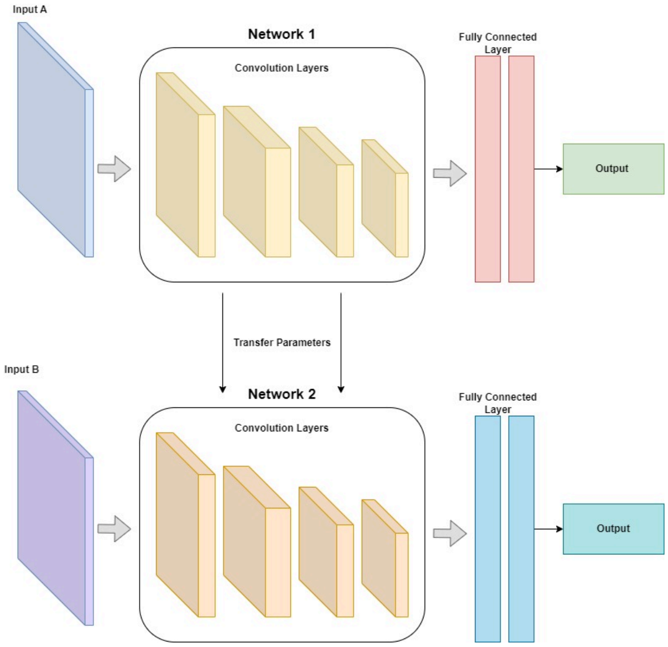


Fig. 3: Concept of Transfer Learning

3.3. Proposed Models

In this research, the pre-trained CNN models that used Chinese Academy of Sciences Micro-expression (CASME II) database in this research work include AlexNet, SqueezeNet, GoogleNet, EfficientNet-b0, ResNet 50 and MobileNet-v2. Besides, all the pre-trained models will be utilized for transfer learning. These pre-trained models are modified at the output layers to classify seven facial ME classes, including Disgust, Fear, Happiness, Repression, Sadness, Surprise and Others.

3.3.1. AlexNet

The entire AlexNet design consists of eight convolutional layers. There are five convolutional layers with max-pooling and three fully connected layers. Every convolutional layer has different kernel sizes and some filters. This input layer is fitted with the image with dimensions of 227×227×3.

Following the first convolutional layer, which has a cross channel normalization layer and a Max-pooling layer, is the first convolutional layer, which has a kernel size of 11×11, 96 filters, and strides of 4. This convolution layer makes use of the ReLU activation layer. This layer's output feature map is 27×27 pixels. The 128 filters in the second convolution layer each have a 5×5 kernel and a stride of 1. It has an output feature map with 13×13. The third convolution layer comes next, which has a 3×3 kernel size, 384 filters, and a stride of 1 with an output feature map of 13×13. A 3×3 kernel with 192 filters makes up the fourth layer, followed by a 13×13 output feature map. The fifth layer has a 128-filter construction, a 3-layer kernel, and a 13-layer feature map as its output. Between the fifth convolution layer and the first fully connected layer is a 3×3 Max-pooling layer with a 6×6 output feature map. 4096 neurons with 50% dropout and ReLU activation are coupled to the first fully connected layer (FC6) and second fully connected layer (FC7). The classifications are carried out in the eighth layer, which is the last fully connected layer using the Softmax activation function. The AlexNet architecture is depicted in Figure 4.

In order to fit the 7 classes of the CASME II dataset, the last fully connected layer is changed for a new fully connected layer with an output size of 7 during the implementation phase. The training settings employ the Stochastic Gradient Descent with Momentum (SGDM) function, and the learning rate is set at 0.0001. The batch size is set at 20 and the model is run for 50 epochs.

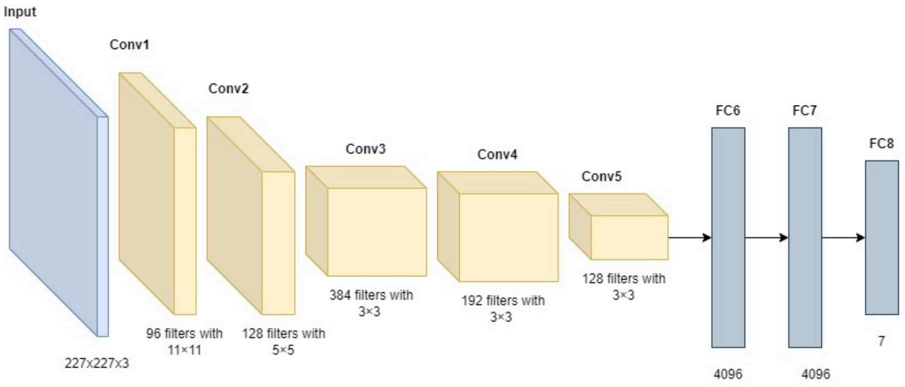


Fig. 4: Architecture of AlexNet

3.3.2. SqueezeNet

Figure 5 depicts the SqueezeNet architecture. SqueezeNet requires image input with 227×227×3 pixels. SqueezeNet has a convolutional layer, eight fire modules, and a final convolutional layer. The first layer of SqueezeNet is a convolution layer with 64 filters, each with a 3 by 3 kernel and a 2-stride. This was followed by the addition of a Max-pooling and a ReLU activation function. The second fire module is started with a convolution with 16 filters and 1 by 1 kernel and continues with a 3 by 3 kernel sizes with 64 filters and a 1 by 1 kernel size with 64 filters expansion convolution layers. Before moving on to the third fire module, the output of these first two layers will be sent to a depth concatenation function. For the third fire module, it begins with 16 filters with 1 by 1 kernel and followed by 64 filters with 1 by 1 kernel and 64 filters with 1 by 1 kernel. A 3 by 3 Max-pooling layer is included at the end of Fire Module 3. The fourth fire module starts out with 32 filters with a 1 by 1 kernel and expands to 128 filters with a 1 by 1 kernel and 128 filters with a 3 by 3 kernel. The fifth fire module starts with a 32 filter with a 1 by 1 kernel, then moves on to 128 filters with a 3 by 3 kernel, and 128 filters with a 1 by 1 kernel. Before moving on to the sixth fire module, the fifth fire module adds a Max-pooling layer. The sixth fire module has 48 filters with 1 by 1 kernel and split into 192 filters with 1 by 1 and 192 filters with 3 by 3 kernel. The seventh fire module also has the same structure as the sixth fire module. The eighth fire module has 64 filters with 1 by 1 kernels at the beginning, 256 filters with 3 by 3 kernels, and 256 filters with 1 by 1 kernels at the end. The ninth fire module starts off with 64 filters that are 1 by 1 kernel, then grows to 256 filters that are 1 by 1 kernel, then to 256 filters that are 3 by 3 kernel. A dropout function is applied before continuing to the tenth convolution layer. The tenth convolution layer has 1000 nodes with 1 by 1 kernel size. A ReLU activation function and a Max-pooling layer are then added after that. SoftMax function is applied at the end of SqueezeNet for classification.

In the implementation phase, the SGDM serves as the optimizer and the last convolution layer (Conv10) is swapped out with a new convolution layer with an output size of 7. With a batch size of 20, the model is trained for 50 epochs with a learning rate of 0.0001.

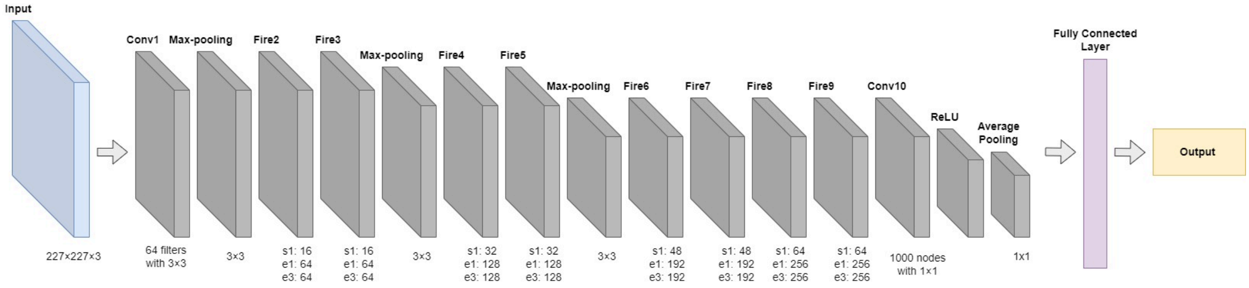


Fig. 5: SqueezeNet Architecture

3.3.3. GoogLeNet

The 22-layer deep CNN known as GoogleNet was created by Google and trained on the ImageNet database. GoogleNet requires input with 224×224×3 in dimensions. In GoogleNet, there are multiple types of filter sizes contained in the inception modules. The first convolution layer, which has 64 filters and a 3×3 kernel, comes first. Following the first convolution layer, there is a 3×3 Max-pooling layer and a ReLU activation function. The second convolution layer, which has 192 filters, and 33 kernel sizes, comes next. After the second convolution layer, a ReLU, cross channel normalization, and a 3×3 Max-pooling layer are added. After then, it moves on to the blocks of the inception module. Every inception block includes inception modules with 1×1, 3×3 and 5×5 kernel. The 3×3 Max-pooling is performed at the input and output of these inception module blocks to generate the final output. An average pooling layer, a dropout layer, a fully connected layer, and the SoftMax function are all included in the classification phase. The GoogleNet architecture is depicted in Figure 6.

A new fully connected layer with an output size of 7 is used to substitute the last fully connected layer in order to accommodate the CASME II dataset. This model is equipped with a SGDM optimizer and learning rate of 0.001. It is trained for 15 epochs with a batch size of 20.

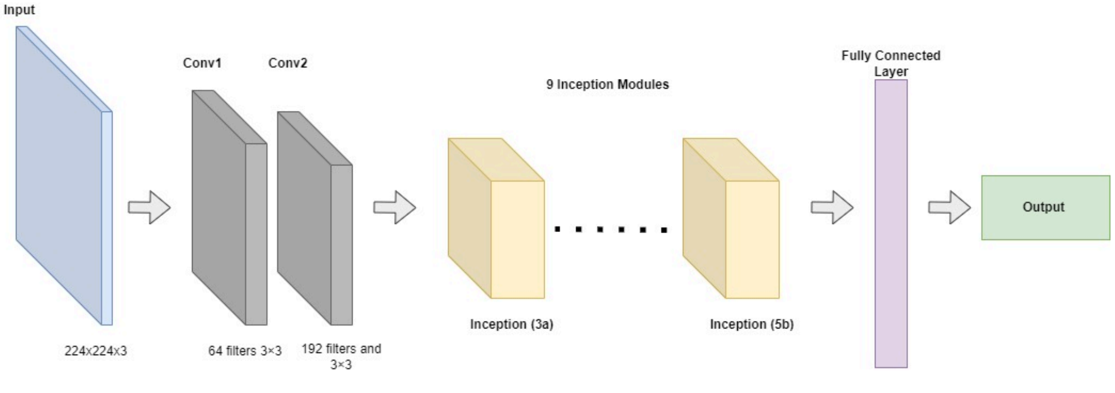


Fig. 6: GoogleNet Architecture

3.3.4. EfficientNet-B0

EfficientNet-B0 is the baseline network in the EfficientNet variant. The image input size for GoogleNet is 224×224×3 in dimensions. It aims to make deep learning on embedded and mobile devices more practical. EfficientNet-B0 has a total of 237 layers of convolution layers. A convolutional stem layer in the network initially processes the input image. A specific number of output channels are used to perform a 3×3 convolution. Each of the several blocks that make up EfficientNet-B0 has a set of operations. Following batch normalization and a non-linear activation function, each block has a depthwise separable convolution. Over the blocks, both the number of filters and the input feature maps' resolution steadily rise. In order to enhance the trade-offs between model size, accuracy, and computational cost, EfficientNet-b0 incorporates width, depth, and resolution scaling. The implementation of global average pooling on the feature maps results in a reduction of the spatial dimensions to a fixed size at the end of the network. The pooled features are flattened and fed following a SoftMax activation function and a fully connected layer for categorizing the classes. The architecture of EfficientNet-b0 is depicted in Figure 7.

The SGDM is utilized for this model during the implementation phase. The learning rate is set to 0.001. A new fully connected layer with an output size of 7 takes the place of the last fully connected layer. The model is run for 10 epochs and the batch size is set to 20.

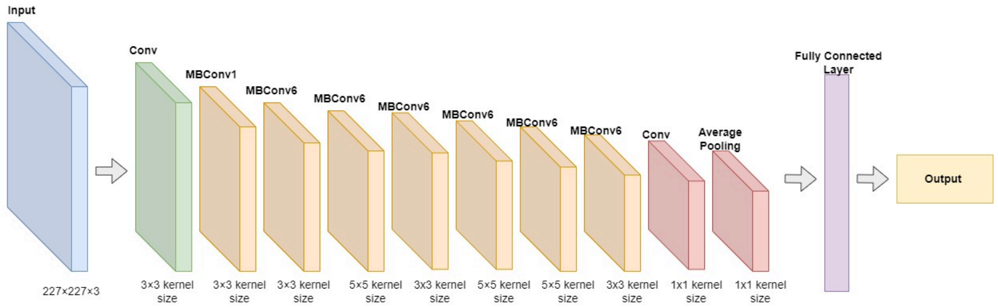


Fig. 7: EfficientNet-b0 Architecture

3.3.5. ResNet-50

ResNet-50 is a 50-layer deep convolution neural network which is based on ResNet-34 architecture, but they have a major difference in the building block. The building block of ResNet-50 was modified into a bottleneck design to overcome the time taken for the training of the layers. It has a SoftMax activation function for classification, a ReLU activation function, a fully connected layer, and five convolution layers that generate various feature maps. The 64 kernels with a 2-stride and the 3×3 max-pooling layer with a 2-stride are merged to form the massive 7×7 kernel convolution that makes up the input component of ResNet-50. The image input is required to be 224×224×3 in dimensions. The second convolution block consists of three repeating convolution layers: 1 by 1 with 64 filters, 3 by 3 with 64 filters, and 1 by 1 with 256 filters. Convolution layers of 128 filters with 1 by 1 kernel, 128 filters with 3 by 3 kernel, and 512 filters with 1 by 1 kernel make up the third convolution block. In the third convolution block, all these convolution layers are repeated four times. The following convolution block consists of three duplicated kernels: 256 filters with 1 by 1 filters, 256 filters with 3 by 3 filters, and 1024 filters with 1 by 1 filters. The fifth convolution block consists of 2048 filters 1 by 1 kernel, 512 filters 1 by 1 kernel, and 512 filters 3 by 3 kernel. There are three repetitions of all three layers. The SoftMax activation function and a fully connected layer with 1000 nodes are used for classification after the sixth convolution block. Figure 8 shows the architecture of ResNet-50 model.

SGDM is applied as the optimizer for ResNet-50 in this experiment. Instead of using the last fully connected layer, another fully connected layer with output size of 7 is added. The model is then trained for 10 epochs with 20 batch sizes and a learning rate of 0.0001.

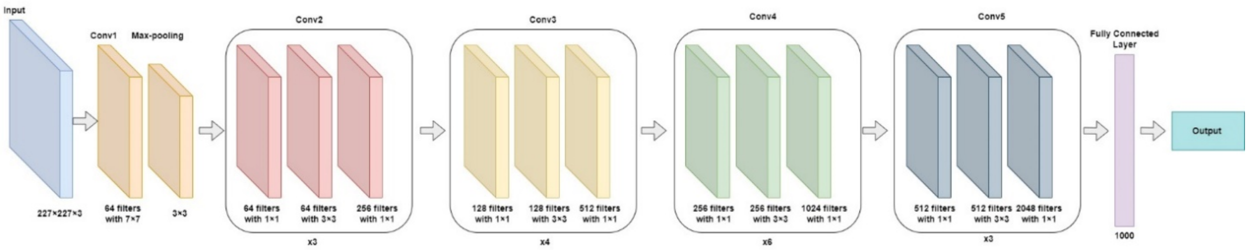


Fig. 8: ResNet-50 Architecture

3.3.6. MobileNet-v2

MobileNet-v2 is a 53 layers deep CNN. It requires image input with 224×224×3 in dimensions. Convolutional layers make up the structure of MobileNet-v2, which come after a fully connected layer and a layer of global average pooling. It consists of many inverted residual blocks. Each inverted residual block contains a linear bottleneck layer, a depthwise convolution layer, and a pointwise convolution layer. The depth-wise convolution layer conducts spatial filtering after the linear bottleneck layer decreases the amount of input sources. Finally, the pointwise convolution layer aggregates the spatial information across all channels. There are two types of blocks in MobileNet-v2 model, one is linear bottlenecks and another one is inverted residuals. By using linear transformations in place of conventional bottlenecks, computing costs are reduced while accuracy is raised. The addition of skip connections reduces the impact of depthwise convolutions on feature representation in the inverted residuals. The concept of the MobileNet-v2 is shown in Figure 9.

During the implementation, the MobileNet-v2 has a learning rate of 0.0001 and the optimizer utilized is SGDM. The epoch for this model is set to 10 times with the batch size of 20.

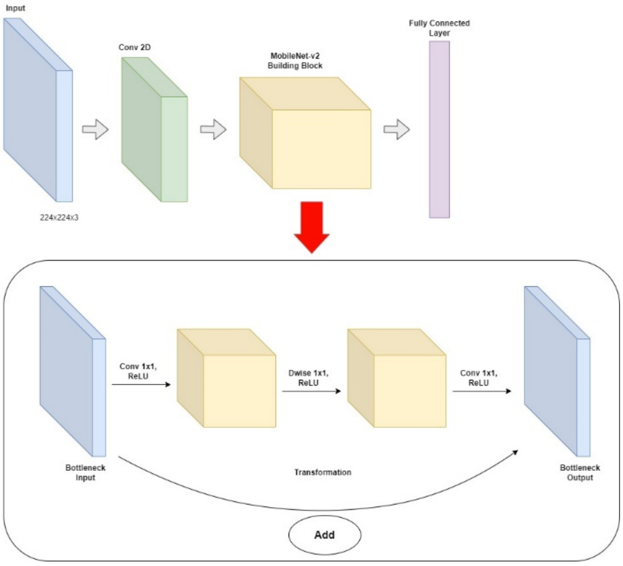


Fig. 9: MobileNet-v2 Architecture

4. Analysis

4.1. Experimental Setup

The models were trained on the Intel Core i7-11800H CPU running at 2.30GHz and the NVIDIA GEFORCE RTX 3060 GPU with 6 GB of dedicated graphic memory. The working environment is MATLAB 2021a.

4.2. Dataset

In this work, CASME II (CASME Database, n.d.) was used to train and evaluate the proposed transfer learning models. An enhanced spontaneous micro-expression database called CASME II was created by Yan et al. (Yan et al., 2014). There are 225 videos obtained from 26 participants in this database. There are a few types of facial expressions labelled in this database: happiness, sadness, disgust, repression, fear, surprise, and others. The samples are collected in a controlled environment laboratory. During the sample collection, each participant was required to watch a short video clip in order to help them elicit their micro-expression. The participants' micro-expressions were taken using a high-speed camera. The camera was set up and faced directly to the participant's face. All the participants' micro-expression samples were recorded at 200 frames per second with a resolution of 280×340 pixels. There are 247 micro-expressions with action units, and emotions labelled were selected from the 3000 facial movements for the database. The dataset has a total of 17124 static images in seven different facial expressions. This dataset includes seven categories of facial expression, which include Happiness, Sadness, Surprise, Disgust, Fear, Repression, and others. The distribution of the number of images of every facial expression was represented in Table 1. Figure 10 shows the sample from the CASME II (CASME Database, n.d.) database.

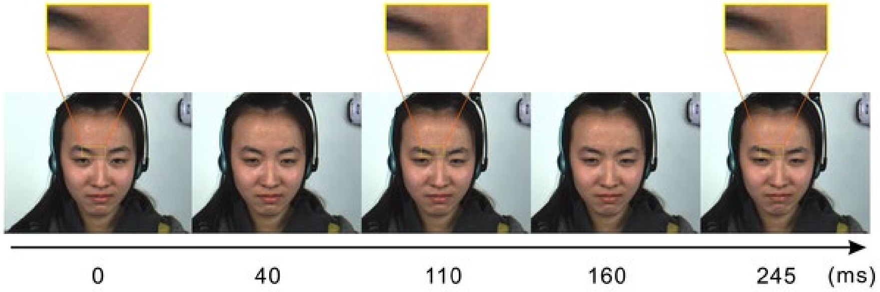


Fig. 10: Sample from CASME II Database (Yan et al., 2014)

Table 1: Distribution of Number of Images

Type of Expression	Number of Images
Happiness	2360
Sadness	150
Surprise	1729
Disgust	4204
Fear	127
Repression	2187
Others	6367
Total	**17124**

5. Result and Discussion

5.1. Experimental Settings

The images were divided into two different sets and resized into various sizes, which are 224×224 pixels for GoogleNet, EfficientNet-b0, ResNet 50 and MobileNet-v2. Then, the images were resized into 227×227 pixels for both AlexNet and SqueezeNet. After the resizing, the images will be split into 70% for training the models and 30% for evaluating the models.

5.2. Experimental Analysis and Discussions

This experiment consists of three settings: Transfer learning without freezing any layers, transfer learning by freezing 20% of layers and transfer learning by freezing 50% of layers. The aims of having three different experiment settings are to investigate how much a pre-trained model can improve its performance with configuration fine-tuning. The settings are described as follows:

- **Setting 1**: Transfer learning without freezing any layers.
- **Setting 2**: Transfer learning by freezing 20% of layers. The first 20% of the learnable layers including the pre-trained models of AlexNet, SqueezeNet, GoogleNet, EfficientNet-b0, ResNet 50 and MobileNet-v2 are frozen with the rest of the layers are trained with the training dataset. The number of classes at the output layers will be set to 7.
- **Setting 3**: Transfer learning by freezing 50% of layers. The first 50% of the learnable layers including the pre-trained models of AlexNet, SqueezeNet, GoogleNet, EfficientNet-b0, ResNet 50 and MobileNet-v2 are frozen with the rest of the layers trained with the training dataset. The number of classes at the output layers will be set to 7.

The pre-processed and cropped version of the ME are used to conduct the experiment. Each experiment setting is implemented for 5 trials and the performances of all trials are averaged to obtain a final performance measurement.

Based on Figure 11, the SqueezeNet outperforms AlexNet, GoogleNet, EfficientNet-b0, ResNet 50 and MobileNet-v2. In this experiment, the SqueezeNet obtained an accuracy of 99.79%. Its accuracy is higher than AlexNet by 0.03%, GoogleNet by 0.43%, EfficientNet-b0 by 0.18%, ResNet 50 by 0.91% and MobileNet-v2 by 1.9%. The reason SqueezeNet achieved such great performance due to the fire modules in this model. The expansion of the fire module into two sections allows the features to be trained evenly by every learnable layers. Hence, it can fully learn the features. The results obtained from the experiments for Setting 1 are recorded in Table 2.

Based on the illustration in Figure 12, GoogleNet outperforms AlexNet, SqueezeNet, EfficientNet-b0, ResNet 50 and MobileNet-v2. In this experiment, GoogleNet obtained an accuracy of 99.31%. Its accuracy is higher than AlexNet by 0.1%, SqueezeNet by 0.2%, EfficientNet-b0 by 0.16%, ResNet 50 by 0.94% and MobileNet-v2 by 1.8%. GoogleNet still managed to outperform other models although 20% of the learnable layers have been frozen due to the complexity of its structure. Since GoogleNet is mostly made up of inception modules that contain many convolution layers for learning, the inception modules are able to learn most of the features. Hence, the 20% freezing work just slightly decreases its efficiency. Table 3 has tabulated the results obtained from the experiments for Setting 2.

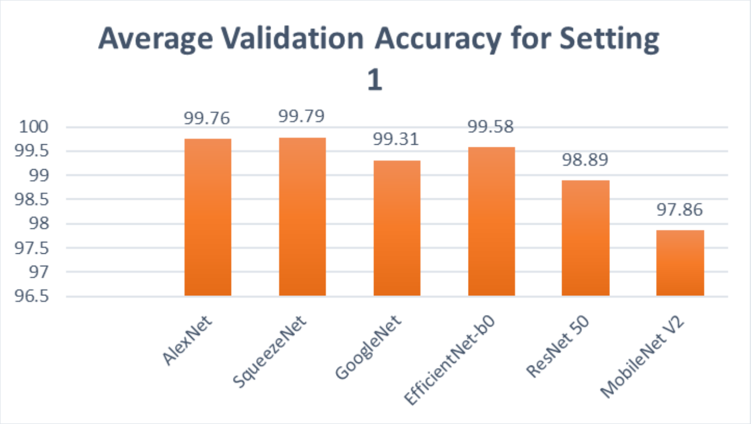


Fig. 11: Comparison of the Models for Setting 1

Table 2: Performance of the Pre-trained Models in Setting 1

Model	AlexNet	SqueezeNet	GoogleNet	EfficientNet-b0	ResNet 50	MobileNet-v2
Validating 1	99.92%	99.67%	99.42%	99.73%	98.70%	98.11%
Validating 2	99.88%	99.98%	99.40%	99.75%	98.95%	97.59%
Validating 3	99.49%	99.82%	99.38%	99.42%	98.95%	98.07%
Validating 4	99.86%	99.81%	99.36%	99.59%	98.81%	97.55%
Validating 5	99.65%	99.65%	99.22%	99.55%	98.97%	98.15%
Average Accuracy	99.76%	99.79%	99.36%	99.61%	98.88%	97.89%

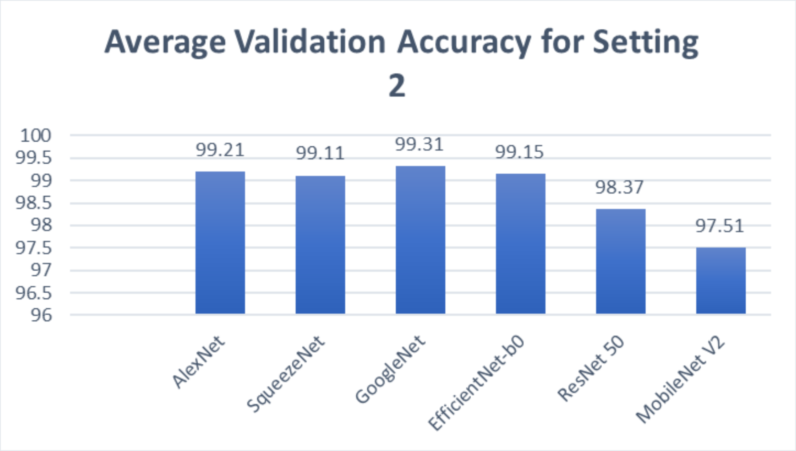


Fig. 12: Comparison of the Models for Setting 2

Table 3: Performance of the Pre-trained Models in Setting 2

Model	**AlexNet**	**SqueezeNet**	**GoogleNet**	**EfficientNet-b0**	**ResNet 50**	**MobileNet-v2**
Validating 1	98.92%	99.86%	98.44%	99.45%	98.09%	97.49%
Validating 2	99.49%	99.49%	99.51%	99.51%	98.73%	96.96%
Validating 3	99.28%	99.16%	99.57%	98.85%	98.48%	97.72%
Validating 4	99.77%	99.26%	99.61%	99.63%	98.48%	97.86%
Validating 5	98.58%	97.78%	99.44%	98.31%	98.05%	97.53%
Average Accuracy	99.21%	99.11%	99.31%	99.15%	98.37%	97.51%

Figure 13 shows AlexNet outperforms SqueezeNet, GoogleNet, EfficientNet-b0, ResNet 50 and MobileNet-v2. In this experiment, AlexNet obtained an accuracy of 99.84%. Its accuracy is higher than SqueezeNet by 0.52%, GoogleNet by 0.53%, EfficientNet-b0 by 0.93%, ResNet 50 by 1.3% and MobileNet-v2 by 2.5%. AlexNet has the best performance in this experiment setting because the feature learning of AlexNet mostly occurs in its fully connected layers unlike other proposed models in this experiment. Thus, it still managed to maintain the learnability although 50% of its learnable layers have been frozen. The results obtained from the experiments for Setting 3 are documented in Table 4.

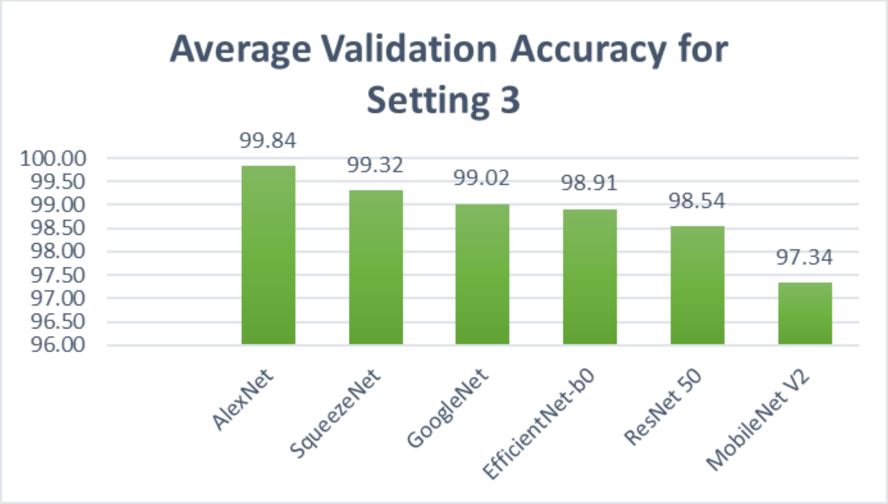


Fig. 13: Comparison of the Models for Setting 3

Table 4: Performance of the Pre-trained Models in Setting 3

Model	AlexNet	SqueezeNet	GoogleNet	EfficientNet-b0	ResNet 50	MobileNet-v2
Validating 1	99.77%	98.33%	98.13%	98.17%	98.19%	97.88%
Validating 2	99.82%	99.69%	99.71%	99.16%	98.72%	97.16%
Validating 3	99.81%	99.63%	98.72%	99.07%	97.31%	96.77%
Validating 4	99.88%	99.42%	99.38%	99.38%	99.59%	97.10%
Validating 5	98.92%	99.53%	99.18%	98.79%	98.89%	97.78%
Average Accuracy	99.84%	99.32%	99.02%	98.91%	98.54%	97.34%

Comparison of performance of the pre-trained models in three different settings is shown in Table 5. SqueezeNet outperformed the other five models in the Setting 1 with an accuracy of 99.79%. Its accuracy is higher than AlexNet by 0.03%, GoogleNet by 0.43%, EfficientNet-b0 by 0.18, ResNet 50 by 0.91% and MobileNet-v2 by 1.9%. In Setting 2, GoogleNet obtained an accuracy of 99.31% and it outperformed SqueezeNet, GoogleNet, EfficientNet-b0, ResNet 50 and MobileNet-v2. Its accuracy is higher than AlexNet by 0.1%, SqueezeNet by 0.2%, EfficientNet-b0 by 0.18%, ResNet 50 by 0.94% and MobileNet-v2 by 1.8%. Then, AlexNet outperformed the other five models with an accuracy of 99.84% in the Setting 2. The AlexNet in the Setting 3 shows the best result as compared to other models in other settings. Its accuracy is higher than SqueezeNet by 0.52%, GoogleNet by 0.53, EfficientNet-b0 by 0.93%, ResNet 50 by 1.3% and MobileNet-v2 by 2.5%. The difference of accuracy between Setting 1 and Setting 2 is 0.48%. Then, the difference of accuracy between Setting 3 and Setting 1 is 0.05% and 0.53% for Setting 2. AlexNet in Setting 3 obtained a higher accuracy because it freezes 50% of the convolution layers. The freezing layer method freezes the extra layers and eliminates any unnecessary training; thus, it is more efficient than the without freezing method and able to achieve higher accuracy.

Table 5: Comparison of Performance of the Pre-trained Models in three different settings

Model	Setting 1	Setting 2	Setting 3
AlexNet	99.76%	99.21%	99.84%
SqueezeNet	99.79%	99.11%	99.32%
GoogleNet	99.36%	99.31%	99.31%
EfficientNet-b0	99.61%	99.15%	98.91%
ResNet 50	98.88%	98.37%	98.54%
MobileNet-v2	97.89%	97.51%	97.34%

The comparison of the pre-trained models used in this work and other model on CASME Ⅱ is illustrated in Figure 14 and presented in Table 6. Zhi et al. (Zhi et al., 2019) utilized a 3D-CNN model with Transfer Learning and Fivefold cross-validation. Then, Wang et al. (Wang et al., 2018) utilized

TLCNN in their work and Sun et al. (Sun et al., 2020) proposed a TS-AUCNN for feature extraction and utilized SVM for classification. As shown in the table, the AlexNet with 50% of freezing layers shows a significant performance of 99.84% accuracy as compared to the model proposed by Zhi et al. (97.6%), Wang et al. (71.2%), Sun et al. (81.8%), SqueezeNet without freezing layer (99.79%) and GoogleNet with 20% of freezing layer (99.31%). Its accuracy is higher than Zhi et al. by 2.24%, Wang et al. by 28.64%, Sun et al. by 18.04%, SqueezeNet without freezing layer by 0.05% and GoogleNet with 20% of freezing layer by 0.53%.

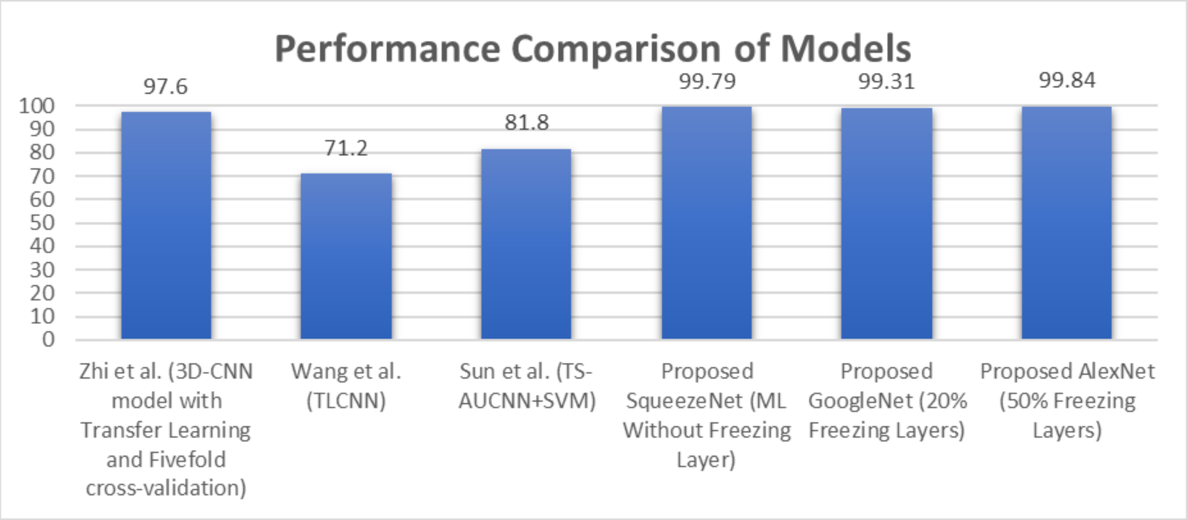


Fig. 14: Comparison of Proposed Models and Other Models on CASME II

Table 6: Comparison of Proposed Models and Other Models on CASME II

Model	Database	Accuracy
3D-CNN model with Transfer Learning and Fivefold cross-validation (Zhi et al., 2019)	CASME II	97.6%
TLCNN (Wang et al., 2018)	CASME II	71.2%
TS-AUCNN+SVM (Sun et al., 2020)	CASME II	81.8%
Proposed SqueezeNet (TL Without Freezing Layer)	CASME II	99.79%
Proposed GoogleNet (20% Freezing Layers)	CASME II	99.31%
Proposed AlexNet (50% Freezing Layers)	CASME II	99.84%

6. Conclusion

In this paper, we presented micro-expressions (MEs) recognition based on the transfer learning of six different types of pre-trained CNN models. These pre-trained CNN models were utilized on the classification of seven different categories of MEs. The seven types MEs include: Disgust, Fear, Happiness, Repression, Sadness, Surprise and Others. The models utilized in this experiment are namely AlexNet, SqueezeNet, GoogleNet, EfficientNet-b0, ResNet 50 and MobileNet-v2. This research has included three different settings of experiment: transfer learning without freezing any layers, transfer learning by freezing 20% of layers and transfer learning by freezing 50% of layers. The experiments were carried out on the publicly available dataset, the Chinese Academy of Sciences

Micro-expression (CASME II) database.

In conclusion, all the proposed pre-trained models managed to achieve state-of-the-art results in three different experiment settings. For the classification of MEs recognition in Setting 1, SqueezeNet surpassed the other five pre-trained models with a better accuracy of 99.79%. GoogleNet had achieved the highest accuracy of 99.31% in Setting 2 while AlexNet obtained higher accuracy of 99.84% in Setting 3. Although this research showed promising results in the classification of MEs, it can be further explored in the future. The purpose of this research is to explore more benefits not only in psychology study but in other fields like marketing field, security purposes, and other applicable fields. Hence, the MEs recognition study should be strengthened by incorporating more micro-expressions databases into the research to balance the number of image samples. The RNN approach can also be incorporated into the MEs recognition study to consider the sequential movement of the facial as well.

Acknowledgements

This work was supported by the Multimedia University internal IR Fund, Malaysia under grant No. MMUI/210110.

References

CASME Database. (n.d.). Retrieved June 28, 2023, from http://casme.psych.ac.cn/casme/e2

Chernykh, V., & Prikhodko, P. (2017). *Emotion Recognition From Speech With Recurrent Neural Networks*.

Fan, Y., Lu, X., Li, D., & Liu, Y. (2016). Video-Based emotion recognition using CNN-RNN and C3D hybrid networks. *ICMI 2016 - Proceedings of the 18th ACM International Conference on Multimodal Interaction*, 445–450. https://doi.org/10.1145/2993148.2997632

Gan, Q., Nie, S., Wang, S., & Ji, Q. (2017). Differentiating between posed and spontaneous expressions with latent regression Bayesian network. *Thirty-First AAAI Conference on Artificial Intelligence*.

Haque, M. A., Rani Alex, J. S., & Venkatesan, N. (2018). Evaluation of Modified Deep Neural Network Architecture Performance for Speech Recognition. *International Conference on Intelligent and Advanced System, ICIAS 2018*. https://doi.org/10.1109/ICIAS.2018.8540636

Jain, D. K., Shamsolmoali, P., & Sehdev, P. (2019). Extended deep neural network for facial emotion recognition. *Pattern Recognition Letters, 120*, 69–74. https://doi.org/10.1016/J.PATREC.2019.01.008

Jain, N., Kumar, S., Kumar, A., Shamsolmoali, P., & Zareapoor, M. (2018). Hybrid deep neural networks for face emotion recognition. *Pattern Recognition Letters, 115*, 101–106. https://doi.org/10.1016/J.PATREC.2018.04.010

Khorrami, P., Le Paine, T., Brady, K., Dagli, C., & Huang, T. S. (2016). How deep neural networks can improve emotion recognition on video data. *Proceedings - International Conference on Image Processing, ICIP, 2016-August*, 619–623. https://doi.org/10.1109/ICIP.2016.7532431

Krestinskaya, O., & James, A. P. (2017). Facial emotion recognition using min-max similarity classifier. *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, 2017-January*, 752–758. https://doi.org/10.1109/ICACCI.2017.8125932

Lasri, I., Solh, A. R., & Belkacemi, M. El. (2019). Facial Emotion Recognition of Students using Convolutional Neural Network. *2019 3rd International Conference on Intelligent Computing in Data Sciences, ICDS 2019*. https://doi.org/10.1109/ICDS47004.2019.8942386

Lim, Y., Ng, K.-W., Naveen, P., & Haw, S.-C. (2022). Emotion Recognition by Facial Expression and Voice: Review and Analysis. *Journal of Informatics and Web Engineering, 1*(2), 45–54. https://doi.org/10.33093/jiwe.2022.1.2.4

Lopes, A. T., de Aguiar, E., De Souza, A. F., & Oliveira-Santos, T. (2017). Facial expression recognition with Convolutional Neural Networks: Coping with few data and the training sample order. *Pattern Recognition, 61*, 610–628. https://doi.org/10.1016/j.patcog.2016.07.026

Madupu, R. K., Kothapalli, C., Yarra, V., Harika, S., & Basha, C. Z. (2020). Automatic Human Emotion Recognition System using Facial Expressions with Convolution Neural Network. *Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020*, 1179–1183. https://doi.org/10.1109/ICECA49313.2020.9297483

Nasri, M. A., Hmani, M. A., Mtibaa, A., Petrovska-Delacretaz, D., Slima, M. Ben, & Hamida, A. Ben. (2020). Face Emotion Recognition from Static Image Based on Convolution Neural Networks. *2020 International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2020*. https://doi.org/10.1109/ATSIP49331.2020.9231537

Pranav, E., Kamal, S., Satheesh Chandran, C., & Supriya, M. H. (2020). Facial Emotion Recognition Using Deep Convolutional Neural Network. *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, 317–320. https://doi.org/10.1109/ICACCS48705.2020.9074302

September, V. N., Lee, J., Ng, K., & Yoong, Y. (2023). *Face and facial expressions recognition system for blind people using ResNet50 architecture and CNN. 2*(2).

Sun, B., Cao, S., Li, D., He, J., & Yu, L. (2020). Dynamic Micro-Expression Recognition Using Knowledge Distillation. *IEEE Transactions on Affective Computing*. https://doi.org/10.1109/TAFFC.2020.2986962

Verma, A., Singh, P., & Rani Alex, J. S. (2019). Modified Convolutional Neural Network Architecture Analysis for Facial Emotion Recognition. *International Conference on Systems, Signals, and Image Processing, 2019-June*, 169–173. https://doi.org/10.1109/IWSSIP.2019.8787215

Wang, S. J., Li, B. J., Liu, Y. J., Yan, W. J., Ou, X., Huang, X., Xu, F., & Fu, X. (2018). Micro-expression recognition with small sample size by transferring long-term convolutional neural network. *Neurocomputing, 312*, 251–262. https://doi.org/10.1016/J.NEUCOM.2018.05.107

Yan, W. J., Li, X., Wang, S. J., Zhao, G., Liu, Y. J., Chen, Y. H., & Fu, X. (2014). CASME II: An Improved Spontaneous Micro-Expression Database and the Baseline Evaluation. *PLOS ONE, 9*(1), e86041. https://doi.org/10.1371/JOURNAL.PONE.0086041

Yani, M., S Si., M. T. B. I., & S.T., M. T. C. S. (2019). Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail. *Journal of Physics: Conference Series, 1201*(1), 12052. https://doi.org/10.1088/1742-6596/1201/1/012052

Zhang, T., Zheng, W., Cui, Z., Zong, Y., & Li, Y. (2019). Spatial-Temporal Recurrent Neural Network for Emotion Recognition. *IEEE Transactions on Cybernetics, 49*(3), 939–947. https://doi.org/10.1109/TCYB.2017.2788081

Zhi, R., Xu, H., Wan, M., & Li, T. (2019). Combining 3D Convolutional Neural Networks with Transfer Learning by Supervised Pre-Training for Facial Micro-Expression Recognition. *IEICE Transactions on Information and Systems, E102.D*(5), 1054–1064. https://doi.org/10.1587/TRANSINF.2018EDP7153