applied sciences

MDPI

Article

AI Creation of Facial Expression Database for Advanced Emotion Recognition Using Diffusion Model and Pre-Trained CNN Models

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, Melaka 75450, Malaysia; 1181200307@student.mmu.edu.my
[2] Centre for Advanced Analytics, CoE for Artificial Intelligence, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, Melaka 75450, 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 from 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, VGG-16 with data augmentation applied is trained on CASME II and tested on the proposed AI-generated dataset, and Inception V3 with 30% freezing layers method is trained on the proposed AI-generated dataset and tested using CASME II. These all successfully achieved state-of-the-art performance. The data augmentation and freezing layers approaches significantly improved the performance of the models. Our proposed approaches achieved state-of-the-art performance and outperformed most of the existing state-of-the-art approaches benchmarked in this study.

Keywords: convolutional neural networks; deep learning; diffusion model; facial expression recognition; micro-expression

Academic Editor: Douglas O'Shaughnessy

Received: 3 February 2026
Revised: 5 March 2026
Accepted: 6 March 2026
Published: 13 March 2026

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 was first explored by Dr. Ekman and Friesen in 1971 [1]. They proposed a Facial Action Code System (FACS) to distinguish between 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-expressions are recognizable and easily captured by the naked eye. They can represent an individual genuine feeling or an acted emotion. Both micro- and macro-expressions allow for effective communication and help

to understand a person's current feelings. Common facial expressions include smiling, 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 are often included in the FER methodology: face detection, feature extraction and facial expression classification. Even though FER is commonly used by researchers, it still faces some critical issues which affect its effectiveness and efficiency. For example, the current existing facial expression datasets that are frequently utilized to conduct studies on FER suffer from 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 expression databases, generative models have been used in some studies. Generative models are a kind of machine learning model that have the ability to produce new data that is similar to the data it was trained on. Generative models focus on understanding the fundamental distribution of the input data, in contrast to discriminative models, which emphasize distinguishing between categories or predicting labels. Once a generative model is trained, it can generate completely new, synthetic data that share similar features with the original dataset. Generative models can be used to generate data such as images, videos, audio, texts, text-to-image, text-to-video, image captioning, and 3D modelling, among others. There are many types of generative models, including the Generative Adversarial Network (GAN), Variational Autoencoder (VAE), autoregressive model, and diffusion model. 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 emotion images for every emotion. Reference [4] employed a conditional generative adversarial network (CGAN) in their work. This 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 the transfer learning approach was utilized to train the proposed generator network. A multi-sequence-based micro-expression (ME) generation approach for ME recognition is introduced by [6]. Another means of facial expression recognition is proposed by [7], using maximum margin Gaussian Mixture Models (GMM). By recording the complex spatio-temporal relations between face muscles, ref. [8] presented a method for recognizing facial expressions. In order to observe children's emotions, ref. [9] proposed an unique approach using GenAI and facial muscle activation.

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 recent works. Pre-trained CNN models are deep learning models which have been trained using a large dataset like ImageNet. This eliminates the need to train the model from scratch, which is time-consuming and uses a large amount of resources. It is much more efficient and leads to better performance. Pre-trained CNN models are commonly used in applications such as object detection, face recognition, image classification, and medical imaging, among others. There are several popular pre-trained CNN models that can be adopted in this work, including VGG, ResNet, AlexNet, SqueezeNet, GoogleNet, Inception, and MobileNet. Xception CNN paired with a K-fold cross-validation technique was adopted

by [10]. In order to identify students' moods based on their facial expressions, ref. [11] proposed a CNN model. Furthermore, ref. [12] employed a two-dimensional (2D) CNN to identify facial emotions and evaluated the model by using a self-collected facial emotion database that contains five emotions. In order to enhance the performance of the CNN model, a unique Venturi Architecture that contains six hidden layers and one output layer for CNN was proposed by [13]. Moreover, a CNN model-derived facial expression recognition algorithm was proposed by [14]. In addition, CNN was used in Naik and Mehta's Hand-over-Face Gesture-based Facial Emotion Method (HFG_FERM) [15]. A new model, called feature redundancy-reduced convolutional neural network (FRR-CNN), has been proposed by [16] to recognize facial expressions and experiments were conducted using CK+ and JAFFE. In order to recognize facial expressions in reality, ref. [17] employed three convolution neural network models, including Light CNN, dual-branch CNN and pre-trained CNN. By employing a residual network, ref. [18] presented a means of micro-expression identification. For the purpose of recognizing facial expressions, ref. [19] presented a Fast Regions with Convolutional Neural Network Features (Faster R-CNN). In addition, ref. [20] 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). Pre-trained architectures were adopted in [21] for facial expression recognition. Furthermore, a multi-modal fusion network (M^3Enet) was proposed by [22] for micro-expression recognition.

Using enhanced feature extraction and preprocessing techniques, ref. [23] presented a real-time face emotion recognition system. Based on textural pattern and convolutional neural networks, ref. [24] presented a method for facial emotion recognition. In addition, a facial expression recognition method using local learning with deep and handcrafted features was presented [25]. Using facial expressions, ref. [26] demonstrate an emotion identification system for drivers while driving. This experiment uses FERDERnet to recognize the facial expressions of drivers while they are driving. Additionally, an Identity-Aware CNN (IACNN) was created by [27] for facial expression recognition. Attention Net (FERAtt) was proposed by [28] for facial expression recognition. Two methods were implemented in their experiment: a model with attention and classification (FERAtt + Cls), and a model with attention, classification, and representation (FERAtt + Rep + Cls). Moreover, ref. [29] used a 3D CNN and transfer learning to recognize facial micro-expressions. Reference [30] adopted (TLCNN) model to recognize micro-expressions (MEs) with a tiny sample size. Furthermore, deep convolution networks were proposed by [31] for facial emotion identification by applying normalization, Action Units (AUs) and CNN. For FER, ref. [32] presented an attention mechanism-based CNN. The micro-movements of the faces are captured, and the textural information of the image is obtained using LBP features. Based on local regions of the face, ref. [33] presented a compact and efficient facial emotion detection network. Reference [34] used an enhanced spatial–temporal learning network (ESTLNet) to conduct dynamic facial expression recognition. Reference [35] adopted a transfer learning approach in their work for FER.

The idea of this research is to introduce a novel approach to facial expression recognition by combining a generative model and pre-trained CNN model. A diffusion model was used to create new AI-generated images in order to create an AI creation database. CASME II was used by the generative model as the 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.

The rest of the paper is organized as follows: Section 2 describes the dataset used, the generative model used, and the other models used in this work. Section 3 defines the

settings and discusses the model's performance. Section 4 concludes the findings of this work and Section 5 suggests some ideas for 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 two types of datasets are utilized: CASME II and the proposed AI-generated dataset. The images obtained from both datasets were pre-processed before being fed into the pre-trained CNN models for the classification task. The preprocessing approaches employed are resizing the image, rescaling the image and changing the image to grayscale. The pre-processed images were 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 distinguish seven different facial expressions, including happiness, sadness, disgust, repression, fear, surprise and others. Lastly, the performance of each model was obtained and compared. An overview of the proposed methodology is presented in Figure 1.

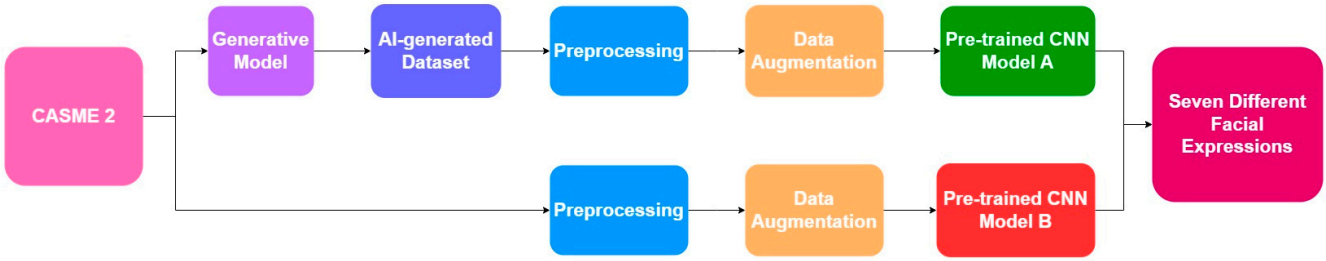


Figure 1. Overview of the proposed methodology.

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 created by [36] in 2014. CASME II is an enhanced version of the CASME dataset, which was developed by [37]. A few types of facial expression are labelled in this database: happiness, sadness, disgust, repression, fear, surprise, and others. The samples were collected in a controlled environment laboratory, and the participants' micro-expressions were taken using a high-speed camera. To elicit micro-expressions, every participant had to view a short video clip throughout the sample-collecting process. The camera was set up to directly face the participant. 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 contains a total of 17,124 static images with seven different facial expressions. This dataset includes seven categories of facial expression: happiness, sadness, surprise, disgust, fear, repression, and others. The distribution of

each facial expression is recorded in Table 1. The sample images, with seven different micro-expressions, are shown in Figure 2.

Table 1. Distribution of images in the CASME II database.

Type of Expression	Number of Images
Happiness	2360
Sadness	150
Surprise	1729
Disgust	4204
Fear	127
Repression	2187
Others	6367
Total	17,124

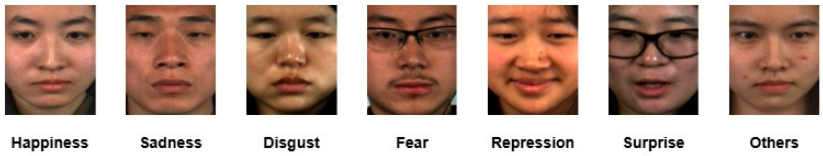


Figure 2. Sample from the CASME II database.

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 was created using the diffusion model. The proposed AI-generated facial expression dataset was generated based on the original CASME II, obtained from the author. The pixel size of the images obtained from CASME II was resized from 280 × 340 to 48 × 48 and turned to grayscale before it was fit into the diffusion model for training purposes. Then, the diffusion model was fine-tuned in order to fulfil the pre-processed images' specification. The diffusion model was well-trained before it was utilized to generate the new facial expression images. The process of creating a new AI-generated facial expression dataset is shown in Figure 3. Moreover, the completely trained model was then used to generate new facial expression images with seven categories of facial expression, including happiness, sadness, surprise, disgust, fear, repression, and others. The newly generated facial expression images are in grayscale with a pixel size of 48 × 48. The generated facial expression dataset consists of 15,464 images. The distribution of each facial expression is recorded in Table 2. The sample images, containing seven different micro-expressions, are shown in Figure 4.

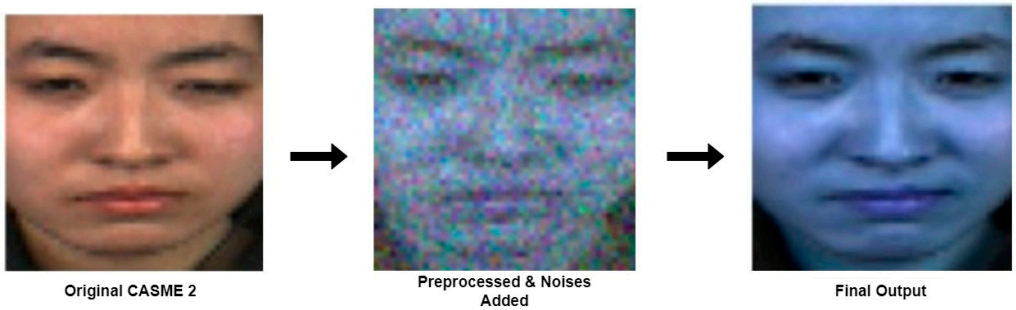


Figure 3. Process of generating new dataset.

Table 2. Distribution of images in the 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	15,464

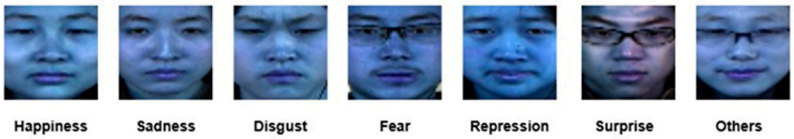


Figure 4. Sample from the proposed AI-generated facial expression dataset.

2.2.3. Why Static Images?

Micro-expressions are rapid and involuntary facial expressions that typically occur within a very short time span, often lasting between 1/25 and 1/5 of a second. Although micro-expressions are inherently dynamic, research has shown that the apex frame, which represents the moment of maximum facial muscle activation, contains the most discriminative visual information within a micro-expression sequence. At this stage, subtle muscle contractions around critical facial regions such as the eyebrows, eyes, and mouth become the most pronounced.

In this study, static frames extracted from micro-expression sequences were used as inputs to the diffusion model. This design allows the model to learn subtle spatial variations in facial features while reducing the computational complexity associated with modelling temporal dynamics. Furthermore, diffusion models are primarily designed for image-generation tasks, making static image representations a suitable input format for learning facial expression features.

2.3. Preprocessing and Parameter Settings

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

In this work, two stages of preprocessing were carried out: one for the diffusion model and another to train the pre-trained CNN models. During the preprocessing for the diffusion model, the images obtained from CASME II was resized from a pixel size of 280 × 340 to a pixel size of 48 × 48 and converted to grayscale format. The images were resized to a smaller size in order to reduce computational complexity and memory requirements during the training of the diffusion model while preserving the essential facial geometry and spatial structure of the images. Micro-expressions are primarily characterized by subtle muscle movements rather than colour variations. Therefore, the model can concentrate on structural face patterns and local texture changes while the complexity of the input is reduced by transforming the images to grayscale.

For the classification stage, the images obtained from both CASME II and the proposed AI-generated dataset were resized from 48 × 48 pixels to 160 × 160 pixels. While preserving compliance with the proposed models' architectural design, resizing the photos to a

larger standardized input size enables the pre-trained CNN models to efficiently extract hierarchical spatial characteristics through convolutional layers. Moreover, the images were resized to 160 × 160 pixels instead of the pre-trained CNN models' original input size due to the limited computational resources available to train the pre-trained CNN models.

Then, data augmentation was utilized to increase dataset variability and reduce the risk of overfitting caused by a limited number of available image samples. To replicate small differences in head orientation and camera positioning, small geometric alterations were implemented, including image rotation within a range of 15 degrees and horizontal flipping. Width and height shifts were applied within the range of 0.1 to represent small spatial displacements that may occur during real-world facial recordings. The brightness of the images was also adjusted between a range of 0.8 and 1.2 in order to simulate slight lighting changes during image capture. In order to retain the semantic significance of the facial expression, these augmentation techniques were specifically limited to restricted ranges. After augmentation, the images were rescaled to a range of 0 to 1 by dividing pixel values by 255. Furthermore, the dataset was further separated into 80% for training and 20% for testing. This split is widely used in image classification studies to guarantee that enough data is provided for training while also providing an independent dataset for testing.

The learning rate for every experiment in this work was set to 0.001 with the batch size set to 32, and the Adaptive Moment Estimation (ADAM) optimizer was the learning algorithm utilized. Preliminary pilot observations revealed that this learning rate offered consistent convergence throughout training, but greater learning rates produced unstable loss behaviour. A batch size of 32 provides a balance between computational efficiency and gradient stability. All experiments were conducted for 100 training epochs to allow for sufficient model learning while preventing excessive overfitting. The Adam optimizer was utilized as the learning algorithm due to its flexible learning rate capacity and superior performance in deep learning optimization problems. As the classification problem includes several emotion categories, the categorical cross-entropy loss function was utilized. Table 3 summarizes all the preliminary experimental hyperparameter settings.

Table 3. Parameter settings.

Parameters	Settings
No. of training	5
Learning Rate	0.001
Train-Test Ratio	80:20
Epochs	100
Batch Size	32
Learning Algorithms	ADAM (Python 3.9, Tensorflow, Keras)
Loss	Categorical Cross-Entropy

2.4. Diffusion Model

Diffusion models are one of the most popular generative models and are mainly used for image generation. In addition to being used for image generation, they are also used for denoising and data generation. Unlike other generative models, diffusion models slowly transform the data into a dense structure. There are two different steps involved in diffusion models: the forward process and reserve process. During the forward process, a diffusion model will take an image and slowly add noise to the image over a series of steps. The noise contained in the image will be increased in every step until the image becomes nearly indistinguishable from random noise. This attempts to simulate a Markov chain, in which data is gradually corrupted. The reverse process is started once the noise is added to 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. Figure 5 represents the proposed diffusion model.

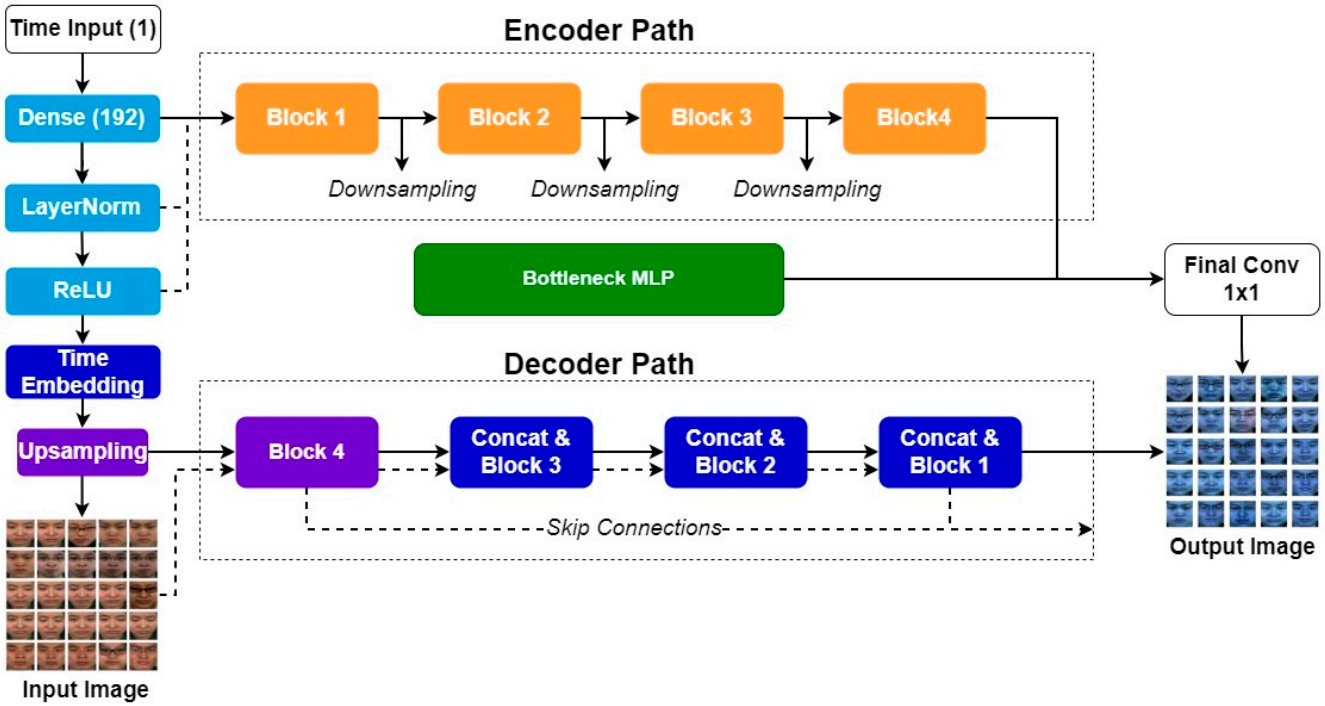


Figure 5. Proposed diffusion model.

To direct the diffusion process toward a specific emotion class, the proposed model incorporates explicit class conditioning. An embedding layer is used to transfer each emotion label to a learnable embedding vector, and then a fully connected projection is applied. Along with the time embedding, this class embedding is inserted into each convolution block of the U-Net architecture.

Consequently, the noise prediction network estimates

$$\in_{\theta} (x_t, t, c) \quad (1)$$

where c represents the target emotion. By conditioning the expected noise on both temporal and class information during reverse diffusion, the denoising trajectory is guaranteed to converge toward the learned distribution of that emotion.

For instance, the relevant embedding vector influences intermediate feature activations when creating the emotion "Repression," directing the reconstruction process toward facial muscle configurations that are statistically linked to repression in the training set. Thus, the model learns a conditional distribution $p(x|c)$, enabling controlled emotion-specific image synthesis.

2.4.1. Forward Process (Diffusion Process)

The forward process is also known as the diffusion process. In the forward process, the model will increasingly destroy the original input image by adding Gaussian noise to it. The noise added to the image will increase until the image becomes indistinguishable from pure noise. Figure 6 illustrates the forward process.

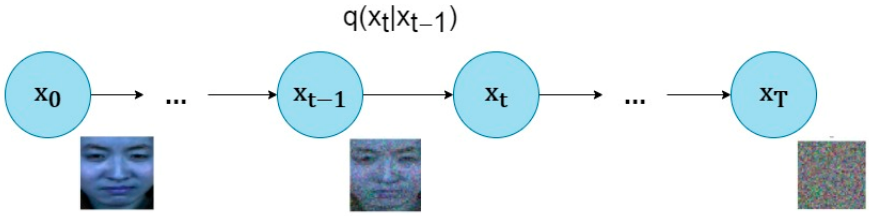


Figure 6. Forward process.

In theory, clean data x_0 is corrupted step-by-step with the introduction of small amounts of Gaussian noise over many time steps $t = 1, 2, \ldots, T$. 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. This is mathematically expressed as

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1},\ \beta_t \mathrm{I}\right) \tag{2}$$

β_t is a small variance term, which 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 the addition of noise in each 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:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{\overline{\alpha}_t}x_0,\ (1-\ \overline{\alpha}_t)I\right) \tag{3}$$

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

$$x_t = \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon \tag{4}$$

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

2.4.2. Reverse Process (Denoising Process)

The reverse process is also known as the denoising process. In the reverse process, the model will try to learn to remove 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 reverse process is presented in Figure 7.

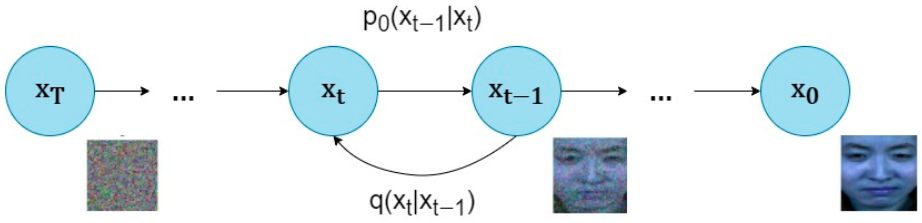


Figure 7. Reverse process.

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

$$p_0(x_{t-1}|x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t,\ t),\ \Sigma_\theta(x_t,\ t)) \tag{5}$$

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

During training, the model is trained to predict the noise ϵ that was added during the forward process using a simple objective function:

$$\mathcal{L}_{simple} = \mathbb{E}_{x_0,\, t,\, \epsilon}\left[\|\epsilon - \epsilon_\theta(x_t,\, t)\|^2\right] \tag{6}$$

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

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha_t}}}\epsilon_\theta(x_t, t)\right) + \sigma_t z \tag{7}$$

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

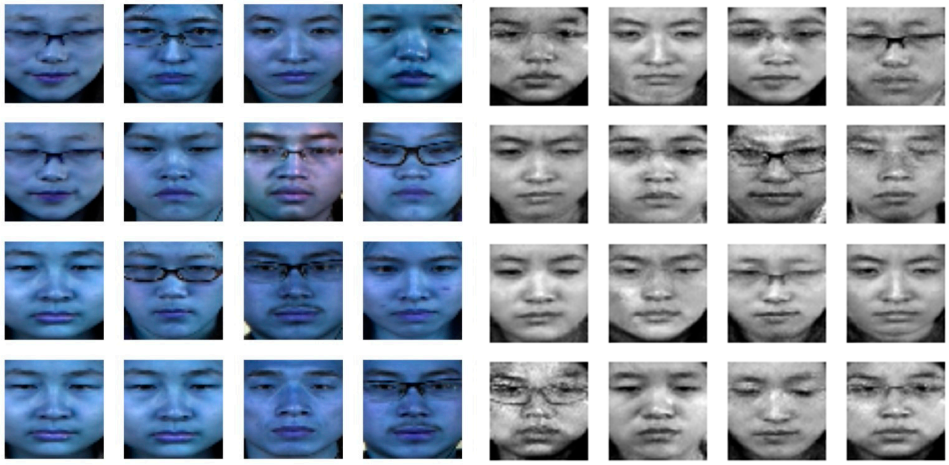
2.4.3. Why the Diffusion Model?

Generative models have been widely used for image synthesis tasks, with GAN being one of the most adopted approaches. GAN-based models have demonstrated strong performance in generating realistic facial images in various computer vision tasks. However, training GANs can be challenging due to issues such as training instability, mode collapse, and sensitivity to hyperparameter selection. These limitations can become more pronounced when working with relatively small datasets such as the CASME II, which contains limited samples of spontaneous micro-expressions.

Diffusion models have recently emerged as a powerful alternative generative framework for image synthesis. Unlike GANs, which rely on an adversarial training process between generator and discriminator networks, diffusion models learn to generate images through a gradual denoising process that models the underlying data distribution. This training strategy tends to provide more stable optimization and improves coverage of the data distribution. As a result, diffusion models have demonstrated convincing performance in producing high-quality and diverse images across a range of computer vision applications.

Diffusion models also have the benefit of better capturing small spatial differences in images. Micro-expressions are characterized by very small muscle movements and fine texture variations around key facial regions. To synthesize realistic facial micro-expression images, diffusion models employ an iterative denoising method that enables the network to gradually rebuild these small structural patterns. Furthermore, diffusion models are appropriate for tasks where gathering a lot of labelled data is challenging because the models have shown high generative power even when trained on comparatively small datasets.

Therefore, diffusion models were adopted in this study as the generative framework for creating the proposed AI-generated facial expression dataset. They are ideal for producing a variety of micro-expression samples that will be useful in subsequent classification tasks due to their consistent training behaviour and robust capacity to simulate diverse image distributions. The output generated from both the diffusion model and the GAN are illustrated in Figure 8.



(a) (b)

Figure 8. (**a**) Output generated by the diffusion model. The presented emotions are clearer. (**b**) Output generated by GAN. The generated facial expressions are incomplete, and the image contains some noise. Furthermore, some of the produced facial features are deformed, including the eyebrows, eyes and mouth.

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 that is recognized as a convolutional neural network (CNN). CNNs originated from the architecture of LeNet, which was invented by [38]. CNNs are applied to analyze input such as pictures or numeric data. CNNs do not require much preprocessing. CNNs were inspired by the neuron networks in the human brain. The convolutional layer of a CNN is used to collect picture features. Max pooling and average pooling layers are the most common pooling layers employed in CNNs. To sort the outcomes into the specified data classifications, a fully connected layer is utilized. A CNN model generally comprises an 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, is responsible for receiving raw data such as images, and the data will be passed to the 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 carried out, is a crucial part of the 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, a 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.

2.5.3. Activation Layer

The activation layer is the layer that is usually implemented 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 the Rectified Linear Unit (*ReLU*), Leaky *ReLU*, Sigmoid function, Hyperbolic Tangent (*Tanh*), and SoftMax

function, but the most commonly used activation function is the Rectified Linear Unit (*ReLU*). *ReLU* is mathematically defined as follows:

$$ReLU(x) = max(0, x) \tag{8}$$

2.5.4. Pooling Layer

Pooling layer is typically included in the construction of CNNs 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 and max pooling are the two different pooling layers. The concept of average pooling and max pooling are illustrated in Figure 9.

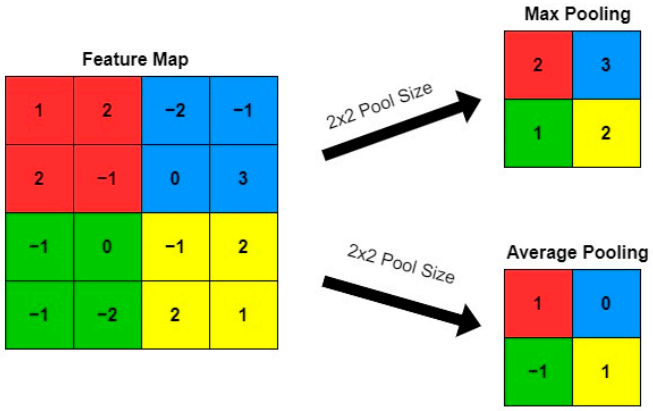


Figure 9. Pooling layers.

2.5.5. Fully Connected Layer (FC Layer)

Occasionally, a thick layer is used in a CNN, referred to as a fully connected layer (FC). It usually appears at the bottom of a neural network. Each intersection in the FC layer's output layer has a direct connection to an intersection in the pooling layer above. Then, the output is passed to a corresponding layer for image classification.

2.5.6. Output Layer

The output layer is the final layer of a CNN. This layer is mainly focused on prediction and classification tasks. Several types of activation functions are applied in the output layer, namely the Sigmoid function, SoftMax function and Linear function. The Sigmoid function is adopted in binary classification, which normally involves two classes. The SoftMax function is employed for multi-class classification, with two or more classes. The linear function is suitable for regression tasks. It is used to predict continuous values, such as stock market prices. The output layer is mathematically defined as follows:

$$y = f(Wx + b) \tag{9}$$

where f represents the activation function (e.g., Softmax, Sigmoid, Linear). In addition, W represents the weights learned during the training process. Then, x is defined as the input vector, which is also known as the 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 an architecture trained on a larger dataset to be applied to a new but similar task. Training a CNN model from scratch often requires a significant amount of time, computational power and labelled data. Transfer learning can eliminate the need to train a CNN model from scratch through leveraging the knowledge gained from a 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 in a new target domain. There are several popular pre-trained CNN models, including AlexNet, SqueezeNet, GoogleNet, ResNet, VGG, Inception, MobileNet, EfficientNet, and DenseNet. The concept of transfer learning is illustrated in Figure 10. The knowledge gained from Domain A is saved and is utilized in Domain B. In conventional CNN models, the data sample is added to the input layer, and passed to the convolution layers to train Network A. After Network A's training on this knowledge is complete, the knowledge will transfer to Network B, and the other set of input will be used to further train the convolution layers in Network B. The learned knowledge will pass through the fully connected layers in Network B for further training and to classify the data into different categories, and will be displayed in the output layer.

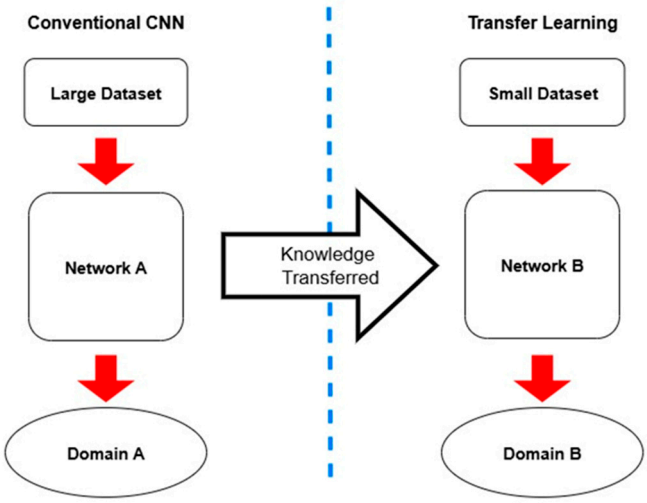


Figure 10. Concept of transfer learning.

2.7. Pre-Trained CNN Models

Pre-trained CNN models are a deep learning architecture that has been trained on a sizable and diverse dataset. This is the model adopted in transfer learning; it eases the need to train a model from scratch and enhances the knowledges of the previously trained model to allow it to perform a new task. Pre-trained CNN models can save a lot of resources, including time, computational resources and data. They are also able to provide state-of-the-art performance. There are a variety of popular pre-trained CNN models that could 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 were adopted include ResNet 50, VGG-16 and Inception V3.

2.7.1. ResNet 50

ResNet-50 [39], 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, and it secured victory in the ImageNet competition that year. Because of its deep and computationally efficient design, ResNet-50 enables high-accuracy image classification and transfer learning applications.

2.7.2. VGG-16

VGG-16 [40], a well-known deep convolutional neural network design, was introduced in the 2014 ILSVRC (ImageNet Large Scale Visual Recognition Challenge). VGG-16 is popular due to its clear and consistent architecture, which makes it powerful and simple to utilize for image classification, feature extraction, and transfer learning.

2.7.3. Inception V3

Inception V3 [41] 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 the Inception model, and it was introduced in 2015. The goal of Inception V3 is to carry out large-scale image recognition tasks more quickly and accurately.

3. Results and Discussion

3.1. Experimental Setup

The models were trained on the Intel® Core™ i7-11800H CPU (Intel, Santa Clara, CA, USA), running at 2.30 GHz, 16 GB RAM, and the NVIDIA Geforce RTX 3060 GPU (NVIDIA, Santa Clara, CA, USA) with 6 GB of dedicated graphic memory. The working environment was MATLAB 2021a.

3.2. Training of Diffusion Model

This section discusses the computational aspects involved in training the diffusion model proposed in this study. The training process was conducted for 20 epochs to allow the model to progressively learn the underlying distribution of the dataset and generate meaningful representations for each class. When compared to conventional deep learning architectures, diffusion models have a comparatively greater computing cost since they involve iterative forward and reverse processes while training. The computational cost of this study was further raised by the need to generate intermediate samples across several emotion classes for the training process.

The training process required a total time of 5 h and 34 min to complete all 20 rounds. The average training time per round was approximately 16 min and 40.4 s, indicating a relatively stable training duration across epochs. The minimum time recorded per round was 16 min and 37.8 s, while the maximum time per round was 16 min and 55.2 s. The small difference between the minimum and maximum training timeframes indicates that there were no notable variations in the computational burden brought on by resource constraints or system instability during the training process. The computational statistics obtained from the training process are summarized in Table 4.

Table 4. Computational statistics.

Computational Aspects	Time Taken
Total training time	5 h 34 min
Average time per epoch	16 min 40.4 s
Minimum time per epoch	16 min 37.8 s
Maximum time per epoch	16 min 55.2 s

3.3. Experimental Results

This study was conducted using four different 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, and VGG-16, were employed in this study. The settings are described as follows:

- Setting 1: Inception V3, ResNet 50, and VGG-16 were trained and tested on the original CASME II. This CASME II is the original dataset, obtained from the original author of this dataset.

- Setting 2: Inception V3, ResNet 50, and VGG-16 were trained and tested on the proposed AI-generated dataset. The proposed AI-generated dataset was generated using a diffusion model 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.
- Setting 4: Inception V3, ResNet 50, and VGG-16 were first trained on the proposed AI-generated dataset and tested using CASME II.

3.3.1. CASME II

CASME II was employed to train and test the pre-trained CNN models in this experiment as the baseline dataset to facilitate comparison with the other datasets. There were five experimental trials for VGG-16, ResNet 50 and Inception V3. The results from every trial were used to obtain the average accuracy for every model.

In Table 5, the performance obtained from each model when trained using CASME II is presented. VGG-16 is the best-performing model trained on CASME II, with the best classification accuracy of 99.33%. Inception V3 and ResNet 50 obtained accuracies of 99.11% and 88.34%, respectively. VGG-16 outperformed both ResNet 50 and Inception V3. VGG-16 has slightly higher accuracy compared to Inception V3, by 0.22%. When compared with ResNet 50, VGG-16 has a very significant difference in accuracy of 10.99%. This is due to VGG-16 having a shallow and compact architecture as compared to ResNet 50, increasing the efficiency of its learning for every facial expression. In this experiment, ResNet 50 has the lowest accuracy. Figure 11 presents a performance graph for every pre-trained CNN model in CASME II.

Table 5. 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

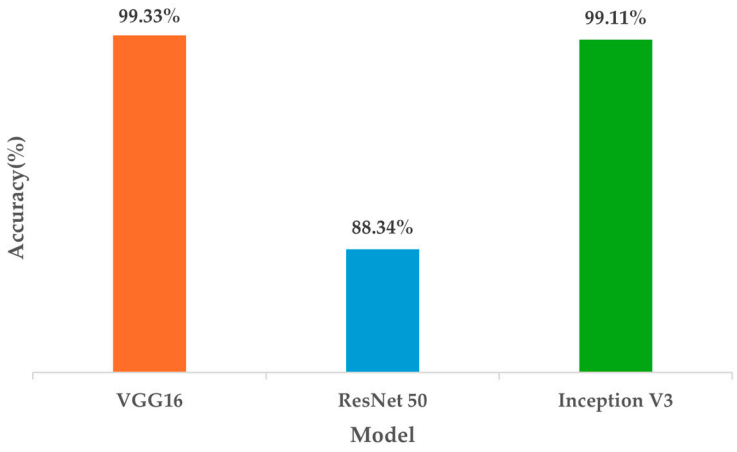


Figure 11. Performance graph for every pre-trained CNN model in CASME II.

3.3.2. Proposed AI-Generated Dataset

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

The performances obtained from each model when trained using our proposed AI-generated dataset are presented in Table 6. In this experiment, Inception V3 showed the best performance, with the highest accuracy compared to the other models. It achieved an accuracy of 99.09% and outperformed VGG-16 and ResNet 50. The performance of Inception V3 was followed by that of ResNet 50, with 98.70%, and VGG-16, with 98.66%. There was minimal difference in the accuracies of ResNet 50 and VGG-16 of 0.04%. When compared with Inception V3, ResNet 50 shows a difference in accuracy of 0.39% and the difference in accuracy between VGG-16 and Inception V3 was 0.43%. The performance of every model is very close when using our proposed AI-generated dataset, as shown in Figure 12.

Table 6. Performance of every pre-trained CNN model on the 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

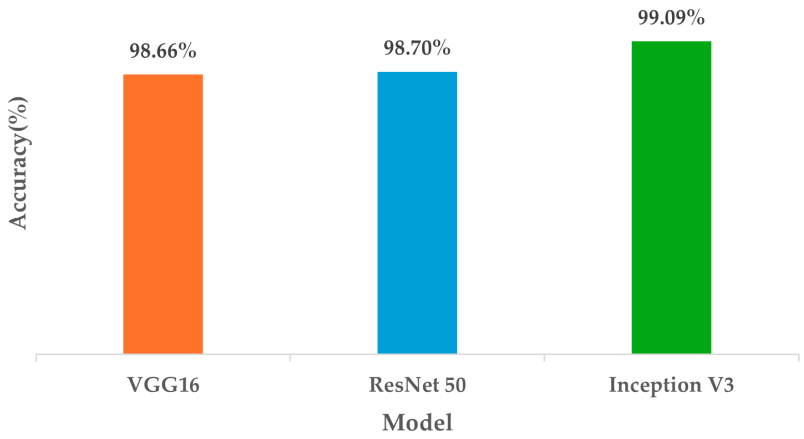


Figure 12. Performance graph for every pre-trained CNN model using the proposed AI-generated dataset.

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

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

Table 7 represents the performance of every model when trained on CASME II and tested using our proposed AI-generated dataset. It can be observed that the best-performing model in this experiment was Inception V3, with the highest accuracy of 99.47%. ResNet 50 achieved an accuracy of 99.24%. It performed slightly worse than Inception V3 in this experiment, with a minimal difference of 0.23%. VGG-16 obtained the lowest accuracy of 98.93% in this experiment. Inception V3 obtained a higher accuracy of 0.54% as compared to VGG-16. When comparing ResNet 50 with VGG-16, there is a small difference of 0.31% between their accuracies. The results are depicted in Figure 13.

Table 7. Performance of every pre-trained CNN model when trained on CASME II and tested on the 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

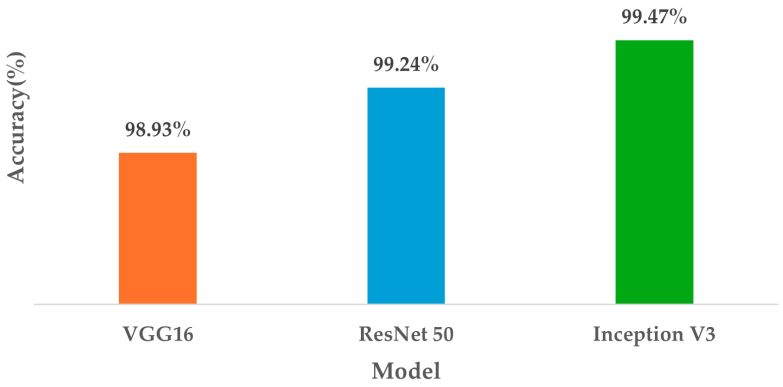


Figure 13. Performance graph for every pre-trained CNN model when trained on CASME II and tested on the proposed AI-generated dataset.

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

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

Table 8 shows the performance of each model when trained on our proposed AI-generated with CASME II is used for testing. Based on the results obtained, Inception V3 showed the best performance in this experiment. It achieved the highest accuracy of 99.48% and outperformed VGG-16 and ResNet 50. VGG-16 and ResNet 50 achieved accuracies of 96.87% and 99.28%, respectively. When comparing Inception V3 and ResNet 50, a minimum difference of 0.20% is obtained. Inception V3 outperformed VGG-16, with a significant difference of 2.70%. VGG-16 showed the worst performance compared to Inception V3 and ResNet 50 in this experiment, as presented in Figure 14.

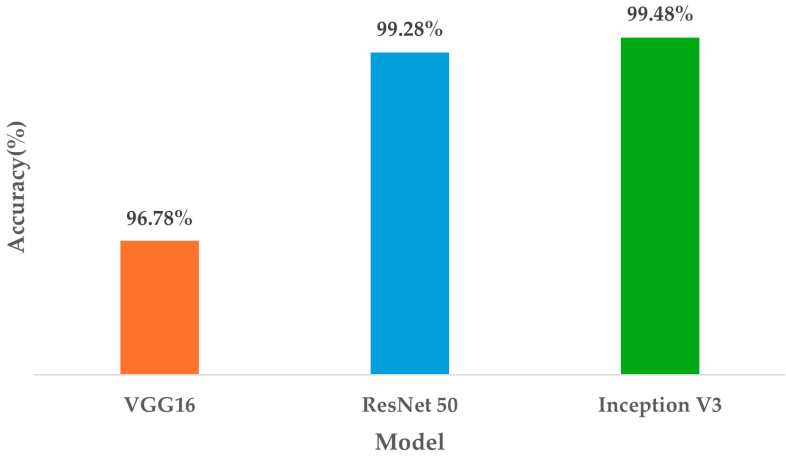


Figure 14. Performance graph for every pre-trained CNN model when trained on the proposed AI-generated dataset and tested on CASME II.

Table 8. Performance of every pre-trained CNN model when trained on the 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

3.4. Ablation Work

In this study, ablation work is carried out in order to improve the performance of the models trained on both CASME II and our proposed AI-generated dataset. Two types of ablations work were 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: Data augmentation is applied to the datasets. Data augmentation is utilized to artificially increase the size, variety and variability of the data sample for the pre-trained CNN models.
- Configuration 2: Transfer learning is performed 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 the remaining layers were applied for training and testing using the datasets.

3.4.1. Data Augmentation

Data augmentation is a method utilized in machine learning in order to synthetically expand the size, variety and variability of the training data for machine learning models and deep learning models. It comprises multiple techniques including flipping, translating, rotating, scaling, zooming, adjusting the image brightness and adding noise to the images. Data augmentation provides a multitude 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 was utilized in both the 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, and could even improve their performance.

CASME II (Data Augmentation)

In this experiment, CASME II was used to train and test every adopted pre-trained CNN model and was utilized as the baseline dataset for comparison with the other datasets. There were five experimental trials 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 the data augmentation technique and training using CASME II are presented in Table 9. The table shows that ResNet 50 achieved an accuracy of 99.20% and outperformed VGG-16 and Inception V3 in this experiment. This result 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, with a difference of 0.69%. VGG-16 obtained the lowest accuracy in this experiment, with a lower accuracy compared to ResNet 50 by 1.97% and a lower accuracy than Inception V3 by 1.28%. All the results are depicted in Figure 15.

Table 9. Performance of every pre-trained CNN model when 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

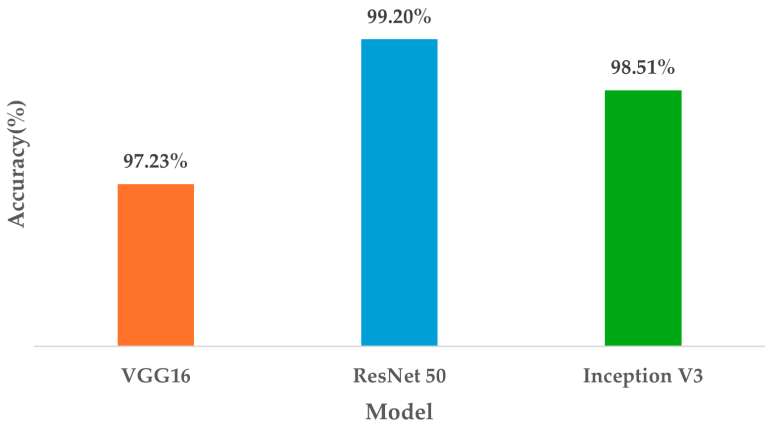


Figure 15. Performance graph for every pre-trained CNN model while trained and tested on CASME II (data augmentation).

Proposed AI-Generated Dataset (Data Augmentation)

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

Table 10 shows each model's performance when trained using our proposed AI-generated dataset. In this experiment, ResNet 50 outperformed VGG-16 and Inception V3 based on its best performance. ResNet 50 achieved 99.03% accuracy, which is the highest accuracy as compared to the other models. Inception V3 and VGG-16 obtained accuracies of 98.54% and 95.73%, respectively. Inception V3's performance fell behind that of ResNet 50, with a gap of 0.49%. VGG-16 with ResNet 50 showed a significant difference in their accuracies of 3.30%. VGG-16 obtained the worst performance in this experiment as compared to ResNet50 and Inception V3. A comparison of the performances is illustrated in Figure 16.

Table 10. Performance of every pre-trained CNN model when 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

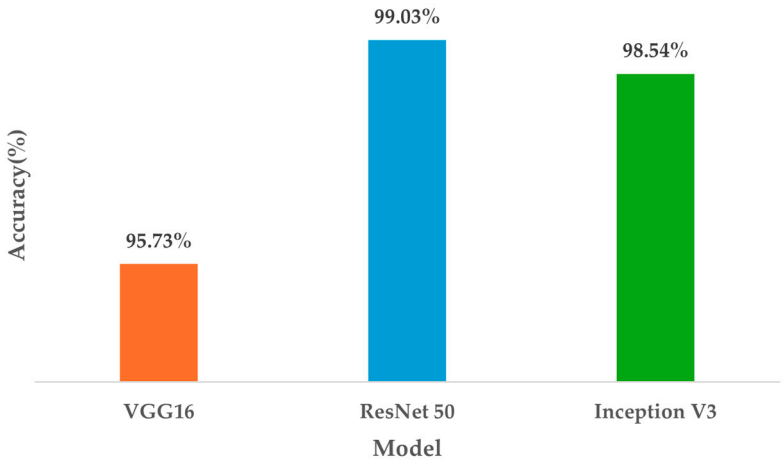


Figure 16. Performance graph for every pre-trained CNN model when trained on the proposed AI-generated dataset (data augmentation).

Training on CASME II and Testing on the Proposed AI-Generated Dataset (Data Augmentation)

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 five 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 11, the performance of 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, 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 worse as compared to ResNet 50; its accuracy is lower than ResNet 50's by 0.05%. When VGG-16 is compared to ResNet 50, there is a difference between both models of 1.12%. Although VGG-16 achieved state-of-the-art performance, it still performed the worst as compared to ResNet 50 and Inception V3, as shown in Figure 17.

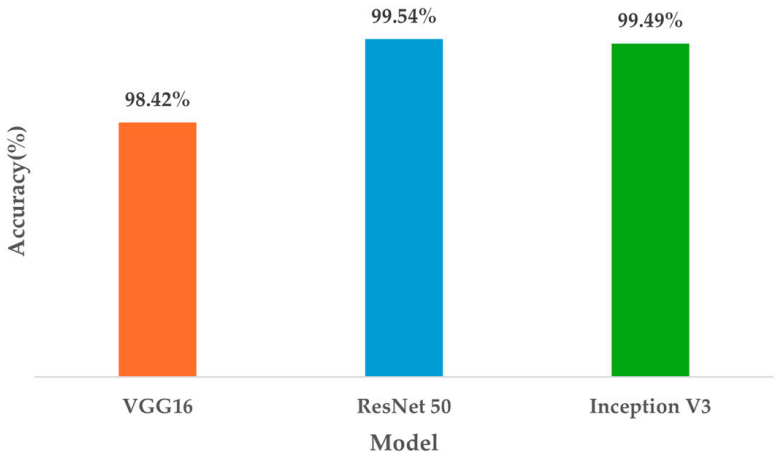


Figure 17. Performance graph for every pre-trained CNN model when trained on CASME II and tested on the proposed AI-generated dataset (data augmentation).

Table 11. Performance of every pre-trained CNN model when 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

Training on Proposed AI-Generated Dataset and Testing on CASME II (Data Augmentation)

In this experiment, data augmentation was applied to our proposed AI-generated dataset and used to train the pre-trained CNN models, while CASME II was applied for testing. A total of five 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 in this experiment, when trained on our proposed AI-generated model and when CASME II is used for testing, is recorded in Table 12. Inception V3 achieved a 99.11% accuracy and the best performance in this experiment. ResNet 50 obtained an outstanding performance, with 99.07% accuracy. It was outperformed by Inception V3 with a minimum difference of 0.04%. VGG-16 obtained an accuracy of 98.38% and was outperformed by Inception V3, with a difference of 0.73%. In this experiment, VGG-16 has the worst performance as compared to Inception V3 and ResNet 50, as represented in Figure 18.

Table 12. Performance of every pre-trained CNN model when 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

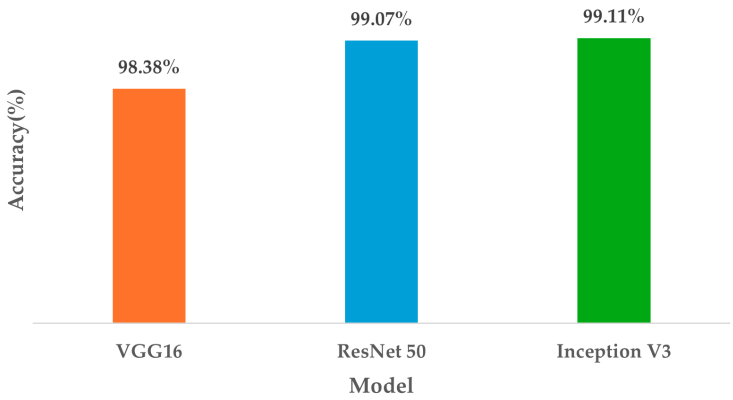


Figure 18. Performance graph for every pre-trained CNN model when trained on the proposed AI-generated dataset and tested on CASME II (data augmentation).

3.4.2. Freezing Layer 30%

A freezing layer is commonly utilized in pre-trained CNN models to prevent certain layers from updating their weights during the 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 the freezing layer technique

including saving computational resources, preserving learned features and preventing overfitting problems.

The freezing layer technique was applied to every adopted pre-trained CNN model in this study. The first 30% learnable layers of every pre-trained CNN model were frozen. The first 30% learnable layers were frozen because this was found to be the most suitable setting for the pre-trained CNN models in this work after trial and error with different freezing layer settings. The freezing layer technique was utilized on the pre-trained CNN models in this work to improve the learning performance of the models.

CASME II (Freezing Layer 30%)

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

Table 13 shows the tabulated performance of every model when 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%. VGG-16 and ResNet 50 achieved accuracies of 96.52% and 93.21% respectively. Both VGG-16 and ResNet 50 were outperformed by Inception V3. When 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 these two models. In this experiment, ResNet 50 presented the poorest performance compared to the other two models. All the results are presented in Figure 19.

Table 13. Performance of every pre-trained CNN model when 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

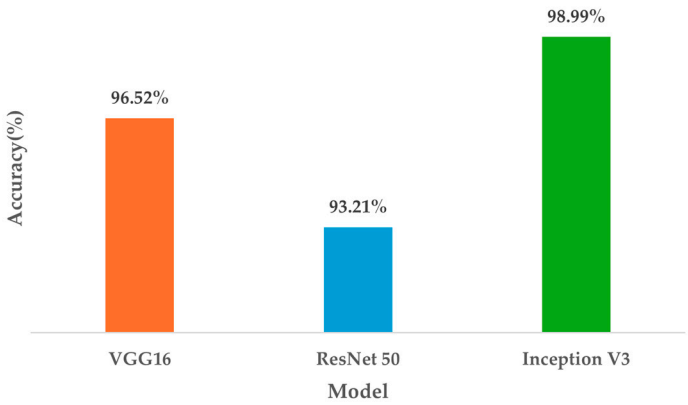


Figure 19. Performance graph for every pre-trained CNN model when trained and tested on CASME II (freezing layer 30%).

Proposed AI-Generated Dataset (Freezing Layer 30%)

In this experiment, our proposed AI-generated dataset was used in the training and testing of the pre-trained CNN models, namely Inception V3, ResNet 50 and VGG-16.

Every model performed five experimental trials and the results obtained from each trial were utilized to obtain the average accuracy.

Table 14 presents the performances of every model when trained and tested on our proposed AI-generated dataset. It can be observed that the best-performing model in this experiment is Inception V3, with the highest accuracy of 99.14%. This result is 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% in accuracy. When 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%. The difference between Inception V3 and VGG-16 is 1.93%. A comparison of the models' performance is illustrated in Figure 20.

Table 14. Performance of every pre-trained CNN model when 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

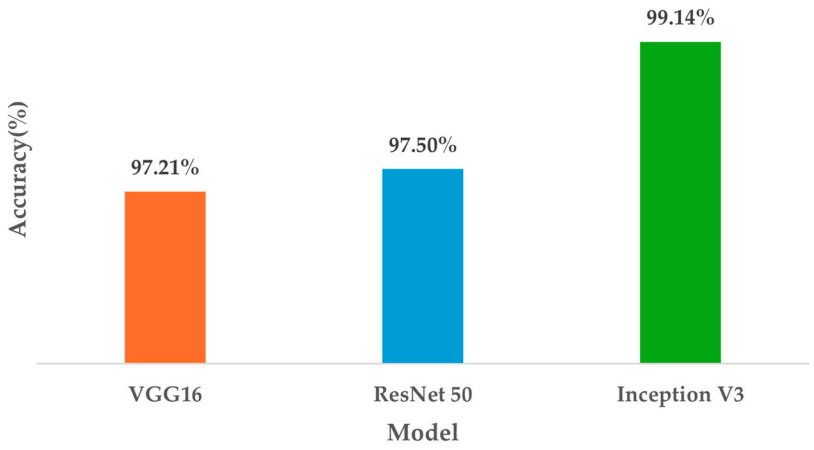


Figure 20. Performance graph for every pre-trained CNN model when trained and tested on the proposed AI-generated dataset (freezing layer 30%).

Training on CASME II and Testing on Proposed AI-Generated Dataset (Freezing Layer 30%)

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

Table 15 presents the performance of each model when 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 successfully surpassed ResNet 50 and VGG-16. In this experiment, VGG-16 achieved an accuracy 99.32% and its performance is behind that of Inception V3. There is a 0.12% difference between Inception V3 and VGG-16. Moreover, ResNet 50 achieved an accuracy of 99.20% and its accuracy is lower than that of Inception V3 by 0.24%. VGG-16 performed slightly better than ResNet 50, by 0.12%. All the models achieved slightly difference accuracies in this experiment, and their performances were very similar, as shown in Figure 21.

Table 15. Performance of every pre-trained CNN model when 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

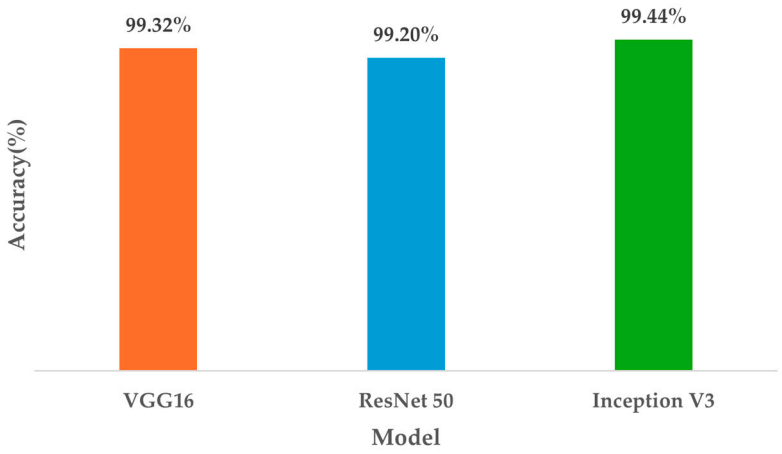


Figure 21. Performance graph for every pre-trained CNN model when trained on CASME II and tested on the proposed AI-generated dataset (freezing layer 30%).

Training on Proposed AI-Generated Dataset and Testing on CASME II (Freezing Layer 30%)

Our proposed AI-generated dataset was employed as the training dataset and CASME II was employed as the testing dataset in this experiment. There were five experimental trials that were undertaken by every model, including Inception V3, VGG-16 and ResNet 50, and the obtained results from every trial were utilized to obtain the average accuracy for every model.

Table 16 shows the performance for each model when trained on our proposed AI-generated with CASME II used for testing. Based on the outcomes, Inception V3 achieved the best results, with an accuracy of 99.53%. 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. VGG-16 has the worst performance in this experiment, although it achieved a state-of-the-art performance. Figure 22 shows the performances of all the proposed models in this experiment.

Table 16. Performance of every pre-trained CNN model when 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

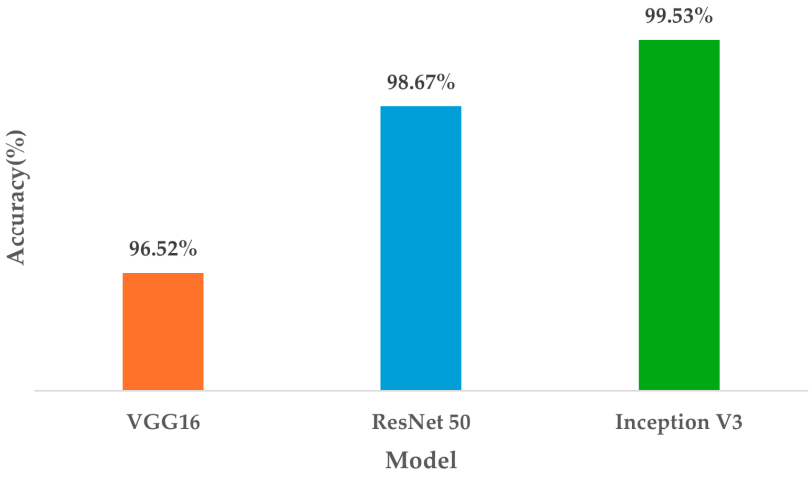


Figure 22. Performance graph for every pre-trained CNN model when trained on the proposed AI-generated dataset and tested on CASME II (freezing layer 30%).

3.5. Performance Comparison with Other Works

In this section, a performance comparison of the proposed approaches and some state-of-the-art approaches is presented. All the existing methods were evaluated based on the existing dataset and their own proposed dataset. Therefore, there were some difficulties in directly comparing the methods in terms of the structure of the database, the preprocessing of the dataset, and the labels of facial expressions, as well as the feature extraction approaches applied to the dataset. The comparisons might not be valuable, but they are still able to demonstrate that our proposed approaches have the ability to achieve competitive and state-of-the-art results compared to the other methods.

Table 17 tabulates a comparison of the performance of the proposed works with other state-of-the-art approaches in terms of facial expression recognition in classification based on an existing dataset, CASME II, and their own proposed dataset. A pre-trained ResNet 50 was deployed in Zhang and Shen's work using CASME II as their dataset. The work proposed by Zhang and Shen achieved an accuracy of 98.41%. The approach proposed by Zhi et al. obtained 97.60% accuracy when using CASME II to train their proposed 3D-CNN. Sun et al. introduced an SVM paired with knowledge distillation in their work and their approach obtained 72.60% accuracy. In Wang et al.'s 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, also trained on CASME II. Our proposed pre-trained VGG-16 achieved 99.33% accuracy and it outperformed the other approaches. Our proposed VGG-16 achieved slightly better accuracy as compared to the results of Zhang and Shen and Zhi et al. While compared with Sun et al.'s and Wang et al.'s works, there is 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 the MMI database and employed a VGG-16 to evaluate their dataset's performance. The performance obtained by their work achieved 98.30% 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 obtained better results compared to the work proposed by Wang et al. In addition, Fan et al. adopted a modified VGG-Net with the CGAN approach. Their modified VGG-Net was trained on the Oulu dataset generated by their proposed CGAN and tested using the generated CK+ dataset. Their approach obtained an accuracy of 90.37%. Two of our proposed approaches use similar methods to Fan et al.'s work. The first approach is ResNet

50, which uses data augmentation and is trained using CASME II and tested using our proposed AI-generated dataset. This approach achieved an outstanding performance, with an accuracy of 99.54%. The second proposed approach is an Inception V3 with 30% freezing layers, trained on our proposed AI-generated dataset and tested on CASME II. The proposed approach achieved 99.53% accuracy. Both of our proposed approaches showed significantly better performance as compared to the work by Fan et al. A performance assessment of our proposed works and other state-of-the-art methods is presented in Figure 23.

Table 17. Performance assessment of the proposed works compared to other state-of-the-art methods.

Model	Dataset	Accuracy
3D-CNN [29]	CASME II	97.60%
TLCNN [30]	CASME II	69.10%
SVM with knowledge distillation [42]	CASME II	72.60%
Pre-trained ResNet 50 [18]	CASME II	98.41%
Modified VGG-Net with CGAN [4]	Trained on AI-generated Oulu and tested 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)	Trained on CASME II and tested on proposed AI-generated dataset	99.54%
Proposed pre-trained Inception V3 (with 30% freezing layers)	Trained on proposed AI-generated dataset and tested on CASME II	99.53%

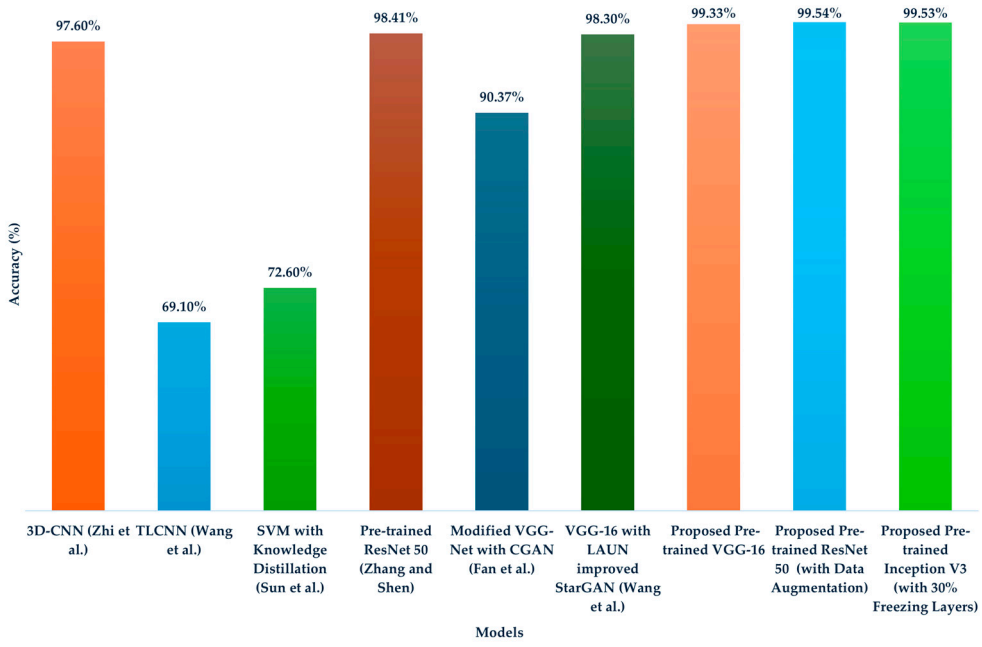


Figure 23. Graph showing a performance comparison between each proposed approach and other state-of-the-art methods [3,4,18,29,30,42].

4. Conclusions

This work is mainly focused on using a generative model to create an AI-generated facial expression dataset for facial expression recognition. Four different experiments were conducted using four different dataset combinations: 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 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 with the same experimental settings. During the data augmentation approach, a VGG-16 model was trained on CASME II and tested on the proposed AI-generated dataset. This proposed approach achieved the highest accuracy compared to other approaches, at 99.54%. When the 30% freezing layers method is utilized, Inception V3 trained on the proposed AI-generated dataset and tested using CASME II was the best-performing approach, with the highest accuracy of 99.53%. In addition, our proposed approaches were also compared with several existing facial expression recognition methods and outperformed other proposed methods. Although this study presented promising performance, there are still many potentials and possibilities for future work, including adopting different types of generative models, adopting different types of deep learning models and using different types of datasets to conduct the same work.

5. Limitations and Future Works

Although the proposed approach presents promising results, several limitations remain and should be considered when interpreting the findings. First, this study relies on a single dataset, CASME II, which may limit the generalizability of the proposed method. While CASME II is a widely used benchmark dataset for micro-expression research, training and evaluation on a single dataset may cause the model to learn dataset-specific characteristics rather than universally applicable facial expression features. As a result, the model's performance may vary when applied to other datasets or real-world environments with different recording conditions, lighting variations, or differing facial characteristics.

Another limitation relates to potential demographic and domain bias within the dataset. The images in CASME II were collected primarily from Asian participants, and the dataset lacks diversity in terms of ethnicity, skin tone, age group, and cultural background. This limited representation may affect the model's ability to generalize to more diverse populations.

Furthermore, the proposed approach utilizes synthetic images generated by diffusion models. While synthetic data can help address data scarcity, it may also reproduce or amplify existing biases present in the original dataset. If the training dataset lacks diversity, the generated samples may inherit similar limitations. Therefore, careful evaluation of the quality and diversity of the synthetic data is necessary to ensure that the generated samples contribute positively to model learning.

In addition, the current study evaluates the generated dataset using a limited set of pre-trained CNN architectures. More advanced deep learning models designed for facial expression recognition may lead to improved performance and better feature representation.

These limitations motivate several potential directions for future research:

- Different types of generative models, such as Generative Adversarial Network (GAN) and Variational Autoencoder (VAE), can be implemented in future work. This approach could allow for a comparison of the quality of the images generated by different types of generative models.
- Utilize image processing methods such as Action Units (AUs), the Histogram of Oriented Gradients (HOG), and Laplacian filters could enhance the quality of the images and make the features more obvious. These approaches would help the models to better learn the features in the images and enhance their accuracy.
- Different types of deep learning architecture include different variants of pre-trained CNN models, hybrid CNN models with attention mechanisms, vision transformers (ViT), the hybrid CNN–Transformer model, 3DCNN, and CNN with Long Short-Term Memory (LSTM). These architectures are suggested for future work because they might improve the FER performance.
- Lastly, the proposed method could employ different facial expression datasets, such as SAMM and SMIC, rather than only CASME II. This could prove that our proposed method is not only workable on CASME II, but it also works when this is replaced with different datasets. This allows for a more comprehensive assessment of the model's generalization capability across different data distributions and recording conditions.

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 and 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

References

1. Ekman, P.; Friesen, W.V. Constants across Cultures in the Face and Emotion. *J. Personal. Soc. Psychol.* **1971**, *17*, 124–129. [CrossRef] [PubMed]
2. Ekman, P.; Dalai, L. *Emotional Awareness: Overcoming the Obstacles to Psychological Balance and Compassion; A Conversation Between the Dalai Lama and Paul Ekman*, 1st ed.; Times Books: New York, NY, USA; Henry Holt and Co.: New York, NY, USA, 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. [CrossRef]
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), Seattle, WA, USA, 10–13 December 2018*; IEEE: New York, NY, USA, 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. [CrossRef]
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), Hyderabad, India, 6–11 April 2025*; IEEE: New York, NY, USA, 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), Shanghai, China, 22–26 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, Portland, OR, USA, 23–28 June 2013; pp. 3422–3429.
9. Solis-Arrazola, M.A.; Sanchez-Yanez, R.E.; Gonzalez-Acosta, A.M.S.; Garcia-Capulin, C.H.; Rostro-Gonzalez, H. Eliciting Emotions: Investigating the Use of Generative AI and Facial Muscle Activation in Children's Emotional Recognition. *Big Data Cogn. Comput.* **2025**, *9*, 15. [CrossRef]
10. 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), Sousse, Tunisia, 2–5 September 2020*; IEEE: New York, NY, USA, 2020; pp. 1–6.
11. 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), Marrakech, Morocco, 28–30 October 2019*; IEEE: New York, NY, USA, 2019; pp. 1–6.
12. 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), Coimbatore, India, 6–7 March 2020*; IEEE: New York, NY, USA, 2020; pp. 317–320.
13. 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), Osijek, Croatia, 5–7 June 2019*; IEEE: New York, NY, USA, 2019; pp. 169–173.
14. 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), Shenyang, China, 29–31 January 2023*; IEEE: New York, NY, USA, 2023; pp. 580–583.
15. 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), Bangalore, India, 2–4 July 2020*; IEEE: New York, NY, USA, 2020; pp. 1–6.
16. Xie, S.; Hu, H. Facial Expression Recognition with FRR-CNN. *Electron. Lett.* **2017**, *53*, 235–237. [CrossRef]
17. Shao, J.; Qian, Y. Three Convolutional Neural Network Models for Facial Expression Recognition in the Wild. *Neurocomputing* **2019**, *355*, 82–92. [CrossRef]
18. 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), Dali, China, 11–13 October 2023*; IEEE: New York, NY, USA, 2023; pp. 772–775.
19. Li, J.; Zhang, D.; Zhang, J.; Zhang, J.; Li, T.; Xia, Y.; Yan, Q.; Xun, L. Facial Expression Recognition with Faster R-Cnn. *Procedia Comput. Sci.* **2017**, *107*, 135–140. [CrossRef]
20. 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), Coimbatore, India, 5–7 November 2020*; IEEE: New York, NY, USA, 2020; pp. 1179–1183.

21. Reghunathan, R.K.; Ramankutty, V.K.; Kallingal, A.; Vinod, V. Facial Expression Recognition Using Pre-Trained Architectures. *Eng. Proc.* **2024**, *62*, 2.
22. Zhao, K.; Liu, X.; Yang, G. M3ENet: A Multi-Modal Fusion Network for Efficient Micro-Expression Recognition. *Sensors* **2025**, *25*, 6276. [CrossRef]
23. 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), Tirunelveli, India, 20–22 August 2020*; IEEE: New York, NY, USA, 2020; pp. 1328–1333.
24. 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), Kuala Lumpur, Malaysia, 24–26 September 2021*; IEEE: New York, NY, USA, 2021; pp. 1–6.
25. 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. [CrossRef]
26. Xiao, H.; Li, W.; Zeng, G.; Wu, Y.; Xue, J.; Zhang, J.; Li, C.; Guo, G. On-Road Driver Emotion Recognition Using Facial Expression. *Appl. Sci.* **2022**, *12*, 807. [CrossRef]
27. 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), Washington, DC, USA, 30 May–3 June 2017*; IEEE: New York, NY, USA, 2017; pp. 558–565.
28. 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), Long Beach, CA, USA, 16–17 June 2019*; IEEE: New York, NY, USA, 2019; pp. 837–846.
29. 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**, *102*, 1054–1064. [CrossRef]
30. 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. [CrossRef]
31. Mohammadpour, M.; Khaliliardali, H.; Hashemi, S.M.R.; AlyanNezhadi, M.M. Facial Emotion Recognition Using Deep Convolutional Networks. In *Proceedings of the 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 22 December 2017*; IEEE: New York, NY, USA, 2017; pp. 0017–0021.
32. Li, J.; Jin, K.; Zhou, D.; Kubota, N.; Ju, Z. Attention Mechanism-Based CNN for Facial Expression Recognition. *Neurocomputing* **2020**, *411*, 340–350. [CrossRef]
33. Jin, X.; Jin, Z. MiniExpNet: A Small and Effective Facial Expression Recognition Network Based on Facial Local Regions. *Neurocomputing* **2021**, *462*, 353–364. [CrossRef]
34. Gong, W.; Qian, Y.; Zhou, W.; Leng, H. Enhanced Spatial-Temporal Learning Network for Dynamic Facial Expression Recognition. *Biomed. Signal Process. Control* **2024**, *88*, 105316. [CrossRef]
35. Kumar, R.; Corvisieri, G.; Fici, T.F.; Hussain, S.I.; Tegolo, D.; Valenti, C. Transfer Learning for Facial Expression Recognition. *Information* **2025**, *16*, 320. [CrossRef]
36. 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. [CrossRef] [PubMed]
37. 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), Shanghai, China, 22–26 April 2013*; IEEE: New York, NY, USA, 2013; pp. 1–7.
38. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
39. 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), Las Vegas, NV, USA, 27–30 June 2016*; IEEE: New York, NY, USA, 2016; pp. 770–778.
40. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
41. 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), Las Vegas, NV, USA, 27–30 June 2016*; IEEE: New York, NY, USA, 2016; pp. 2818–2826.
42. Sun, B.; Cao, S.; Li, D.; He, J.; Yu, L. Dynamic Micro-Expression Recognition Using Knowledge Distillation. *IEEE Trans. Affect. Comput.* **2022**, *13*, 1037–1043. [CrossRef]