STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–15.
Published online in International Academic Press (www.IAPress.org)

Mixed Emotion Recognition Through Facial Expression using Transformer-Based Model

Limas Jaya Akeh*, Gede Putra Kusuma

Computer Science Department, BINUS Graduate Program, Bina Nusantara University, Indonesia

Abstract Basic facial expressions such as Angry, Disgust, Surprised, Happy, Scared, and Sadness can express emotions. However, in conversations, compound emotions can form Mixed Facial emotions, a combination of basic emotions that is much more complex. Mixed emotion recognition is a recent study that has not been researched enough, even though datasets such as the Indonesian Mixed Emotion Dataset (IMED) already contain mixed emotions. This research aims to implement and fine-tune Transformer-based models such as Vision Transformer, Swin Transformer v2, and ConvNet-based models such as ConvNeXt architecture to identify and recognize mixed emotions through human faces using the IMED Dataset. Various configurations with fine-tuned hyperparameters are tested and vary between each model. The result shows that Vision Transformer architecture outperforms other models in Mixed Emotion Recognition from Facial expressions and reaches up to 79.37% Testing accuracy compared to Swin Transformer v2 model with 65.36% Testing accuracy and ConvNext with 74.77% Testing accuracy.

Keywords Emotion Recognition, Mixed Emotions, Facial Expression, Vision Transformer, Deep Learning

DOI: 10.19139/soic-2310-5070-2103

1. Introduction

Humans are social creatures that require communication with each other, this is done to convey their opinions and feelings. One way of nonverbal communication that humans use to convey their feelings is through facial expressions. According to De La Torre and Cohn J.F., facial expressions can describe emotions, personality, and intentions [1]. Previous Research papers suggest that facial expressions can be categorized into various basic emotions, such as Anger, Disgust, Surprise, Happy, Afraid, and Sad [2, 3]. However, recent findings indicate that this may not be the case, for example, heavy smokers smiled and frowned while watching a burning cigarette, which indicates there may be more than one emotion shown by the smoker [4, 5].

Research conducted in the field of SSP (Social Signal Processing) in the introduction and analysis of interactions between humans and computers uses *Artificial Intelligence* with various methods in the *Facial Emotion Recognition* task [6]. This task examines how computer systems can recognize emotions through facial expressions. Examples include the use of Facial Emotion Recognition to recognize the emotional state and attention of students in learning at educational institutes such as universities or the use of Facial Emotion Recognition to detect the state and immersion of players in entertainment such as video games. Researchers such as Kolakowska et. al. [7] can see player responses and emotions while playing games and obtain indirect feedback from players and increase player satisfaction.

Recent studies have performed Facial Emotion Recognition tasks using Transformer-based Models such as Vision Transformer and Swin Transformer with outstanding performance, Vision Transformer model by Li et. al.

*Correspondence to: Limas Jaya Akeh (Email: limas.akeh@binus.edu). Computer Science Department, BINUS Graduate Program, Master of Computer Science, Bina Nusantara University. Jakarta, Indonesia 11480.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

[12] reached 88.62% test accuracy in RAF-DB Dataset and Swin Transformer model by Han et. al. [11] managed to reach 99.38% test accuracy in CK+ Dataset, proving that Transformer-based models are superior in FER Tasks due to its capability and scalability by using Attention mechanism. However, recent models try to disprove that, such as adapted ConvNeXt models which are able to achieve higher accuracy and outperform other FER approaches according to FerNeXt [13] and EmoNeXt [14].

However, not many experiments or classifications of Facial Emotion Recognition have been carried out on datasets that have Mixed or Compound Emotions, even though Pessoa, L. [8] explained that the emotions conveyed by a person are a complex topic from a network of neurons in the brain that are interconnected and can originate from several different brain regions. These basic emotions can create a combination of several emotions which is called ”Mixed Emotion” or ”Compound Emotion” [9], where even basic emotions that should be opposites, such as sadness and happiness, can occur simultaneously. The emergence of datasets that contain mixed emotions such as IMED Dataset (Indonesian Mixed Emotion Dataset) [10] allows researchers to understand and conduct experiments to recognize Mixed emotions from Facial Emotion Recognition tasks.

Therefore, in this research paper, we analyze, implement, and optimize Transformer-based models by performing various hyperparameter finetuning to recognize Mixed Emotion using the IMED Dataset, comparing the performance of each model such as Vision Transformer and Swin Transformer, compare its performance to ConvNeXt Model and finally evaluate and test the best performing models in Facial Emotion Recognition task.

2. Literature Review and Related Work

Research related to Facial Emotion Recognition through facial expressions has been carried out using several methods including SVM and AAM, followed by CNN.

In the original IMED Dataset research by Liliana et. al. [10], the author validated basic emotions using the SVM (Support Vector Machine) method and produced an average precision of 84.473%, average recall of 85.807%, average accuracy of 83.472%, and F1-Score of 0.846.

Another research by Liliana et. al. [15] uses Facial Point Detection as a method for determining unique points on the face using the AAM (Active Appearance Method) method, after that, Semantic Feature extraction converts FP (Facial points) to produce FC (Facial components). This FC will be input into FFC (Fuzzy facial Component) which will produce input parameters for the FEC (Fuzzy Emotion Classification) model. This model produces an average accuracy of 87% on the IMED Dataset.

Research by Jala et. al. [16] developed a Convolutional Neural Network (CNN model that uses the Indonesian Mixed Emotion Dataset (IMED) and implemented it on real-time data. First, this research carries out data preprocessing using the Viola-Jones method and feeds it to train the model. From the experimental results, the CNN model created can achieve 99.76% accuracy on the IMED Dataset with a learning rate of 0.001 and epoch 300, however, when tested with faces other than the IMED Dataset via webcam, the accuracy drops to 93.63%.

Chowanda, Andry [17] experimented by creating his own CNN architecture called ARCH-1. The architecture created uses the Ensemble method and separation between the Convolution layers and the Residual layers, the model then processes the data in parallel to obtain benefits from both approaches, the model is able to store long-term information using residual connections while having the robustness of convolution layers. From this architecture, the model can reach up to 97.1% test on the IMED Dataset.

Nafis et. al. [18] implemented the YOLOv3 model which is able to recognize facial expressions in real time, including in videos. This model produces an accuracy of 79%, precision 80%, recall 79%, and F1-Score 77% on the IMED Dataset.

From the research above, many researchers still only carry out Facial Emotion Recognition on basic emotions, even though the IMED (Indonesian Mixed Emotion Dataset) dataset already contains mixed facial emotions or compound emotions. but recent studies generally use more complex architecture such as Transformer-based models such as Vision Transformer and Swin Transformer, or Convolution-based Models such as ConvNeXt and YOLOv3. The research above also still uses the finetuned vanilla Vision Transformer model, even though there are already other ViT variants such as Swin Transformer. Other models such as ResNet and CNN which have been developed into ConvNeXt have not been explored enough in Facial Emotion Recognition research. ConvNeXt follows the existing mechanism in Transformer to improve the performance of Convolution Networks such as ResNet and CNN.

Therefore, this research will lead to the implementation of Transformer-based models such as the Vision Transformer as a baseline model, model variants such as the Swin Transformer as a comparison, and Convolution-based models such as ConvNeXt to classify Facial Mixed Emotion Recognition against mixed emotions that are already available in the IMED dataset. After the implementation, the models will be compared between each other regarding the accuracy and performance of each model compared to each other classifying mixed emotions.

3. Proposed Methods

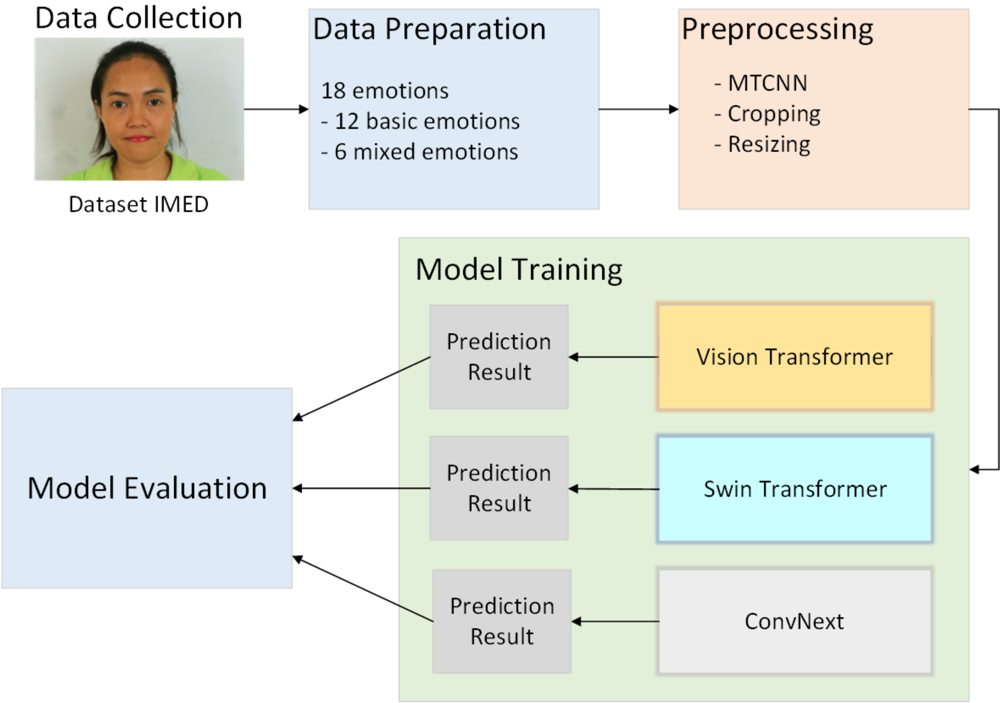


Figure 1. Experiment Design

The proposed methods are divided into several main steps including Data Collection, Data Preparation, Data Preprocessing, Model Training, and Model Evaluation. Each step is shown in Figure 1 above. The first step is Data Collection, where datasets are collected to conduct the experiment, datasets such as FER-2013 are not sufficient because the label only contains basic emotions, and hence IMED Dataset is chosen for this experiment. After collecting the data, they are prepared before being processed in the next step. In data preprocessing, several methods are applied to the data, including MTCNN to obtain the face segment of each image, cropping, resizing, and storing it to train the models, The models (Vision Transformer, Swin Transformer, and ConvNeXt) used the PyTorch library to ensure consistency and easy reproduction of this study; however, each model linear head classifier is modified to match and fit the data, finally, the model is then trained with the preprocessed data using several prepared configurations. After training, the models are evaluated and compared with each other to obtain the result of the experiment.

3.1. Data Collection

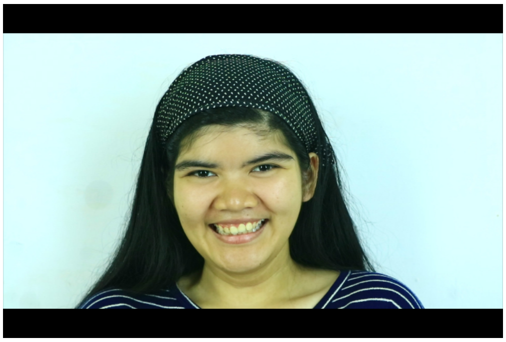


Figure 2. Example data with Happy label

To conduct the experiment, the Dataset used is IMED (Indonesian Mixed Emotion Dataset) which consists of 19 categories of emotions, where it contains 12 complex mixed emotions (e.g. Angrily-Disgusted, Happy-Disgust, etc.) and 7 basic single emotions (Angry, Disgust, Fear, etc.), all of the data are taken from 15 Indonesian subjects (9 females and 6 males) where each subject perform various facial expressions and validated by experts, the dataset contains a total of 281 videos and 30,254 images with size of 720x480 pixels made from each frame of each video.

3.2. Data Preparation

Not all data are used for this experiment, for example, Neutral is not used because it is not considered one of the basic emotions, and the amount of data is negligible compared to other emotions, to ensure authenticity and reduce complexity, the authors decided to not use Neutral because of the sheer low amount of data contained in the dataset. Because of that, the data used are 18 categories of emotions, which consist of 12 mixed emotions, and 6 single emotions which are Angry, Disgust, Fear, Happy, Sad and Surprised. The data is then randomly shuffled and stored into Training, Testing, and Validation Dataset with the ratio of 60:20:20 and the data are split evenly with stratified sampling according to the emotion class.

3.3. Data Preprocessing

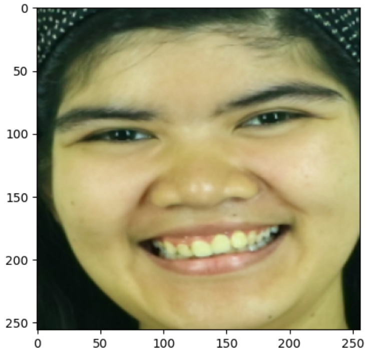


Figure 3. Example of MTCNN Preprocessing Result, Cropped and resized to 256 x 256 pixels

Several methods are used in the preprocessing pipeline, such as MTCNN, Cropping, and resizing the image. In this study, we deliberately try not to add augmented data because this study aims to research mixed emotion from real-life faces, which means that data are processed to include facial data and remove noise from the dataset without additional preprocessing, MTCNN is used to ensure that facial data and facial landmarks are fed to models because, in the dataset, there are regions in the image that do not contain information or facial features (such as white background, the subject clothes, and image border). MTCNN might produce false positives which can result in incorrect preprocessing, but based on our observations, this occurs very rarely because of the IMED dataset's high image quality and front-facing view of subject faces.

After processing all of the datasets using MTCNN, the dataset now only contains faces, but due to the limitation of using MTCNN, some faces are not recognized which reduces the data available for training. Figure 3 shows an example of a sample after preprocessing. Because MTCNN obtains facial features from images, the size is variable between each data, to ensure consistency and can be fed to the model easily, all data is then resized to 256x256, this will slightly impact recognition performance as sometimes facial features will look stretched, but overall the effect toward model performance is negligible since most of the images bounding box from MTCNN has similar size. The distribution of the dataset after preprocessing is shown below in Figure 4.

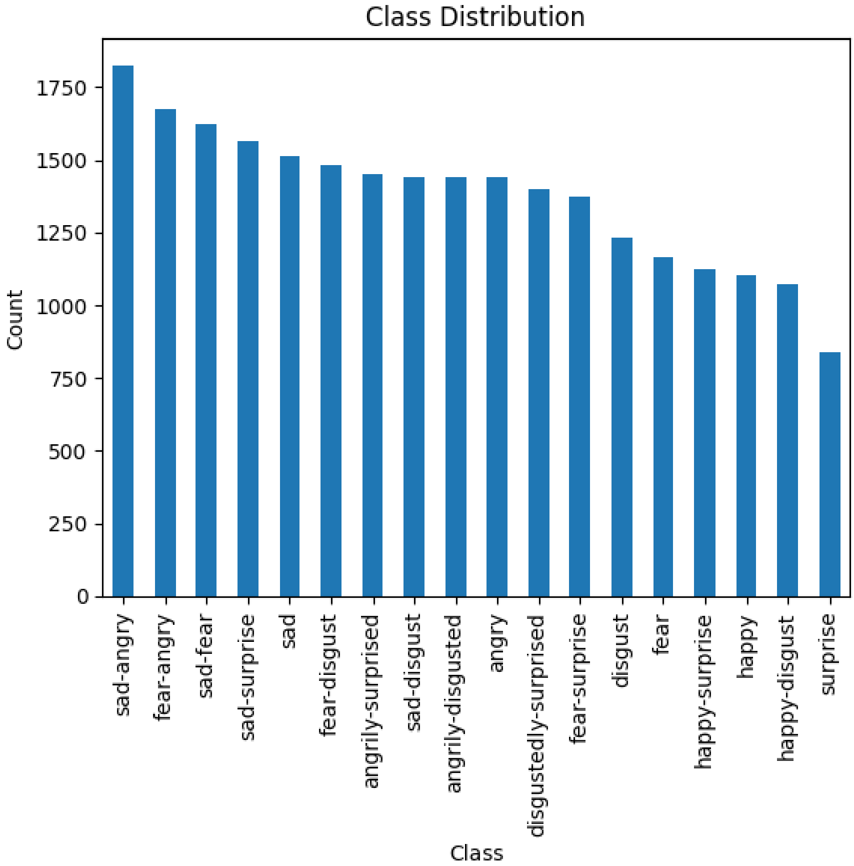


Figure 4. Class Distribution of 18 emotions used in this experiment

3.4. Model Training

In this research, 3 different models are implemented using the PyTorch library including Vision Transformer (ViT) Base-16 Model, ConvNeXt Base Model, and Swin Transformer v2 Model. Models are trained by fine-tuning hyperparameters such as changing the initial Learning rate and Optimizer. Furthermore, for each model, 3 different configurations are used, the hyperparameter tuning for each experiment is shown in Table 1.

Table 1. Initial Hyperparameter

Hyperparameter	**ViT/B-16**			**ConvNeXt/B**			**SwiTv2/B**		
	A	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
Optimizer	Adam	SGD	SGD	Adam	SGD	SGD	Adam	SGD	SGD
Epoch	100	100	100	100	100	100	100	100	100
Patience	10	10	10	10	10	10	10	10	10
Batch Size	32	32	32	32	32	32	32	32	32
Initial LR	0.001	0.01	0.1	0.001	0.01	0.1	0.001	0.01	0.1
Decay LR	No	No	Yes	No	No	Yes	No	No	Yes
Step Size	-	-	20	-	-	20	-	-	20

For each model, three different optimizers are used: the Adam optimizer, Stochastic Gradient Descent (SGD), and SGD with a decaying learning rate. This study aims to observe the effect of each optimizer on each model. To ensure consistency, a few parameters (Epochs, Patience, and Batch Size) are deliberately set to the same values. Since each model is trained using the Early Stopping method, training will end prematurely if the model's validation accuracy does not improve after 10 consecutive iterations. Our observations indicate that a batch size of 32 strikes a balance between training speed and computational requirements. Lower batch sizes result in slower training and worse performance, while batch sizes higher than 32 place significant strain on computational resources. Learning rates were selected based on initial training experiments; Adam starts with a learning rate of 0.001 due to its adaptive nature, which adjusts the learning rate dynamically based on the first and second moments of the gradients, allowing it to correct itself naturally [19]. On the other hand, SGD requires a higher initial learning rate to converge faster, and for SGD with a decaying learning rate, the learning rate is initially set higher to enable faster convergence and to fine-tune model weights in later iterations [20].

3.4.1. Vision Transformer

The first model is Vision Transformer by Dosovitskiy et. al. [21], which is a Transformer-based model that uses Transformer Encoder to encode information from images to a series of patches (chunks of image) alongside each patch positional information. This information means that the image is now represented as the combination of weight, context, and the relationship between each patch as matrices, and employs the Self-Attention mechanism to determine which patches are more important relative to each other and result in Attention score, a higher score means that patch is more relevant in determining the class of the image, furthermore, it also captures long-range dependencies, to calculate the Attention score, we can use the following formula:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

Where for each patch, compute the dot product between each query (Q) across all keys (K^T), this calculates the score of how much attention this patch should pay compared to other patches, because the score may result in large values especially if the d_k value is large, the score is scaled with $\sqrt{d_k}$ so that the value is stable, applying softmax function will convert the value into probabilities, if the value is high, then it indicates that the patch has similar context toward another patch, finally, multiply the result by the value matrix (V) to obtain matrices that represent a compressed information that contains the context and the position of the patch.

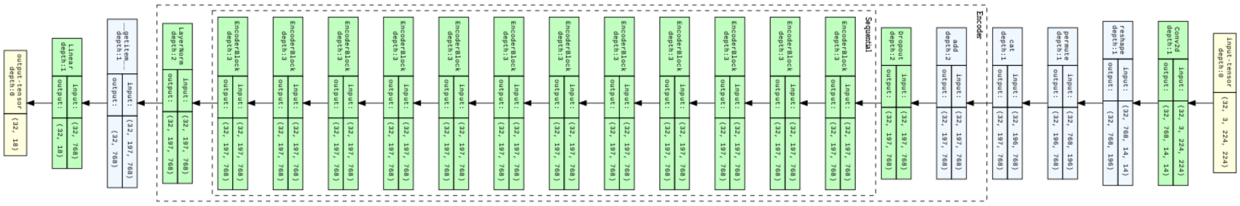

input-tensor depth:0 (32, 3, 224, 224)
Conv2d depth:1 input: (32, 3, 224, 224) output: (32, 768, 14, 14)
reshape depth:1 input: (32, 768, 14, 14) output: (32, 768, 196)
permute depth:1 input: (32, 768, 196) output: (32, 196, 768)
cat depth:1 input: (32, 196, 768) output: (32, 197, 768)
Encoder
add depth:2 input: (32, 197, 768) output: (32, 197, 768)
Dropout depth:2 input: (32, 197, 768) output: (32, 197, 768)
Sequential
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
EncoderBlock depth:3 input: (32, 197, 768) output: (32, 197, 768)
LayerNorm depth:2 input: (32, 197, 768) output: (32, 197, 768)
__getitem__ depth:1 input: (32, 197, 768) output: (32, 768)
Linear depth:1 input: (32, 768) output: (32, 18)
output-tensor depth:0 (32, 18)

Figure 5. Vision Transformer Modified Architecture

With the help of PyTorch Library, this experiment uses a pre-trained *ViT_B_16* Model, and only modifies the linear head classifier matching the number of emotions, the resulting model can be seen in figure 5. and retrains using the IMED Dataset.

3.4.2. Swin Transformer v2

The second Transformer-based model is Shifted Windows Transformer v2 (Swin Transformer v2) by Liu et. al. [22], which is one of the variant adaptations of the original Vision Transformer architecture [21], this is an improved version of the previous version by the same author, Liu et. al. [23]. Swin Transformer architecture is different compared to Vision Transformer, where Vision Transformer splits an image into same-size tokens, Swin Transformer merges patches using patch-merging layers, effectively reducing the number of tokens. This reduces the number of tokens by a multiple of 4, downsampling the image, and capturing the global context more efficiently. Swin Transformer performs local self-attention by using non-overlapping shifting windows across multiple patches, thus reducing the complexity and computation needed. Vision Transformer is computationally expensive when dealing with high-resolution images, formula 1 describes how Global self-attention works because it needs to compute the dot product between each query (Q) across all keys (K^T), on higher-resolution images, this matrix will be huge which is expensive to compute. Because of that, in the original Swin Transformer, the formula is modified to the following:

$$Attention_{window}(Q, K, V) = softmax(\frac{Q_{window}K^T_{window}}{\sqrt{d_k}} + B)V_{window} \tag{2}$$

In equation 2, Swin Transformer calculates self-attention locally within each window, and with added relative position bias (B), the model can adjust attention scores based on how close or far apart each patch is within a certain window. In the improved version [22], the model uses scaled cosine attention, which replaces the dot-product similarity with cosine similarity, the formula becomes:

$$Sim(q_i, k_j) = \frac{cos(q_i, k_j)}{\tau} + B_{ij} \tag{3}$$

$$Attention_{window}(Q, K, V) = softmax(Sim(q_i, k_j))V_{window} \tag{4}$$

Scaled cosine attention solves some problems in the previous Swin Transformer model [23], such as attention maps frequently being dominated by a few pixel pairs. With Scaled Cosine similarity, where B_{ij} is the relative position bias between pixel i and j and τ is a learnable scalar, non-shared across layers that adjusts during training, providing more stable and optimized training, this makes the model easier to scale and better transfer learning across different window resolutions.

With the help of the PyTorch library, This experiment uses a pre-trained *swin_v2_b* Model and also only modifies the linear head classifier matching the number of emotions, the resulting model can be seen in figure 6.

3.4.3. ConvNeXt

The final model used as a comparison is the ConvNeXt model [24], which only uses a pure Convolutional Network model instead of Transformer-based models. ConvNeXt is a standard ConvNet that incorporates design

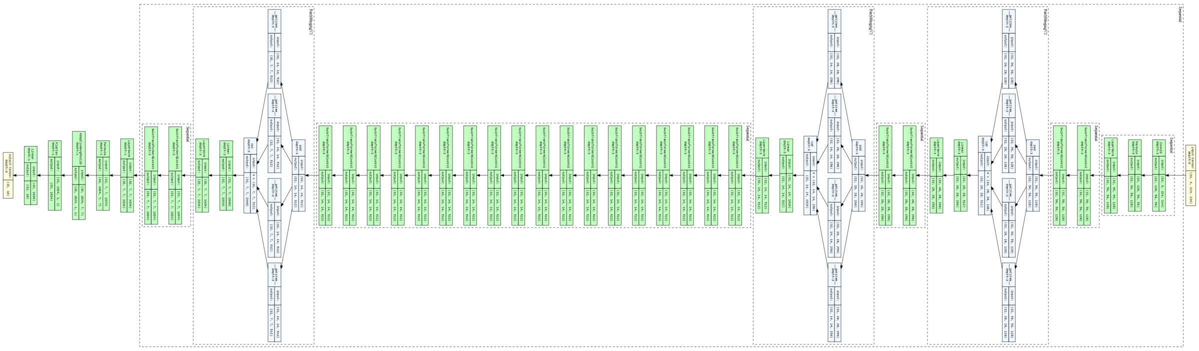


Figure 6. Swin Transformer v2 Modified Architecture

elements of a Vision Transformer, while still retaining the effectiveness of a CNN, with several improvements such as non-overlapping 4x4, stride-4 convolution layer that mimics patch embedding of Vision Transformer to reduce complexity while retaining most of the information, large kernel size (using 7x7 depthwise convolution layer) that enables long-range dependencies, capturing global information and spatial relationships, this mimics how Attention mechanism captures relationships between different patches, and micro design adjustments such as fewer activation functions (one activation function per block), replacing ReLU with GELU activation function, and substituting Batch Normalization (BN) with Layer Normalization (LN). All of this improvements are made to create a ConvNet model and design elements of Vision Transformer, named ConvNeXt [24].

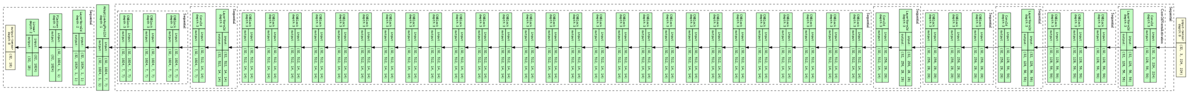


Figure 7. ConvNeXt Modified Architecture

With the help of the PyTorch library, This experiment uses a pre-trained *convnext_base* Model and also only modifies the linear head classifier matching the number of emotions, the resulting model can be seen in figure 7.

On each iteration of training, all models will be evaluated according to Validation Accuracy, which is the accuracy obtained when carrying out detection on the Validation dataset. This is done to ensure that the trained model is not overfitting and has good accuracy in predicting the emotional class of an image. After every iteration, the best-performing model is saved.

3.5. Model Evaluation

After training is finished, all of the models are tested using the Test Dataset, which has been purposefully left behind on classifying never-before-seen data. Model performance measurements are evaluated through the Multi-Class Confusion Matrix by paying attention to performance such as Test Accuracy, namely the accuracy obtained when carrying out detections on the Test dataset that has not been seen by the model before, if there is a significant discrepancy between the test accuracy and the validation accuracy, it indicates that the model is struggling to generalize effectively to unseen data (Overfitting/Underfitting). Other indicators including Weighted Average Recall, Weighted Average Precision, and F1-Score further determine the model performance. The results are then evaluated and compared with each other to determine the best-performing model.

4. Results and Discussion

Results of each model training are presented in this section, for each model, 3 experiments are run using different optimizers, including Adam, SGD, and SGD with decaying Learning Rate. The objective is to find the highest validation accuracy from each experiment and consider the best result from each model.

4.1. Training and Validation Result

Models are trained using the initial hyperparameter explained in Table 1, 3 different settings are used by changing optimizer, initial Learning Rate, and adjusting the decay rate. 3 Models will be trained which are Vision Transformer model, Swin Transformer v2 model, and ConvNeXt model. In the next section, each model training and validation result will be shown and discussed.

4.1.1. ViT/B-16 Model

Table 2. Experiment Result ViT-Base 16 Model

Configuration	Epoch	Train		Validation	
		Acc.	Loss	Acc.	Loss
Adam	90	0.8281	0.5996	0.7824	0.7276
SGD	100	0.7296	0.9744	0.7113	1.0393
SGD+StepLR	37	0.8460	0.6377	**0.7957**	0.9973

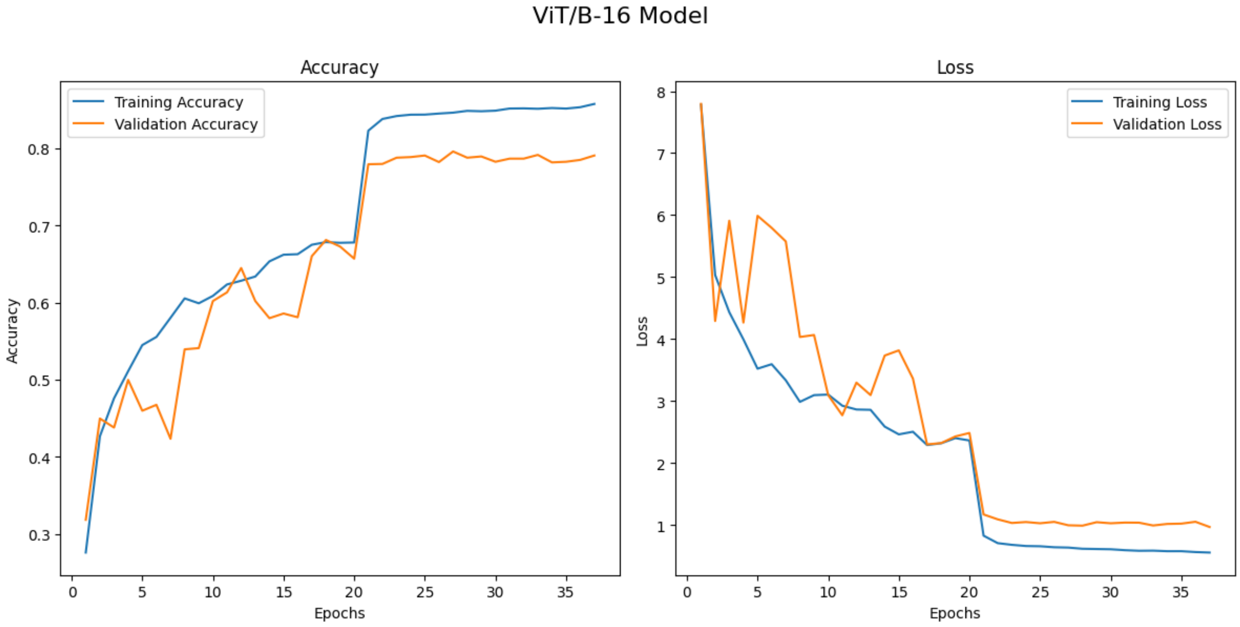


Figure 8. Training and Validation Result of ViT/B-16 Model with SGD Learning Rate 0.1 and Decay of 0.1 every 20 epochs

From the training above, it is shown that Vision Transformer using SGD Optimizer with 0.1 Learning Rate and decaying Learning Rate by 0.1 every 20 epochs trained the best model (Configuration SGD+StepLR). Note that the 3 configurations show different iterations due to the Early Stopping mechanism after 10 consecutive iterations of no improvements toward Validation Accuracy. The Model managed to reach 84.60% Training Accuracy, 0.6377 Training Loss, 79.57% Validation Accuracy, and 0.9973 Validation Loss. The training process takes roughly 298 seconds every epoch and is shown below in Figure 8.

4.1.2. Swin Transformer v2/B Model

Table 3. Experiment Result Swin Transformer v2 Model

Configuration	**Epoch**	**Train**		**Validation**	
		Acc.	**Loss**	**Acc.**	**Loss**
Adam	77	0.4748	1.7363	0.6076	1.3916
SGD	64	0.4238	1.9303	0.5240	1.7201
SGD+StepLR	73	0.5001	1.8406	**0.6528**	1.1259

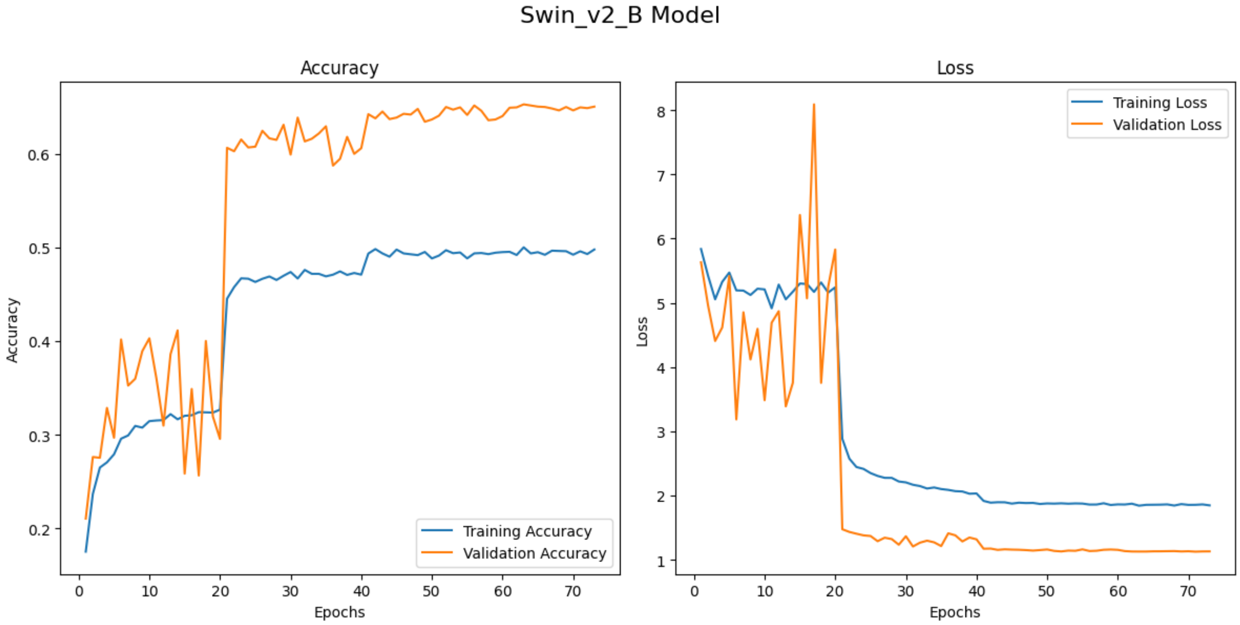


Figure 9. Training and Validation Result of Swin_v2_B Model with SGD Learning Rate 0.1 and Decay of 0.1 every 20 epochs

Swin Transformer v2 also performs best using SGD Optimizer with 0.1 Learning Rate and decaying Learning Rate by 0.1 every 20 epochs trained the best model (Configuration SGD+StepLR). However, because of the high Learning Rate (0.1), the result varies wildly during the early iterations because of huge weight adjustment after every iteration, but once reaching Epoch 20, the Learning Rate decays to 0.01, reaching local maxima. The Swin Transformer v2 Model reached 50.01% Training Accuracy, 1.8406 Training Loss, 65.28% Validation Accuracy, and 1.1259 Validation Loss. The training takes roughly 350 seconds every epoch, which is consideribly slower than Vision Transformer, this mostly happens because the dimension of the image is small enough (256x256), where Swin Transformer would perform multiple layers of attention in this small image compared to Vision Transformer. If the image resolution is higher, Vision Transformer would suffer due to scaling quadratically with the number of patches because of global self-attention. The training process is shown below in Figure 9.

4.1.3. ConvNeXt Base Model

Table 4. Experiment Result ConvNeXt Base Model

Configuration	Epoch	Train		Validation	
		Acc.	Loss	Acc.	Loss
Adam	82	0.6083	1.2861	0.7158	0.9918
SGD	98	0.5373	1.5841	0.6306	1.3934
SGD+StepLR	60	0.6356	1.2953	**0.7626**	0.7885

The final model, ConvNext also performs best using SGD Optimizer with 0.1 Learning Rate and decaying Learning Rate by 0.1 every 20 epochs trained the best model (Configuration SGD+StepLR). Similar to Swin Transformer, because of the high Learning Rate (0.1), the result varies wildly during the early iterations because of huge weight adjustments after every iteration, but once reaching Epoch 20, the Learning Rate decays to 0.01, thus reducing the weight update. The ConvNeXt Base Model reached 63.56% Training Accuracy, 1.2953 Training Loss, 76.26% Validation Accuracy, and 0.7885 Validation Loss. The training takes roughly 245 seconds every epoch, which is expected because of ConvNext model architecture simplicity compared to ViT and Swin Transformer. The training process is shown below in Figure 10.

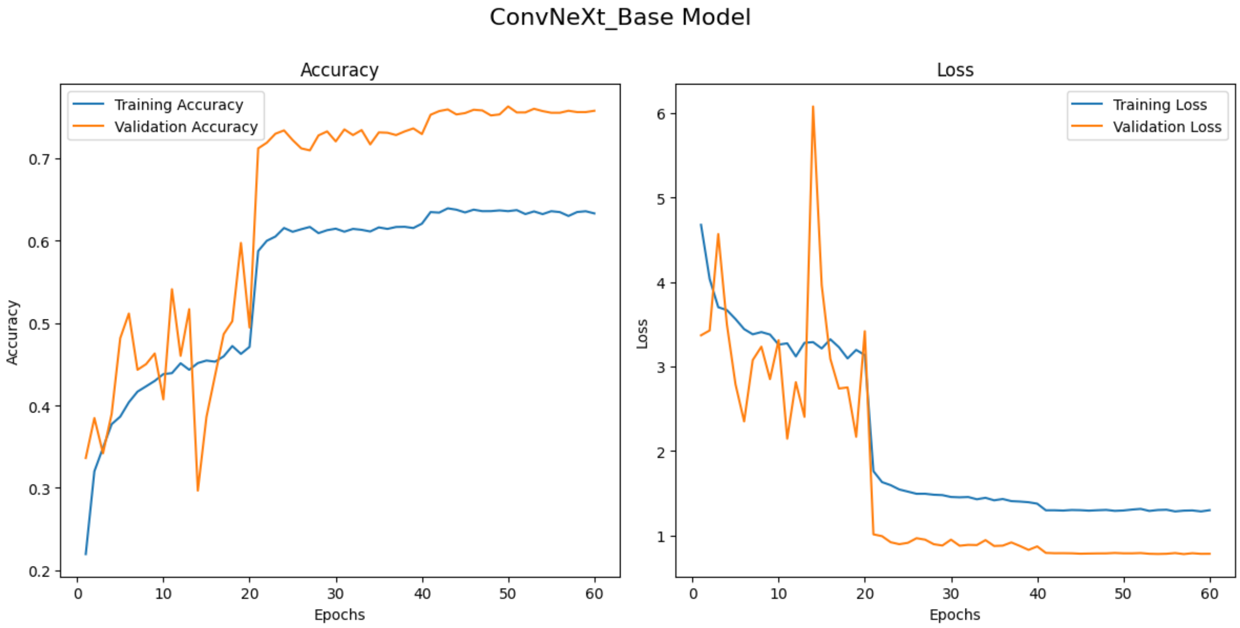


Figure 10. Training and Validation Result of ConvNeXt_Base Model with SGD Learning Rate 0.1 and Decay of 0.1 every 20 epochs

4.1.4. *Training and Validation Summary*

Table 5. Summary of Training and Validation from models

Model	Configuration	Train		Validation	
		Acc.	Loss	Acc.	Loss
ViT	SGD+StepLR	**0.8460**	**0.6377**	**0.7957**	0.9973
Swin v2	SGD+StepLR	0.5001	1.8406	0.6528	1.1259
ConvNeXt	SGD+StepLR	0.6356	1.2953	0.7626	**0.7885**

After performing training and validation using the 3 models, the results are summarized in Table 5 above. Based on the summary, Vision Transformer performs generally better, beating both Swin Transformer and ConvNeXt model in regards to Training Accuracy with 84.60%, Training Loss with 0.6377, and Validation Accuracy with 79.57%. The model ConvNeXt has 63.56% Training Accuracy, 1.2953 Training Loss, 76.26% Validation Accuracy, and 0.7885 Validation Loss, beating Vision Transformer in regards to Validation Loss significantly indicating ConvNeXt can deliver more consistent results, albeit predicts the correct emotion less frequently. The last model is Swin Transformer v2 with a Training Accuracy of 50.01%, Training Loss of 1.8406, Validation Accuracy of 65.28%, and Validation Loss of 1.1259, performing the worst compared to other models because Swin Transformer tends to converge much later, exhausting the training patience and underfit due to low resolution and low amount of data on certain labels.

4.2. *Testing Result*

After training the models, the best configuration for each model is tested against the Test Dataset to ensure that the model can recognize facial emotion on never-seen data. The Test dataset consists of 20% of overall data that has been divided evenly across categories of emotions, we will use weighted Average Recall, Weighted Average Precision, Weighted Average F1-Score, and Overall Accuracy.

4.2.1. *Vision Transformer Model Testing Result*

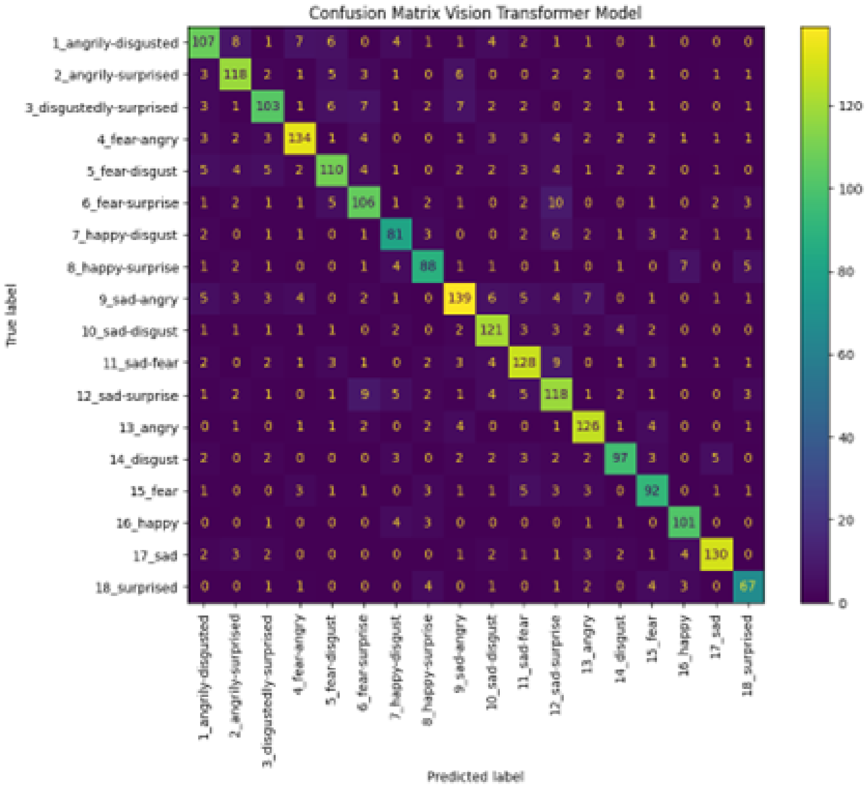
Using the '*scikit-learn*' library, we can visualize Vision Transformer Testing Result using Heatmap which can be seen in Figure 11, looking at each category Weighted Average F1-Score, it shows that Vision Transformer struggles the most in classifying Sad-Surprise emotions with a F1-Score of 72.39%, followed by Happy-Disgust emotions with a F1-Score of 75.35%. From the Testing Result, Vision Transformer model achieved 79.49% Weighted Average Precision, 79.37% Weighted Average Recall, 79.37% Weighted Average F1-Score, and Overall Test Accuracy of 79.37%. Comparing the slight difference between Validation and Test Accuracy, this means that the Vision Transformer model can perform well on unseen data.

4.2.2. *Swin Transformer v2 Model Testing Result*

Swin Transformer v2 is struggling in a few categories such as Angrily-Disgusted emotions (F1-Score of 59.09%), Sad-Surprise emotions (F1-Score of 53.81%), and Sad-Fear emotions (F1-Score of 58.92%) with many misclassifications shown in Figure 12. From the Testing Result, the Swin Transformer v2 model achieved 68.12% Weighted Average Precision, 65.36% Weighted Average Recall, 65.59% Weighted Average F1-Score, and Overall Test Accuracy of 65.36%. Overall, Swin Transformer v2 performs the worst compared to the other models.

4.2.3. *ConvNeXt Model Testing Result*

ConvNeXt struggles on different categories than Swin Transformer, which is Sad-Surprise emotions (F1-Score of 65.19%) and Fear-Surprise emotions (F1-Score of 66.90%). On other emotions, ConvNeXt reaches an average of 70-80% F1-Score shown in Figure 13. From the Testing Result, the ConvNeXt model achieved 75.31% Weighted Average Precision, 74.77% Weighted Average Recall, 74.71% Weighted Average F1-Score, and Overall Test



	1_angrily-disgusted	2_angrily-surprised	3_disgustedly-surprised	4_fear-angry	5_fear-disgust	6_fear-surprise	7_happy-disgust	8_happy-surprise	9_sad-angry	10_sad-disgust	11_sad-fear	12_sad-surprise	13_angry	14_disgust	15_fear	16_happy	17_sad	18_surprised
1_angrily-disgusted	107	8	1	7	6	0	4	1	1	4	2	1	1	0	1	0	0	0
2_angrily-surprised	3	118	2	1	5	3	1	0	6	0	0	2	2	0	1	0	1	1
3_disgustedly-surprised	3	1	103	1	6	7	1	2	7	2	2	0	2	1	1	0	0	1
4_fear-angry	3	2	3	134	1	4	0	0	1	3	3	4	2	2	2	1	1	1
5_fear-disgust	5	4	5	2	110	4	1	0	2	2	3	4	1	2	2	0	1	0
6_fear-surprise	1	2	1	1	5	106	1	2	1	0	2	10	0	0	1	0	2	3
7_happy-disgust	2	0	1	1	0	1	81	3	0	0	2	6	2	1	3	2	1	1
8_happy-surprise	1	2	1	0	0	1	4	88	1	1	0	1	0	1	0	7	0	5
9_sad-angry	5	3	3	4	0	2	1	0	139	6	5	4	7	0	1	0	1	1
10_sad-disgust	1	1	1	1	1	0	2	0	2	121	3	3	2	4	2	0	0	0
11_sad-fear	2	0	2	1	3	1	0	2	3	4	128	9	0	1	3	1	1	1
12_sad-surprise	1	2	1	0	1	9	5	2	1	4	5	118	1	2	1	0	0	3
13_angry	0	1	0	1	1	2	0	2	4	0	0	1	126	1	4	0	0	1
14_disgust	2	0	2	0	0	0	3	0	2	2	3	2	2	97	3	0	5	0
15_fear	1	0	0	3	1	1	0	3	1	1	5	3	3	0	92	0	1	1
16_happy	0	0	1	0	0	0	4	3	0	0	0	0	1	1	0	101	0	0
17_sad	2	3	2	0	0	0	0	0	1	2	1	1	3	2	1	4	130	0
18_surprised	0	0	1	1	0	0	0	4	0	1	0	1	2	0	4	3	0	67

Figure 11. Heatmap of Vision Transformer Testing Result

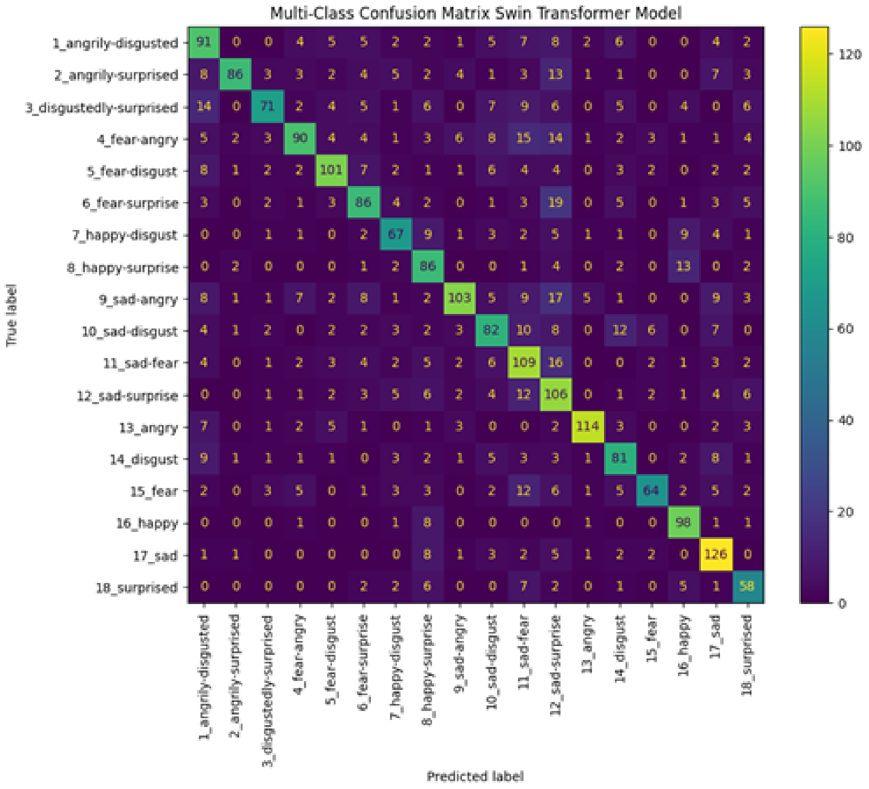
Accuracy of 74.77%. Furthermore, looking at ConvNeXt Validation Accuracy compared to the Test Accuracy indicates a well-generalizing model.

4.2.4. Testing Summary

Table 6. Summary of Testing the models using Test Dataset

Model	Test			
	Accuracy	Precision	Recall	F1-Score
ViT	**79.37%**	**79.49%**	**79.37%**	**79.37%**
Swin v2	65.36%	68.12%	65.36%	65.59%
ConvNeXt	74.77%	75.31%	74.77%	74.71%

The results and comparison between each model are shown in Table 6 above. It is shown that the Vision Transformer model performed the best with 79.37% Average Test Accuracy, 79.49% Average Precision, 79.37% Average Recall, and 79.37% F1-Score. ConvNeXt model in second place with 74.77% Average Accuracy, 75.31% Average Precision, 74.77% Average Recall, and 74.71% Average F1-Score. Swin Transformer v2 in third place with 65.36% Average Accuracy, 68.12% Average Precision, 65.36 % Average Recall, 65.59 % Average F1-Score, significantly lower than both Vision Transformer and ConvNeXt model. This shows that both Vision Transformer and ConvNeXt models are good candidates to perform Facial Mixed Emotion Recognition tasks, with Vision



	1_angrily-disgusted	2_angrily-surprised	3_disgustedly-surprised	4_fear-angry	5_fear-disgust	6_fear-surprise	7_happy-disgust	8_happy-surprise	9_sad-angry	10_sad-disgust	11_sad-fear	12_sad-surprise	13_angry	14_disgust	15_fear	16_happy	17_sad	18_surprised
1_angrily-disgusted	91	0	0	4	5	5	2	2	1	5	7	8	2	6	0	0	4	2
2_angrily-surprised	8	86	3	3	2	4	5	2	4	1	3	13	1	1	0	0	7	3
3_disgustedly-surprised	14	0	71	2	4	5	1	6	0	7	9	6	0	5	0	4	0	6
4_fear-angry	5	2	3	90	4	4	1	3	6	8	15	14	1	2	3	1	1	4
5_fear-disgust	8	1	2	2	101	7	2	1	1	6	4	4	0	3	2	0	2	2
6_fear-surprise	3	0	2	1	3	86	4	2	0	1	3	19	0	5	0	1	3	5
7_happy-disgust	0	0	1	1	0	2	67	9	1	3	2	5	1	1	0	9	4	1
8_happy-surprise	0	2	0	0	0	1	2	86	0	0	1	4	0	2	0	13	0	2
9_sad-angry	8	1	1	7	2	8	1	2	103	5	9	17	5	1	0	0	9	3
10_sad-disgust	4	1	2	0	2	2	3	2	3	82	10	8	0	12	6	0	7	0
11_sad-fear	4	0	1	2	3	4	2	5	2	6	109	16	0	0	2	1	3	2
12_sad-surprise	0	0	1	1	2	3	5	6	2	4	12	106	0	1	2	1	4	6
13_angry	7	0	1	2	5	1	0	1	3	0	0	2	114	3	0	0	2	3
14_disgust	9	1	1	1	1	0	3	2	1	5	3	3	1	81	0	2	8	1
15_fear	2	0	3	5	0	1	3	3	0	2	12	6	1	5	64	2	5	2
16_happy	0	0	0	1	0	0	1	8	0	0	0	0	1	0	0	98	1	1
17_sad	1	1	0	0	0	0	0	8	1	3	2	5	1	2	2	0	126	0
18_surprised	0	0	0	0	0	2	2	6	0	0	7	2	0	1	0	5	1	58

Figure 12. Heatmap of Swin Transformer v2 Testing Result

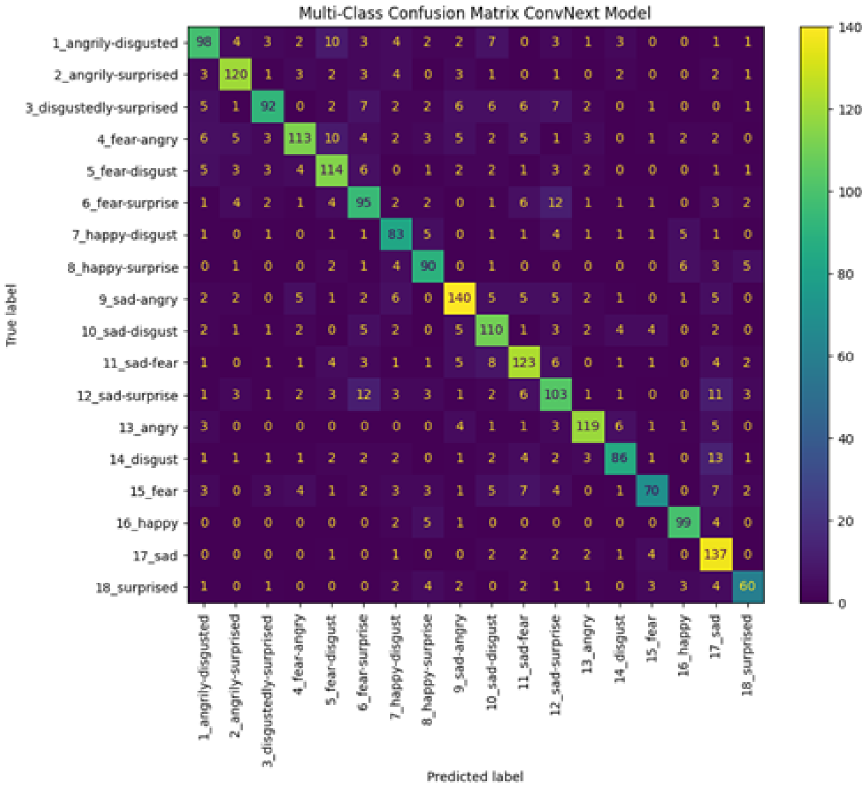
Transformer being slightly better than ConvNeXt. Furthermore, All of the models' Testing Accuracy is similar to their respective Validation Accuracy, which means that each model can generalize well on unseen data.

5. Conclusion and Future Works

This research aims to evaluate and compare 3 models in classifying mixed emotions from facial expressions using the IMED Dataset. After experimenting, all models perform the best using a Learning Rate of 0.1, with a Stochastic Gradient Descent (SGD) Optimizer and decaying Learning Rate. Based on the test results, it can be concluded the model that has the best performance is the Vision Transformer model with an Average Testing Accuracy performance of 79.37%, outperforming both the Swin Transformer v2 model with an Average Testing Accuracy of 65.36%, and ConvNeXt model with an Average Testing Accuracy of 74.77%.

However, we have identified several areas which can be improved further, such as augmenting and exploring other preprocessing techniques such as adjusting image contrast to increase data quality, which in turn will improve model performance. This research is limited to using sufficient preprocessing techniques such as Cropping and Resizing to ensure that Mixed Emotion recognition can be researched further with minimum image processing and no augmented data. In the IMED Dataset, only a few preprocessing techniques are used to ensure that facial points are fed to the model correctly. We can also add or mix other datasets to further prove the generalizability of these models in classifying mixed emotions, this experiment only uses IMED Dataset to confirm that the models are capable of classifying mixed emotions from facial expressions.

Additionally, exploring options by experimenting with hyperparameter finetuning, the configurations used in this research are limited due to limitations in currently available computing power and time frame, training is limited



	1_angrily-disgusted	2_angrily-surprised	3_disgustedly-surprised	4_fear-angry	5_fear-disgust	6_fear-surprise	7_happy-disgust	8_happy-surprise	9_sad-angry	10_sad-disgust	11_sad-fear	12_sad-surprise	13_angry	14_disgust	15_fear	16_happy	17_sad	18_surprised
1_angrily-disgusted	98	4	3	2	10	3	4	2	2	7	0	3	1	3	0	0	1	1
2_angrily-surprised	3	120	1	3	2	3	4	0	3	1	0	1	0	2	0	0	2	1
3_disgustedly-surprised	5	1	92	0	2	7	2	2	6	6	6	7	2	0	1	0	0	1
4_fear-angry	6	5	3	113	10	4	2	3	5	2	5	1	3	0	1	2	2	0
5_fear-disgust	5	3	3	4	114	6	0	1	2	2	1	3	2	0	0	0	1	1
6_fear-surprise	1	4	2	1	4	95	2	2	0	1	6	12	1	1	1	0	3	2
7_happy-disgust	1	0	1	0	1	1	83	5	0	1	1	4	1	1	1	5	1	0
8_happy-surprise	0	1	0	0	2	1	4	90	0	1	0	0	0	0	0	6	3	5
9_sad-angry	2	2	0	5	1	2	6	0	140	5	5	5	2	1	0	1	5	0
10_sad-disgust	2	1	1	2	0	5	2	0	5	110	1	3	2	4	4	0	2	0
11_sad-fear	1	0	1	1	4	3	1	1	5	8	123	6	0	1	1	0	4	2
12_sad-surprise	1	3	1	2	3	12	3	3	1	2	6	103	1	1	0	0	11	3
13_angry	3	0	0	0	0	0	0	0	4	1	1	3	119	6	1	1	5	0
14_disgust	1	1	1	1	2	2	2	0	1	2	4	2	3	86	1	0	13	1
15_fear	3	0	3	4	1	2	3	3	1	5	7	4	0	1	70	0	7	2
16_happy	0	0	0	0	0	0	2	5	1	0	0	0	0	0	0	99	4	0
17_sad	0	0	0	0	1	0	1	0	0	2	2	2	2	1	4	0	137	0
18_surprised	1	0	1	0	0	0	2	4	2	0	2	1	1	0	3	3	4	60

Figure 13. Heatmap of ConvNeXt Testing Result

to 3 initial hyperparameter configurations for each model. Parameters that can be adjusted include adjusting the initial Learning Rate, higher Patience before stopping the training early, using other optimizers such as Adagrad or RMSProp, and higher batch size.

Furthermore, implementing and evaluating other Transformer-based model architectures such as DeiT (Data-Efficient Image Transformer) or CvT (Convolutional Vision Transformer), or comparing other Convolution-based model architectures such as EfficientNet or RepVGG will further enhance studies in recognizing mixed emotions from facial expression.

Acknowledgement

The authors would also like to extend our gratitude to Dewi Yanti Liliana et. al. for allowing and providing access to IMED Dataset.

REFERENCES

1. De la Torre, F., and Cohn, J. F., *Facial expression analysis*, Visual analysis of humans: Looking at people, pp. 377–409, 2011.
2. Siedlecka, E., and Denson, T. F., *Experimental methods for inducing basic emotions: A qualitative review*, Emotion Review, vol. 11, no. 1, pp. 377–409, 2019.
3. Paul Ekman, *Basic emotions*, Handbook of cognition and emotion, vol. 98, pp. 45–60, 1999.
4. Larsen, J. T., and McGraw, A. P., *Further evidence for mixed emotions*, Journal of personality and social psychology, vol. 100, no. 6, pp. 1095, 2011.
5. Williams, P., and Aaker, J. L., *Can mixed emotions peacefully coexist?*, Journal of consumer research, vol. 28, no. 4, pp. 636–649, 2002.

6. Vijayalakshmi, A., and Mohanaiah, P., *Literature survey on emotion recognition for social signal processing*, Advances in Communication, Signal Processing, VLSI, and Embedded Systems: Select Proceedings of VSPICE 2019, pp. 345–360, 2019.
7. Kołakowska, A., Landowska, A., Szwoch, M., Szwoch, W., and Wrobel, M. R., *Emotion recognition and its applications*, Human-Computer Systems Interaction: Backgrounds and Applications 3, pp. 51–62, 2014.
8. Pessoa, L., *Understanding emotion with brain networks*, Current opinion in behavioral sciences, vol.19, pp. 19–25, 2018.
9. Larsen, J. T., and McGraw, A. P., *The case for mixed emotions*, Social and Personality Psychology Compass, vol. 8, no. 6, pp. 263–274, 2014.
10. Liliana, D. Y., Basaruddin, T., and Oriza, I. I. D., *The indonesian mixed emotion dataset (imed) a facial expression dataset for mixed emotion recognition*, Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality, pp. 56–60, 2018.
11. Han, S., Chang, H., Shi, Z., and Hu, S., *Facial Expression Recognition Algorithm Based on Swin Transformer*, 2023 9th International Conference on Systems and Informatics (ICSAI), pp. 1–6, 2023.
12. Li, H., Sui, M., Zhao, F., Zha, Z., and Wu, F., *MVT: mask vision transformer for facial expression recognition in the wild*, arXiv preprint arXiv:2106.04520, 2021.
13. El-Khashab, O., Hamdy, A., and Mahmoud, A., *FerNeXt: Facial Expression Recognition Using ConvNeXt with Channel Attention*, 2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), pp. 1–8, 2023.
14. El Boudouri, Y., and Bohi, A., *EmoNeXt: an Adapted ConvNeXt for Facial Emotion Recognition*, 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6, 2023.
15. Liliana, D. Y., Basaruddin, T., and Widyanto, M. R., *Fuzzy emotion recognition using semantic facial features and knowledge-based fuzzy*, International Journal of Engineering and Technology, vol. 11, no. 2, pp. 177–186, 2019.
16. Jala, A. B., Purboyo, T. W., and Nugrahaeni, R. A., *Implementation of convolutional neural network (cnn) algorithm for classification of human facial expression in indonesia*, 2020 International Conference on Information Technology Systems and Innovation (ICITSI), pp. 256–262, 2020.
17. Chowanda, Andry, *Separable convolutional neural networks for facial expressions recognition*, Journal of Big Data, vol. 8, no. 1, pp. 132, 2021.
18. Nafis, A. F., Navastara, D. A., and Yuniarti, A., *Facial expression recognition on video data with various face poses using deep learning*, 2020 12th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 362–367, 2020.
19. Kingma, D. P., *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, 2014.
20. Stich, S. U., *Local SGD converges fast and communicates little*, arXiv preprint arXiv:1805.09767, 2018.
21. Dosovitskiy et. al., *An image is worth 16x16 words: Transformers for image recognition at scale*, arXiv preprint arXiv:2010.11929., 2020.
22. Liu et. al., *Swin transformer v2: Scaling up capacity and resolution*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12009–12019, 2022.
23. Liu et. al., *Swin transformer: Hierarchical vision transformer using shifted windows*, Proceedings of the IEEE/CVF international conference on computer vision, pp. 10012–10022, 2021.
24. Liu et. al., *A convnet for the 2020s*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 11976–11986, 2022.