

Channel-Wise Attention Based Xception to Recognise Facial Emotions of Children with ASD

by

**Farjahan Akter Bobby (201014007)
Mahfuja Akter Shifa (201014001)
Rudaba Sultana (183014015)**

Capstone project (CSE499) submitted in partial fulfillment of the requirement for the degree of

Bachelor of Science in Computer Science and Engineering

Under the supervision of

Md. Aktaruzzaman Pramanik



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF LIBERAL ARTS BANGLADESH**

SPRING 2025

© *Farjahan Akter Bobby, Mahfuja Akter Shifa, Rudaba Sultana*
All rights reserved

DECLARATION

Project Title Channel-Wise Attention Based Xception to Recognise Facial Emotions of Children with ASD
Authors Farjahan Akter Bobby, Mahfuja Akter Shifa, Rudaba Sultana
Student IDs 201014007, 201014001, 183014015
Supervisor Md. Aktaruzzaman Pramanik

We declare that this project/thesis report entitled *Channel-Wise Attention Based Xception to Recognise Facial Emotions of Children with ASD* is the result of our own work except as cited in the references. The project/thesis report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Farjahan Akter Bobby
201014007

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Mahfuja Akter Shifa
201014001

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Rudaba Sultana
183014015

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Date: May 20, 2025



Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Mohammadpur, Dhaka - 1207

CERTIFICATE

This is to certify that the project/thesis report entitled **Channel-Wise Attention Based Xception to Recognise Facial Emotions of Children with ASD**, submitted by **Farjahan Akter Bobby** (Student ID: 201014007), **Mahfuja Akter Shifa** (Student ID: 201014001), **Rudaba Sultana** (Student ID: 183014015) are undergraduate students of the **Department of Computer Science and Engineering** has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfill the requirements for its acceptance in partial fulfillment for the degree of *Bachelor of Science in Computer Science and Engineering*.

Md. Aktaruzzaman Pramanik, Lecturer
Dept. of CSE, University of Liberal Arts Bangladesh

Md. Nazmul Abdal, Lecturer
Dept. of CSE, University of Liberal Arts Bangladesh

Fatema Khan, Lecturer
Dept. of CSE, University of Liberal Arts Bangladesh

Place: Dhaka

Date: May 20, 2025

ACKNOWLEDGEMENTS

We would like to express our deep and sincere gratitude to our research supervisor, *Md. Aktaruzzaman Pramanik*, for giving us the opportunity to conduct research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity and motivation have deeply inspired us. He has taught us the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

We are greatly indebted to our honorable teachers of the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh who taught us during the course of our study. Without any doubt, their teaching and guidance have completely transformed us to the persons that we are today.

We are extremely thankful to our parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing us for our future. We would like to say thanks to our friends and relatives for their kind support and care.

Finally, we would like to thank all the people who have supported us to complete the project work directly or indirectly.

Farjahan Akter Bobby, Mahfuja Akter Shifa and Rudaba Sultana

University of Liberal Arts Bangladesh

Date: May 20, 2025

Dedicated to my family
the heart of my world
Their unconditional love,
silent sacrifices, and
unwavering faith in me
shaped the person I am
today.

– *Farjahan Akter Bobby*

To my beloved mother, and
my dear siblings,
for their unconditional love,
endless support, and belief in
me every step of the way.

– *Mahfuja Akter Shifa*

To
my dearest mother,
my strength, my forever
best friend, and my
greatest blessing.

– *Rudaba Sultana*

ABSTRACT

In computer vision tasks, convolutional networks are at the core. However, traditional CNN, including the Xception model, focuses on extracting spatial information while giving equal importance to all channels within a local receptive field and fails to capture long-range dependencies, which cause performance degradation, and this model overfits and memorizes noise on small datasets due to its capacity. This observation leads to the proposal of a channel-wise attention-based Xception architecture to improve the performance. The model was then used on an ASD-specific data set to recognize their emotions through facial expressions. The analysis shows that, this model outperforms, achieving macro precision of 96.60%, macro recall of 96.57%, macro F1-score of 96.58% and macro-AUC 99.89%. Thus, the proposed channel-wise attention-based Xception architecture is very robust, and it can perfectly distinguish all the emotions. Finally, an application was proposed utilizing this architecture through which caregivers or parents can recognize ASD children's emotions.

Keywords: class-wise attention, Xception, autism spectrum disorder, facial emotion, ASD

Contents

1	Introduction	1
1.1	Problem statement	2
1.2	Motivation, aims and objectives	2
1.2.1	Motivation	2
1.2.2	Aim and objective	3
2	Literature Review	4
2.1	Relevant Work using Deep Learning Model	4
2.2	Xception	8
2.2.1	Xception Architecture	8
2.2.2	Standard Convolution vs Depthwise Separable Convolution	9
2.2.3	Residual Connection	12
2.2.4	Review of Xception Architecture	12
2.3	Squeeze-and-Excitation Network (SENet)	13
2.4	Evaluation Metrics for Multiclass	14
2.5	Summary of Literature Review	16
3	Methodology	19
3.1	Dataset	19
3.1.1	Data Collection	19
3.1.2	Data Preprocessing	21
3.1.2.1	Undersampling	22
3.1.2.2	Oversampling	22
3.1.2.3	Data split	23
3.2	Proposed Model	23
3.2.1	Proposed Depthwise Separable Convolution	23

3.2.2	Entry Flow of the proposed Xception	25
3.2.3	Middle Flow of the proposed Xception	27
3.2.4	Exit Flow of the proposed Xception	27
3.2.5	Algorithmic Functionality of proposed model	28
3.3	Budget Planning	30
3.4	Modern Tools and Technologies Used	30
4	Result analysis	32
4.1	Xception Base Model Evaluation	32
4.1.1	Loss and Accuracy Curve	32
4.1.2	Classification Report and Confusion Matrix Analysis	33
4.1.3	AUC-ROC Curve Analysis	35
4.2	Proposed Model Evaluation	36
4.2.1	Loss and Accuracy Curve	37
4.2.2	Classification Report and Confusion Matrix Analysis	38
4.2.3	AUC-ROC Curve Analysis	40
4.3	Comparison Between Models	40
4.3.1	Class wise Performance Comparison	40
4.3.2	Metric-wise comparison	41
4.3.3	Computational Efficiency Comparison	42
4.4	Proposed Web Application	43
5	Conclusions	46
5.1	Social and Sustainability Issues	46
5.1.1	Social Impact	46
5.1.2	Sustainability Strategy	47
5.2	Brief Summary	48
5.3	Future works	49
	References	50

List of Figures

2.1	Xception Model Architecture	9
2.2	Standard Convolution	10
2.3	Depthwise Separable Convolution	10
2.4	Squeeze-and-Excitation Block Diagram	14
3.1	Data size visualization across different classes.	20
3.2	Few samples of facial emotions per class	21
3.3	Data size across different classes after increasing data size per class. .	22
3.4	Proposed Depthwise Separable Convolution Block.	24
3.5	The entry flow of the proposed Xception model.	26
3.6	Middle flow of the proposed Xception model.	27
3.7	Exit flow of the proposed Xception model.	28
3.8	Algorithmic functionality of proposed depthwise separable convolution.	29
3.9	Algorithmic functionality of proposed Xception Architecture.	29
3.10	Total budget for our capstone project	30
4.1	Loss and accuracy curve of the Xception model.	33
4.2	Classification report of the Xception model.	34
4.3	Confusion matrix of the Xception model.	35
4.4	AUC-ROC curve of the Xception model.	36
4.5	Train and validation loss of the proposed model.	37
4.6	Training and validation accuracy of the proposed model.	38
4.7	Classification model of the proposed model.	39
4.8	Confusion matrix of the proposed model.	39
4.9	AUC-ROC curve of the proposed model.	40
4.10	Home interface of the proposed web application.	44

4.11 The application is analyzing the uploaded image. 44

4.12 Output interface of emotion recognition. 45

List of Tables

2.1	Performance Metrics	15
2.2	Summary of literature review on the FER task of ASD children. . . .	16
3.1	Mathematical representation of our proposed depthwise separable convolution.	25
4.1	Hyperparameter of the Xception Model.	32
4.2	Hyperparameter of the Proposed Model	36
4.3	Class-wise Performance Comparison of the Base and Proposed Model	41
4.4	Metric-wise comparison between models.	42
4.5	Model Complexity Comparison.	42

Chapter 1

Introduction

For feature extraction of computer vision tasks and to invent more accurate AI systems, the majority of authors applied hands-on feature extraction methods such as local binary pattern (LBP), Gabor filter) [Debnath et al. \(2022\)](#), [Z. Wang et al. \(2021\)](#), sparse learning [Talaat et al. \(2024\)](#), etc. However, for feature extraction, CNN is most promising [Niu et al. \(2020\)](#) and attained excellent outcomes. Nevertheless, traditional CNN, including Xception architecture, mostly focuses on local receptive fields, giving equal importance to all channels. A significant challenge in Xception is that, unlike spatial features, it does not explicitly consider which channels carry more critical information. Furthermore, this model focuses on local receptive fields and, as a result, fails to capture long-range dependencies. This could be mitigated through attention mechanisms [Vaswani et al. \(2017\)](#) [Woo et al. \(2018\)](#).

Building on this, several researchers have increasingly adopted attention mechanisms in various CNN architectures for computer vision tasks like facial emotion recognition (FER) [Rajan et al. \(2022\)](#) [Zhuang et al. \(2022\)](#) [Y. Zhang et al. \(2022\)](#) [Huang et al. \(2023\)](#). Also, [Mao et al. \(2024\)](#) introduced a window-based attention mechanism, [Z. Wang et al. \(2021\)](#) applied an attention branch, and [Le et al. \(2023\)](#) [Kortli et al. \(2020\)](#) applied a self-attention mechanism for both global and local feature extraction. Even for the FER task [Talaat et al. \(2024\)](#) used the Xception model, which reached almost 95.23% accuracy after applying an attention mechanism, but with not unsatisfactory AUC and F1-score, and overfitted as the model is heavy and the dataset is small.

Recognizing these limitations becomes even more critical when focusing on sensitive applications such as facial emotion recognition for children with Autism Spectrum Disorder (ASD) due to poor facial expression. As a result, people cannot recognize the emotions of children with ASD and cannot meet their needs or provide proper treatment. In the human-computer interaction (HCI) field, FER has been applied in areas like medical treatment, psychological analysis, etc. However, in ASD children, extracting critical facial features is more important, and this area needs more attention to assist caregivers or therapists directly. Because the performance of the existing models is far from satisfactory. This urged us to work on this research.

Since in computer vision for FER tasks, the attention mechanism has been proven very effective, and the Xception model gives better performance compared to other traditional classification models due to the depthwise separable convolutional power but overfits and fails to capture long-term dependencies, to mitigate these problems, this paper presents an Channel-wise attention mechanism-based Xception model for recognizing the emotions of special children to improve the performance.

1.1 Problem statement

The Xception architecture improves efficiency through depthwise separable convolutions. To capture long-range dependencies of the Xception model, it fails to perform better; additionally, it does not explicitly consider which channels carry more critical information. Based on precision, recall, and AUC scores, the Xception model had comparatively low performance in distinguishing between different emotions of ASD children. To mitigate this limitation, an improved Xception model is proposed for better accuracy.

1.2 Motivation, aims and objectives

1.2.1 Motivation

Our motivation is to build an improved Xception architecture that will combine channel-wise attention mechanisms to enhance feature extraction for FER tasks in children with Autism Spectrum Disorder (ASD), from the vital need to support not

only ASD children but also their caregivers and therapists by bridging a critical communication gap. Because the Xception model has some limitations for which it struggles to give a better performance, which motivates us to improve the Xception model. Again, research shows that while 33% of ASD children respond normally in terms of emotions, approximately 46% struggle to express or understand, and around 20% face more severe challenges [Faruk et al. \(2023\)](#). This emotional disconnect often leaves caregivers and therapists unsure how to respond effectively [Uddin & Ashrafun \(2023\)](#). Unlike many existing approaches that focus primarily on therapy for ASD individuals [Pavez et al. \(2023\)](#); [Saranya & Anandan \(2022\)](#), very few applications directly assist caregivers or therapists [Talaat et al. \(2024\)](#). In addition, existing models often overfit and fail to capture class-specific emotional nuances. Our goal is to build a reliable and adaptive model that will help caregivers and therapists to understand ASD children's emotional needs.

1.2.2 Aim and objective

- To implement an enhanced deep learning model (Xception architecture) of facial emotion recognition for ASD children.
- To optimise the deep learning model and strengthen interpretability or feature extraction by integrating an attention mechanism

Chapter 2

Literature Review

Facial emotion recognition plays a pivotal role in supporting emotional development and therapeutic interventions for children with ASD. Given the distinct and often subtle ways in which autistic children express emotions, developing accurate, real-time, and contactless systems remains a significant research challenge. Although various deep learning and hybrid approaches have been explored, many existing systems struggle to perform effectively in real-world ASD scenarios, primarily due to the absence of autism-specific datasets and the reliance on models originally designed for neurotypical facial expressions, which fail to capture the unique emotional patterns seen in autistic children.

2.1 Relevant Work using Deep Learning Model

To begin with, [Talaat et al. \(2024\)](#) proposed a deep learning framework combining a sparse kernel autoencoder with a convolutional neural network to recognize facial emotions in children with Autism Spectrum Disorder (ASD). The model architecture includes an autoencoder for both feature extraction and selection, ensuring dimensionality reduction and emphasizing critical facial features. The encoded features are passed to a pre-trained CNN (MobileNet, ResNet, or Xception), with the Xception model achieving performance such as 95.23% accuracy, 93.2% precision, 94.21% recall, F1-score 93.31% and an AUC of 91.34%. The classification targets six fundamental emotions: anger, happy, neutral, fear, sadness, and surprise. The dataset used comprises curated facial images of autistic children, total 1200 training and 220 testing images, with a notable class imbalance like joy and sadness having

the highest representation. The authors also used data augmentation and a small-scale labeled dataset with cleaned and preprocessed images. Despite achieving high accuracy, the study highlights limitations such as the training accuracy is 96.32% and validation accuracy is 88.54% with a loss 1.31. This validation loss is increasing over the epochs indicating overfitting problem. Also, this report did not evaluated class-wise metrics which is crucial for multiclass. And the result can be biased for some classes. These issues suggest a need for refined models capable of handling subtle facial cues in ASD-specific emotional expressions.

[Komuraiah et al. \(2024\)](#) developed a web-based facial emotion recognition system using CNN and the Flask framework to support emotion recognition in children with Autism. The system detects six core emotions such as joy, fear, surprise, sadness, anger, and natural through real-time webcam input. A CNN model was trained on 15,340 labeled images using augmentation such as flipping, rotation, and noise to enhance generalization. The model obtained a low training loss of 0.24 and reported perfect recall, precision, and F1-score for some emotion classes. However, the macro-average F1-score was 91%, indicating variation in performance across classes. This suggests the model may struggle to distinguish certain emotions, which is critical for therapeutic accuracy.

In contrast, [El Rhatassi et al. \(2024\)](#) proposed a facial emotion recognition system tailored for individuals with autism, aiming to support emotional understanding and personalized care through a chatbot interface. The system used convolutional neural networks (CNNs) and vision transformers (ViTs) to detect seven emotions such as happy, neutral, angry, sad, surprise, fear, and disgust which is trained on FER2013 and tested on a smaller autism-specific dataset. The CNN architecture included four convolutional blocks with batch normalization, ReLU activations, dropout, and softmax output. Data augmentation and Bayesian hyperparameter optimization improved generalization. CNN achieved 74% accuracy on FER2013, outperforming ViT's 69%, and 65% on the autism dataset, performing best on happy and neutral. However, a key limitation emerged: autistic individuals often express emotions atypically, making it questionable to train on neurotypical datasets like FER2013. This mismatch affected performance, particularly for subtle emotions like fear and disgust.

[Chung & Oh \(2019\)](#) developed a CNN based facial expression recognition system aimed at assisting individuals with Autism Spectrum Disorder (ASD) in

understanding others' emotions. The model was trained using the Facial Expression Recognition Challenge dataset from Kaggle, which contains 48×48 pixel grayscale images labeled with seven emotions such as happy, angry, neutral, scared, sad, surprised, and disgust. Face detection was performed using the Haar Cascade Classifier in OpenCV, and emotion classification was handled using a CNN model based on Arriaga et al. (2017), achieving an overall accuracy of 69.5%. The relatively low model accuracy highlights a significant gap in robustness and reliability for real-world usage, especially for applications involving ASD users who rely heavily on accurate and consistent emotional feedback. Furthermore, the dataset was generic and not ASD-specific, which may limit the model's effectiveness in recognizing subtle or atypical expressions often exhibited by individuals with ASD.

[Awatramani & Hasteer \(2020\)](#) proposed a CNN-based facial emotion recognition model to assist children with Autism Spectrum Disorder (ASD) in identifying human emotions. The model was trained on the FER-2013 dataset, which includes 28,709 training and 3,589 validation grayscale images sized 48×48 pixels, covering seven emotions. The architecture comprised 10 convolutional layers, ReLU activations, batch normalization, dropout (0.5), and global average pooling. The model achieved a validation accuracy of 67.5%, with noticeable difficulty in recognizing underrepresented emotions like disgust.

Building on CNN based model, [Saranya & Anandan \(2022\)](#) proposed a hybrid deep learning approach combining Facial Action Coding System (FACS), Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) for detecting Autism Spectrum Disorder (ASD) in children through facial emotion analysis. The model extracts fine-grained facial features using FACS-CNN, and then uses LSTM to capture time-series emotional cues. The system was trained on AFFECT-NET and validated with KFAD-2020 datasets, involving autistic and non-autistic children's facial images. Image preprocessing included resizing, normalization, dropout (0.4), and ReLU activation. The proposed model achieved 92.5% accuracy, 91.5% precision, 91.5% recall, and an approximate F1-score of 91.5%, outperforming baseline models like MLP (48.67%), CNN (67.75%), and LSTM (71.56%). However, for real-world application more robust model is need.

[Silva et al. \(2021\)](#) suggested a system for identifying emotions in kids with autism spectrum disorder (ASD) using a hybrid model that integrates Intel® RealSense™ 3D sensor with a Support Vector Machine (SVM) classifier. The system

extracts facial landmarks and Facial Action Units (AUs) to detect six emotions happiness, anger, sadness, fear, surprise, and neutral according to geometric and motion-based features. A non-linear SVM with RBF kernel achieved 93.6% accuracy, lacked deep temporal learning and relied on a relatively small, predominantly neurotypical dataset, further restricting its real-world application for autistic children.

[Pavez et al. \(2023\)](#) proposed a smart mirror system, Emo-mirror using convolutional neural networks (CNNs). This study's objective was to enhance traditional emotion therapy methods by integrating intelligent, contactless computer vision technology. The system was designed to detect five basic emotions such as happy, sad, angry, scared, and neutral via facial expressions captured through a camera and processed in real-time to provide therapeutic feedback. To process the data, the authors employed image preprocessing techniques that included resizing facial images to 200×200 pixels, grayscale conversion, and data augmentation using horizontal flipping. Stochastic Gradient Descent (SGD) with momentum and a learning rate of 0.01 was used as the optimizer, with dropout layers (rate = 0.5) and ReLU activation functions for regularization and performance. Two CNN architectures, VGG16 and ResNet50, were trained and evaluated on the FER2013 and CK+ datasets over 100 epochs. Evaluation was performed using accuracy and confusion matrix analyses. Among the tested configurations, ResNet50 with the CK+ dataset achieved the highest accuracy of 89.9%, while VGG16 yielded 93.3% on the same dataset.

Earlier classical methods like [Rani \(2019\)](#) proposed a facial emotion recognition system for autistic children using Local Binary Pattern (LBP) for feature extraction from grayscale facial images and used two classifiers: Support Vector Machine (SVM) and a two-layer feedforward Neural Network (NN). The system was trained on a very small dataset of 25 facial images, collected from online sources, representing four emotions: happy, sad, neutral, and angry. LBP+SVM achieved 90% accuracy, outperforming the LBP+NN model, which reached 70%.

[Y. Wang et al. \(2025\)](#) proposed an emotion recognition system tailored for children with autism, focusing solely on atypical facial expressions, convolutional vision transformer (CvT). Recognizing that autistic facial expressions are often atypical, characterized by reduced dynamics and facial asymmetry, they developed an autism-specific dataset containing over 24,000 facial images collected from real

rehabilitation environments and basic four emotions, such as happy, sad, calm, and angry. The CvT model achieved an accuracy of 79.12%, substantially outperforming a pure Vision Transformer (ViT), which reached only 48%. This performance boost highlights the advantage of combining convolutional operations with self-attention to better capture local and global facial cues in children with autism.

2.2 Xception

The Xception model, also known as ‘Extreme Inception,’ is designed to improve computational cost by reducing the number of parameters and FLOPs compared to the standard convolution. To achieve that, it leverages depthwise separable convolution. Depthwise separable convolution decouples channel-wise and spatial feature learning.

2.2.1 Xception Architecture

Chollet (2017) proposed the Xception architecture that contains three main parts. These are entry flow, middle flow, and exit flow (Figure 2.1). Entry flow extracts the lower-level feature using the combination of depthwise separable convolutional, standard convolutional layers, and residual blocks. Whereas the middle flow consists of eight repeated blocks of depthwise separable convolutional and ends with exit flow. In the exit flow, it uses global average pooling instead of a fully connected layer to reduce overfitting. The main backbone of Xception is that it uses the depthwise separable convolutional neural network, and each flow consists of these depthwise separable convolutions. Consisting of 36 layers, each layer is followed by a batch normalization and ReLU activation function.

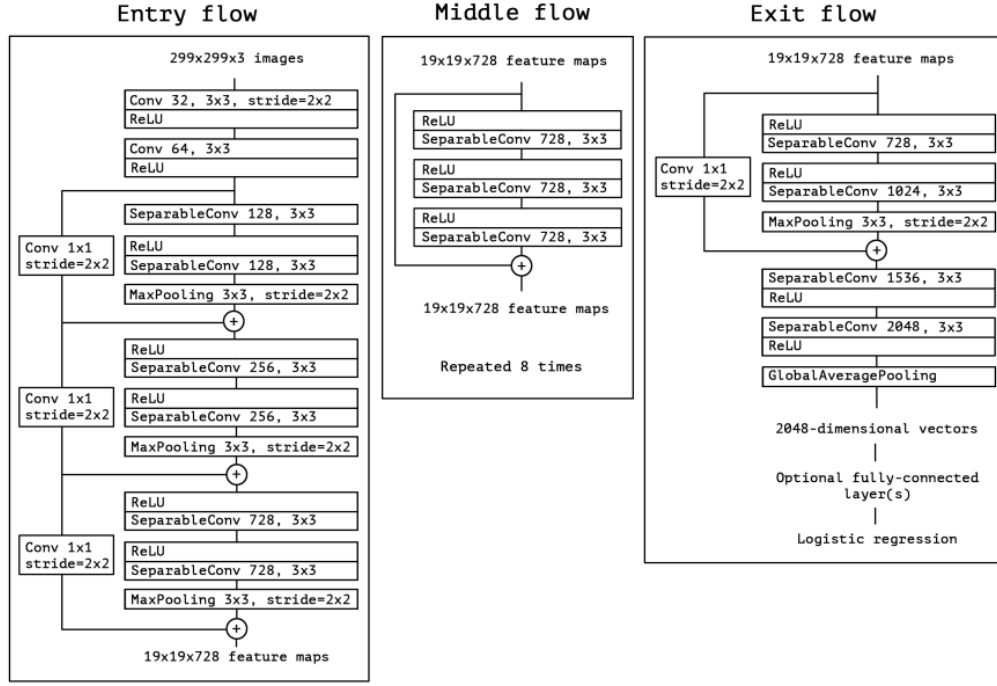


Figure 2.1: Xception Model Architecture as depicted in [Chollet \(2017\)](#).

2.2.2 Standard Convolution vs Depthwise Separable Convolution

Standard Convolution: In a standard convolution, a set of kernels or filters convolves all channels at once and produces a single output feature map 2.2. For an input feature $X \in \mathbb{R}^{H \times W \times C_{in}}$, a standard convolution applies a filter $K \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$, computing the output feature map $Y \in \mathbb{R}^{H' \times W' \times C_{out}}$. Thus, the equation of the standard convolution is

$$Y_{i,j,c} = \sum_{m=1}^k \sum_{n=1}^k \sum_{d=1}^{C_{in}} K_{m,n,d,c} \cdot X_{i+m-1,j+n-1,d}$$

Here, k , H , W , C_{in} , and C_{out} are the kernel size, height, width, the number of input channels, and the number of output channels, respectively. The value at spatial location (i,j) in the c -th output channel is represented by $Y_{i,j,c}$.

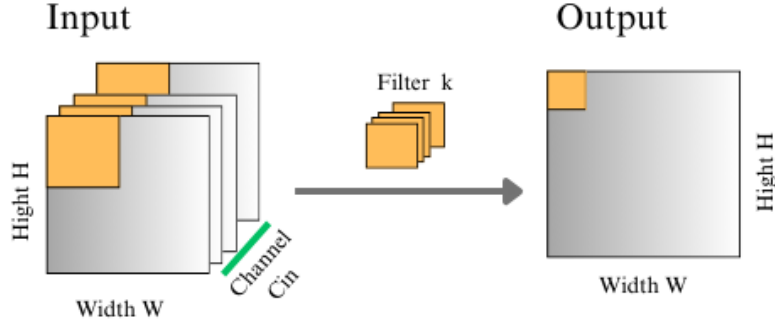


Figure 2.2: Standard Convolution

Depthwise Separable Convolution: Depthwise separable convolution has two parts Figure 2.3. 1) Depthwise Convolution, 2) Point-wise Convolution.

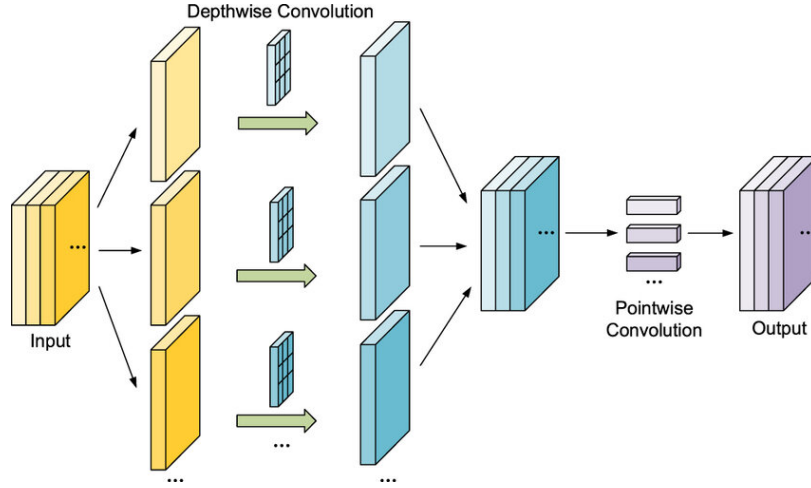


Figure 2.3: Depthwise Separable Convolution, illustrated by Luo et al. (2023).

1. **Depthwise Convolution:** In depthwise convolution, unlike standard convolution, filters convolve all the input channels separately but simultaneously. Then stack all the channels. If the input has $H \times W$ an image size with C_{in} number of channels and kernel K traverses all the channels separately at a time, then for an input dimension $X \in \mathbb{R}^{H \times W \times C_{in}}$, the depthwise convolution employs C_{in} separate filters, each of kernel size $K_{depthwise} \in \mathbb{R}^{k \times k \times 1}$, where the number of the output channel is the same as the input channel C_{in} . Thus the output for $Y_{depthwise} \in \mathbb{R}^{H' \times W' \times C_{in}}$ is

$$Y_{depthwise,i,j,C_{in}} = \sum_{m=1}^k \sum_{n=1}^k K_{depthwise,m,n,1} \cdot X_{i+m-1,j+n-1,C_{in}}$$

2. **Point-wise Convolution:** Pointwise is applied to combine all the output features extracted through depthwise convolution. It uses 1×1 convolution across channels C_{in} to do this operation. The pointwise convolution maps C_{in} input channels to C_{out} output channels using the filter $K_{pointwise} \in \mathbb{R}^{1 \times 1 \times C_{in} \times C_{out}}$, The output $Y \in \mathbb{R}^{H' \times W' \times C_{out}}$ is computed as follows:

$$Y_{i,j,c} = \sum_{d=1}^{C_{in}} K_{pointwise,1,1,d,c} \cdot Y_{depthwise,i,j,d}$$

Efficiency Comparison: Standard vs. Depthwise Separable Convolution:

1. **Standard Convolution:** The computational cost and parameter cost for standard convolution are as follows:
 - (a) **Computational cost:** $Cost_{standard} = C_{in} \times H' \times W' \times C_{out} \times k \times k$
 - (b) **Parameter cost:** $Params_{standard} = C_{in} \times C_{out} \times k \times k$
2. **Depthwise Separable Convolution:** The computational cost and parameter cost for Depthwise Separable Convolution are as follows:
 - (a) **Computational cost:**
 - Depthwise convolution: $Cost_{depthwise} = C_{in} \times H' \times W' \times k \times k$
 - Point-wise convolution: $Cost_{pointwise} = C_{in} \times H' \times W' \times C_{out}$

Thus, the total Computational cost is

$$Cost_{separable} = Cost_{depthwise} + Cost_{pointwise}$$

$$Cost_{separable} = H' \times W' \times C_{in} \times (k \times k + C_{out})$$

- (b) **Parameter cost:**

$$Params_{separable} = C_{in} \times k \times k + C_{in} \times C_{out}$$

From the equation, we can see that the ratio for both computational cost and parameter cost is $\frac{Cost_{separable}}{Cost_{standard}}$ or $\frac{Params_{separable}}{Params_{standard}} = \frac{k \times k + C_{out}}{k \times k \times C_{out}}$ which implies that depthwise separable convolution is more efficient than standard convolution as both computational cost and parameter cost are fewer than standard convolution.

2.2.3 Residual Connection

In Xception architecture, residual connection is used to enable dimension matching and enhance gradient flow. Usually in residual blocks, the output $F(x)$ is added with the input x . But when dimension mismatch occurs due to different channel sizes of input and output, a 1×1 is used to adjust the size, mitigate the vanishing gradient problem, and allow the model to train for deeper architecture. The final outcome becomes $y = F(x) + W_{1 \times 1} \cdot x$. Here $W_{1 \times 1}$ represents the 1×1 convolution weight.

2.2.4 Review of Xception Architecture

In CNNs, the network learns important features by capturing both spatial and channel information. However, most research focuses on extracting spatial information while providing equal importance to all channels within a local receptive field.

Similarly, in Xception, providing equal importance to all channels, feature extraction is performed hierarchically based on local patterns [X. Wang et al. \(2018\)](#). Though Xception improves efficiency through depthwise separable convolution, still it makes the task less effective for global context understanding as its receptive field grows gradually. Therefore, long-range dependencies across channels remain weak. As a result, understanding global features becomes challenging. Which causes texture bias towards the model [Geirhos et al. \(2018\)](#) and degrades the generalization power of complex images. So long-range dependencies are important. Also, as it is a heavy model, on a small dataset it may memorize the noise and cause an overfitting problem [C. Zhang et al. \(2021\)](#).

On the contrary, SENet introduces a channel-wise attention mechanism to find out which channel is more important alongside spatial information [Hu et al. \(2018\)](#). Also brings global understanding by applying a global average pooling operation during the squeeze step which summarizes each channel's global representation and gives an output in one single value for every channel. This way, SENet mitigates the challenge of long-range interdependencies. At the same time when dealing with small or moderate data, this attention mechanism reduces the overfitting problem by forcing the model to concentrate on discriminative features rather than noise (Hu et al, 2018).

2.3 Squeeze-and-Excitation Network (SENet)

To provide channel-wise attention mechanisms and to improve channel interdependencies explicitly, squeeze-and-excitation network (SENet) is proposed by [Hu et al. \(2018\)](#). This mechanism is achieved by squeeze-and-excitation blocks.

This architecture includes three operations 2.4. Such as squeeze operation, excitation operation, and channel recalibration.

- **Squeeze operation:** To extract each channel's global representation, the squeeze operation helps. At first, it summarizes each channel's global representation and gives an output in one single value for every channel. This is achieved by global average pooling (GAP). For an input feature map $X \in \mathbb{R}^{H \times W \times C}$. GAP produces an average value for each channel across all spatial location such that

$$z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W X_{i,j,c}$$

- **Excitation operation:** After getting a single or squeezed vector (z) for each channel through GAP, an excitation operation is performed. Here, channel dependencies are obtained through a fully connected neural network. Through this channel mechanism, the important and less important channels are distinguished. For this mechanism, 2 fully connected (FC) layers are used.

$$S = \sigma(W_2 \cdot \delta \cdot (W_1 \times z))$$

Here, $W_1 \in \mathbb{R}^{\frac{c}{r} \times c}$ and $W_2 \in \mathbb{R}^{c \times \frac{c}{r}}$. Squeezed feature vector (z) is fed into the 1st (FC) layer W_1 . This layer reduces the number of channels by a factor of r , then ReLU is applied $\delta(W_1 z)$. This transformation leverages the dimensionality of z from c to $\frac{c}{r}$. 2nd FC layer (W_2) reconstructs the original number of channels $\frac{c}{r}$ to c which is to $1 \times 1 \times c$. This passes through a sigmoid activation function (σ). The range of this activation is from 0-1. When the value is close to 0, it carries less important channels; otherwise, it is more important.

- **Channel recalibration:** In here, learned channel attention weights S from squeeze-and-excitation are applied to the original feature map X to focus

on the most important feature. If the X' is recalibrated feature map, then $X' = S \cdot X$

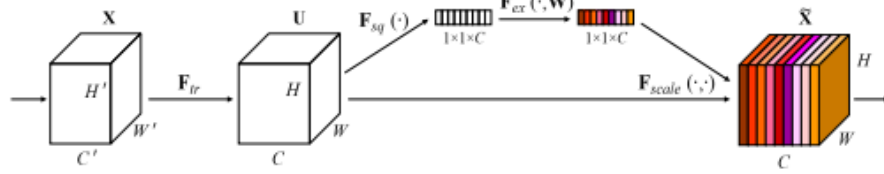


Figure 2.4: Squeeze-and-Excitation Block Diagram adopted from [Hu et al. \(2018\)](#).

2.4 Evaluation Metrics for Multiclass

We evaluated the performance accuracy, loss, recall, precision, F1 score, AUC score and Confusion matrix. In this research, recall, precision, and f1-score were calculated separately for each class to evaluate the performance metric following the one-vs-rest method for classifying multi-class. Along with that, the AUC score is determined by employing the “macro” average.

For each class c , (where $c = 1, 2, 3, \dots, k$ and N = total number of samples):

- **True positive (TP_c):** The number of samples that are positive and correct predictions of class c .
- **False positive (FP_c):** The number of samples that are not of that specific class but are recognized as positive. Instances were wrongly predicted as class C .
- **False negative (FN_c):** the number of samples that are of class c but are misclassified as negative.

Following these, a variety of performances are measured following the formula of Table 2.1 for multiclass classification. Because only accuracy is not enough to evaluate a model’s performance for multiclass classification. To get a more detailed and insightful evaluation of a model’s performance, recall, precision, and F1-score play a vital role.

Table 2.1: Performance Metrics

Metrics	Expression	Eq. No.
Precision _c	$\frac{TP_c}{TP_c + FP_c}$	(1)
Recall _c	$\frac{TP_c}{TP_c + FN_c}$	(2)
F1-Score _c	$\frac{2}{\frac{1}{\text{Precision}_c} + \frac{1}{\text{Recall}_c}}$	(3)
Macro-average of Precision	$\frac{\sum_{c=1}^k \text{Precision}_c}{k}$	(4)
Macro-average of Recall	$\frac{\sum_{c=1}^k \text{Recall}_c}{k}$	(5)
Macro-average of F1-Score	$\frac{2}{\frac{1}{\text{Macro Precision}} + \frac{1}{\text{Macro Recall}}}$	(6)
Accuracy	$\frac{\sum_{c=1}^k TP}{N}$	(7)

Here, precision measures the proportion of positive predictions that are correctly classified instances 2.1(1), whereas recall measures the proportion of positive cases that are correctly predicted by the model 2.1(2). Again, the F1-score is the harmonic mean of recall and precision which is a single value that balances both precision and recall 2.1(3).

2.5 Summary of Literature Review

Table 2.2: Summary of literature review on the FER task of ASD children.

Author	Dataset(s)	Data preprocessing	Architecture	Result	Limitation
Y. Wang et al. (2025)	Own Dataset	Faces cropped, aligned	Convolutional ViT (CvT)	Accuracy 79.12%	Performance isn't satisfactory; Data East China specific; 33 participants.
Komuraiah et al. (2024)	Autistic Children Image Dataset	Removed stock and duplicate photos and applied image augmentation.	Custom CNN	Accuracy 91%, Precision 1.00, Recall 1.00, F1 score 1.00, Macro-avg 91%	Macro-avg is lower, suggesting mass misclassification in certain classes.
El Rhatassi et al. (2024)	FER-2013, Autism Dataset	Resized, normalized, augmented; facial features manually analyzed	CNN + Vision Transformer	Accuracy 65% on autism dataset	Performance isn't satisfied; Trained on FER-2013, tested on ASD data.
Talaat et al. (2024)	Kaggle (own dataset)	Remove duplicate images and apply image augmentation.	Xception with Sparse Auto Encoder	Accuracy 95.23%, Train acc. 96.31%, Validation acc. 88.54% , F1-score 93.31%, AUC 91.34%	Overfitting; multi-class evaluation not presented, discriminatory power isn't strong.
Pavez et al. (2023)	FER2013 and CK+ datasets	Image resizing, grayscale conversion, and data augmentation were used.	VGG16 and ResNet50	CK+ Dataset: Accuracy of VGG16: 93.3%	Therapy-based (mimic emotion) isn't sufficient, as it cannot support spontaneous emotional needs.
Silva et al. (2021)	Own Dataset	Extracted 78 facial landmarks, normalized	Intel RealSense + SVM (RBF)	Accuracy 93.6%, AUC 99%, MCC 92.2%	Therapy-based (mimic emotion) isn't sufficient, because it cannot support spontaneous emotional needs.
Awatramani & Hasteer (2020)	FER-2013 dataset	Data augmented; faces cropped and normalized	Custom CNN	Accuracy 67.5%	Overfitting risk
Chung & Oh (2019)	FER-2013 dataset	Haar cascade (OpenCV)	Custom CNN	Accuracy 69.5%	Performance is not satisfactory for real world
Rani (2019)	Own Dataset	Grayscale, cropped, resized, LBP extracted	SVM with LBP	Accuracy 90%	Only accuracy isn't enough as the data size is 25 high overfitting may arise, which is questionable for real world application

From Table 2.2 we can analyze that the accuracy achieved by [Y. Wang et al. \(2025\)](#) is not satisfied; also, data was collected from a single institute from 33 children which is East China specific, only containing four emotions, which is an insufficient representation of the heterogeneity within autism spectrum disorders, particularly regarding symptom severity, verbal ability, and comorbid conditions. [El Rhatassi et al. \(2024\)](#) trained on the FER-2013 dataset and tested ASD-specific dataset. This approach is questionable because facial expression differs from typical expressions to atypical expressions of an ASD individual. As a result, it had poor performance. [Chung & Oh \(2019\)](#), [Awatramani & Hasteer \(2020\)](#) trained and tested their model on the FER-2013 dataset, which contains typical or neurological expression. It had poor performance, too, and it is therapy-based. [Pavez et al. \(2023\)](#), [Silva et al. \(2021\)](#) had comparatively better accuracy, and the data were not trained on an ASD-specific dataset. Their domain is therapy-based (mimicking emotion shown on the screen), which is not sufficient because it cannot support spontaneous emotional needs in everyday life, where caregivers or parents can benefit. [Rani \(2019\)](#) worked on an ASD-specific dataset containing four emotions, where the data size is 25. It achieved 90% accuracy, but the problem is that for multiclass classification, only accuracy is not enough, as due to the small dataset, it can overfit the model. [Talaat et al. \(2024\)](#), [Komuraiah et al. \(2024\)](#) also worked on a specific ASD dataset collected from Kaggle. [Komuraiah et al. \(2024\)](#) had recall 1.00, precision 1.00, F1 score 1.00, and macro-avg 91%. Lower macro-avg suggests mass misclassification in certain classes, which causes emotion overlapping. On the other hand, [Talaat et al. \(2024\)](#) had 95.23% accuracy but with 91.34% AUC and validation accuracy of 88.53%, over the epochs it's increasing, and classwise evaluation isn't accomplished here. It means discriminatory power over the classes is not strong, the model is highly overfitting, and there is room for improvement.

As a dataset owned by [Talaat et al. \(2024\)](#), which is an ASD-specific dataset and aligns with our goal, we are going to utilize this dataset. From the literature we found, Xception improves efficiency through depthwise separable convolution compared to the traditional CNN mentioned in Table 2.2. However, it still makes the task less effective for global context understanding as its receptive field grows gradually and overfits. Therefore, long-range dependencies across channels remain weak. As a result, understanding global features becomes challenging. This degrades the

generalization power of complex images. So, to utilize its efficiency, we intend to work on the Xception model to improve emotion recognition in autistic children, allowing the model to learn more discriminative and relevant facial regions. Also, we found out that when attention mechanisms were added to traditional CNN architecture, the performance improved.

Motivated by their findings, the present study aims to further refine and expand this direction by not only adopting an Xception-based architecture but also integrating an attention mechanism to enhance the model's focus on class-specific facial features, especially where emotional overlap is common (e.g., sadness vs. anger). Inspired by Talaat et al.'s architecture, we seek to improve emotion recognition in autistic children by incorporating channel-wise attention, allowing the model to learn more discriminative and relevant facial regions. The goal is to support caregivers and parents, thus aiding in the emotional development and personalized care of children with ASD.

Chapter 3

Methodology

3.1 Dataset

3.1.1 Data Collection

For this research, the dataset has been collected from Kaggle.

- This dataset contains 831 images of 6 different emotions of autistic children. These emotions are anger, neutral, sad, fear, happy, and surprise.
- Anger, neutral, sad, fear, happy, and surprise contain 69, 33, 392, 55, 214 and 68 data respectively. A summary of the datasets is given in the Figure 3.1.

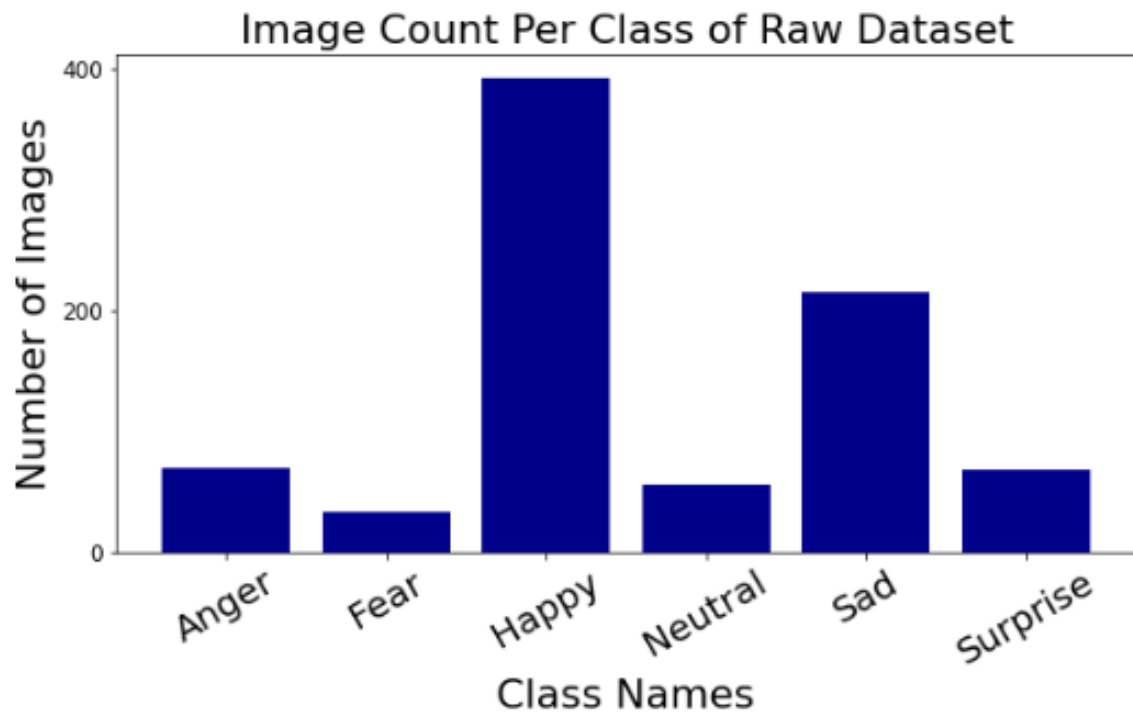


Figure 3.1: Data size visualization across different classes.

- The dataset is clean and the dimension of each photo is different. A few samples are given in Figure 3.2.

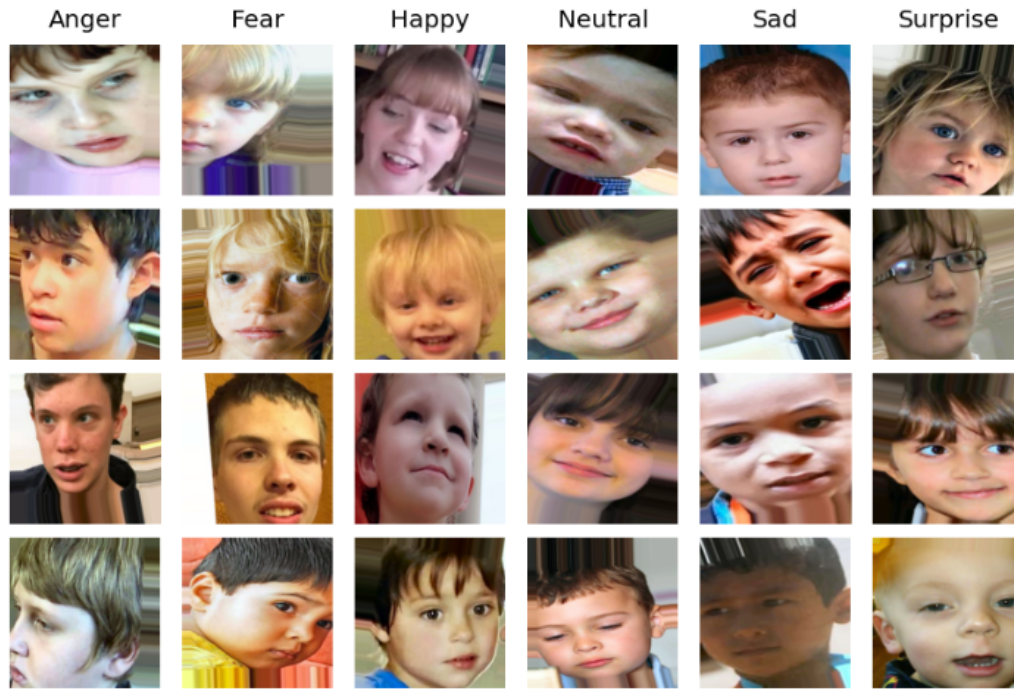


Figure 3.2: Few samples of facial emotions per class

3.1.2 Data Preprocessing

The dataset preprocessing steps we followed are given below:

- The data has been resized to 299×299 using Python OpenCV.
- After that, the data size was increased 7 times for all classes using the augmentation techniques in Figure 3.3. As a result, the total data size went from 831 to 5350.

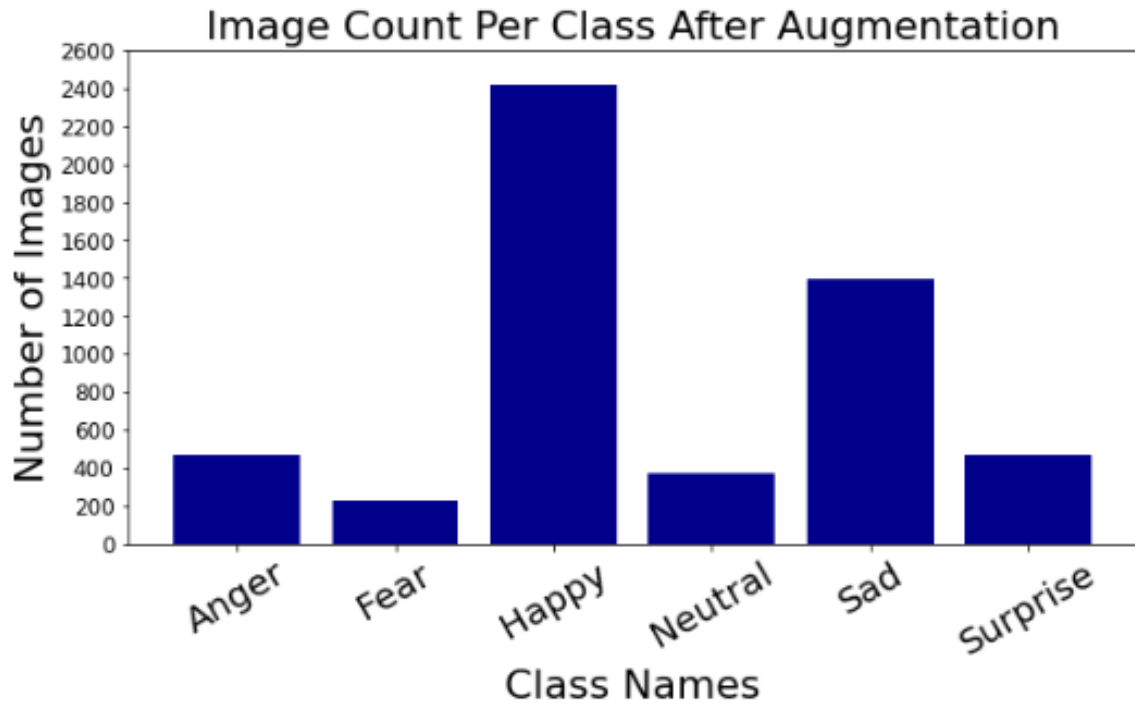


Figure 3.3: Data size across different classes after increasing data size per class.

- The dataset is imbalanced. To balance the dataset, both under-sampling and oversampling techniques were used.

3.1.2.1 Undersampling

Undersampling technique means the process of decreasing the number of samples of the majority class, with corresponds to the minority class. In our dataset, through the under-sampling techniques, the number of the majority class, which means 'happy', was decreased first, corresponding to the number of samples of the second majority class, 'sad'. Consequently, the dataset decreased to 4,334 from 5,350.

3.1.2.2 Oversampling

On the other hand, the oversampling technique means the process of increasing the number of samples of the minority classes that correspond to the majority class. After the undersampling techniques, the number of minority classes (anger, fear, neutral, and surprise) was increased through the oversampling technique, which resembles the number of majority classes (happy and sad). As a result, a properly

balanced dataset was prepared in this way, and the dataset size increased from 4,334 to 8,380. This technique has been adopted because we had few data.

- To make the data set robust and diverse, the following parameters were adopted for augmentation in both undersampling and oversampling techniques:
 - I. Rotation Range is used to rotate the images within a given range of 20. This range is adopted so that even after rotating data for a large number of datasets, we get the data that will reflect the actual data.
 - II. Using shear range, data was shifted clockwise. The range 0.2 is chosen so that images don't get shrunk.
 - III. Horizontal flip is used to flip the image horizontally.
 - IV. Height and width shift simply shifted the images along the x-axis and y-axis, respectively. The shift range was 0.2.
 - V. Through the brightness range, images were brightened or darkened randomly.
 - VI. The data was randomly zoomed in or zoomed out following the mentioned range by Zoom range.

3.1.2.3 Data split

- After balancing, the data was carefully cleaned for better performance.
- Finally, the dataset was divided into train, test, and validation. Here the proportion is 80:10:10. This implies that 80% of the data is used for training, 10% is used for validation, and the remaining 10% is used for testing.

3.2 Proposed Model

3.2.1 Proposed Depthwise Separable Convolution

Xception architecture (discussed in section 2.2.1), leverages the depthwise separable convolution and focuses on spatial feature extraction but may fail to capture long-range dependencies, which causes performance degradation and overfitting in

small datasets. To mitigate this problem, we proposed an enhanced deep learning architecture by integrating SENet (a channel-wise attention mechanism) with the Xception model.

SENet (see section 2.4) captures channel-wise dependencies and suppresses less important channels and highlights relevant regions. This strengthens the model’s ability to extract the most relevant emotion-specific features. Specifically, in our proposed model, we integrated the SENet within the depthwise separable convolution block (between the depthwise convolution and pointwise convolution) of the Xception model Figure 3.4. In here, through the depthwise convolution, all the channels were stacked together, then the SE (Squeeze-and-Excitation) block was applied to distinguish more important and less important channels and then the learned channel attention weights were recalibrated. Through this process, the model captures the most relevant spatial and fine-grained details, reducing the problem of passing weakly activated features. Finally, this extracted information was combined through pointwise convolution.

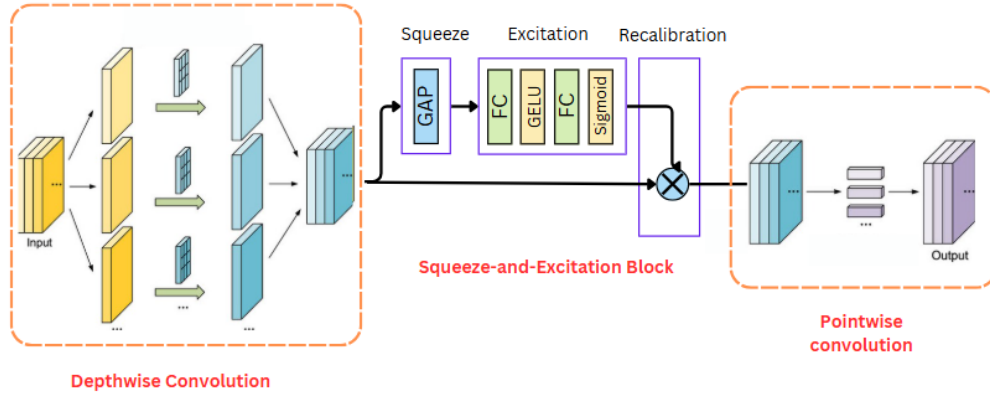


Figure 3.4: Proposed Depthwise Separable Convolution Block.

Within the SE block we used GELU activation function 3.1.

$$\text{GELU}(x) = x \cdot \Phi(x) = x \cdot \frac{1}{2} \left[1 + \text{erf} \left(\frac{x}{\sqrt{2}} \right) \right] \quad (3.1)$$

GELU’s balanced combination of smooth gradients (unlike ReLU’s discontinuous derivative at 0), non-linearity and adaptive gating (through its probabilistic weighting) makes it uniquely powerful. The GELU activation function smoothly

transits the gradient flow, minimizes the risk of dead neurons and allows smaller negative values to pass through smaller activations. However, ReLU sets all the negative values to zero.

Mathematically, we represented our proposed model in Table 3.1.

Table 3.1: Mathematical representation of our proposed depthwise separable convolution.

Block Name	Equation
Step 1. Depthwise Convolution	$U_{depthwise,h',w',c} = \sum_{m=1}^k \sum_{n=1}^k K_{depthwise,m,n,1} \cdot X_{h'+m-1,w'+n-1,C_{in}}$
Step 2. SE Block	$z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W U_{depthwise,h',w',c}$ $S = \sigma(W_2 \cdot \delta(W_1 \cdot z))$ $U'_{depthwise,h',w',c} = S_c \cdot U_{depthwise,h',w',c}$
Step 3. Pointwise Convolution	$Y_{h',w',c} = \sum_{d=1}^{C_{in}} K_{pointwise,1,1,d,c} \cdot U'_{depthwise,i,j,d}$
Combined steps	$Y = \text{Pointwise}(\text{SE}(\text{Depthwise}(X)))$

We used this proposed depthwise separable convolution in the entry, middle and exit flow.

3.2.2 Entry Flow of the proposed Xception

In the entry flow in Figure 3.5, we took an image size 100×100 and fed it to the architecture and replaced the channel number of the last block with 512 channels instead of 728 channels. Consequently, following the shape changing equations 3.2,

3.3and 3.4 it reached to $7 \times 7 \times 512$ feature channel.

$$\left[\frac{I - K + 2P}{S} + 1 \right] \times \left[\frac{I - K + 2P}{S} + 1 \right] \quad (3.2)$$

$$\frac{\text{previous layer's shape}}{2} \times \frac{\text{previous layer's shape}}{2} \quad (3.3)$$

$$\left[\frac{I - K}{S} + 1 \right] \times \left[\frac{I - K}{S} + 1 \right] \quad (3.4)$$

Here, I stands for input shape, K for kernel size, P for padding, and S for stride. Also, euqations 3.2, 3.3and 3.4 stands for the shape of the convolution layer, the shape of the max pooling layer, and the shape of the residual connection respectively.

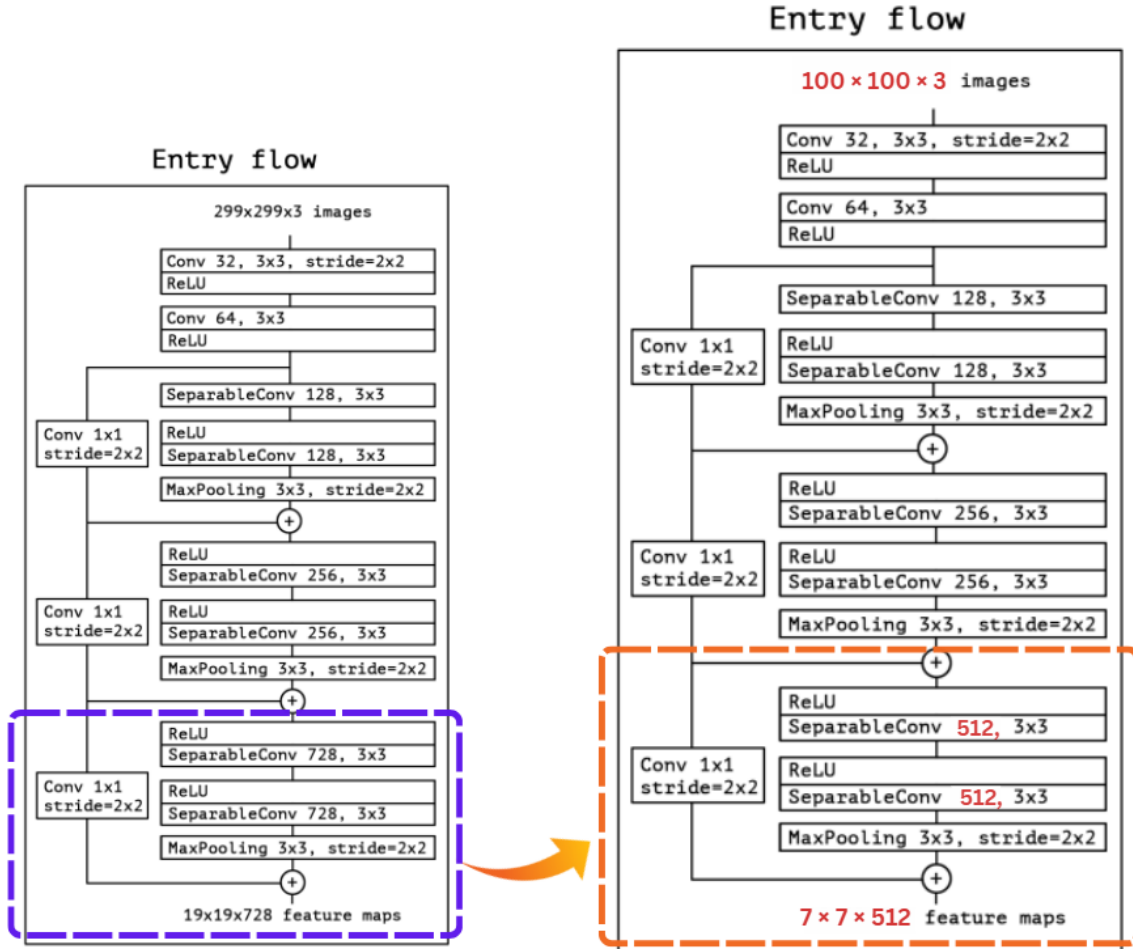


Figure 3.5: The entry flow of the proposed Xception model.

3.2.3 Middle Flow of the proposed Xception

In the middle flow, we used the GELU activation function and 5×5 kernel in 512 channels instead of 7×7 . This middle flow was repeated just once Figure 3.6. Middle flow extracts high-level complex features and to avoid the problem of dead neurons in the middle flow, GELU is used instead of ReLU.

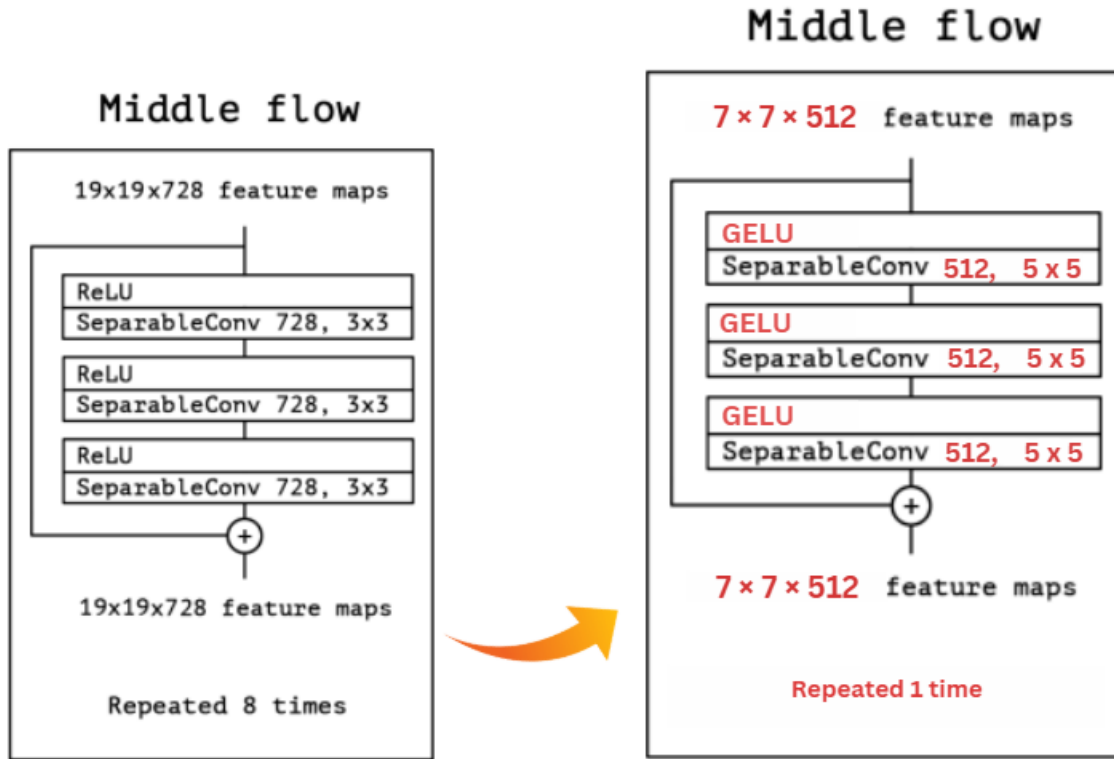


Figure 3.6: Middle flow of the proposed Xception model.

3.2.4 Exit Flow of the proposed Xception

In the exit flow (Figure 3.7) we used only 512 and 1024 channel based separable convolution blocks and integrated them with 5×5 kernels instead of 3×3 and global average pooling were replaced with a flatten and dense layer. The global average pooling layer represents all the feature map values with just one single value by averaging, which may result in the loss of capturing complex spatial feature and minimize the discriminating power of the model. But extracting fine-grained details is crucial for facial emotion recognition, so we introduced flatten and dense layers instead of GAP. This flatten converts multidimensional features

into a 1D vector and the dense layer introduces non-linear functionality, which helps to retain more complex and fine-grained details.

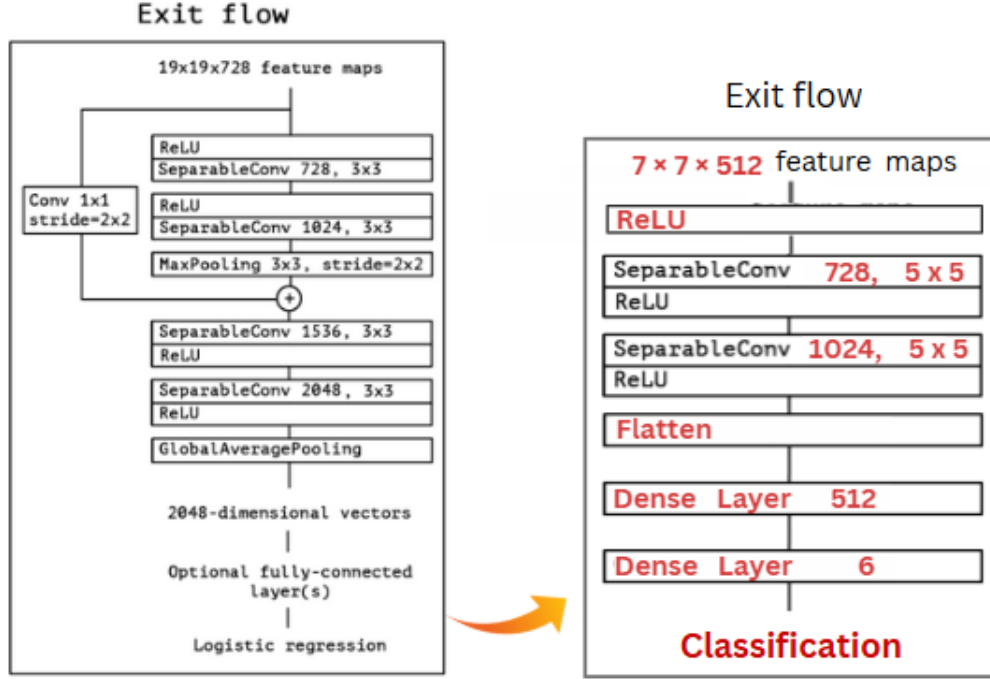


Figure 3.7: Exit flow of the proposed Xception model.

Furthermore, we used 3×3 kernels in the entry flow, as entry flow captures lower-level feature and it helps to capture fine-grained details. Furthermore, 5×5 kernels were used in the middle and exit flow to capture broader spatial areas and global features rather than just fine-grained details. Finally, the model classified six emotions.

3.2.5 Algorithmic Functionality of proposed model

Algorithmic functionality of proposed depthwise separable convolution and proposed Xception model is given in Figure 3.8 and in Figure 3.9 respectively.

Algorithm 1: Proposed Depthwise Separable Convolution Functionality

```
function Squeeze_Excite_block (X, ratio):
    Channel ← Shape (X) [-1]
    X ← GlobalAveragePooling (X)
    X ← FullyConnected (X, reduce_channel=True)
    X ← GELU (X)
    X ← FullyConnected (X, expand_channel=True)
    X ← Sigmoid (X)
    X ← Rescale (1, 1, channel) X
    X ← Multiply (X, X)
    return X

function DepthwiseSeparableConV_block (X, filter, kernel, padding):
    // Depthwise Convolution
    X ← DepthwiseConv2D (kernel, padding) (X)
    X ← BatchNormalization (X)
    X ← ReLU (X)
    // Squeeze and Excitation Block
    X ← Squeeze_Excite_block (X, ratio=16)
    //Pointwise Convolution
    X ← PointwiseConv1x1 (filter, kernel, padding) (X)
    return X
```

Figure 3.8: Algorithmic functionality of proposed depthwise separable convolution.

Algorithm 2: Proposed Xception Model's Functionality

```
function Entry_flow (input_tensor X):
    // all same as base model
    // change last Depthwise Separable Convolution block
    X ← DepthwiseSeparableConV_block (X, filter=512, kernel=5, padding=same)
    X ← BatchNormalization (X)
    X ← ReLU (X)
    X ← Add (previous_layer, X)
    return X

function Middle_flow (previous_layer):
    repeat 1 time:
        repeat 3 times:
            X ← GELU (previous_layer)
            X ← DepthwiseSeparableConV_block (X, filter=512, kernel=5, padding=same)
            X ← BatchNormalization (X)
        return X
    previous_layer ← Add (previous_layer, X)
    return previous_layer

function Exit_flow (X):
    X ← ReLU (X)
    X ← DepthwiseSeparableConV_block (X, filter=728, kernel=5, padding=same)
    X ← ReLU (X)
    X ← DepthwiseSeparableConV_block (X, filter=1029, kernel=5, padding=same)
    X ← Flatten
    X ← FullyConnected (filter=512) (X)
    X ← FullyConnected (units=6) (X)
    return X
```

Figure 3.9: Algorithmic functionality of proposed Xception Architecture.

In the final step, we have trained this proposed model and compared it with the base model and with the performance of the author [Talaat et al. \(2024\)](#), where they trained the Xception model integrating a sparse autoencoder. In the section 4,

these models were evaluated through loss and accuracy, recall, precision, f1-score, confusion matrix, macro average AUC, and FLOPs.

3.3 Budget Planning

The total cost for our research project is given in the Figure 3.10

Category	Descriptions	Quantity/People	Unit Cost (Tk)	Total Cost (Tk)
Course	Online Course	3 members	8000	8000
Equipment	Laptop Repair	1 member	4000	4000
	Cooling Fan	1 member	550	550
Supplies	Paper Printing	240 pages	5/page	1200
	Notebooks	5	120/notebook	600
	Pens	18 sets	20/set	360
Connectivity	Internet Subscription	12 months	3000/month	36000
Food	Meals during offline sessions	3 members x 48 trips	100/meal	14,400
Travel for meeting with Supervisor	Member 1 Travel Fare	48 trips	130/trip	6240
	Member 2 Travel Fare	48 trips	100/trip	4800
	Member 3 Travel Fare	48 trips	80/trip	3840
Total Budget for the Project				79990

Figure 3.10: Total budget for our capstone project

3.4 Modern Tools and Technologies Used

Our project made effective use of modern tools and technologies to strengthen work collaboration, organization, and productivity. Below are the key software and platforms used:

- **Hardware:** Desktop computer powered by Intel Core i5 with 8 gigabytes of RAM.

- **Jupyter Notebook:** We used Jupyter Notebook to develop and test deep learning models to combine code, outputs, and documentation in one interface.
- **Mendeley:** We used Mendeley as a reference management tool to arrange, reference and maintain research documents and bibliographic information systematically.
- **Overleaf:** We used Overleaf as a primary tool to write LaTeX-based report writing to ensure high-quality formatting and version management.
- **Canva:** We used Canva to design our materials, materials such as model diagrams and research posters.
- **PowerPoint:** We used PowerPoint to prepare our presentation slides.
- **Google Drive:** We used Google Drive for storing and sharing research files, datasets, models, and other essential documents in an orderly and accessible manner.
- **Google Meet:** We used Google Meet to carry out online meetings and research discussions with our team members and supervisor.
- **WhatsApp Group:** We created a WhatsApp group to enable fast communication and smooth coordination among team members during our whole research process.

Chapter 4

Result analysis

4.1 Xception Base Model Evaluation

This base model was trained for 60 epoch keeping the batch size 32, adam optimizer, learning rate at 0.001. All other hyperparameters are mentioned in Table 4.1 that has been used to train.

Table 4.1: Hyperparameter of the Xception Model.

Hyperparameter	Value
Batch Size	32
Epoch	60
Learning Rate	0.001
Optimizer	Adam
Activation	ReLU, Softmax
Early Stopping patience	10
Reduce learning rate factor	0.2
Reduce learning rate Patience	5
Reduce learning rate Min learning rate	1e-6

4.1.1 Loss and Accuracy Curve

Loss and accuracy curves help to analyze how a model learns during training and validation. Whether it overfits or underfits during training. To train the model, 80% data were fed in the training set, and 10% data were fed for validation.

Figure 4.1 shows the validation and training loss and accuracy curve over 60 epochs. Here, the loss curve shows that both validation and training loss decreased

significantly within the initial 25 epochs. The validation loss had minor fluctuations within 5 to 25 epochs while training loss decreased smoothly, which indicates the model struggled to generalize on the validation dataset in early epochs. Both validation and training loss stabilized after almost 23 epochs. However, when focused on the accuracy curve, it shows a noticeable zigzag pattern within 25 to 28 epochs, indicating high overfitting. After that, there is a consistent gap between the training and validation accuracy curves, which is fine but slightly overfitting.

After training for 60 epochs, this model obtained a training accuracy of 90.39% with a loss of 0.22 and a validation accuracy of 87.77% with a loss of 0.28 Figure 4.1. The inference time of the model is 53.14 seconds, and the FLOPs is 16.72 GFLOPs.

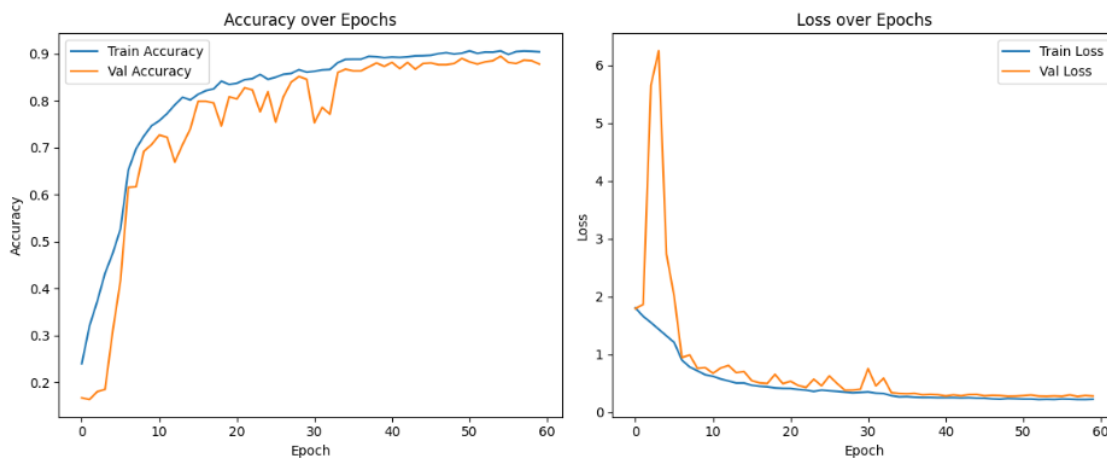


Figure 4.1: Loss and accuracy curve of the Xception model.

After the training process, using the test dataset model was evaluated. The test dataset contains 10% of the overall dataset. The test dataset was evaluated through a confusion matrix, classification report, and AUC-ROC curve.

4.1.2 Classification Report and Confusion Matrix Analysis

If we focus on the classification report (Figure 4.2, it shows that the precision, recall, and f1-score for anger, sad, fear, and surprise are very poor. Specifically, for the emotion sad it is 65% to 67% which means the model can't properly recognize emotions. As a result, it effected the overall accuracy. This base model's test accuracy is 86.88%. However, for the real-world scenario, this result is not acceptable.

Classification Report:				
	precision	recall	f1-score	support
Anger	0.7926	0.7589	0.7754	141
Fear	0.8784	0.9220	0.8997	141
Happy	1.0000	0.9858	0.9929	141
Neutral	0.9928	0.9716	0.9821	141
Sad	0.6577	0.6950	0.6759	141
Surprise	0.9051	0.8794	0.8921	141
accuracy			0.8688	846
macro avg	0.8711	0.8688	0.8697	846
weighted avg	0.8711	0.8688	0.8697	846

Figure 4.2: Classification report of the Xception model.

From the confusion matrix (Figure 4.3), we can see that emotion overlapping occurred here mostly between sad and anger. Anger was misclassified as sadness, and the false positive rate is 30 and the false negative rate is 33. Similarly, sad was predicted as fear and anger. However, neutral and happy were predicted much better than other emotion.

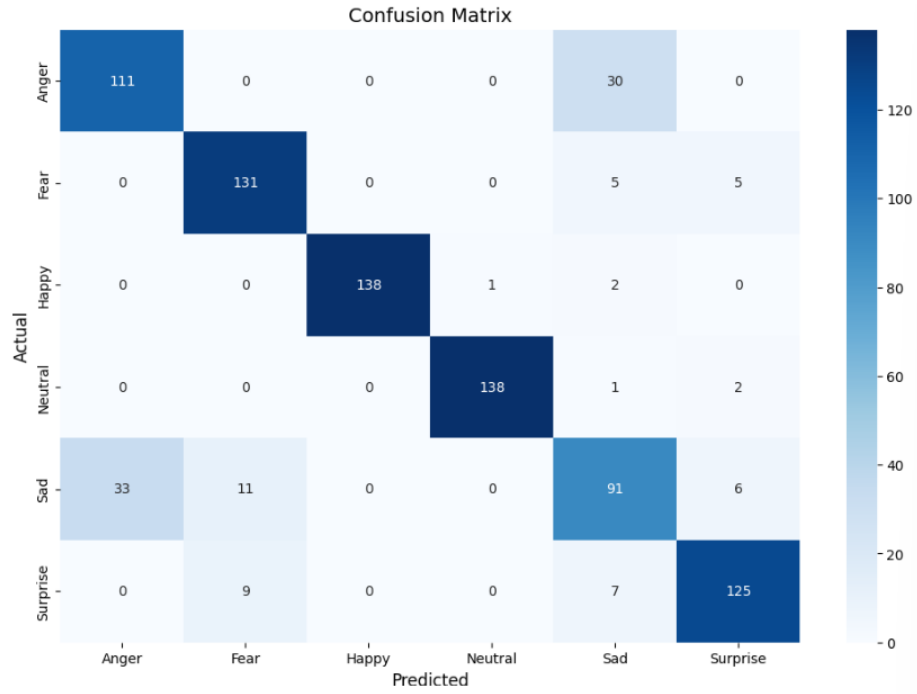


Figure 4.3: Confusion matrix of the Xception model.

4.1.3 AUC-ROC Curve Analysis

To give a deeper look, we have also plotted the ROC-AUC curve (Figure 4.4). Though the macro average AUC is 98.31% but for sad, AUC is 93.35% and for anger it is 97.6%. Which means this model failed to accurately distinguish emotions across the classes.

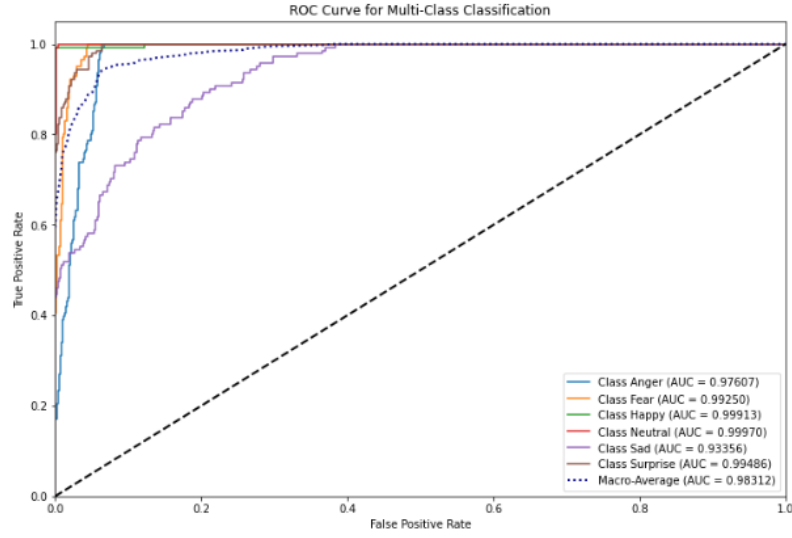


Figure 4.4: AUC-ROC curve of the Xception model.

4.2 Proposed Model Evaluation

We adopted a batch size of 32, Adam optimizer, GELU, ReLU, SoftMax activation function. To avoid overfitting, a callback function was used, keeping the min delta at 0.0001, the reduction factor at 0.2, and the patience at 3, all in the ReduceLROn-Plateau function, and some other hyperparameters to train the model, which is presented in Table 4.2.

Table 4.2: Hyperparameter of the Proposed Model

Hyperparameter	Value
Batch Size	32
Epoch	60
Learning Rate	0.0001
Optimizer	Adam
Activation	ReLU, GELU, Softmax
Dropout	0.3
Early Stopping patience	10
Reduce learning rate factor	0.2
Reduce learning rate Patience	3
Reduce learning rate Min Delta	0.0001

4.2.1 Loss and Accuracy Curve

We trained our proposed model with 80% data for 60 epochs. After training for 60 epochs with a learning rate of 0.0001, from the loss and accuracy curves in Figure 4.5 and 4.6 respectively, we can see that both train and validation loss is decreasing consistently, indicating that the model is learning effectively. From epoch near 23, the loss stabilizes, and the validation loss does not spike, and the gap is very small, which implies the model is well optimised and it is not overfitting. From Figure 4.6, we can see that there exists the slightest fluctuation compared to the base model, and after a certain epoch, it stabilizes. Our proposed model achieved a training accuracy of 97.99% with a loss of 0.057 and a validation accuracy of 96.64% with a loss of 0.1044. The inference time of the model is 9.146 seconds, and the FLOPs are 0.77 GFLOPs.

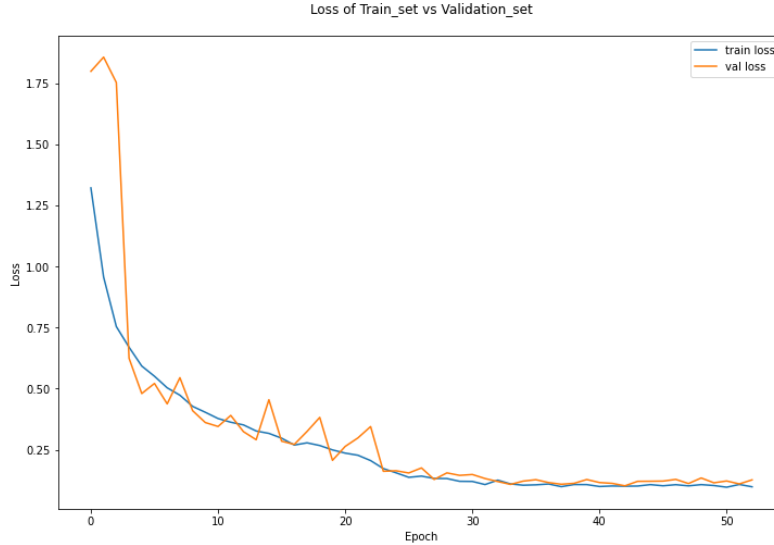


Figure 4.5: Train and validation loss of the proposed model.

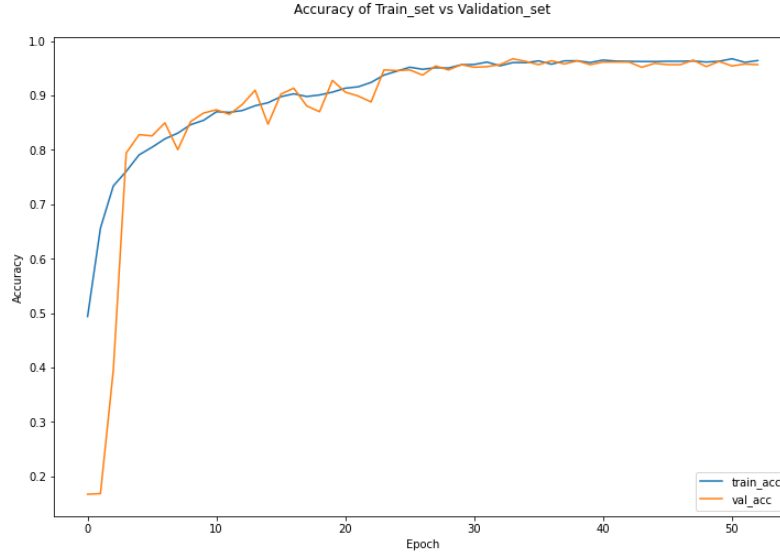


Figure 4.6: Training and validation accuracy of the proposed model.

After the training process, the model was evaluated on the test dataset. The test dataset contains 10% of overall dataset. The test dataset was evaluated through a classification report, confusion matrix, and AUC-ROC curve.

4.2.2 Classification Report and Confusion Matrix Analysis

Depending on this confusion matrix, if we focus on the classification report in Figure 4.7, our proposed model achieved precision, recall, and F1-scores ranging from 95 to 100%, which implies the model is extremely accurate at predicting emotions. By comparison, 'sad' had a relatively low range of 91 to 92 63%. However, the macro average precision, recall, and f1-score are 96.60%, 96.57%, and 96.58%, respectively, which indicates that the model is accurately identifying positive instances. It highlights the model's ability and robust classification performance.

Classification Report:				
	precision	recall	f1-score	support
Anger	0.9514	0.9716	0.9614	141
Fear	1.0000	1.0000	1.0000	141
Happy	0.9856	0.9716	0.9786	141
Neutral	0.9714	0.9645	0.9680	141
Sad	0.9167	0.9362	0.9263	141
Surprise	0.9710	0.9504	0.9606	141
accuracy			0.9657	846
macro avg	0.9660	0.9657	0.9658	846
weighted avg	0.9660	0.9657	0.9658	846

Figure 4.7: Classification model of the proposed model.

From the confusion matrix in Figure 4.8, we can analyze that fear had been accurately identified. On the other hand, Surprise and Sad had comparatively more errors. Here, the error for surprise is 11, where 7 were confused with sad. Similarly, Sad had 11 errors, where 6 samples were confused with Anger. Overall, fear, neutral, happy, and anger were predicted well compared to sadness and surprise.

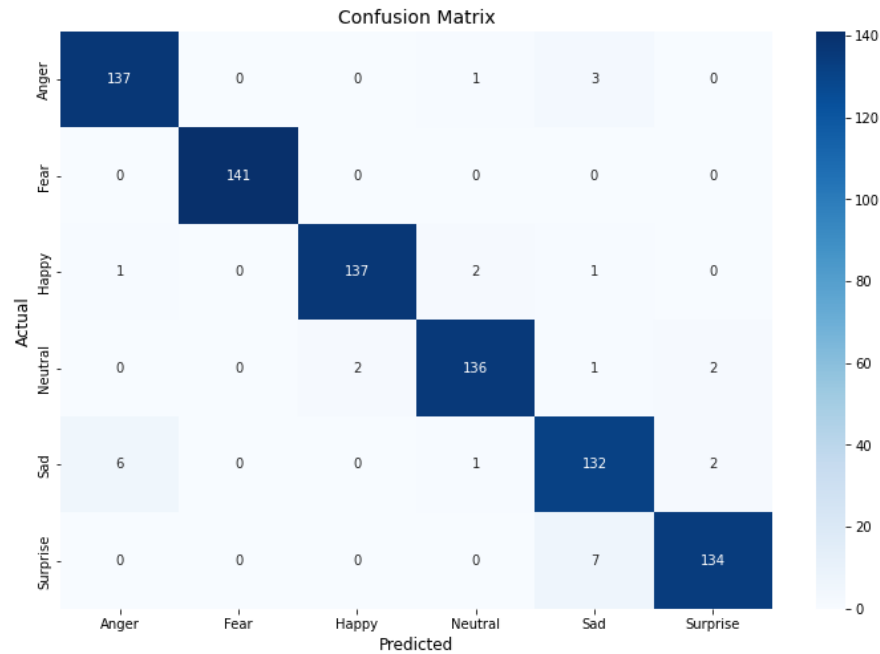


Figure 4.8: Confusion matrix of the proposed model.

4.2.3 AUC-ROC Curve Analysis

From the ROC curve in Figure 4.9, we can see that AUC score for fear is 1. That means model is accurately distinguishing this emotion. At the same time, the AUC score for anger, happy, neutral, sad and surprise is up to 99% and the macro average AUC is 99.89%, which indicates the model can distinguish between emotions across all classification thresholds.

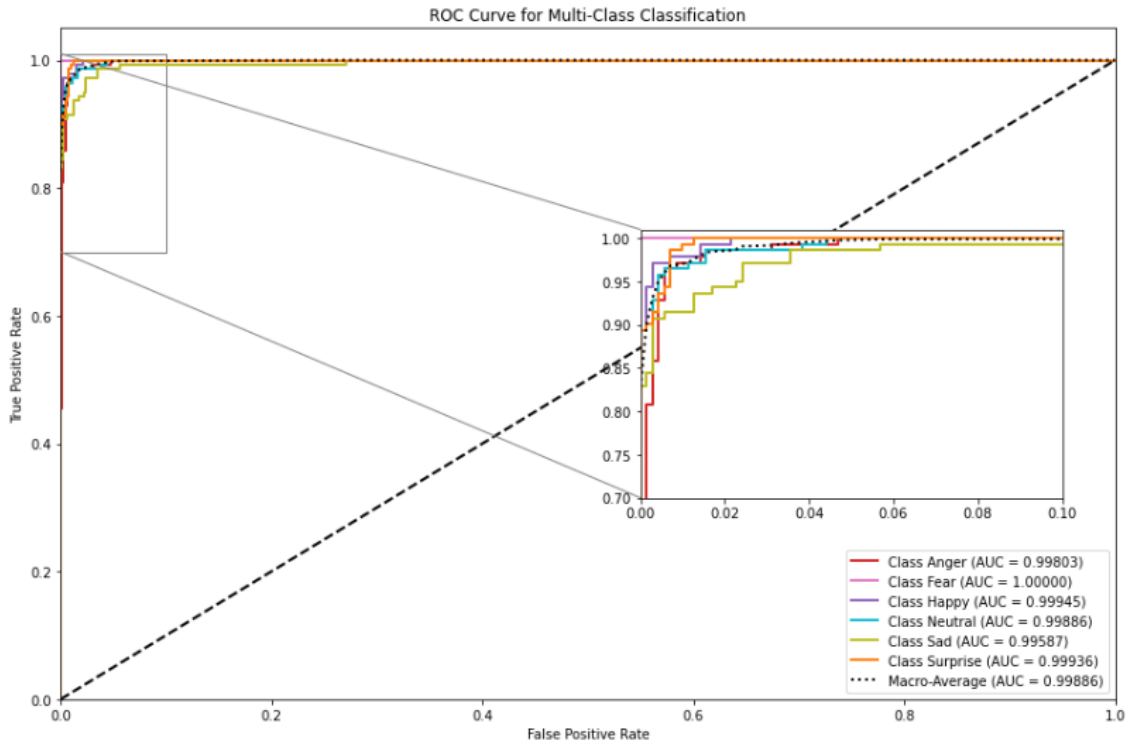


Figure 4.9: AUC-ROC curve of the proposed model.

4.3 Comparison Between Models

4.3.1 Class wise Performance Comparison

From class-wise metrics comparison, we can see that, in six emotions, anger, fear, sad and surprise had a notable improvement, with precision, recall, and F1-score increasing by 7.8% to near 25%. Where the base Xception model was struggling significantly to classify the emotion sad, our proposed model successfully classified sad with an improvement from 24.12% to 25.9% with precision, recall, and F1-score. The same goes for the emotion anger. Which means our proposed model

can effectively address this challenge. A slight decrease was observed in two classes, happy and neutral, and the range is 0.14% to 1.44%. However, precision, recall, and F1-score for happy and neutral ranges from 96.45% to 98.56% in our proposed model. So, this minor reduction does not significantly impact the overall performance. Also, focusing on all the emotion’s metric-wise performance, it shows that our model’s class-specific discrimination power is very strong, and the model is a robust and reliable system for multi-class emotion recognition.

Table 4.3: Class-wise Performance Comparison of the Base and Proposed Model. Here, the base model, proposed model, precision, and improvement are denoted as A, B, Prec., and Imp. respectively.

Emotion	Prec. (A)(%)	Prec. (B)(%)	Imp. (%)	Recall (A)(%)	Recall (B)(%)	Imp. (%)	F1-score (A)(%)	F1-score (B)(%)	Imp. (%)
Anger	79.26	95.14	+15.88	75.89	97.16	+21.27	77.54	96.14	+18.6
Fear	87.84	100	+12.16	92.2	100	+7.8	89.97	100	+10.03
Happy	100	98.56	1-1.44	98.58	97.16	-1.42	99.29	97.86	-1.43
Neutral	99.28	97.14	-0.14	97.16	96.45	-0.71	98.21	96.8	-1.41
Sad	65.77	91.67	+25.9	69.50	93.62	+24.12	67.59	92.63	+25.04
Surprise	90.51	97.1	+6.59	87.94	95.04	+7.1	89.21	96.06	+6.85

4.3.2 Metric-wise comparison

If we compare our proposed model with the model of [Talaat et al. \(2024\)](#), where they introduced an attention mechanism (sparse autoencoder) with the Xception model and with the base model, we can also find from Table 4.4 that there is a drastic improvement in our proposed model. Though the gap of accuracy between our proposed model and [Talaat et al. \(2024\)](#)’s model is 1.34% but if we look at the precision, recall, F1-score, and AUC, the gap is 3.4%, 2.36%, 3.25% and 8.55% respectively. This indicates that our proposed model is very robust, and it can perfectly distinguish emotions compared to the existing model we review. This proposed model provided it performs robustly by learning fine-grained details, coarse-grained details, and hierarchical features.

Table 4.4: Metric-wise comparison between models.

Metric	Xception Base Model	Xception with Sparse Autoencoder Talaat et al. (2024)	Proposed Xception Model	Improvement
Accuracy	86.88%	95.23%	96.57%	+ 1.34%
Precision	87.11%	93.2%	96.60%	+ 3.4%
Recall	86.88%	94.21%	96.57%	+ 2.36%
F1-Score	86.97%	93.33%	96.58%	+ 3.25%
AUC	98.29%	91.34%	99.89%	+ 8.55%

4.3.3 Computational Efficiency Comparison

For real-world deployment and resource-constrained environments, it is very important to evaluate not only class based performance but also computational efficiency.

Therefore, in this section we are presenting the computational comparison of our proposed Xception model against the base model of Xception, focusing on floating-point operations, inference time, and trainable parameters.

Table 4.5: Model Complexity Comparison.

Metric	Xception model	Proposed Xception Model
Inference Time (in seconds)	53.14	9.146
FLOPs (in billion)	16.72	0.77
Trainable parameter (in million)	20.82	28.62

From Table 4.5, we can see that the inference time for the Xception model is 53.14 seconds, FLOPs is 16.72 billion, and the trainable parameter is approximately 20.82 million. On the other hand, for our proposed model, the inference time is 9.146 seconds, FLOPs is 0.77 billion, and the trainable parameter is approximately 28.62 million. This proposed model is 5.8 times faster, and in terms of FLOPs, it is ~ 22 times more efficient, making the model suitable for latency-sensitive tasks in emotion recognition tasks. Although the trainable parameter of our proposed model is higher, the computational cost is lighter and faster; it indicates that the parameters are being used efficiently due to the design strategy of our model and the model is well optimized.

Thus, adding the Squeeze-and-Excitation (SENet) attention mechanism into the Xception model, it's giving channel-wise wise attention, extracting each channel's global representation more efficiently. Our proposed methodology has great potential to recognize the emotions of special children and mitigate the problems of overlap of emotions.

4.4 Proposed Web Application

In this application, the user has to upload an image of the facial emotions of children with autism. They can upload or directly capture the images through a webcam. When an image is submitted, it will be analyzed and generate the output with a short note at the top of the drop-down expander. Along with that, a percentage will also be displayed for all the emotions. From this, the user will get to know the accurate result of recognition.

Figure 4.10, shows the homepage of the proposed application. The user can drag or upload an image and then submit it. On the right side, the interface will show the outcome with a percentage. Next, Figure 4.11 shows that the uploaded image is being loaded to process and analyze. In the final step, Figure 4.12 shows the outcome. For example, the uploaded image is recognized as surprised with 100% accuracy.

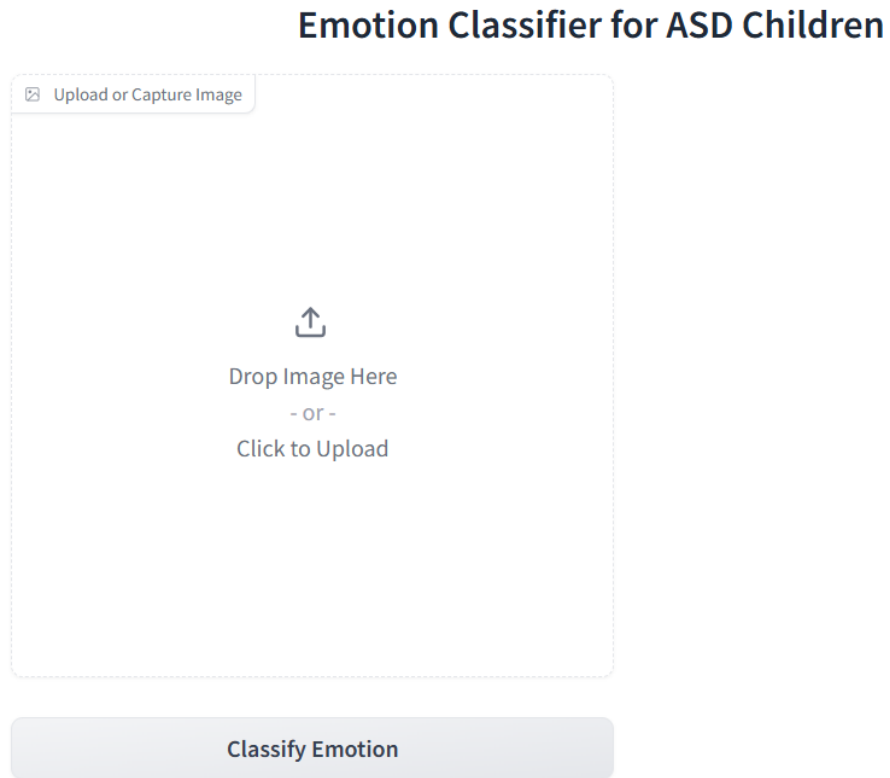


Figure 4.10: Home interface of the proposed web application.

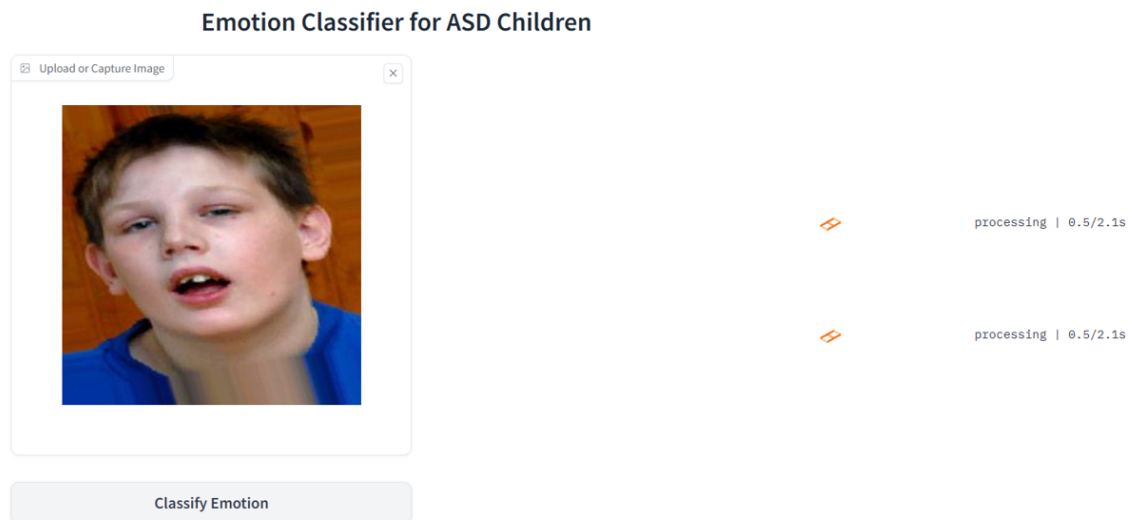


Figure 4.11: The application is analyzing the uploaded image.

Emotion Classifier for ASD Children

Upload or Capture Image



Classify Emotion

🧠 Predicted Emotion: Fear

📋 Detailed Probabilities:

Fear: 98.18%

Sad: 1.82%

Surprise: 0.00%

Neutral: 0.00%

Happy: 0.00%

Anger: 0.00%

Figure 4.12: Output interface of emotion recognition.

Chapter 5

Conclusions

5.1 Social and Sustainability Issues

5.1.1 Social Impact

Supporting the emotional well-being of children with Autism Spectrum Disorder (ASD) is important not just for families, but for society as a whole. Children with ASD often find it hard to express their emotions, which can make it difficult to form friendships, learn new skills, and feel a sense of belonging. These struggles can lead to loneliness, emotional distress, self-harm, and even serious mental health challenges [Maskey et al. \(2013\)](#). Alarming, studies have brought attention to the widespread occurrence of suicidal actions among the ASD individuals [Jackson et al. \(2018\)](#). Their parents also face heavy emotional and financial burdens, often feeling isolated due to public misunderstandings and stigma [Alshekaili et al. \(2019\)](#) [Lecavalier et al. \(2006\)](#) [Upoma et al. \(2020\)](#).

To support this need, we have developed a specialized Facial Emotion Recognition (FER) system tailored for children with ASD. By accurately recognizing their unique emotional expressions, our system strengthens emotional bonds within families and promotes healthier communication. Beyond individual families, our work has the potential to drive positive societal change. It can foster greater understanding, acceptance, and inclusion of individuals with ASD in schools, communities, and public spaces. By bridging emotional gaps, our system helps build a more compassionate society where every child is understood, valued, and supported.

5.1.2 Sustainability Strategy

Sustainable AI engineering involves creating systems that maintain long-term effectiveness while reducing energy demands. Balancing technological prowess with environmental mindfulness plays a key role in securing AI's long-term contributions to worldwide innovation.

Our project introduces a deep learning-based approach to help children with ASD improve the recognition of their facial emotions. Many children with ASD often have trouble expressing emotions in ways people easily understand. This can cause frustration, misunderstandings, and emotional pain. By creating a system that can understand these emotions accurately, caregivers and healthcare workers can respond in the best way, leading to better mental health for children. Early detection of emotional states allows for timely support, which is essential for their future health.

By delivering high performance while using resources wisely, the project supports long-term sustainability. The system uses less computing power while staying accurate, so people and small organizations with fewer technological resources can benefit. This approach makes AI tools easier to use, helping more people to take part in AI-assisted healthcare and learning. Moreover, this tool supports ASD children to take part in school and social life, reducing misunderstandings and making sure they feel included.

As AI continues to advance, the combination of efficient methods for emotion recognition can enhance accessibility and inclusivity in various fields. The project introduces an advanced AI system using Xception and attention mechanisms to read emotions from facial expressions.

Making AI efficient, open for growth, and future-ready helps develop smarter systems. These features help to grow healthcare and AI while building reliable and adaptable technology for everyone. The project promotes building tools that are flexible for helping society, leading to stronger industries and new ideas. Making the best use of resources is a key part of the project's plan for long-term success. The

system runs effectively without wasting too much power or computing capacity, reducing its impact on the environment. This approach proves that new technology can develop without causing damage to nature.

Choosing eco-friendly AI development keeps technological progress balanced, responsible, and helpful for society. It lays the foundation for future AI to be both effective and responsible, so its benefits are accessible to all communities.

5.2 Brief Summary

This paper addresses the problem of the Xception model that focuses on local receptive fields, giving equal importance to all channels, and fails to capture long-distance correlation due to lack of attention modeling. This causes texture bias towards the model Geirhos et al. (2018) and degrades the generalization power of complex images and struggles to give better accuracy on small datasets. Recognizing these limitations becomes even more critical when focusing on sensitive applications such as facial emotion recognition for children with Autism Spectrum Disorder (ASD) due to poor facial expression. Depending on that, the main objectives of this study were to implement an enhanced deep learning model (Xception architecture) to recognize emotion through facial expression of ASD children. At the same time optimise the deep learning model and strengthen interpretability or feature extraction by integrating an attention mechanism. To achieve our objectives, we proposed a depthwise separable convolutional neural network including a Squeeze-and-Excitation block (SE block) with a GELU activation function between the depthwise convolution and pointwise convolution. So that, when the feature is extracted parallelly through depthwise convolution and stacked together, the SE block can give attention to each channel to find out which channel carries the most important feature. After extraction, all the feature were extracted through pointwise convolution. Next, this proposed depthwise separable convolutional neural network were applied in Xception architecture. The three-building block (Entry flow, middle flow, and exit flow) of Xception architecture were modified by including 5×5 kernels, mixed activation functions (ReLU + GELU), and increasing and decreasing layer and channel numbers as well. After applying this attention-based Xception model to an ASD-specific dataset, we found that compared to the base model of

Xception architecture and existing attention based Xception model, our model has outperformed in distinguishing all the classes. Specifically, when the base model F1-score for sad, anger, and surprise was 67.59%, 77.54% and 89.21% our base model achieved it 92.63%, 96.14% and 96.06%. The base model had 86.88% accuracy with FLOPs of 16.72 billion and an inference time of 53.14 seconds, whereas our proposed model achieved 96.57% accuracy with FLOPs of 0.77 billion and an inference time of 9.146 seconds. The result suggests that, his proposed model is 5.8 times faster, and in terms of FLOPs, it is ~ 22 times more efficient, making the model suitable for latency-sensitive tasks in emotion recognition tasks. Thus, adding the Squeeze-and-Excitation (SENet) attention mechanism into the Xception model, its giving channels wise attention, extracting each channel’s global representation more efficiently. Our proposed methodology has great potential to recognize the emotions of special children and mitigate problems of overlap of emotion.

5.3 Future works

We intend to include data privacy, as in real life the application will collect the images of ASD children, preserve them, and then self-train the dataset for classification. At the same time, we want to include an IoT-based emotion recognition system so that whenever children move, that surveillance-car-like system will also move along with them, detect faces, and recognize emotions in real time.

References

- Alshekaili, M., Al-Balushi, N., Al-Alawi, M., Mirza, H., Al-Huseini, S., Al-Balushi, M., ... others (2019). Risk factors underlying depressive symptoms among parents/primary care providers of kids with autism spectrum disorder: A study from muscat, oman. *Perspectives in Psychiatric Care*, 55(4), 600–606.
- Awatramani, J., & Hasteer, N. (2020). Facial expression recognition using deep learning for children with autism spectrum disorder. In *2020 ieee 5th international conference on computing communication and automation (iccca)* (pp. 35–39).
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1251–1258).
- Chung, S., & Oh, U. (2019). Exploring the design space of an augmented display for conveying facial expressions for people with autism. In *2019 ieee international symposium on mixed and augmented reality adjunct (ismar-adjunct)* (pp. 435–437).
- Debnath, T., Reza, M. M., Rahman, A., Beheshti, A., Band, S. S., & Alinejad-Rokny, H. (2022). Four-layer convnet to facial emotion recognition with minimal epochs and the significance of data diversity. *Scientific Reports*, 12(1), 6991.
- El Rhatassi, F. E., El Ghali, B., & Daoudi, N. (2024). Deep learning approaches for recognizing facial emotions on autistic patients. *International Journal of Electrical & Computer Engineering* (2088-8708), 14(4).
- Faruk, M. O., Rahman, M. S., Rana, M. S., Mahmud, S., Al-Neyma, M., Karim, M. S., & Alam, N. (2023). Socioeconomic and demographic risk factors of autism spectrum disorder among children and adolescents in bangladesh: Evidence from a cross-sectional study in 2022. *PloS one*, 18(8), e0289220.
- Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2018). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations*.

- Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8), 2011–2023.
- Huang, Z.-Y., Chiang, C.-C., Chen, J.-H., Chen, Y.-C., Chung, H.-L., Cai, Y.-P., & Hsu, H.-C. (2023). A study on computer vision for facial emotion recognition. *Scientific reports*, 13(1), 8425.
- Jackson, S. L., Hart, L., Brown, J. T., & Volkmar, F. R. (2018). Brief report: Self-reported academic, social, and mental health experiences of post-secondary students with autism spectrum disorder. *Journal of autism and developmental disorders*, 48, 643–650.
- Komuraiah, B., Perna, S., Mudapu, V., Sridasyam, A., & Koyyada, N. (2024). Web-based facial emotion recognition system for enhancing social interaction in autistic spectrum disorder children using cnn & flask framework. In *2024 15th international conference on computing communication and networking technologies (icccnt)* (pp. 1–7).
- Kortli, Y., Jridi, M., Al Falou, A., & Atri, M. (2020). Face recognition systems: A survey. *Sensors*, 20(2), 342.
- Le, N., Nguyen, K., Tran, Q., Tjiputra, E., Le, B., & Nguyen, A. (2023). Uncertainty-aware label distribution learning for facial expression recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 6088–6097).
- Lecavalier, L., Leone, S., & Wiltz, J. (2006). The impact of behaviour problems on caregiver stress in young people with autism spectrum disorders. *Journal of intellectual disability research*, 50(3), 172–183.
- Luo, Z., Peng, Y., Dong, X., & Qian, H. (2023). Rotating machinery fault diagnosis using dimension expansion and antisymnet lightweight convolutional neural network. *Measurement Science and Technology*, 34(11), 115005.
- Mao, J., Xu, R., Yin, X., Chang, Y., Nie, B., Huang, A., & Wang, Y. (2024). Poster++: A simpler and stronger facial expression recognition network. *Pattern Recognition*, 110951.
- Maskey, M., Warnell, F., Parr, J. R., Le Couteur, A., & McConachie, H. (2013). Emotional and behavioural problems in children with autism spectrum disorder. *Journal of autism and developmental disorders*, 43, 851–859.
- Niu, H., Xu, W., Akbarzadeh, H., Parvin, H., Beheshti, A., & Alinejad-Rokny, H. (2020). Deep feature learnt by conventional deep neural network. *Computers & Electrical Engineering*, 84, 106656.

- Pavez, R., Diaz, J., Arango-Lopez, J., Ahumada, D., Mendez-Sandoval, C., & Moreira, F. (2023). Emo-mirror: a proposal to support emotion recognition in children with autism spectrum disorders. *Neural Computing and Applications*, 35(11), 7913–7924.
- Rajan, V., Brutti, A., & Cavallaro, A. (2022). Is cross-attention preferable to self-attention for multi-modal emotion recognition? In *Icassp 2022-2022 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 4693–4697).
- Rani, P. (2019). Emotion detection of autistic children using image processing. In *2019 fifth international conference on image information processing (iciip)* (pp. 532–535).
- Saranya, A., & Anandan, R. (2022). Facial action coding and hybrid deep learning architectures for autism detection. *Intelligent Automation & Soft Computing*, 33(2).
- Silva, V., Soares, F., Esteves, J. S., Santos, C. P., & Pereira, A. P. (2021). Fostering emotion recognition in children with autism spectrum disorder. *Multimodal Technologies and Interaction*, 5(10), 57.
- Talaat, F. M., Ali, Z. H., Mostafa, R. R., & El-Rashidy, N. (2024). Real-time facial emotion recognition model based on kernel autoencoder and convolutional neural network for autism children. *Soft Computing*, 28(9), 6695–6708.
- Uddin, M. J., & Ashrafun, L. (2023). Parenting in context: parents’ experiences of caring for a child with autism in bangladesh. *International Journal of Anthropology and Ethnology*, 7(1), 10.
- Upoma, T. F., Moonajilin, M. S., Rahman, M. E., & Ferdous, M. Z. (2020). Mothers initial challenges having children with autism spectrum disorders in bangladesh. *Bangladesh Journal of Medical Science*, 19(2), 268–272.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 7794–7803).
- Wang, Y., Pan, K., Shao, Y., Ma, J., & Li, X. (2025). Applying a convolutional vision transformer for emotion recognition in children with autism: Fusion of facial expressions and speech features. *Applied Sciences*, 15(6), 3083.
- Wang, Z., Zeng, F., Liu, S., & Zeng, B. (2021). Oaenet: Oriented attention ensemble for accurate facial expression recognition. *Pattern Recognition*, 112, 107694.

- Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the european conference on computer vision (eccv)* (pp. 3–19).
- Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3), 107–115.
- Zhang, Y., Wang, C., Ling, X., & Deng, W. (2022). Learn from all: Erasing attention consistency for noisy label facial expression recognition. In *European conference on computer vision* (pp. 418–434).
- Zhuang, X., Liu, F., Hou, J., Hao, J., & Cai, X. (2022). Transformer-based interactive multi-modal attention network for video sentiment detection. *Neural Processing Letters*, 54(3), 1943–1960.