AI Creation of Facial Expression Database for Advanced Emotion Recognition

Jia Jun Ho , Wee How Khoh * , Ying Han Pang , Hui Yen Yap , Fang Chuen Lim Alvin

Posted Date: 4 February 2026

doi: 10.20944/preprints202602.0339.v1

Keywords: convolution neural networks; deep learning; facial expression recognition; generative model; micro-expression

 doi:10.20944/preprints202602.0339.v1

Article

AI Creation of Facial Expression Database for Advanced Emotion Recognition

Jia Jun Ho [1], Wee How Khoh [2,*], Ying Han Pang [2], Hui Yen Yap [2] and Fang Chuen Lim Alvin [2]

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

[2] Centre for Advanced Analytics, CoE for Artificial Intelligence, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, 75450, Melaka, Malaysia

* Correspondence: whkhoh@mmu.edu.my; Tel.: +60-62523465

Abstract

With applications in psychology, security, and human-computer interaction, facial expression recognition (FER) has become an essential tool for non-verbal communication. Current research often categorizes expressions into micro and macro types, yet existing datasets suffer inconsistent labelling for classes, limited diversity of the databases, and insufficient scale for the currently available datasets. To address these gaps, this work proposes a novel framework combining the Diffusion model with pre-trained CNNs. Leveraging original images from established datasets, CASME II, we generate synthetic facial expressions to augment training data, mitigating bias and inconsistency. The synthetic dataset is evaluated using ResNet 50, VGG16 and Inception V3 architectures. Inception V3 trained on the proposed AI-generated dataset and tested using CASME II achieved the highest accuracy of 99.48%. VGG-16 with data augmentation applied was trained on CASME II and tested on the proposed AI-generated dataset achieved 99.54%. While 30% freezing layers method is utilized, Inception V3 trained on the proposed AI-generated dataset and tested using CASME II obtained an accuracy of 99.53%. The data augmentation and freezing layers approaches have significantly improved the performance of the models. Our proposed approaches have achieved state-of-the-art performance and outperformed most of the existing state-of-the-art approaches benchmarked in this study.

Keywords: convolution neural networks; deep learning; facial expression recognition; generative model; micro-expression

1. Introduction

In our daily lives, facial expressions are quite important. Facial expression is a kind of non-verbal communication for human that helps humans to express their feelings or messages in a more useful and effective way. Facial expression is first explored by Dr. Ekman and Friesen in 1971 [1]. They proposed a Facial Action Code System (FACS) to distinct the facial expressions by using action units (AUs). There are two categories of facial expressions, which are micro expression and macro expression. Micro expression is a form of reflexive behaviour in which an individual expresses their actual feelings via facial muscle movements. In contrast, macro expression is recognizable and easy to be captured by raw eyes. It can represent an individual genuine feeling or acted emotion. Both micro and macro expressions provide effective communication and help to understand a person's current feelings. The common facial expressions are smile, sad, fear, angry, surprise, disgust, confused and neutral [2].

Facial expression recognition (FER) is a cutting-edge technology within the fields of computer vision and affective computing that utilizes facial features taken from images or videos in order to automatically identify and interpret human emotions. Three main stages were often included in the FER methodology: face detection, feature extraction and facial expression classification. Even though

 doi:10.20944/preprints202602.0339.v1

FER is commonly practiced by researchers, it is still facing some critical issues which affect the effectiveness and efficiency of FER. For example, the current existing facial expression datasets that frequently utilized to conduct the studies on FER is suffered from several limitations such as inconsistent labelling for classes, limited diversity of the databases, and insufficient scale for the currently available datasets.

In order to overcome the limitations of facial expressions database, generative models have been used in some studies. Generative model is a kind of machine learning model that has the ability to produce new data which is similar to the data it was trained on. Generative models focus on understanding the fundamental distribution of input data, in contrast to discriminative models, which emphasise distinguishing between categories or predicting labels. Once the generative model is trained, it can generate completely new, synthetic data that share similar features to the original dataset. Generative models can be used to generate data like images, videos, audio, texts, text-to-image, text-to-video, image captioning, 3D modelling and other possible data. There are many types of generative models including Generative Adversarial Network (GAN), Variational Autoencoder (VAE), Autoregressive model, Diffusion model others. Generative models are rapidly being employed across a variety of areas such as scientific research, healthcare, simulation, art and design, content creation, natural language processing (NLP) etc. Generative models were deployed to generate new facial expression images in this work. In the following study, a LAUN upgraded StarGAN for face emotion recognition is proposed [3]. StarGAN and LAUN improved StarGAN were utilized to create a series of higher quality fake facial emotions images for every emotion. [4] employed a conditional generative adversarial network (CGAN) in their work. It is an unsupervised domain adaption for the recognition of face emotions. Furthermore, [5] proposed an unsupervised learning micro-expression generative adversarial network (ULME-GAN) to generate micro-expression sequences. To improve accuracy, an AU-matrix re-encoding (AUMR) was deployed, and transfer learning approach was utilized to train their proposed generator network. Then, a multi-sequence based micro-expression (ME) generation approach for ME recognition is introduced by [6]. A facial expression recognition is proposed by [7] using maximum margin Gaussian Mixture Models (GMM). By recording Complex Spatio-Temporal Relations between face muscles, [8] suggested a method for recognizing facial expressions.

The performance and resilience of FER systems have been significantly improved by the recent developments in deep learning architectures, especially CNN. To elevate the performance of FER, pre-trained CNN models are proposed in several most recent works. Pre-trained CNN models are deep learning models which have been trained using a large dataset like ImageNet. It eliminates the need of train from scratch, which is time consuming and cost large amount of resources. It is much more efficient and has better performance. Pre-trained CNN models are commonly used in applications such as object detection, face recognition, image classification, medical imaging and others. There are several popular pre-trained CNN models that can be adopted in this work including VGG, ResNet, AlexNet, SqueezeNet, GoogleNet, Inception, MobileNet and others. In the following study, Xception CNN paired with a K-fold cross-validation technique is adopted by [9]. In order to identify students' moods based on their facial expressions, [10] proposed a CNN model. Furthermore, [11] employed a 2-dimensional (2D) CNN to identify the facial emotion and evaluated the model by using a self-collected facial emotion database which contains five emotions. In order to enhance the performance of CNN model, an unique Venturi Architecture that contains 6 hidden layers and one output layer for CNN was proposed by [12]. Moreover, a CNN model derived facial expression recognition algorithm was proposed by [13]. Besides, CNN was used in Naik and Mehta's Hand-over-Face Gesture based Facial Emotion Method (HFG_FERM) [14]. A new model called feature redundancy-reduced convolutional neural network (FRR-CNN) has been proposed by [15] to recognize facial expressions and experimented using CK+ and JAFFE. In order to recognise face expressions in the wild, [16] employed three convolution neural network models including Light CNN, dual-branch CNN and pre-trained CNN. By employing a residual network, [17] presented micro-expression identification. For the purpose of recognizing facial expressions, [18] presented a

 doi:10.20944/preprints202602.0339.v1

Fast Regions with Convolutional Neural Network Features (Faster R-CNN). Besides that, [19] 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).

With enhanced feature extraction and preprocessing techniques, [20] presented a real-time face emotion recognition system. Based on textural pattern and convolutional neural networks, [21] presented facial emotion recognition. In addition, a facial expression recognition by using local learning with deep and handcrafted features was presented [22]. By using facial expressions, [23] demonstrate an emotion identification system for drivers while driving. This experiment uses FERDERnet to recognise the facial expressions of drivers while they are driving. Additionally, an Identity-Aware CNN (IACNN) is created by [24] for the facial expression recognition. Then, Attention Net (FERAtt) was proposed by [25] for facial expression recognition. There are two methods implemented in this experiment: a model with attention and classification (FERAtt+Cls), and a model with attention, classification, and representation (FERAtt+Rep+Cls). Moreover, [26] used a 3D CNN and transfer learning to recognize facial micro-expression. [27] adopted (TLCNN) model to recognize micro-expression (ME) with a tiny sample size. Furthermore, deep convolution networks were proposed by [28] for face emotion identification by applying normalization, Action Units (AUs) and CNN. For FER, [29] presented an attention mechanism-based CNN. The micro movements of the faces are captured, and the texture information of the image is obtained using LBP features. Based on face local regions, [30] presented a compact and efficient facial emotion detection network. [31] used an enhanced spatial-temporal learning network (ESTLNet) to conduct dynamic facial expression recognition.

The idea of this research is to introduce a novel approach for facial expression recognition by combining the generative model and pre-trained CNN model. Diffusion model was used to generate new AI generated images in order to create an AI creation database. CASME Ⅱ was used by generative model as training data in order to generate new AI images. The new AI generated images will be fed into the pre-trained CNN models for classification, and the performance of each pre-trained CNN model will be compared to each other.

The rest of the paper is organized as the following: Section 2 describes the dataset used, generative model used, and the models used in this work. Section 3 defines the settings used and discussed the model's performance in this work. Section 4 concludes the findings of this work and Section 5 suggests some ideas for the future work.

2. Methodology

2.1. Methodology Overview

In this research, an AI creation facial expression database will be created for facial expression recognition. This research uses a generative AI model to create an AI-generated dataset and pre-trained CNN models are utilized to assess the AI-generated dataset. CASME Ⅱ is the dataset utilized in this work. Firstly, CASME Ⅱ is utilized to train the Diffusion model in order to produce a new AI-generated facial expression dataset. In this work, there are two types of datasets utilized: CASME Ⅱ and the proposed AI-generated dataset. The images obtained from both datasets were pre-processed before feeding into the pre-trained CNN models for the classification task. The preprocessing approaches employed are resizing the image, rescaling and turning into grayscale. The pre-processed images are used to train and test every single pre-trained CNN model adopted in this work. This work can be segmented into four sections: train and test using CASME Ⅱ, train and test using the proposed AI-generated dataset, train using CASME Ⅱ and test using the proposed AI-generated dataset, and train using the proposed AI-generated dataset and test using CASME Ⅱ. Inception V3, VGG-16 and ResNet 50 are the pre-trained CNN models proposed in this work. Inception V3, VGG-16 and ResNet 50 carried out the classification task to distinct the seven different facial expressions including happiness, sadness, disgust, repression, fear, surprise and others. Lastly, the performance

 doi:10.20944/preprints202602.0339.v1

for each model was obtained and compared to each other. The proposed methodology's overview is given in Figure 1.

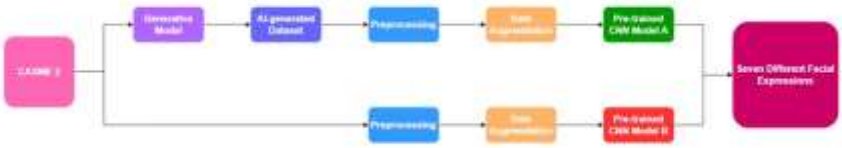


Figure 1. The proposed methodology's overview.

2.2. Datasets

2.2.1. CASME Ⅱ

The Chinese Academy of Sciences Micro-expression 2 (CASME Ⅱ) is an enhanced spontaneous micro-expression database which created by [32] in 2014. CASME Ⅱ is an enhanced version of CASME dataset, which was developed by [33]. 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, and the participants' micro-expressions were taken using a high-speed camera. To elicit their micro-expression, every participant had to view a short video clip throughout the sample collecting process. The camera was set up and faced directly to the participant's face. The micro-expression samples from every subject were recorded at a speed of 200 fps with a pixel size of 280×340. Out of 3000 facial movements in the database, 247 micro-expressions labelled with action units and emotions were chosen. 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 each facial expression's image count was recorded in Table 1. The sample images with 7 different micro-expressions are shown in Figure 2.

Figure 2. Sample from CASME Ⅱ database.

Table 1. Distribution of number of images for CASME Ⅱ database.

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

2.2.2. Proposed AI-Generated Dataset

In this work a, self-proposed AI-generated facial expression dataset is introduced. The proposed AI-generated dataset is created using Diffusion model. The proposed AI-generated facial expression

 doi:10.20944/preprints202602.0339.v1

dataset is generated based on the original CASME Ⅱ obtained from the author. The pixel size of the images obtained from CASME Ⅱ is resized from 280×340 to 48×48 and turned into grayscale before fit into the Diffusion model for training purpose. Then, the Diffusion model is fine-tuned in order to fulfil the pre-processed images specification. The diffusion model is well trained before it is utilized to generate the new facial expression images. The process of generating new AI-generated facial expression datasets is shown in Figure 3. Moreover, the completely trained model is then used to generate new facial expression images that contains seven categories of facial expression, which include Happiness, Sadness, Surprise, Disgust, Fear, Repression, and others. The newly generated facial expression images are in grayscale with pixel sizes of 48×48. The generated facial expression dataset consists of 15464 images. The distribution of each facial expression's image count was recorded in Table 2. The sample images with 7 different micro-expressions are shown in Figure 4.

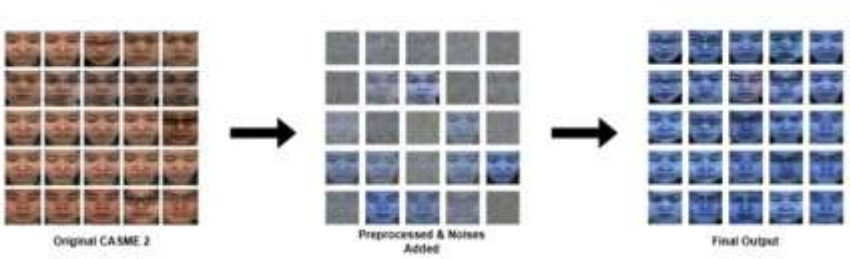


Figure 3. Process of generating new dataset.

Figure 4. Sample from proposed AI-generated facial expression dataset.

Table 2. Distribution of number of images for proposed AI-generated facial expression dataset.

Type of Expression	Number of Images
Happiness	2780
Sadness	1632
Surprise	1845
Disgust	3120
Fear	1760
Repression	2187
Others	2140
Total	15464

2.3. Preprocessing

Raw data often contains unnecessary and noisy information such as background change. Hence, preprocessing is introduced in this case. It is often known as data cleaning and data organizing process. Preprocessing is a series of processes of preparing raw data before feeding it into a machine learning model or deep learning model. It provides cleaner and precise data for the models and helps them to have a better learning of the data.

In this work, there are two set of preprocessing carried out. One for the Diffusion model and the other for the pre-trained CNN models. During the preprocessing for Diffusion model, the images obtained from CASME Ⅱ is resized from pixel size of 280×340 to 48×48. Then, the images were turned

 doi:10.20944/preprints202602.0339.v1

from RGB to grayscale. These approaches were taken due to the resources constraints to train the Diffusion model.

In the preprocessing for the pre-trained CNN models, the images obtained from both CASME Ⅱ, and our proposed AI-generated dataset are first resized from 48×48 pixels to 160×160 pixels. Then, data augmentation is utilized to artificially raise the number of images to help the architecture to have a better learning of the features. The image is rotated 15 degrees and horizontally flipped. Moreover, width shift and height shift were applied with the range settings of 0.1. The brightness of the images was also adjusted between the range of 0.8 to 1.2. After all these adjustments, the images' fill mode is set to nearest and rescaled to 1/255. Furthermore, the dataset was further separated into 80% for training and 20% for testing.

2.4. Diffusion Model

Diffusion model is one of the most popular generative models which mainly used for image generation. It not only has the usage of image generation but also denoising and data generation. Unlike other generative models, it transforms the data slowly into a dense structure. There are two different steps involved in diffusion model: forward process and reserve process. During forward process, diffusion model will take an image and slowly add noise to the image over a series of steps. The noise of the image will be increased in every step until the image becomes nearly indistinguishable from the random noise. It attempts to simulate a Markov chain, in which data gets gradually corrupted. The reverse process is started once the noise is added into the images. This process is also called denoising because it will try to eliminate the added noise from the image and reconstruct the image. In order to denoise the image, a neural network is employed. The neural network will be trained to recognize the original image based on the input noise and try to restore the original state of the image.

2.4.1. Forward Process (Diffusion Process)

Forward process is also known as diffusion process. In Forward process, the model will increasingly destroy the original input image by adding Gaussian noises into it. The noises will keep adding into the image until the image becomes blurry from pure noise. Figure 5 illustrates the process of Forward Process.

In theory, a clean data x_0 is step by step corrupted with the introduce of small amounts of Gaussian noise over many time steps $t = 1, 2, \dots, T$. A Markov chain might be applied to explain this, in which the current noisy sample x_t relies solely on the prior x_{t-1} and so on. It is mathematically expressed as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathrm{I}) \quad (1)$$

β_t is a small variance term, it controls the amount of noise introduced at each step. This process turns the structured data into random noise over a number of steps.

Instead of modelling noise addition in every step, the whole forward process is defined in a single closed-form formula that directly ties the noisy sample x_t to the original data x_0:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I) \quad (2)$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. By using the sampling equation, a noisy version of the input at any timestep t

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \quad (3)$$

$\epsilon \sim \mathcal{N}(0, \mathrm{I})$ represents a random noise vector. As t increases, the term $\sqrt{\bar{\alpha}_t}$ declines, indicating that the original data contributes less and the sample is controlled by noise.

 doi:10.20944/preprints202602.0339.v1

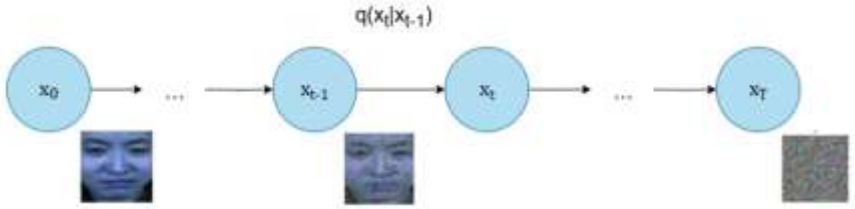


Figure 5. Forward process.

2.4.2. Reverse Process (Denoising Process)

Reverse process is also known as denoising process. In Reverse process, the model will try to learn removing noise from the image and restore the image back to its original form. Once the model is trained, it gains the ability to produce new images by applying the step-by-step reverse diffusion method. The process of Reverse Process is presented in Figure 6.

In theory, it starts with random noise and the reverse step from x_t to x_{t-1} is given by a Gaussian:

$$p_0(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1};\, \mu_\theta(x_t,t), \Sigma_\theta(x_t,t)\right) \tag{4}$$

where a neural network is parameterized by θ predicts both mean μ_θ and variance Σ_θ. Since the actual reverse distribution is unidentified, the network is taught to estimate it.

While in the training, the model is trained to predict the noise ϵ that was inserted to the forward process using a simple objective function:

$$\mathcal{L}_{simple} = \mathbb{E}_{x_{0,t,\epsilon}}[\|\epsilon - \epsilon_\theta(x_t,t)\|^2] \tag{5}$$

The network attempts to identify the source of a noisy sample x_t. Once trained, the process is reversed using the formula:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha_t}}}\epsilon_\theta(x_t,t)\right) + \sigma_t z \tag{6}$$

z represents as a small random noise sample, while σ_t represents as a variance term. By repeating this process from $t = T$ to $t = 0$, noise is progressively eliminated, producing a realistic and excellent sample.

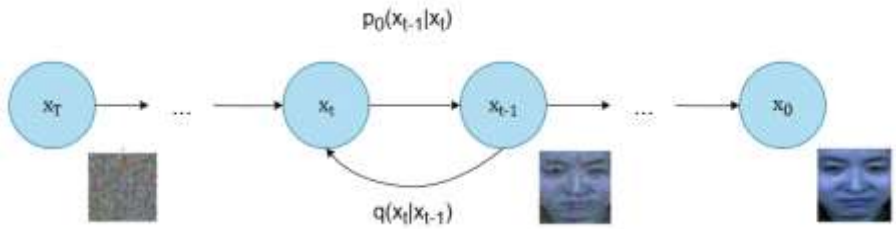


Figure 6. Reverse process.

2.5. Convolutional Neural Network (CNN)

Object identification, facial recognition, image classification, medical imaging, and other tasks are commonly performed using a deep learning model recognized as a convolutional neural network (CNN). It originated from the architecture of LeNet, which was invented by [34]. CNN is applied to analyse input such as pictures or numeric data. CNN does not require much preprocessing. CNN is designed according to the human brain neuron networks. The convolutional layer of CNN is used to collect picture features. Max pooling and average pooling layers; the common pooling layers

 doi:10.20944/preprints202602.0339.v1

employed in CNN. To classify the outcomes of data classifications, a fully connected layer is utilized. A CNN model generally comprises of input layer, convolution layers, pooling layers, fully connected layers, and an output layer.

2.5.1. Input Layer

An input layer, the first layer of a CNN model, which responsible for receiving raw data such as images and the data will be passed to convolution layer to extract the features from the data.

2.5.2. 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 things it requires. The input data is processed using convolutional techniques to extract important characteristics and capture spatial correlations. In the convolution procedure, kernel 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.

2.5.3. Activation Layer

An activation layer is the layer that usually utilized after every convolution layer or fully connected layer. It is utilized in order to integrate non-linearity into the architecture. Activation functions come in a variety of forms, such as Rectified Linear Unit (ReLU), Leaky ReLU, Sigmoid function, Hyperbolic Tangent (Tanh), and SoftMax function, but the most common exploited activation function is Rectified Linear Unit (ReLU). ReLU is mathematically defined as:

$$ReLU(x) = max\,(0, x) \tag{7}$$

2.5.4. Pooling Layer

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. Average pooling as well as Max pooling are the two different pooling layers. The concept of Average pooling and Max pooling are illustrated in Figure 7.

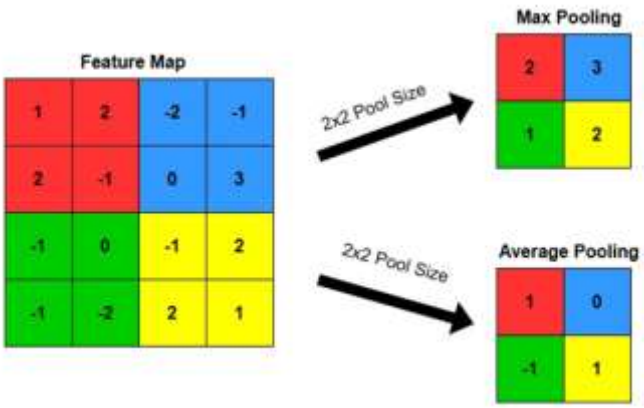


Figure 7. Pooling layers.

2.5.5. Fully Connected Layer (FC Layer)

Occasionally, a thick layer in CNN is used to refer to a fully connected layer (FC). It usually appears at the bottom of a neural network. Each intersection in FC layer's output layer has a direct

 doi:10.20944/preprints202602.0339.v1

connection to an intersection in the pooling layer above. Afterward, the output will be passed to a corresponding layer for image classification.

2.5.6. Output Layer

An output layer is the final layer of CNN. This layer is mainly focus on the prediction and classification tasks. There are several types of activation functions applied in output layer, namely Sigmoid function, SoftMax function and Linear function. Sigmoid function is adopted in binary classification where normally occurs two classes classification. Meanwhile, SoftMax function is employed for multi-class classification where normally occurs two or more classes classification. Moreover, linear function is suitable for regression tasks. It is used to predict continuous value such as stock market price. Output layer is mathematically defined as:

$$y = f(Wx + b) \tag{8}$$

where f is represented as the activation function (e.g. Softmax, Sigmoid, Linear). Besides, W is represented as the weights learned during the training process. Then, x is defined as the input vector, which is also known as features. Lastly, b is defined as the bias term.

2.6. Transfer Learning

In CNN, there is a methodology known as transfer learning. It enables the application of an architecture trained on a larger dataset to a new but similar task. Training a CNN model from scratch often requires a significant amount of time, computational power and labelled data. With the use of transfer learning, it can eliminate the need to train a CNN model from scratch and leverages the knowledge gained from the previously trained architecture which has been trained on a sizable dataset (i.e., ImageNet, which contains over 1000 object classes). Then, the previously trained model is utilized to a new target domain. There are several popular pre-trained CNN models including AlexNet, SqueezeNet, GoogleNet, ResNet, VGG, Inception, MobileNet, EfficientNet, DenseNet etc. The concept of transfer learning is as illustrated in Figure 8. The knowledge gained from the Domain A is saved and is utilized it in Domain B. In conventional CNN model, the data sample will be added into the input layer, and passed into the convolution layers to train the Network A. After the trained knowledge has completed in Network A, the knowledge will transfer to the Network B, and the other set of input will be used to further train the convolution layers in the Network B. The learned knowledge will pass through the fully connected layers in Network B in order to further training and classifying the data into different category and it will be displayed at the output layer.

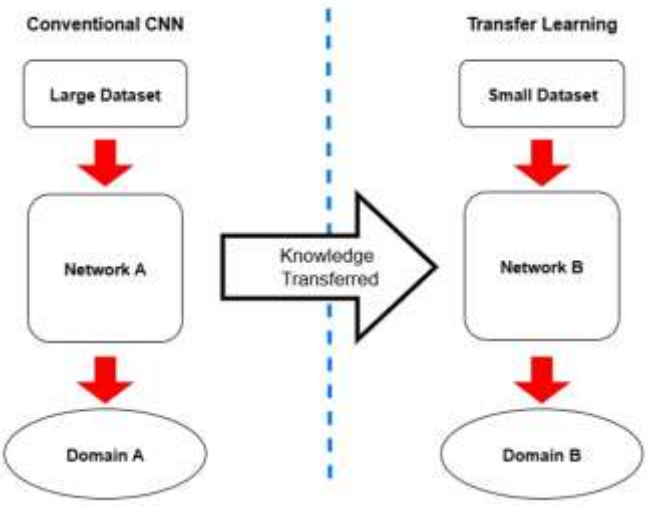


Figure 8. Concept of transfer learning.

 doi:10.20944/preprints202602.0339.v1

2.7. Pre-Trained CNN Models

Pre-trained CNN models are the deep learning architecture which has been trained on a sizable and diverse dataset. It is the model adopted transfer learning; it eases the need of train from scratch and enhances the knowledges form the previous trained model in order to perform a new task. Pre-trained CNN models can save up a lot of resources including time, computational resources and data. It also able to provide state-of-the-art performance. There are a variety of popular pre-trained CNN models to be deployed in FER, namely AlexNet, SqueezeNet, VGG-16, VGG-19, ResNet, Inception, MobileNet and EfficientNet. In this study, the pre-trained CNN models that are chosen to adopt including ResNet 50, VGG-16 and Inception V3.

2.7.1. ResNet 50

ResNet-50, a deep convolution neural network architecture with 50 layers, is extensively utilized for image recognition applications. Microsoft Research debuted it in 2015 as a part of the ResNet family, which secured victory in the ImageNet competition that year [35]. The use of "skip connections", which promote gradient flow during backpropagation and mitigate the vanishing gradient problem that occasionally occurs in very deep networks, is one of ResNet-50's standout features. With 49 convolutional layers and a final fully connected layer, the architecture is composed of a series of bottleneck blocks, each with three layers: a 1×1 convolution for dimensionality decreasing, a 3×3 convolution, and another 1×1 convolution for restoring dimensions. Because of its deep and computationally efficient design, ResNet-50 enables high accuracy image classification and transfer learning applications. In Figure 9, the architecture of ResNet 50 is explained.

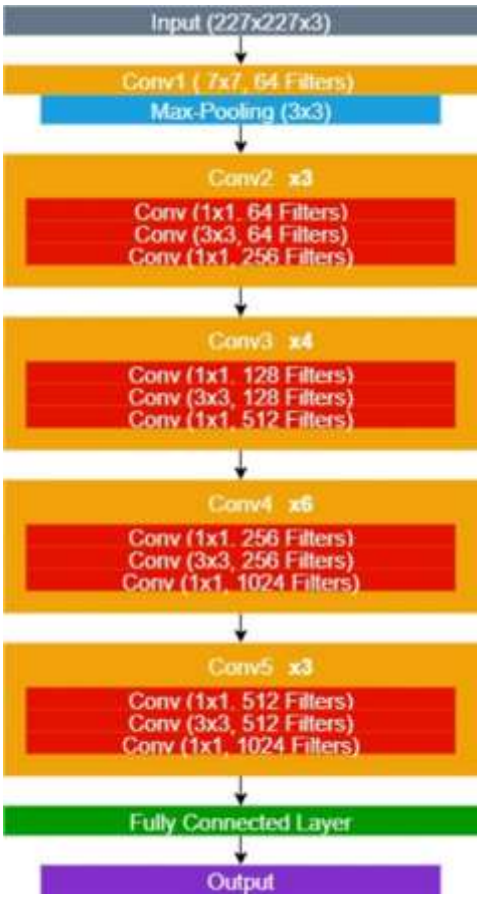


Figure 9. Architecture of ResNet 50.

 doi:10.20944/preprints202602.0339.v1

2.7.2. VGG-16

VGG-16 [36], a well-known deep convolutional neural network design was introduced in the 2014 ILSVRC (ImageNet Large Scale Visual Recognition Challenge). It is developed by the Visual Geometry Group at the University of Oxford. It has 16 weight layers, containing 3 fully connected and 13 convolutional layers, and a SoftMax layer for categorization. The primary idea of VGG-16 is the recurrent application of tiny 3×3 convolution filters, which enables the network to capture intricate characteristics while keeping the number of parameters under control. A max-pooling layer is placed after every of the five blocks that make up these levels in order to minimise spatial dimensions. VGG-16 is popular for its clear and consistent architecture, which makes it powerful and simple to utilize for image classification, feature extraction, and transfer learning issues. The architecture of VGG-16 is presented in Figure 10.

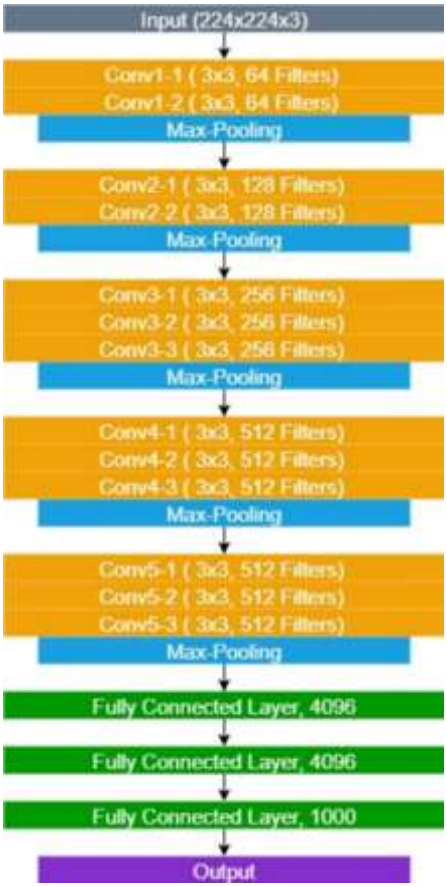


Figure 10. Architecture of VGG-16.

2.7.3. Inception V3

Inception V3 [37] is also known as GoogleNet, it is a deep CNN architecture developed by Google as a part of the Inception series. Inception V3 is the third version of Inception model, and it was introduced in 2015. The goal of Inception V3 was to carry out large-scale image recognition tasks more accurate and faster at the same time. Inception V3 is begin with an input size of 299×299×3 and goes through a few convolution layers and pooling layers. It then passes through several kinds of Inception modules, including Inception Modules A, B and C. The network identifies multi-scale spatial patterns by using comparable convolutional paths with various filter sizes such as 1×1, 3×3 and 5×5, and pooling layers. The large convolutions are factorized into smaller convolutions to enhance the performance. For example, a 5×5 layer is replaced with two 3×3 layers. The feature maps are down-sampled from 35×35 to 17×17 and subsequently to 8×8 using Reduction-A and Reduction-B modules. This increases depth while keeping important information. Batch normalization and

 doi:10.20944/preprints202602.0339.v1

auxiliary classifiers are utilized across the network to improve training stability and regularization. The last stage consists of a global average pooling, a dropout layer for overfitting prevention, fully connected layers, and SoftMax activation for class probabilities. In Figure 11, the Inception V3 architecture's summary is illustrated.

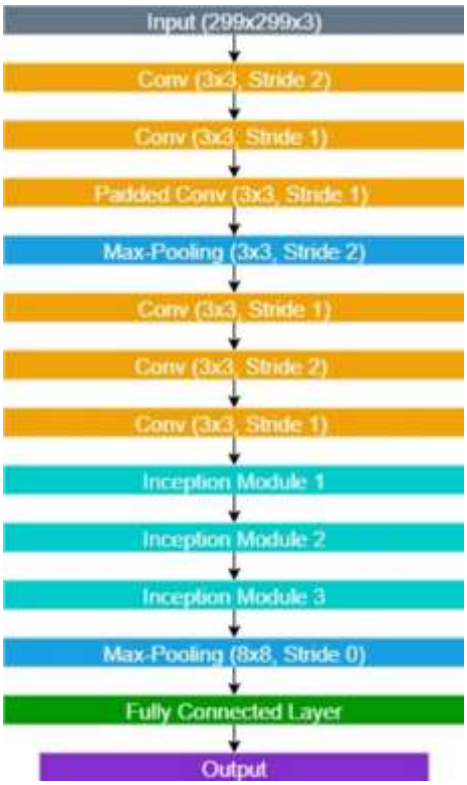


Figure 11. Overview of Inception V3 architecture.

3. Results and Discussion

3.1. 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.

3.2. Experimental Results

This study was conducted using four different of dataset settings: train and test on CASME Ⅱ, train and test on proposed AI-generated dataset, train on CASME Ⅱ and test on proposed AI-generated dataset, and train on proposed AI-generated dataset and test using CASME Ⅱ. For the evaluation of every dataset's performance, pre-trained CNN models, namely Inception V3, ResNet 50, VGG-16 and were employed in this study. The settings are described as follows:

- **Setting 1:** Inception V3, ResNet 50, and VGG-16 were training and testing on the original CASME II. This CASME II is the original dataset that obtained from the original author of this dataset.
- **Setting 2:** Inception V3, ResNet 50, and VGG-16 were training and testing on the proposed AI-generated dataset. The proposed AI-generated dataset is generated using Diffusion model that trained on CASME II.
- **Setting 3:** Inception V3, ResNet 50, and VGG-16 were first trained using CASME II and then tested on the proposed AI-generated dataset.

 doi:10.20944/preprints202602.0339.v1

- **Setting 4:** Inception V3, ResNet 50, and VGG-16 were first trained on the proposed AI-generated dataset and tested using CASME II.

3.2.1. CASME Ⅱ

CASME Ⅱ is employed to train and test the pre-trained CNN models in this experiment as the baseline dataset in order to compared with the other datasets. There are 5 trials of experiments for VGG-16, ResNet 50 and Inception V3, obtained results from every trial were applied to obtain the average accuracy for every model.

In Table 3, the performance obtained from each model while trained using CASME Ⅱ is presented. VGG-16 is the best performed model while trained on CASME Ⅱ with the best classification accuracy reported 99.33%. Meanwhile, Inception V3 and ResNet 50 obtained 99.11% and 88.34%, respectively. VGG-16 has outperformed both ResNet 50 and Inception V3. VGG-16 has slightly higher accuracy as compared to Inception V3 by 0.22%. While comparing with ResNet 50, VGG-16 has a much significantly difference in accuracy by 10.99%. This is due to VGG-16 has a shallow and compact architecture as compared to ResNet 50 and allow it to have a much efficient learning for every facial expression. In this experiment, ResNet 50 has the lowest accuracy.

Table 3. Performance of every pre-trained CNN model on CASME Ⅱ.

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	99.13	99.35	99.30	99.42	99.46	99.33
ResNet 50	85.62	86.49	90.17	89.42	89.98	88.34
Inception V3	98.16	99.32	99.45	99.23	99.39	99.11

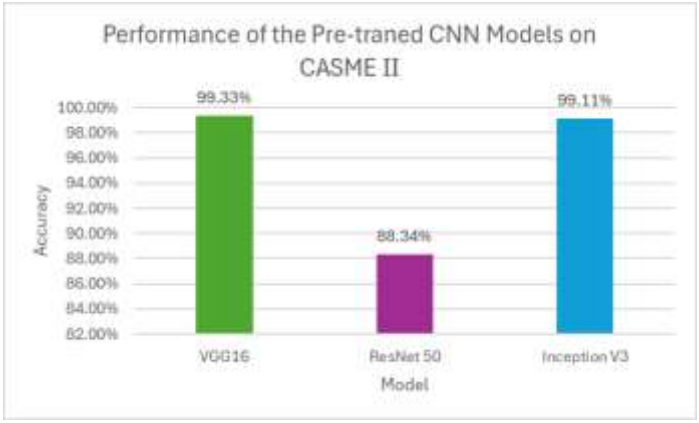


Figure 12. Performance graph for every pre-trained CNN model in CASME Ⅱ.

3.2.2. Proposed AI-Generated Dataset

In this experiment, our proposed AI-generated dataset is applied to train and test every pre-trained CNN model. There are 5 trials of experiments for VGG-16, ResNet 50 and Inception V3, obtained results from each trial were applied to obtain the average accuracy for every model.

The performance obtained from each model while trained using our proposed AI-generated dataset is presented in Figure 4. In this experiment, Inception V3 has the best performance with the highest accuracy among other models. It achieved an accuracy of 99.09% and outperformed VGG-16 and ResNet 50. The performance of Inception V3 is followed by ResNet 50 with 98.70% and VGG-16 with 98.66%. The is a minimum difference in accuracies between ResNet 50 and VGG-16 by 0.04%. When comparing with Inception V3, ResNet 50 has a difference in accuracy by 0.39% and the

 doi:10.20944/preprints202602.0339.v1

difference between VGG-16 and Inception V3 is 0.43%. The performance of every model is very close while using our proposed AI-generated dataset.

Table 4. Performance of every pre-trained CNN Model on proposed AI-generated dataset.

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	98.91	99.11	99.28	97.34	98.66	98.66
ResNet 50	98.17	98.58	99.15	98.25	99.34	98.70
Inception V3	98.73	99.32	99.42	98.64	99.36	99.09

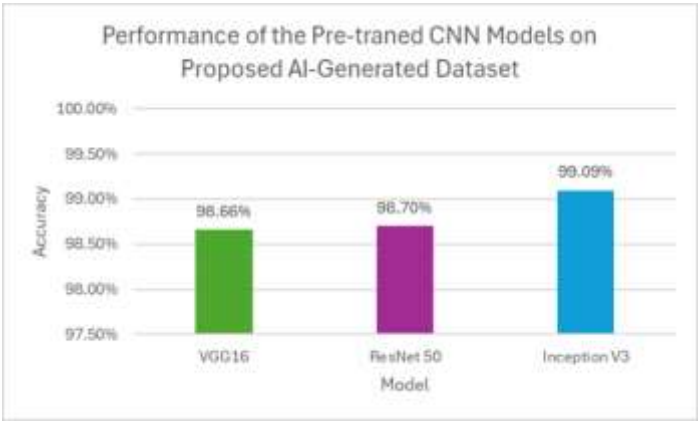


Figure 13. Performance graph for every pre-trained CNN model in proposed AI-generated dataset.

3.2.3. Training on CASME Ⅱ and Testing on Proposed AI-Generated Dataset

In this experiment, CASME Ⅱ is used to train the pre-trained CNN models and our proposed AI-generated dataset is used for testing the models. There are 5 trials of experiments for Inception V3, VGG-16 and ResNet 50, the results obtained from each trial were applied to obtain the average accuracy for every model.

Figure 5 represents the performance of every model while trained on CASME Ⅱ and tested using our proposed AI-generated dataset. From the observation, the best performed model in this experiment is Inception V3 with the highest accuracy of 99.47%. Meanwhile, ResNet 50 achieved an accuracy of 99.24%. It performed slightly worse than Inception V3 in this experiment by a minimal difference of 0.23%. On the other hand, VGG-16 has the lowest accuracy of 98.93% in this experiment. While compared with Inception V3, Inception V3 has a higher accuracy of 0.54% as compared to VGG-16. While compared ResNet 50 with VGG-16, there is a small difference of 0.31% between their accuracy.

Table 5. Performance of every pre-trained CNN model while trained on CASME Ⅱ and tested on proposed AI-generated dataset.

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	98.93	99.33	97.74	99.25	99.40	98.93
ResNet 50	98.31	99.35	99.55	99.40	99.60	99.24
Inception V3	99.36	99.56	99.54	99.43	99.46	99.47

 doi:10.20944/preprints202602.0339.v1

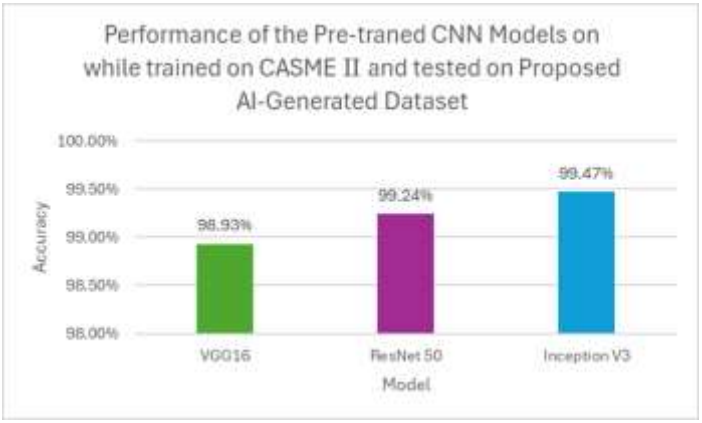


Figure 14. Performance graph for every pre-trained CNN model while trained on CASME Ⅱ and tested on proposed AI-generated dataset.

3.2.4. Training on Proposed AI-Generated Dataset and Testing on CASME Ⅱ

In this experiment, our proposed AI-generated dataset is employed to train the pre-trained CNN models and tested on CASME Ⅱ. There are 5 trials of experiments for Inception V3, ResNet 50, VGG-16, results obtained from each trial were applied to obtain the average accuracy for every model.

Based on Figure 6, the performance for each model while trained on our proposed AI-generated and CASME Ⅱ is used for testing in this experiment. Based on the results obtained, Inception V3 showed the best performance in this experiment. It achieved the highest accuracy with 99.48% and outperformed VGG-16 and ResNet 50. VGG-16 and ResNet 50 achieved 96.87% and 99.28%, respectively. For the comparison between Inception V3 and ResNet 50, a minimum difference of 0.20% is shown. Then, Inception V3 outperformed VGG-16 by a significant difference of 2.70%. VGG-16 has the worst performance as compared to Inception V3 and ResNet 50 in this experiment.

Table 6. Performance of every pre-trained CNN model while trained on proposed AI-generated dataset and tested on CASME Ⅱ.

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	94.79	97.55	98.37	94.44	98.74	96.78
ResNet 50	99.42	99.46	99.51	98.66	99.33	99.28
Inception V3	99.48	99.50	99.55	99.38	99.47	99.48

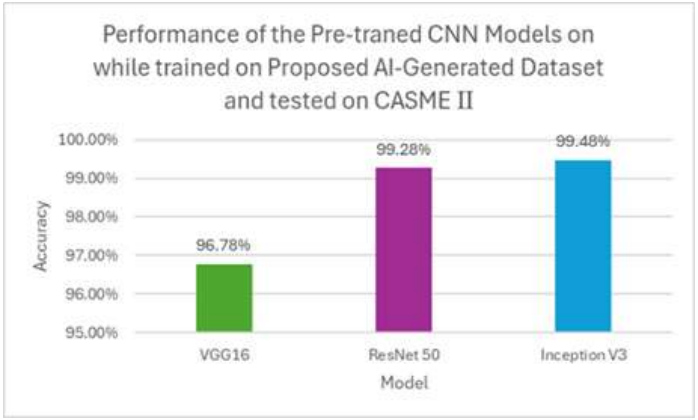


Figure 15. Performance graph for every pre-trained CNN model while trained on proposed AI-generated dataset and tested on CASME Ⅱ.

 doi:10.20944/preprints202602.0339.v1

3.3. Ablation Work

In this study, ablation work is conducted in order to improve the performance of the models training on both CASME Ⅱ and our proposed AI-generated dataset. There are two types of ablations works conducted in this study: applying data augmentation to the datasets and freezing 30% of the learnable layers of the pre-trained CNN models. The settings are described as follows:

- **Configuration 1:** Applying data augmentation to the datasets. Data augmentation is utilized to artificially increase the size, variety and variability of data sample for the pre-trained CNN models.
- **Configuration 2:** Transfer learning by freezing 30% of the layers. The first 50% of the learnable layers in the pre-trained CNN models, namely Inception V3, VGG-16 and ResNet 50 were frozen, and remaining layers were applied to train and test using the datasets.

3.3.1. Data Augmentation

Data augmentation is a method utilized in machine learning in order to synthetically expand the size, variety and variability of training data for machine learning models and deep learning models. It takes up multiple techniques including flipping, translating, rotating, scaling, zooming, adjusting the image brightness and adding noise to the images. Data augmentation provides tons of benefits including reducing overfitting problems, improving the size of dataset, improving model generalization and improving the performance of model training.

In this work, data augmentation is utilized to both CASME Ⅱ dataset and our proposed AI-generated dataset in order to synthetically increase the number of images and improve the scale of the datasets. This can help to provide better model learning conditions for the pre-trained CNN models employed in our work, even improve their performance.

CASME II

In this experiment, CASME Ⅱ is used to train and test every adopted pre-trained CNN model and utilized as the baseline dataset in order to compared with the other datasets. There are 5 trials of experiments for Inception V3, VGG-16, and ResNet 50. The results obtained from each trial were applied to obtain the average accuracy for every model.

The results obtained from every model while utilizing data augmentation technique and trained using CASME Ⅱ were tabulated in Table 7. Based on the table, it presents that ResNet 50 achieved 99.20% and outperformed VGG-16 and Inception V3 in this experiment. It is followed by Inception V3 with an accuracy of 98.51%, and Inception V3 is followed by VGG-16 with 97.23% accuracy. ResNet 50 performed better than Inception V3 by a difference of 0.69%. Meanwhile, VGG-16 has the lowest accuracy in this experiment. It has lower accuracy as compared to ResNet 50 by 1.97% and lower than Inception V3 by 1.28%.

Table 7. Performance of every pre-trained CNN model while trained and tested on CASME Ⅱ (Data Augmentation).

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	92.47	98.04	98.42	98.56	98.66	97.23
ResNet 50	99.07	99.18	99.39	99.50	98.87	99.20
Inception V3	94.69	99.19	99.60	99.55	99.52	98.51

 doi:10.20944/preprints202602.0339.v1

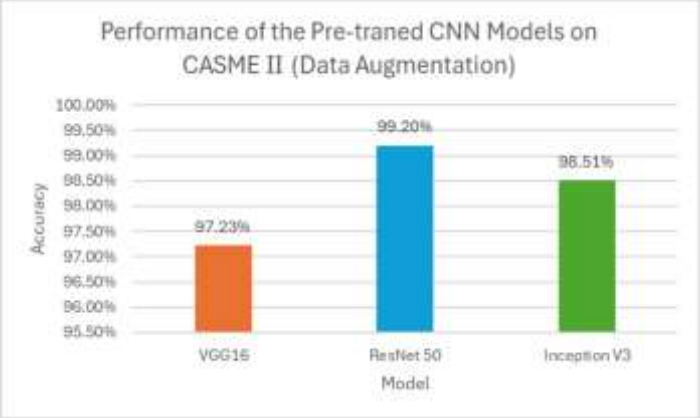


Figure 16. Performance graph for every pre-trained CNN model while trained and tested on CASME Ⅱ (Data Augmentation).

Proposed AI-Generated Dataset

Our proposed AI-generated dataset was utilized as the training and testing dataset for the pre-trained CNN models in this experiment. There are three different models, namely Inception V3, VGG-16 and ResNet 50. Each model performed a total of 5 trials of experiments and the achieved results from every trial were utilized to tabulate the average accuracy for every model used.

In Table 8, it shown each model's performance while trained using our proposed AI-generated dataset. In this experiment, ResNet 50 has outperformed VGG-16 and Inception V3 based on its best performance. ResNet 50 achieved 99.03% of accuracy, which is the highest accuracy as compared to the other models. In the meantime, Inception V3 and VGG-16 obtained 98.54% and 95.73% respectively. Inception V3 performance was fell behind of ResNet 50 with a gap of 0.49%. While comparing VGG-16 with ResNet 50, they showed a significant difference of 3.30%. VGG-16 has the worst performance in this experiment as compared to ResNet 50 and Inception V3.

Table 8. Performance of every pre-trained CNN model while trained on the proposed AI-generated dataset (Data Augmentation).

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	92.51	95.46	96.29	97.52	96.88	95.73
ResNet 50	98.77	98.90	99.13	99.22	99.15	99.03
Inception V3	95.61	99.00	99.27	99.33	99.47	98.54

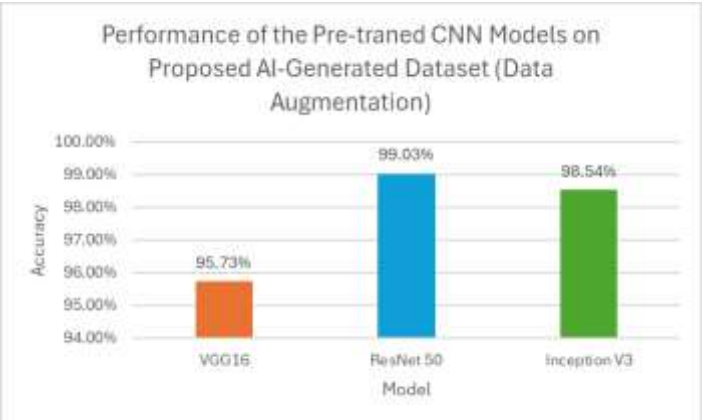


Figure 17. Performance graph for every pre-trained CNN model while trained on the proposed AI-generated dataset (Data Augmentation).

 doi:10.20944/preprints202602.0339.v1

Training on CASME II and testing on Proposed AI-generated Dataset

CASME Ⅱ was utilized to train the pre-trained CNN models in this experiment, and our proposed AI-generated dataset was used to test the models. A total of 5 trials were conducted for each model including VGG-16, ResNet 50 and Inception V3. Their performances were tabulated and used to obtain the average accuracy.

As shown in Table 9, the performance for every model while training on CASME Ⅱ and testing on our proposed AI-generated dataset was recorded. Based on the results recorded in the table, it shows that ResNet 50 has an outstanding performance with the highest accuracy of 99.54%. VGG-16 and Inception V3 were outperformed by ResNet 50 in this experiment. Inception V3 performed slightly poorer as compared to ResNet 50, its accuracy is lower than ResNet 50 by 0.05%. While VGG-16 compared to ResNet 50, there is a difference between both models by 1.12%. Although VGG-16 has reached the state-of-the-art performance, it still performs the worst as compared to ResNet 50 and Inception V3.

Table 9. Performance of every pre-trained CNN model while trained on CASME Ⅱ and tested on the proposed AI-generated dataset (Data Augmentation).

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	98.32	97.82	98.86	98.16	98.94	98.42
ResNet 50	99.63	99.51	99.40	99.59	99.58	99.54
Inception V3	99.35	99.42	99.44	99.55	99.67	99.49

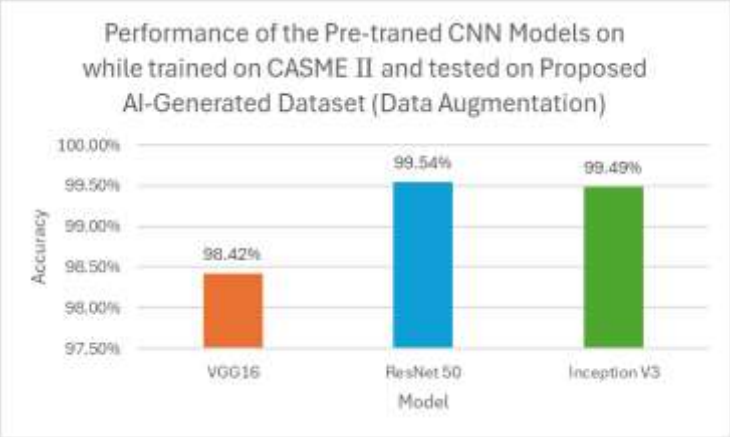


Figure 18. Performance graph for every pre-trained CNN model while trained on CASME Ⅱ and tested on the proposed AI-generated dataset (Data Augmentation).

Training on Proposed AI-generated Dataset and testing on CASME II

In this experiment, data augmentation was applied to our proposed AI-generated dataset, and it is used to train the pre-trained CNN models while CASME Ⅱ was applied for testing. A total of 5 trials were conducted for each model including VGG-16, ResNet 50 and Inception V3. Their performances were tabulated and used to obtain the average accuracy.

The performance of every model while trained on our proposed AI-generated and CASME Ⅱ is used for testing in this experiment is recorded in Table 10. Inception V3 achieved 99.11% of accuracy and shows the best performance in this experiment. ResNet 50 obtained an outstanding performance with 99.07% accuracy. It has outperformed by Inception V3 with a minimum difference of 0.04%. Then, VGG-16 obtained an accuracy of 98.38% and it outperformed by Inception V3 by a difference

 doi:10.20944/preprints202602.0339.v1

of 0.73%. In this experiment, VGG-16 has the worst performance as compared to Inception V3 and ResNet 50.

Table 10. Performance of every pre-trained CNN model while trained on the proposed AI-generated dataset and tested on CASME Ⅱ (Data Augmentation).

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	97.37	98.39	98.53	98.48	99.12	98.38
ResNet 50	98.12	99.22	99.30	99.35	99.35	99.07
Inception V3	99.36	97.79	99.44	99.44	99.50	99.11

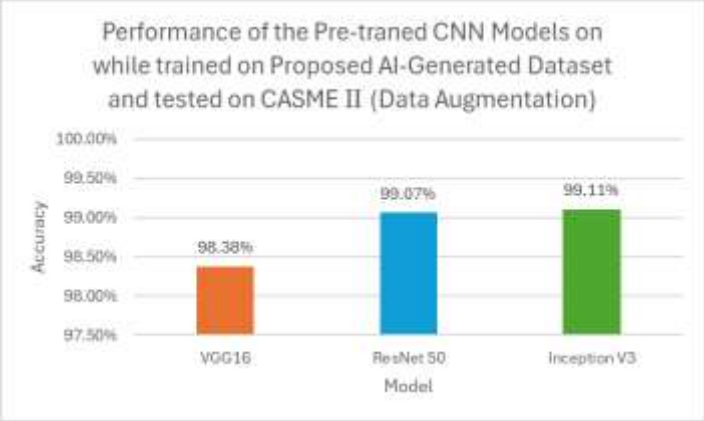


Figure 19. Performance graph for every pre-trained CNN model while trained on the proposed AI-generated dataset and tested on CASME Ⅱ (Data Augmentation).

3.3.2. Freezing Layer 30%

Freezing layer is commonly utilized in pre-trained CNN models to prevent certain layers from updating their weights during training process. The objective of this method is to apply the knowledge learned by the model from a large dataset like ImageNet to a newer and smaller dataset. The benefits of applying freezing layer technique including saving computational resources, preserving learned features and preventing overfitting problem.

Freezing layer technique is utilized to every adopted pre-trained CNN model in this study. The first 30% learnable layers of every pre-trained CNN model were frozen in this study. The first 30% learnable layers were frozen because it is the most suitable settings for the pre-trained CNN models in this work after many trials and errors with different freezing layer settings. Freezing layer technique is utilized on the pre-trained CNN models in this work to improve the learning performance of the models.

CASME II

CASME Ⅱ is applied for the training and testing of every pre-trained CNN model and acts as the baseline dataset in order to compared with the other datasets. There are three pre-trained models, namely Inception V3, ResNet 50 and VGG-16. Each model performed a total of 5 trials of experiments, and their average accuracies were obtained.

In Table 11, it tabulated the performance of every model while trained and tested on CASME Ⅱ. Based on the obtained results, Inception V3 shows a state-of-the-art performance with the highest accuracy of 98.99% in this experiment. VGG-16 and ResNet 50 achieved 96.52% and 93.21% respectively. Both VGG-16 and ResNet 50 were outperformed by Inception V3. While Inception V3 is compared with VGG-16, both models show a significant difference of 2.47%. On the other hand, the difference between Inception V3 and ResNet 50 is more significant. There is a 5.78% gap between

 doi:10.20944/preprints202602.0339.v1

these two models. In this experiment, ResNet 50 presented the poorest performance among the other two models.

Table 11. Performance of every pre-trained CNN model while trained and tested on CASME Ⅱ (Freezing Layer 30%).

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	98.65	95.78	92.04	97.57	98.54	96.52
ResNet 50	86.76	90.65	93.92	96.65	98.05	93.21
Inception V3	98.38	99.40	98.20	99.40	99.56	98.99

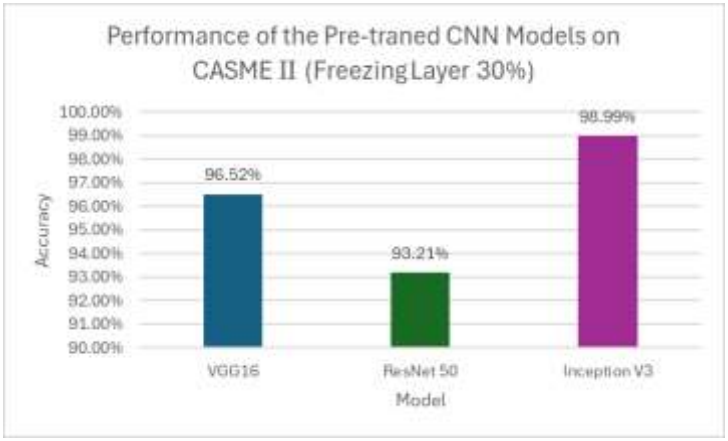


Figure 20. Performance graph for every pre-trained CNN model while trained and tested on CASME Ⅱ (Freezing Layer 30%).

Proposed AI-generated Dataset

In this experiment, our proposed AI-generated dataset is used on the training and testing of the pre-trained CNN models, namely Inception V3, ResNet 50 and VGG-16. Every model performed 5 trials of experiments and the results obtained from each trial is utilized to obtain the average accuracy.

As presented in Table 12, the performances of every model while trained and tested on our proposed AI-generated dataset is recorded. From the observation, the best performed model in this experiment is Inception V3 with the highest accuracy of 99.14%. Then, it followed by ResNet 50 with an accuracy of 97.50%. ResNet 50 is followed by VGG-16 with an accuracy of 97.21%. The performances of both ResNet 50 and VGG-16 are very close with a minor difference of 0.29% accuracy. While Inception V3 is compared to both ResNet 50 and VGG-16, it shows major differences. The difference between Inception V3 and ResNet 50 is 1.64%. Then, the difference between Inception V3 and VGG-16 is 1.93%.

Table 12. Performance of every pre-trained CNN model while trained and tested on the proposed AI-generated dataset (Freezing Layer 30%).

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	92.65	95.69	99.02	99.36	99.34	97.21
ResNet 50	96.44	97.56	97.72	97.88	97.90	97.50
Inception V3	98.87	99.46	99.50	99.57	98.31	99.14

 doi:10.20944/preprints202602.0339.v1

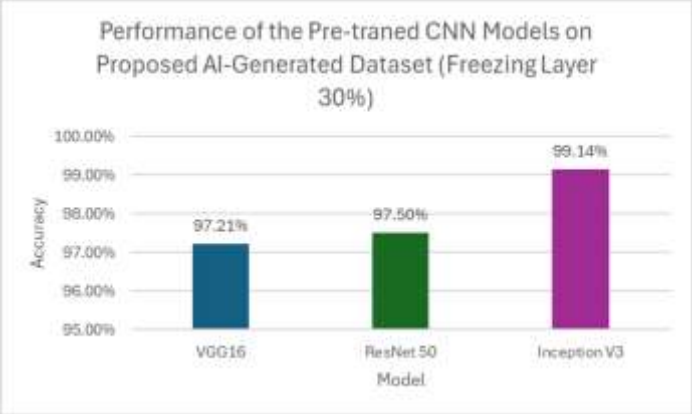


Figure 21. Performance graph for every pre-trained CNN model while trained and tested on the proposed AI-generated dataset (Freezing Layer 30%).

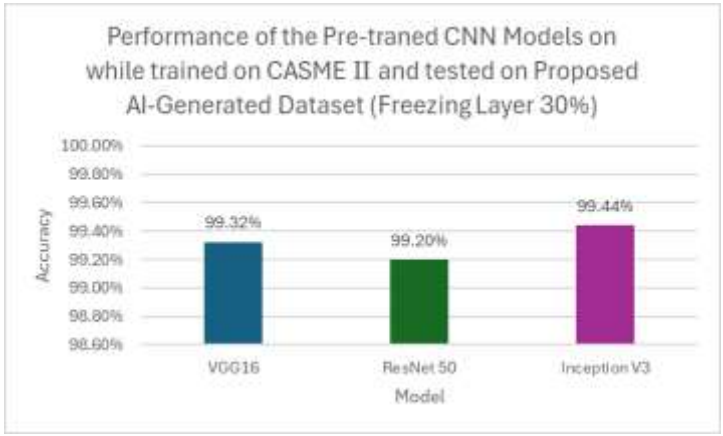
Training on CASME II and testing on Proposed AI-generated Dataset

In this experiment, the pre-trained CNN models were trained on the CASME Ⅱ and tested on our proposed AI-generated dataset. There are 5 trials of experiments for VGG-16, ResNet 50 and Inception V3, results taken from each trial were employed to achieve the average accuracy for every model.

In Table 13, it presents the performance of each model while trained using CASME Ⅱ and tested using our proposed AI-generated dataset. Based on the table, Inception V3 has the best performance with the highest accuracy of 99.44%. It has successfully surpassed ResNet 50 and VGG-16. In this experiment, VGG-16 achieved 99.32% and its performance is behind of Inception V3. There is a 0.12% difference between Inception V3 and VGG-16. Moreover, ResNet 50 achieved 99.20% and its accuracy is lower than Inception V3 by 0.24%. VGG-16 performed slightly better than ResNet 50 by 0.12%. All the models have slightly difference accuracy in this experiment, and their performances are very similar.

Table 13. Performance of every pre-trained CNN model while trained on CASME Ⅱ and tested on the proposed AI-generated dataset (Freezing Layer 30%).

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	99.15	99.29	99.07	99.52	99.59	99.32
ResNet 50	98.06	99.39	99.47	99.52	99.55	99.20
Inception V3	99.30	99.41	99.47	99.50	99.53	99.44



 doi:10.20944/preprints202602.0339.v1

Figure 22. Performance graph for every pre-trained CNN model while trained on CASME Ⅱ and tested on the proposed AI-generated dataset (Freezing Layer 30%).

Training on Proposed AI-generated Dataset and testing on CASME II

Our proposed AI-generated dataset is employed as the training dataset and CASME Ⅱ is employed as the testing dataset in this experiment. There are 5 trials of experiments conducted by every model including Inception V3, VGG-16 and ResNet 50 and obtained results from every trial were utilized to obtain the average accuracy for every model.

Based on Table 14, the performance for each model while trained on our proposed AI-generated and CASME Ⅱ is used for testing in this experiment. Based on the outcomes obtained, Inception V3 achieved the best results with an accuracy of 99.53% in this experiment. It is followed by ResNet 50 with an accuracy of 98.67% and ResNet 50 is followed by VGG-16 with an accuracy of 96.52%. Inception V3 outperformed ResNet 50 by 0.86% and significantly outperformed VGG-16 by 3.01%. In this experiment, ResNet 50 performed slightly poorer as compared to Inception V3. Meanwhile, VGG-16 has the worst performance in this experiment although it has presented a state-of-the-art performance.

Table 14. Performance of every pre-trained CNN model while trained on the proposed AI-generated dataset and tested on CASME Ⅱ (Freezing Layer 30%).

Models	Accuracy (%)					
	1	2	3	4	5	Average
VGG-16	98.57	93.67	98.24	93.85	98.26	96.52
ResNet 50	97.90	99.29	97.71	99.20	99.27	98.67
Inception V3	99.51	99.52	99.50	99.55	99.57	99.53

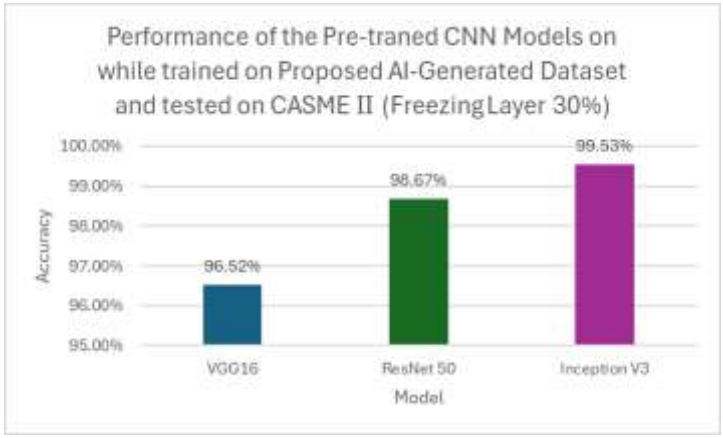


Figure 23. Performance graph for every pre-trained CNN model while trained on the proposed AI-generated dataset and tested on CASME Ⅱ (Freezing Layer 30%).

3.4. Performance Comparison with Other Works

In this section, the performance comparison of the proposed approaches with some state-of-the-art approaches was performed. All the existing methods were evaluated based on the existing dataset and their own proposed dataset. Therefore, it posed some difficulties to directly compare with the other methods in term of the structure of the database, preprocessing of the dataset, labels of the facial expressions, as well as the feature extraction approaches applied to the dataset. The comparison might not be valuable, but the comparison is still able to demonstrate that our proposed approaches have the capability to achieve competitive and state-of-the-art results compared to other methods compared.

 doi:10.20944/preprints202602.0339.v1

In Table 15, it tabulates the comparison of the performance of the proposed works with the other state-of-the-art approaches for facial expression recognition in classification based on existing dataset, CASME Ⅱ and their own proposed dataset. A pre-trained ResNet 50 was deployed in Zhang and Shen's work while using CASME Ⅱ as their dataset, The work proposed by Zhang and Shen achieved 98.41%. Meanwhile, the approach proposed by Zhi et al. obtained 97.60% of accuracy while using CASME Ⅱ to train their proposed 3D-CNN. Sun et al. introduced a SVM paired with knowledge distillation in their work and their approach obtained 72.60% accuracy. In Wang et al. proposed approach, a Transferring Long-term Convolutional Neural Network (TLCNN) was trained on CASME Ⅱ and the model achieved an accuracy of 69.10%. In our proposed work, a pre-trained VGG-16 was adopted, and it also trained on CASME Ⅱ. Our proposed pre-trained VGG-16 achieved 99.33% and it has outperformed the other approaches above. Our proposed VGG-16 achieved slightly better accuracy as compared to the works by Zhang and Shen and Zhi et al. While compared with Sun et al.'s and Wang et al.'s works, there are a significant difference in performance between our proposed approach and their approaches.

Furthermore, Wang et al. proposed a LAUN improved StarGAN to generate an AI-generated dataset based on MMI database and employed a VGG-16 to evaluate their dataset performance. The performance obtained by their work is 98.30% in accuracy. In our proposed approach, Inception V3 with 30% freezing layers was trained using our own proposed AI-generated dataset. It managed to obtain an accuracy of 99.14% and it has better results as compared to the work proposed by Wang et al. Besides, Fan et al. adopted a modified VGG-Net with CGAN approach. Their modified VGG-Net was trained on Oulu dataset generated by their proposed CGAN and tested using generated CK+ dataset. Their approach obtained an accuracy of 90.37%. Meanwhile, there are two of our proposed approaches which have similar methods to Fan et al's work. The first approach is ResNet 50 applied with data augmentation while trained using CASME Ⅱ and tested using our proposed AI-generated dataset. This approach achieved an outstanding performance with an accuracy of 99.54%. Then, the second proposed approach is an Inception V3 applied with 30% freezing layers while trained on our proposed AI-generated dataset and tested on CASME Ⅱ. The performance obtained by this proposed approach is 99.53% in accuracy. Both of our proposed approaches have significantly better performance as compared to the work by Fan et al.

Table 15. Performance assessment of the proposed works against other state-of-the-art methods.

Model	Dataset	Accuracy
3D-CNN [26]	CASME II	97.60%
TLCNN [27]	CASME II	69.10%
SVM with Knowledge Distillation [38]	CASME II	72.60%
Pre-trained ResNet 50 [17]	CASME II	98.41%
Modified VGG-Net with CGAN [39]	Trained on AI-generated Oulu and test on AI-generated CK+	90.37%
VGG-16 with LAUN improved StarGAN [3]	AI-generated MMI	98.30%
Proposed Pre-trained VGG-16	CASME II	99.33%
Proposed Pre-trained ResNet 50 (with Data Augmentation)	Training on CASME II and Tested on Proposed AI-generated Dataset	99.54%
Proposed Pre-trained Inception V3 (with 30% Freezing Layers)	Training on Proposed AI-generated Dataset and Tested on CASME II	99.53%

 doi:10.20944/preprints202602.0339.v1

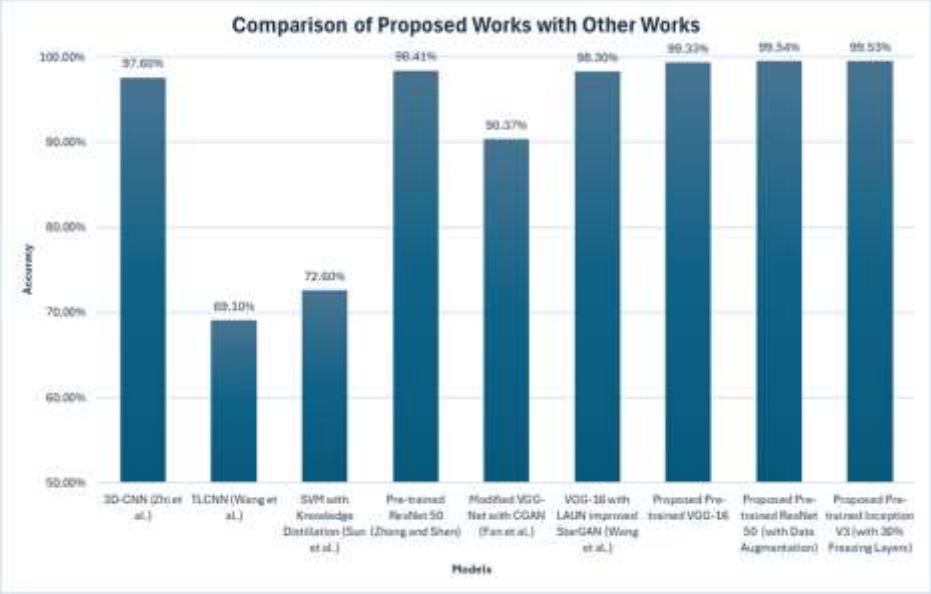


Figure 24. Graph of performance comparison between every proposed approach and other state-of-the-art methods.

4. Conclusions

In this work, it is mainly focused on using a generative model to create an AI-generated facial expression dataset to perform facial expression recognition. There are four different experiments using four different dataset combinations had been conducted: pre-trained CNN models trained and tested on CASME Ⅱ, pre-trained CNN models trained and tested on the proposed AI-generated dataset, pre-trained CNN models trained on CASME Ⅱ and tested on the proposed AI-generated dataset, and pre-trained CNN models trained on the proposed AI-generated dataset and tested on CASME Ⅱ. All the experiments were conducted on the same pre-trained CNN models including VGG-16, ResNet 50, and Inception V3. The experiments focused on using the pre-trained CNN models to evaluate the performance of the datasets.

In conclusion, all the proposed architectures managed to achieve outstanding performances in all the performed experiments. Based on the obtained results, Inception V3 trained on the proposed AI-generated dataset and tested using CASME Ⅱ achieved the highest accuracy of 99.48% while compared with VGG-16 and ResNet 50 that have the same experimental settings. During the application of data augmentation approach, a VGG-16 model was trained on CASME Ⅱ and tested on the proposed AI-generated dataset. This proposed approach has the highest accuracy compared to other approaches proposed with an accuracy of 99.54%. While 30% freezing layers method is utilized, Inception V3 trained on the proposed AI-generated dataset and tested using CASME Ⅱ is the best performed approach with the highest accuracy of 99.53%. Apart from that, our proposed approaches were also compared with several existing facial expression recognition methods and have outperformed other proposed methods. Although this study presented promising performance but there are still available many potentials and possibilities to be applied in future work including adopting different types of generative models, adopting different types of deep learning models and trying different types of datasets to conduct the same work.

5. Future Works

Although the proposed approaches present promising results, it still has space for some potential future exploration that can be applied in facial expression recognition. There are some limitations and problems faced including time consuming for the image generation by the generative model, computational limitations for heavy load tasks, and quality of the generated image. Furthermore, the current AI-created facial expression dataset is based on the original CASME Ⅱ. The

 doi:10.20944/preprints202602.0339.v1

images obtained from the participants of this dataset are Asians. It lacks subjects with different ethnicity, skin tone, age, and cultural backgrounds. Moreover, this study only focused on using a few pre-trained CNN models to conduct the evaluation. Hence, more suitable deep learning architectures that capable of conducting facial expression recognition can be adopted and provide better performances as compared to the current approaches. Particularly, these limitations and problems have inspired the following future directions in order to be addressed:

- Different types of generative models such as Generative Adversarial Network (GAN) and Variational Autoencoder (VAE) can be implemented in future work. This approach is to compare the quality of image generation by different types of generative models.
- Utilize image processing such as Action Units (AUs), Histogram of Oriented Gradients (HOG), Laplacian filters to enhance the quality of the images and make the features to be more obvious. These approaches will help the models to have better learning of the features in the images and enhance their accuracy performance.
- Different types of deep learning architecture include different variants of pre-trained CNN models, hybrid CNN model with attention mechanisms, vision transformers (ViT), hybrid CNN-Transformer model, 3DCNN, and CNN with Long Short-Term Memory (LSTM). These architectures were suggested for the future work in order to perform because they might improve the performance of FER.
- Lastly, the proposed method can employ different types of facial expression datasets rather than only CASME II. This is to prove that our proposed method is not only workable on CASME II, but it also works while replace with different datasets.

Author Contributions: Conceptualization, J.J.H.; Data Curation, J.J.H.; Methodology, J.J.H.; Software, J.J.H.; Validation, J.J.H.; Writing – Original Draft, J.J.H.; Project Administration, W.H.K. and Y.H.P.; Supervision, W.H.K. and Y.H.P.; Writing – Review & Editing, H.Y.Y. and F.C.L.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data in this study are available upon request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
AUs	Action Units
CASME II	The Chinese Academy of Sciences Micro-expression 2
CNN	Convolutional Neural Network
FER	Facial Expression Recognition
GAN	Generative Adversarial Network
HOG	Histogram of Oriented Gradients
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
NLP	Natural Language Processing
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TLCNN	Transferring Long-term Convolutional Neural Network
VAE	Variational Autoencoder

 doi:10.20944/preprints202602.0339.v1

References

1. Ekman, P.; Friesen, W.V. Constants across Cultures in the Face and Emotion. *Journal of Personality and Social Psychology* **1971**, *17*, 124–129, doi:10.1037/h0030377.
2. Ekman, P. ; Dalai, L. Emotional Awareness: Overcoming the Obstacles to Psychological Balance and Compassion; a Conversation between the Dalai Lama and Paul Ekman; 1. ed.; Times Books, Henry Holt and Co: New York, 2008; ISBN 9780805087123.
3. Wang, X.; Gong, J.; Hu, M.; Gu, Y.; Ren, F. Laun Improved Stargan for Facial Emotion Recognition. *IEEE Access* **2020**, *8*, 161509–161518, doi:10.1109/ACCESS.2020.3021531.
4. Fan, Y.; Lam, J.C.K.; Li, V.O.K. Unsupervised Domain Adaptation with Generative Adversarial Networks for Facial Emotion Recognition. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data); IEEE: Seattle, WA, USA, December 2018; pp. 4460–4464.
5. Zhou, J.; Sun, S.; Xia, H.; Liu, X.; Wang, H.; Chen, T. ULME-GAN: A Generative Adversarial Network for Micro-Expression Sequence Generation. *Appl Intell* **2024**, *54*, 490–502, doi:10.1007/s10489-023-05213-z.
6. Chen, Y.; Zhong, C.; Huang, P.; Cai, W.; Wang, L. Improving Micro-Expression Recognition Using Multi-Sequence Driven Face Generation. In Proceedings of the ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); IEEE: Hyderabad, India, April 6 2025; pp. 1–5.
7. Tariq, U.; Yang, J.; Huang, T.S. Maximum Margin GMM Learning for Facial Expression Recognition. In Proceedings of the 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG); April 2013; pp. 1–6.
8. Wang, Z.; Wang, S.; Ji, Q. Capturing Complex Spatio-Temporal Relations among Facial Muscles for Facial Expression Recognition. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition; June 2013; pp. 3422–3429.
9. Nasri, M.A.; Hmani, M.A.; Mtibaa, A.; Petrovska-Delacretaz, D.; Slima, M.B.; Hamida, A.B. Face Emotion Recognition from Static Image Based on Convolution Neural Networks. In Proceedings of the 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP); IEEE: Sousse, Tunisia, September 2020; pp. 1–6.
10. Lasri, I.; Solh, A.R.; Belkacemi, M.E. Facial Emotion Recognition of Students Using Convolutional Neural Network. In Proceedings of the 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS); IEEE: Marrakech, Morocco, October 2019; pp. 1–6.
11. Pranav, E.; Kamal, S.; Satheesh Chandran, C.; Supriya, M.H. Facial Emotion Recognition Using Deep Convolutional Neural Network. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS); IEEE: Coimbatore, India, March 2020; pp. 317–320.
12. Verma, A.; Singh, P.; Rani Alex, J.S. Modified Convolutional Neural Network Architecture Analysis for Facial Emotion Recognition. In Proceedings of the 2019 International Conference on Systems, Signals and Image Processing (IWSSIP); IEEE: Osijek, Croatia, June 2019; pp. 169–173.
13. Luo, Y.; Wu, J.; Zhang, Z.; Zhao, H.; Shu, Z. Design of Facial Expression Recognition Algorithm Based on Cnn Model. In Proceedings of the 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA); IEEE: Shenyang, China, January 29 2023; pp. 580–583.
14. Naik, N.; Mehta, M.A. An Improved Method to Recognize Hand-over-Face Gesture Based Facial Emotion Using Convolutional Neural Network. In Proceedings of the 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT); IEEE: Bangalore, India, July 2020; pp. 1–6.
15. Xie, S.; Hu, H. Facial Expression Recognition with FRR-CNN. *Electronics Letters* **2017**, *53*, 235–237, doi:10.1049/el.2016.4328.
16. Shao, J.; Qian, Y. Three Convolutional Neural Network Models for Facial Expression Recognition in the Wild. *Neurocomputing* **2019**, *355*, 82–92, doi:10.1016/j.neucom.2019.05.005.
17. Haiwei, Z.; Zhuofan, S. Micro-Expression Recognition Based on Residual Network. In Proceedings of the 2023 IEEE 5th International Conference on Civil Aviation Safety and Information Technology (ICCASIT); IEEE: Dali, China, October 11 2023; pp. 772–775.
18. Li, J.; Zhang, D.; Zhang, J.; Zhang, J.; Li, T.; Xia, Y.; Yan, Q.; Xun, L. Facial Expression Recognition with Faster R-Cnn. *Procedia Computer Science* **2017**, *107*, 135–140, doi:10.1016/j.procs.2017.03.069.

 doi:10.20944/preprints202602.0339.v1

19. Madupu, R.K.; Kothapalli, C.; Yarra, V.; Harika, S.; Basha, C.Z. Automatic Human Emotion Recognition System Using Facial Expressions with Convolution Neural Network. In Proceedings of the 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA); IEEE: Coimbatore, India, November 5 2020; pp. 1179–1183.
20. John, A.; Mc, A.; Ajayan, A.S.; Sanoop, S.; Kumar, V.R. Real-Time Facial Emotion Recognition System with Improved Preprocessing and Feature Extraction. In Proceedings of the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT); IEEE: Tirunelveli, India, August 2020; pp. 1328–1333.
21. Mukhopadhyay, M.; Dey, A.; Shaw, R.N.; Ghosh, A. Facial Emotion Recognition Based on Textural Pattern and Convolutional Neural Network. In Proceedings of the 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON); IEEE: Kuala Lumpur, Malaysia, September 24 2021; pp. 1–6.
22. Georgescu, M.-I.; Ionescu, R.T.; Popescu, M. Local Learning with Deep and Handcrafted Features for Facial Expression Recognition. *IEEE Access* **2019**, *7*, 64827–64836, doi:10.1109/ACCESS.2019.2917266.
23. Xiao, H.; Li, W.; Zeng, G.; Wu, Y.; Xue, J.; Zhang, J.; Li, C.; Guo, G. On-Road Driver Emotion Recognition Using Facial Expression. *Applied Sciences* **2022**, *12*, 807, doi:10.3390/app12020807.
24. Meng, Z.; Liu, P.; Cai, J.; Han, S.; Tong, Y. Identity-Aware Convolutional Neural Network for Facial Expression Recognition. In Proceedings of the 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017); IEEE: Washington, DC, DC, USA, May 2017; pp. 558–565.
25. Fernandez, P.D.M.; Pena, F.A.G.; Ren, T.I.; Cunha, A. Feratt: Facial Expression Recognition with Attention Net. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); IEEE: Long Beach, CA, USA, June 2019; pp. 837–846.
26. Zhi, R.; Xu, H.; Wan, M.; Li, T. Combining 3d Convolutional Neural Networks with Transfer Learning by Supervised Pre-Training for Facial Micro-Expression Recognition. *IEICE Trans. Inf. & Syst.* **2019**, *E102.D*, 1054–1064, doi:10.1587/transinf.2018EDP7153.
27. Wang, S.-J.; Li, B.-J.; Liu, Y.-J.; Yan, W.-J.; Ou, X.; Huang, X.; Xu, F.; Fu, X. Micro-Expression Recognition with Small Sample Size by Transferring Long-Term Convolutional Neural Network. *Neurocomputing* **2018**, *312*, 251–262, doi:10.1016/j.neucom.2018.05.107.
28. Mohammadpour, M.; Khaliliardali, H.; Hashemi, S.Mohammad.R.; AlyanNezhadi, Mohammad.M. Facial Emotion Recognition Using Deep Convolutional Networks. In Proceedings of the 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI); IEEE: Tehran, December 2017; pp. 0017–0021.
29. Li, J.; Jin, K.; Zhou, D.; Kubota, N.; Ju, Z. Attention Mechanism-Based CNN for Facial Expression Recognition. *Neurocomputing* **2020**, *411*, 340–350, doi:10.1016/j.neucom.2020.06.014.
30. Jin, X.; Jin, Z. MiniExpNet: A Small and Effective Facial Expression Recognition Network Based on Facial Local Regions. *Neurocomputing* **2021**, *462*, 353–364, doi:10.1016/j.neucom.2021.07.079.
31. Gong, W.; Qian, Y.; Zhou, W.; Leng, H. Enhanced Spatial-Temporal Learning Network for Dynamic Facial Expression Recognition. *Biomedical Signal Processing and Control* **2024**, *88*, 105316, doi:10.1016/j.bspc.2023.105316.
32. Yan, W.-J.; Li, X.; Wang, S.-J.; Zhao, G.; Liu, Y.-J.; Chen, Y.-H.; Fu, X. Casme II: An Improved Spontaneous Micro-Expression Database and the Baseline Evaluation. *PLoS ONE* **2014**, *9*, e86041, doi:10.1371/journal.pone.0086041.
33. Yan, W.-J.; Wu, Q.; Liu, Y.-J.; Wang, S.-J.; Fu, X. CASME Database: A Dataset of Spontaneous Micro-Expressions Collected from Neutralized Faces. In Proceedings of the 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG); IEEE: Shanghai, China, 2013; pp. 1–7.
34. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324, doi:10.1109/5.726791.
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Las Vegas, NV, USA, June 2016; pp. 770–778.

 doi:10.20944/preprints202602.0339.v1

36. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* **2014**.
37. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Las Vegas, NV, USA, June 2016; pp. 2818–2826.
38. Sun, B.; Cao, S.; Li, D.; He, J.; Yu, L. Dynamic Micro-Expression Recognition Using Knowledge Distillation. *IEEE Trans. Affective Comput.* **2022**, *13*, 1037–1043, doi:10.1109/TAFFC.2020.2986962.
39. Fan, Y.; Lam, J.C.K.; Li, V.O.K. Unsupervised Domain Adaptation with Generative Adversarial Networks for Facial Emotion Recognition. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data); IEEE: Seattle, WA, USA, December 2018; pp. 4460–4464.