Human Face Emotion Recognition in Thermal Images Using cGAN and EfficientNet with Attention Mechanisms

Tahir Aman Shuba

A Thesis Submitted to the Department of Computer Science and Engineering,

College of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

October, 2025

Adama, Ethiopia

Human Face Emotion Recognition in Thermal Images Using cGAN and EfficientNet with Attention Mechanisms

Tahir Aman Shuba

Advisor: Sintayehu Hirpassa (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering,

College of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

October, 2025

Adama, Ethiopia

DECLARATION

I declare that this Master Thesis entitled "**Human Face Emotion Recognition in Thermal Images Using cGAN and EfficientNet with Attention Mechanisms**" is my own work and has not been submitted to any university for similar purpose. The references used in this Thesis are duly recognized by proper citations.

Tahir Aman Shuba	______________________	_______________
Name of student	Signature	Date

RECOMMENDATION OF ADVISORS

I, the major advisor of this Thesis, hereby certify that I have closely advised the student while developing this Thesis and read the draft Thesis entitled "**Human Face Emotion Recognition in Thermal Images Using cGAN and EfficientNet with Attention Mechanisms"** prepared under my guidance by **Tahir Aman Shuba**. Therefore, I recommend the submission of the Thesis to the department for further review and evaluation.

Sintayehu Hirpassa (Ph.D.)	___________________	_________________
Major Advisor	Signature	Date

APPROVAL SHEET

I, the major advisor of this Thesis, hereby certify that the recommendation and suggestion given by the research review committee are appropriately incorporated into the final Thesis entitled "**Human Face Emotion Recognition in Thermal Images Using cGAN and EfficientNet with Attention Mechanisms"** by **Tahir Aman Shuba.**

Sintayehu Hirpassa (Ph.D.)	____________________	__________________
Major Advisor	Signature	Date

APPROVAL OF BOARD OF REVIEWERS

We, the undersigned, members of the Board of Reviewers of the research open defense by **Tahir Aman Shuba** have read and evaluated the Thesis entitled "**Human Face Emotion Recognition in Thermal Images Using cGAN and EfficientNet with Attention Mechanisms"** and assessed the understanding of the candidate about the research. This is, therefore, to certify that the Thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

Chairperson	Signature	Date
Reviewer 1	Signature	Date
Reviewer 2	Signature	Date

Final approval and acceptance of the Thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
College Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

I begin by expressing my deepest and profound gratitude to Allah for granting me numerous blessings that have helped make actualization of this achievement a reality. I am especially thankful for the gift of sustained health and the unwavering resilience that empowered me throughout this journey.

I would like to extend my deepest and sincerest gratitude to my esteemed advisor, **Dr. Sintayehu Hirpassa (Ph.D.)**, whose guidance played a critical role in successful accomplishment of this work. His constant encouragement, insightful guidance, and expert guidance played a key role in realization of this work; its accomplishment could not have been achieved but for him. I also extend my warmest and sincerest appreciations to **Dr. Mesfin Abeba (Ph.D.)** and **Dr. Bahiru Shifaw (Ph.D.)** for their diligent supervision and useful recommendations, whose contribution helped make a significant impact in enhancing this study and shaping its development.

Lastly, I extend my deepest and sincerest appreciations to my fellow group members of the Artificial Intelligence Special Interest Group, my lovely family, and my supportive friends. Their encouragement and patience helped instill my determination, and I value their contribution immensely.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATION

LIST OF ACRONYMS

AI	Artificial Intelligence
CFTD	Comprehensive Facial Thermal Dataset
cGAN	Conditional Generative Adversarial Networks
CNN	Convolutional Neural Network
CV	Computer Vision
CLAHE	Contrast Limited Adaptive Histogram Equalization
D	Discriminator
DenseNet	Densely Connected Convolutional Network
DL	Deep Learning
FER	Facial Emotion Recognition
FID	Frechet Inception Distance
FN	False Negative
FP	False Positive
G	Generator
GANs	Generative Adversarial Networks
GAP	Global Average Pooling
GMP	Global Max Pooling
GPU	Graphic Processing Unit
IR	Infrared
L	Loss
ML	Machine Learning
ResNet	Residual Network
RAM	Random Access Memory
SE	Squeeze-and-Excitation
tFER	Thermal Face Emotion Recognition
TN	True Negative
TP	True Positive
UNI-T	Uni-Trend Technology
YOLO	You Only Look Once

ABSTRACT

Human use emotions to express their feelings and effectively interact with others. Human express emotions through hands, voice, gestures, and most importantly, facial expressions. Facial emotion recognition widely used in human-computer interaction, security, and healthcare. Traditionally based on posed visible images, such methods fail to capture natural emotions and are prone to lighting conditions. Visible spectrum imaging has several limitations, including sensitivity to lighting conditions, facial obstructions, and potential variations in facial expressions. Thermal imaging is less affected by obstructions of the face and changes in illumination. Thermal facial emotion recognition offers a robust alternative to conventional visible-spectrum methods by mitigating the effects of lighting variability and deceptive facial expressions. However, the use of thermal images for emotion recognition has not been extensively explored, primarily due to the limited availability of thermal image datasets. This research investigated the impact of enhanced preprocessing, synthetic data generation, and advanced deep learning architectures on thermal emotion classification. In this paper we used bilateral filters, Gaussian Blur and CLAHE for image preprocessing, data augmentation, cGANs for synthetic data generation, and EfficientNet-B5with CBAM for feature extraction. We utilized CBAM on the EfficientNet-B5 blocks. Our proposed method passed the extracted features to ResNet-18 for the classification of human facial emotions of thermal images into five emotional expressions such as happy, sad, angry, natural, and surprise. The proposed method used comprehensive facial thermal datasets for training and testing to compare its performance on baseline work. We employed the Fréchet Inception Distance to evaluate the realism and quality of the generated images. Accuracy, precision and F1-score metrics are utilized to assess the performance of the model. Experimental results demonstrated a significant improvement in recognition performance, achieving an accuracy of 98.81%. These findings highlight the effectiveness of the proposed pipeline for thermal facial emotion recognition and suggest promising potential for deployment in real-world conditions.

Keywords: Emotion recognition, Face Emotion Recognition, Facial expression, Face recognition, Thermal image

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Emotion represents a feeling-dominated state which affects behavioral reactions as well as psychological and physical alterations. They arises from a blend of psychological and physiological processes, influenced by both internal and external stimuli,and serve as a means for humans to communicate and express feelings effectively. Due to rapid advancement in technology in the past decade, machines are now capable of identifying and categorizing human emotions (Krishna et al., 2019; Agung et al., 2024).

Humans express their emotions through various means, including the movement of hands, pitch of voice, gestures, and above all, through facial expressions, which account for 55% of emotional communication (Krishna et al., 2019). The human face provides crucial information on emotional states and behaviors, as facial expressions are essential for understanding others' motives during social interactions. Consequently, facial recognition has emerged as a critical area for decoding human emotions. Among the diverse techniques in this domain, Facial Emotion Recognition has demonstrated impressive results (Agung et al., 2024). Machine learning facilitates FER by recognizing emotions from images or videos, thereby leveraging valuable insights into the underlying emotional states of individuals that are pertinent to social interactions. With FER aiding in the interpretation of others' feelings and intentions, it has high relevance in many areas, including healthcare, education, workplace environments, robotics, communication, as well as psychology, among others (Jacintha et al., 2019).

Facial responses are often good predictors of the health or emotional state of the person and are typically best categorized among one of the seven universal emotions: aggression, anxiety, astonishment, sadness, hatred, joy, or indifference. Emotion identification is the act of determining the emotional status of the person, but the act of making an attempt at discovering the emotional experience of the individual is referred to as emotion detection. Emotion-reading ability among humans is highly disparate, and the application of technology for the facilitation of better emotion identification is itself an area where growth has been occurring at high rates (Dada et al., 2023a).

Facial expressions act as a mirror to human emotions, serving as a nonverbal means of communication. Automated emotion recognition has emerged as a reliable tool for detecting deception during interrogations and is also utilized in assessing patient pain levels, stress analysis, and counseling. There are significant limitations associated with facial emotion recognition through visible images, including the simplicity with which individuals can feign expressions and the considerable influence of lighting conditions on the quality of images (Thampi et al., 2021).

Recently, extensive research has been conducted to create a system for recognizing facial emotions through machine learning and deep learning methodologies. This advancement not only contributes to the enhancement of facial emotion recognition systems but also introduces new parameters for analysis. Images used for classifying facial expressions are taken from cameras operating within the visible light spectrum, as these cameras are widely found in both handheld devices and attachments for portable gadgets like smartphones and tablets, which are affordable. While numerous studies have been undertaken to recognize facial expressions from images taken with standard visible light cameras, there is still much to explore in this area(Bhattacharyya et al., 2021), Classifying facial expressions can be challenging due to everyday factors such as shadows, reflections, and low-light conditions. Various objects in the background, including scenery and other elements, can complicate the task of isolating the face for expression analysis. The use of thermal images can mitigate these challenges, as they focus on the thermal distribution in facial muscles, leading to improved accuracy in facial expression classification.

Thermal imaging relies on the principle of heat radiation emitted from the skin of the face. Thermal images are employed to evaluate both immediate and chronic stress conditions, as these factors significantly impact blood flow. Variations in blood circulation result in changes in facial temperature, and these temperature fluctuations aid in detecting and refining the thermal imaging process (Chaitanya, 2020).

The method of transforming heat into visible images is referred to as thermal imaging. Infrared energy given off by an object is referred to as the heat signature of the object, which is proportional to its temperature.Generally, the temperature of the human body and face ranges from 35.5 to 37.5 °C, leading to a stable thermal signature. The unique patterns of superficial blood vessels, veins, and tissue structures beneath the skin significantly contribute to these thermal patterns, making them distinctive for each individual. Facial emotion recognition using thermal imaging is unaffected by skin color or facial features. Temperature on the face elevates when there is an external sound, artifact, or event (Thampi et al., 2021).

Facial thermal signatures are non-intrusive physiological signals and hence have immense potential for emotion recognition. Emotion detection using thermal images might be more feasible in particular situations, such as at night or during daytime low-light conditions. However, it must be noted that, on the whole, emotion recognition from thermal images is tougher compared to visible images (S. Wang et al., 2014).

Presently, much emphasis has been paid to pattern recognition, computer vision, and related studies of facial expressions. The facial emotions may reveal the person's health and internal well-being, which are commonly classified into seven traditional classes, namely aggression, anxiety, astonishment, sadness, hatred, happiness, and indifference. The recognition of human emotions is also called emotion recognition, where people's performance in identifying other people's emotions greatly varies. The application of technology to assist in recognizing emotions is a field of emerging research; the specific task of identifying an individual's emotional experience is known as emotion detection (Dada et al., 2023a).

For decades, much emphasis has been placed on human emotions, starting with pioneering research by Darwin on emotional expression in humans and animals. The area has since been enriched by the involvement of psychologists, computer scientists, biologists, and sociologists, among others. Facial expressions are an important part of human interaction, which is instinctive and usually displays one's thoughts and feelings without even saying a word. The face is important to social communication and interaction: it expresses the personality, emotion, feelings, and sentiments of an individual. As a generally recognized characteristic feature of being human, there have risen systems that try to create an interface much more usable and easier towards user experience (Dada et al., 2023b).

This study proposed human face emotion recognition in thermal images using conditional Generative Adversarial Networks and EfficientNet with an attention mechanisms.

1.2. Motivation of the Study

The motivation for this study arises from the potential to improve the capabilities of human face emotion recognition through the use of thermal images. This motivation is constituted of several factors:

The conventional methods of emotion recognition usually suffer from issues in dealing with lighting changes, facial obstructions and differences in the expressive behavior of different individuals. The present work aims to overcome these drawbacks using thermal imagery to build a more robust solution.

Progress in technological innovation, especially regarding the deep learning methodologies of EfficientNet and attention mechanisms, offers a great opportunity to enhance the processes of feature extraction and representation in thermal imagery. This study exploited recent technological advancements to develop a more effective model for emotion recognition.

Thirdly, accurate emotion recognition holds substantial potential in applications such as mental health monitoring, security systems, human computer interaction, and marketing campaigns. Improvement in emotion recognition through thermal imaging technology could enable innovative applications that rely on robust and reliable emotional information.

Lastly, the lack of datasets with labeled thermal images is a big hurdle in training deep-learning models. In order to tackle this challenge, this research employed preprocessing, cGANs to generate synthetic data, integrate efficient-net with attention mechanism thereby augmenting existing datasets, generating synthetic data, feature extraction and improving both the model training and overall performance.

1.3. Statement of the Problem

Emotion recognition using facial expressions has gained significant attention through the use of visible-spectrum images accompanied by convolutional neural networks and machine learning techniques. Emotion recognition based on visible images is often unreliable due to frequent interference by illumination and intentional manipulation of facial expressions Aiswarya K. Prabhakaran (2021). By comparison, thermal imaging provides a more robust solution due to its ability to capture thermal emissions radiating from the human body and making it less prone to variations in exterior lighting and less vulnerable to intentional facial manipulation. Though thermal image-based emotion recognition has numerous advantages, applications have remained largely uninvestigated due to a shortage of thermal image datasets. Previous studies, such as Assiri & Hossain (2023) research, does not fully account for the most relevant facial features; Aiswarya K. Prabhakaran (2021) relies upon a small dataset lacking augmentation; Chaitanya (2020) presents poor precision rates for certain emotions; and Elbarawy et al. (2019) make use of datasets where respondents only display a narrow range of expressions. Prasad & Chandana (2023) lack of preserving fine facial features, relies upon a small subjects and lacking augmentation. In response to these shortcomings, this research employed preprocessing, cGANs for synthetic data generation and integrated EfficientNet with an attention mechanism for feature extraction. Such a methodology expands the thermal image datasets and highlights most relevant parts in thermal images and hence improves facial emotion recognition accuracy.

1.4. Research Questions

In solving the problem stated above, attempts are made to answer the following questions:-

RQ1: What is the impact of applying Bilateral Filter and CLAHE on enhancing the quality of thermal images and their role in improving thermal facial emotion recognition accuracy?

RQ2: What is the impact of generated synthetic thermal facial images in mitigating data scarcity and improving the classification performance of the thermal facial emotion recognition?

RQ3: What is the impact of integrating EfficientNet with attention mechanisms in enhancing discriminative feature extraction and improving classification accuracy in thermal facial emotion recognition?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of the study is human face emotion recognition in thermal images using cGAN and EfficientNet with attention mechanisms.

1.5.2. Specific Objective

The following specific objectives are addressed to achieve the general objective.

- To review related works and identify current trends to come up with gaps in face emotion recognition in thermal images.
- To evaluate the impact of Bilateral Filter and CLAHE in enhancing the quality of the thermal images and improving the accuracy of thermal facial emotional recognition
- To evaluate the impact of generated synthetic thermal facial images in mitigating data scarcity and improving the classification performance of the thermal facial emotional recognition
- To evaluate the impact of integrating EfficientNet with an attention mechanism in enhancing discriminative feature extraction and improving classification accuracy in thermal facial emotional recognition

1.6. Significance of the Study

The significance of this thesis work extends across multiple domains as outlined below:

Advancement in Emotion Recognition Technology: Emotion recognition technology might improve through this research by sticking to the study of thermal images, in such a way that machine can understand human feelings better and make human computer interaction more natural.

Applications in Security and Surveillance: Improvements in the technologies of emotion recognition can help to improve the security frameworks by detecting unusual behavior or potential threats through emotional signals, thus creating a safer environment.

Healthcare Innovations: This is particularly useful in healthcare for the monitoring of patient conditions, the assessment of pain levels, and the improvement of patient care through appropriate timely interventions.

Social Robotics and AI: The results obtained in this research will guide the developing of more empathetic artificial intelligence systems and social robots, by enhancing their ability to understand human emotions, which will finally result in a better user experience and quality of interaction.

1.7. Scopes and Limitation of the Study

1.7.1. Scope of the Study

This study focused on the development of high accuracy and robust facial emotion recognition using thermal imaging and state of the art deep learning techniques. The study focused on the utilization of thermal images that provide dependable heat patterns, independent of the lighting conditions and manipulative facial expressions, as an alternative to visible spectrum images in FER. The study investigated the use of cGANs to increase the quality and diversity of training datasets, combined EfficientNet with attention mechanisms for the feature extraction and evaluated the overall performance of the thermal facial emotional recognition. This study reviewed current thermal image datasets concerning emotion recognition.

1.7.2. Limitations of the Study

The classification comprised just covered a small set of emotional classes like happy, sad, angry, surprised, and neutral, thereby excluding more complicated or subtle emotional states.

The data set was not varied in terms of ethnicity, age groups, and lighting. This can create bias and affect the model's robustness across different demographics.

This study relied on thermal imagery alone for emotion recognition and does not combine other complementary modalities such as visible spectrum images, audio, facial landmarks, or physiological signals.

1.8. Contributions of the Study

The application of cGANs greatly increases datasets diversity through the creation of synthesized thermal face images, solving the problem of small thermal image datasets availability for facial emotion recognition. This strategy enhances generalization capacity and optimizes feature extraction by supplying deep learning architectures with an increased number of variations, particularly with the application of EfficientNet, i.e., the EfficientNet-B5 model, which is highly efficient and scalable. The incorporation of channel and spatial attention mechanisms enables the model to focus on salient areas of the face, improving the emotion recognition accuracy by facilitating it to better capture subtle changes in thermal facial expressions.

Moreover, the models was trained on both original and synthetic data and subsequently tested on the original thermal images, demonstrating that computer generated thermal images can significantly improve the validity and robustness of facial emotion classification.

This approach demonstrates the beneficial effect of data augmentation on the models' generalization capability. Furthermore, thermal imaging presents an effective alternative to the problem of low light or variable lighting conditions to which visible light cameras are susceptible. The study discussed to what extent deep learning models retain their efficacy with different temperatures and environmental conditions, offering interesting perspectives on the robustness of emotion recognition in thermal images for practical applications. The research is significant in the domain of HCI since the technology promotes affective computing through improved ability of machines to recognize emotions in a broader range of situations. The possible applications are extensive and include areas such as security, healthcare, driver monitoring, and education, where proper emotional recognition may prove essential. In general, this research facilitated the future implementation of noninvasive emotion detection via thermal imaging, thereby establishing the basis for revolutionary applications in a number of domains.

1.9. Organizations of the Study

This research is organized into seven chapters, and each is structured to be a different phase of the research process. Chapter one is an introduction and it highlights the significance of the study and discusses the particular problem that inspired this research. Chapter Two is a thorough literature review of pertinent literature, including key areas like deep learning and machine learning methods, data augmentation methods, face recognition concepts, and in-depth analysis of current research in the field of thermal face emotion recognition. Chapter Three explains the methodology followed in the research, covering datasets collection process, the preprocessing of data, development environment tools, baseline model selection and implementation, as well as measures adopted for the performance evaluation. Chapter Four details the core contribution of the study by elaborately describing the adopted model's structure, model learning functions used along with a complete pseudo code walk through of the training procedure. Chapter Five provides a detailed explanation of the implementation of the model, including example code snippets, the actual values of the hyper parameters during the experiments, and a detailed discussion of the experiment class. Chapter Six presents the experimental results obtained using the proposed method, which discusses the results in full and compares them with those produced by previous studies in order to determine the weaknesses and strengths of this methodology. Finally, Chapter Seven is the conclusion, which gives an overview of the key findings of the research, offers key insights, and provides directions for future research based on the results and any remaining open questions.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Overview of Thermal Face Emotion Recognition

Facial emotion recognition via thermal means utilizes infrared thermal imaging technology to identify and measure emotional states through the changes in facial temperature (Chirag Kumar Kyal, 2021). The technique provides a contact free and noninvasive means of gauging emotional responses, making it highly convenient in scenarios where conventional approaches might not be feasible (Assiri & Hossain, 2023).

2.2. Thermal Face Emotion Recognition

Thermal Face Emotion Recognition takes advantage of thermal imaging to detect human emotions based on facial temperature variations. The emerging field faces a number of challenges such as the requirement for subject independence, limitation due to datasets availability, and demand for complex algorithms. New studies have introduced new frameworks and datasets that enhance the accuracy and robustness of tFER systems, thus making practical applications in various fields feasible.

Classical models often struggle to generalize across different subjects,which leads to poor recognition performance of diverse emotional expressions (Rooj et al., 2024).In addition, the limited availability of thermal facial image datasets restricts the development of robust and reliable emotion recognition systems (Rooj et al., 2024; Abuhussein et al., 2024).

Feature extraction is a critical part of enhancing recognition accuracy since the detection of subtle variations in temperature can represent various emotional states (Rooj et al., 2024; Pavel et al., 2023).

Despite the great promise achieved by tFER, there are still difficulties in reaching universality for various populations and emotional conditions.

2.3. Thermal Face Emotion Recognition Techniques

Thermal facial emotion recognition detects human emotions with thermal imaging technology, which reads facial surface temperature change (Assiri & Hossain, 2023). Emerging technologies in deep learning techniques and large datasets availability have greatly promoted tFER precision and usability across various fields.

2.3.1. Traditional Thermal Face Emotion Recognition Techniques

Classic methods of facial emotion recognition based on thermal imaging. Conventional methods of facial emotion recognition from thermal imaging utilize the temperature gradients on the face to classify and detect human emotions (Assiri & Hossain, 2023). The methods are especially applicable where visible light-based systems encounter difficulties, particularly under low-light conditions. The procedures entail thermal image capture, preprocessing, and the use of machine learning algorithms for the recognition of emotional states. While traditional approaches to facial emotion recognition using thermal imaging have shown promise, they face challenges such as limited datasets sizes and a need for greater understanding of thermal features (Chirag Kumar Kyal, 2021).

2.3.2. Techniques of Thermal Face Emotion Recognition Using Deep Learning

Computer Vision is a subfield of AI that deals with enabling computers and machines to understand and analyze visual information, such as images and videos. Like human sensing, these systems derive meaningful information from visual input (Bharat, 2023).

Computer Vision allows computers to read and make sense of the visual world in ways not too different from human perception. Starting from acquiring images or video frames via cameras or sensors, the raw visuals are preprocessed to increase the quality and reliability of the data.

Machine Learning, focuses on creating algorithms that allow computers to learn from data. Algorithms like these allow machines to look at the data, make decisions, and produce knowledgeable predictions or decisions (Bharat, 2024). In a manner analogous to human learning through the interpretation of sensory information, Machine Learning aims to derive decisions based on data inputs, with algorithms functioning as the fundamental driving forces. Deep

Learning, a subset of Machine Learning, employs neural networks to emulate the operational mechanisms of the human brain within computational systems. Within these networks, neurons are systematically organized and trained to utilize extensive datasets, thereby enabling deep learning models to effectively process and analyze information in real-time, similarly to how algorithms facilitate Machine Learning.

Deep Learning models are composed of several layers of interconnected nodes in neural networks, where each layer passes information to the next in the hierarchy for analyzing and identifying patterns. By capturing high-level and low-level data, these models distill complex datasets into more available and meaningful categories.

Image classification, as part of image analysis, refers to the classification of whole images, not just the identification of unique parts. CNNs, a type of deep learning architecture, are especially good at tasks related to images because they can learn complex hierarchies of features. The ability of CNNs to recognize complex patterns enables them to achieve incredibly accurate image classifications; thus, they are considered the state of the art machine learning approach to visual object recognition. CNNs have become the predominant machine learning methodology for the task of visual object recognition (Huang et al., 2016).

Facial recognition technology encompasses the processes of identifying and verifying individuals by examining their facial features. This innovation has diverse applications, ranging from enhancing security measures through authentication and access controls to facilitating engaging filters in entertainment and aiding law enforcement agencies in recognizing suspects via surveillance recordings (Bharat, 2023).

Beyond recognition, Computer Vision also enables image creation and manipulation. Generative models, such as GANs, can produce highly realistic images, supporting artistic creativity, content generation, and data augmentation to enhance machine learning model training.

2.4. Bilateral Filtering

Bilateral filtering is an effective image processing technique used in many image enhancement applications, such as stereo vision. It is a nonlinear filtering technique where spatial and content-based information is combined. While it reduces noise and smooths images, it does a good job at edge preservation. As an efficient image-processing approach, bilateral filtering not only protects image edges but also excels at removing salt and pepper noise. What characterizes this approach, though, is its ability to retain the sharpness of edges but reduce the noise, a fundamental feature of many image applications. In the field of denoising and enhancement in images, the clarity of edges in the processing result very often significantly enhances visual quality (Wu & Zhong, 2024). With facial emotion recognition from thermal images, it is crucial to preserve the subtle facial details around points like the eyes, mouth, and forehead, as they contain unique distinguishing features for emotion classification purposes(Assiri & Hossain, 2023).

2.5. Gaussian Blur

Gaussian blur is a filter that minimizes image detail and noise by averaging pixel values within a given radius. Gaussian blur is a very common method used in image processing for noise reduction and image smoothing. It works by convolving an image with a Gaussian function, which has the effect of blurring the image while to some degree retaining edges. This process is very crucial in various applications, including facial detection, image restoration, and 3D scene reconstruction.

Blur can be brought about by very many factors, including motion that is encountered while taking a photograph with a camera or when one uses a non-focused camera in photographing (Andriyani, 2024).

$$G(x, y) = \frac{1}{2\pi\sigma^2} \cdot exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Equation 2.1: Gaussian Blur Equation

- μ is the mean or "center" of the distribution, which shifts the peak of the bell curve along the x-axis.
- σ\sigma is the standard deviation, which controls the width of the bell curve. A smaller σ results in a steeper curve, while a larger σ produces a flatter curve.
- *e* is the base of the natural logarithm, approximately equal to 2.71828.
- *x* and y Gaussian distributions. the standard deviation
- σ represents the standard deviation of the distribution in both the *x* and *y* directions. It measures the spread or dispersion of the Gaussian distribution. A smaller σ\sigma results in

a steeper peak, indicating that the values are tightly clustered around the mean. A larger σ results in a flatter curve, indicating that the values are more spread out from the mean.

- Π is a mathematical constant approximately equal to 3.14159. It is used here as part of the normalization factor to ensure that the total area under the Gaussian surface equals 1. This is important for probability density functions.

2.6. Contrast Limited Adaptive Histogram Equalization

Histogram of an image is useful in improving the image. Image histogram plays a role in image enhancement. The image histogram in image processing is the method by which occurrences of all the intensity values are displayed in the image, histogram equalization being a powerful yet simple contrast enhancement method (Noureddine et al., 2024).

CLAHE is one such amazing advancement in the arena of image enhancement techniques that effectively overcomes the drawbacks of conventional histogram equalization techniques by emphasizing local contrast while simultaneously suppressing the excessive boosting of noise. The method has seen extensive use across many domains, specifically medical imaging and machine learning, thereby demonstrating its versatility and efficiency (Swadi et al., 2024).

CLAHE is a technique used in the field of image processing to enhance contrast. This method improves the quality of images overall, making it easier for models like GANs to learn more relevant features, thus greatly increasing classification accuracy.

2.7. Generative Adversarial Network

A GAN is a type of machine learning framework that draws inspiration from noise contrastive estimation and the loss function used in modern GANs, as introduced by Ian Goodfellow and his colleagues (Aggarwal et al., 2021). The practical application of GANs began in 2017 with the generation of human faces, aiming to enhance image quality and produce better representations at higher intensities. The concept behind adversarial networks was initially inspired by a blog post written by Olli Niemitalo in 2010, and the idea is now known as Conditional GAN.

The working functionality of GANs involves three basic principles: First, the generative model generates data with the help of probabilistic representations; second, model training happens within adversarial settings; And Third, Deep learning neural networks and AI algorithms have to be utilized for training the whole system. Though GAN was originally designed for unsupervised ML tasks, its working principle is found quite satisfactory in semi-supervised learning and reinforcement learning. All these factors combined make GANs versatile solutions in many fields of industries, such as health, engineering, and banking (Aggarwal et al., 2021).

Generative adversarial networks were recently been introduced as a novel method for training generative models. Those networks consist of two rival models: a generative model G, that estimates the hidden data distribution, and a discriminative model D, that estimates the probability that a given sample is coming from the train data rather than from G. Both G and D can be realized as nonlinear mapping functions, for instance,a multi-layer perceptron (Mirza & Osindero, 2014).

To learn a generator distribution p_g over data x, the generator constructs up a mapping function from a prior noise distribution p_z (z) to data space as G (z; θg). And the discriminator, D(x; θd), is outputting a single scalar representing the probability that x came from train data instead of p_g. G and D, are simultaneously trained: we adapt the parameters of G such that to minimize log(1 −D(G(z))), while simultaneously adapting the parameters of D such that to minimize logD(X). It can be seen of as a two player min-max game with value function V (G, D):

$$\min_G \max_D V(D,G) = \mathbb{E}_{x,y\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z,y\sim p_z}[\log\left(1 - D\left(G(z)\right)\right)]$$

Equation 2.2: GAN Equation

Where,

- $D(x)$ is the probability that x is real.
- $G(z)$ aims to generate data that fools D.

2.7.1. Conditional Generative Adversarial Nets

Generative adversarial networks can be extended to a conditional model by conditioning the discriminator and the generator on additional information, denoted as y. The additional information can include elements like the class label or information from another modality.

Conditioning is achieved by adding y to the discriminator and the generator as another input layer. In the generator, the prior noise $p_z(z)$ and y are mapped concurrently to a common hidden representation. The structure of adversarial training consists of giving large amounts of flexibility to that representation. The inputs x and y are sent to a discriminative function for the discriminator to enable the model to classify real and synthetic samples conditioned only (again modeled by a MLP in this case) (Mirza & Osindero, 2014). The objective function of a two-player mini-max game would be the following:-

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x,y\sim p_{\text{data}}}[\log D(x,y)] + \mathbb{E}_{z,y\sim p_z}[\log\left(1 - D(G(z,y),y)\right)$$

Equation 2.3: cGAN equation

Where:

- x is the real data and y is the conditional information (e.g., a label).
- z is the random noise vector sampled from the latent space.
- G(z, y) is the generator's output conditioned on y.
- D(x, y) is the discriminator's estimate of the probability that xx is real, conditioned on y.
- The generator G tries to minimize log(1−D(G(z,y),y)), and the discriminator D tries to maximize its ability to distinguish real data (x, y) from fake data (G(z, y), y).

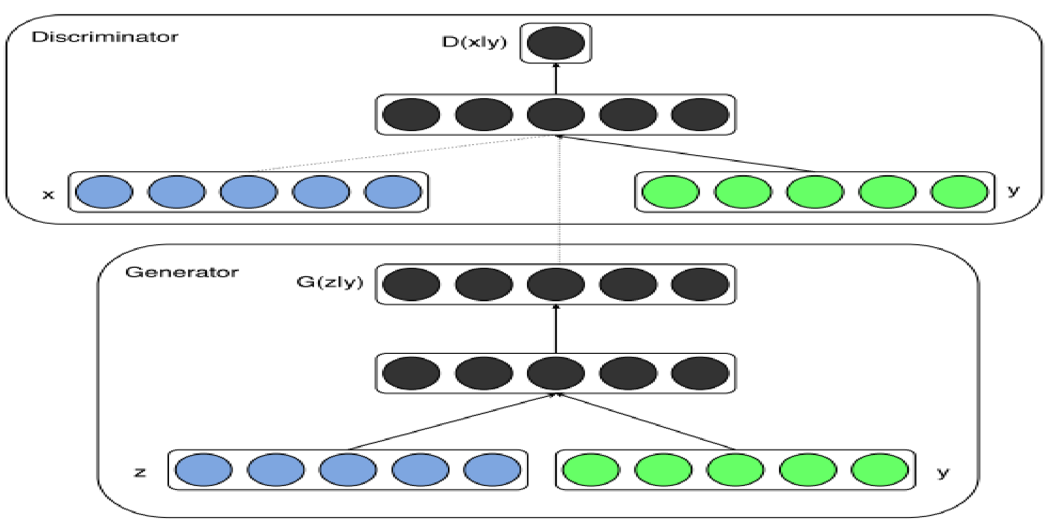

Figure 2.1: Conditional Generative Adversarial Nets

(Mirza & Osindero, 2014)

2.8. EfficientNet

EfficientNet is a CNN architecture designed to scale network depth, width, and resolution uniformly using a compound scaling method. Unlike traditional approaches that scale these factors independently, EfficientNet employs fixed scaling coefficients to adjust all dimensions proportionally (Tan & Le, 2019).

This compound scaling approach is based on the principle that larger input images require deeper networks to expand the receptive field and wider networks to capture detailed patterns more effectively. The foundation of EfficientNet-B0 includes inverted bottleneck residual blocks from MobileNetV2, enhanced with squeeze-and-excitation blocks to improve feature representation.

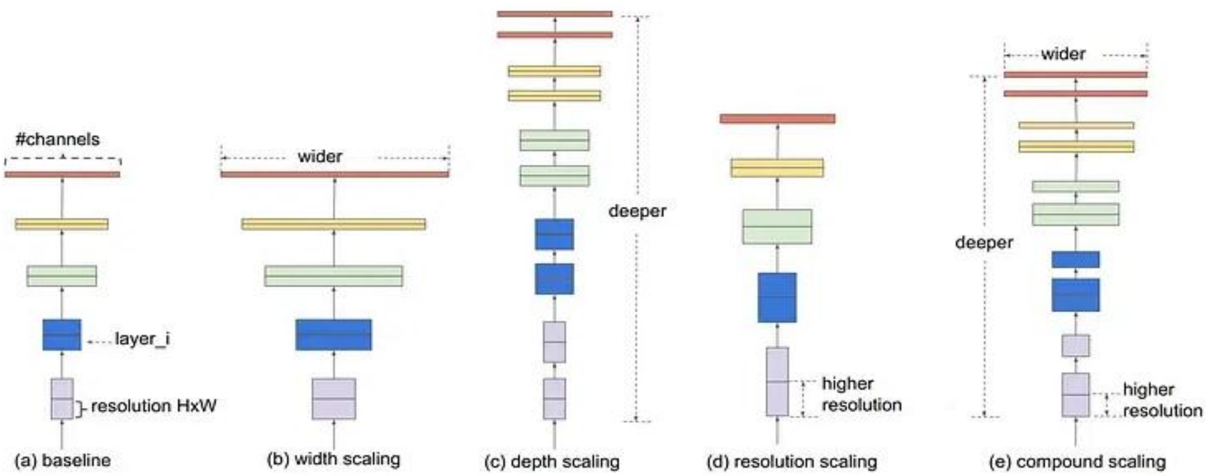

Figure 2.2: Different scaling methods and compound scaling
(Tan & Le, 2019)

2.9. Attention Mechanism

Attention is one of the most important cognitive functions of humans. The critical point in human perception is that we do not process all available information simultaneously. Instead, humans focus on specific parts of the information when needed and discard other perceivable details. Such a selection allows an individual to select high-value information out of huge data using limited cognitive resources. Attention can enhance the efficiency and accuracy of the processing of perceptual information. It has become a very fundamental notion in deep learning, inspired by human biological systems that focus on relevant details when handling large volumes of information. Attention mechanisms have become quite widespread in recent times, with the development of deep neural networks across various fields. Beyond improving performance, attention mechanisms also serve as a tool to interpret the often complex behavior of neural architectures (Niu et al., 2021).

The attention mechanism assigns varying weights to different input elements, allowing the model to concentrate more precisely on crucial information and thereby enhance its performance (Y. Wang et al., 2024). In addition to identifying focal points, attention also enriches the representation of relevant features (Mirza & Osindero, 2014; Woo et al., 2018).
The attention mechanism helps improve the manner in which the model learns salient features by enabling the model to concentrate on the salient parts of the feature map and ignore the less important ones.

2.9.1. Squeeze and Excitation Blocks

A Squeeze and Excitation block is a computational unit, build on top of the a transformation F_{tr} that maps an input $X \in R^{H' \times W' \times C'}$ to feature maps $U \in R^{H \times W \times C}$ (Hu et al., 2017) .

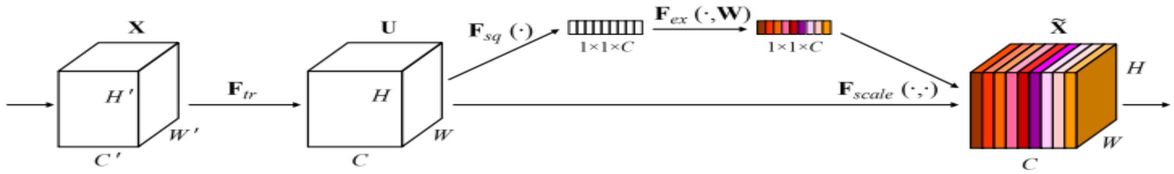

Figure 2.3: A Squeeze-and-Excitation block

(Hu et al., 2017)

2.9.2. Channel Attention

The channel attention mechanism dynamically reweights the features in each channel based on their importance, enabling the model to focus on the features that are most applicable to the given application (Y. Wang et al., 2024).

Channel attention is a technique employed in deep CNNs to focus on the most relevant channels in a feature map. This enhances network performance by reducing the processing of noise and irrelevant information (Hu et al., 2018). It allows the model to prioritize on more important features of face like shape and texture of eye, mouth and nose, resulting in better preservation of these features in the emotional facial thermal image.

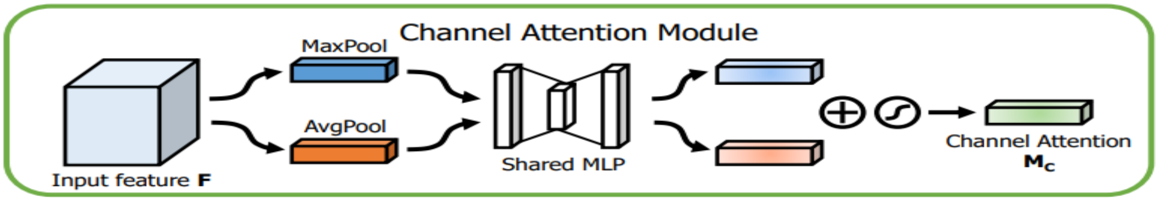

Figure 2. 4: Channel attention

(S. Woo et al., 2018)

2.9.3. Spatial Attention

The spatial attention mechanism assigns different weights to information from different locations, allowing the model to focus on areas that are most relevant for the task (Y. Wang et al., 2024).Spatial attention is a method used in deep CNNs to pay attention to the most important areas in a feature map. Instead of focusing on particular channels, spatial attention finds out "where" the most important features are in the image (S. , P. J. , L. J.-Y. , & K. I. S. Woo, 2018). This mechanism enables the model to emphasize areas like the eye, mouth, and forehead facial regions that are crucial for emotional cues while inhibiting less informative background areas. Within thermal facial emotion recognition, spatial attention serves to maintain the localized thermal patterns for emotional expressions, thus enhancing the model's concentration and classification precision.

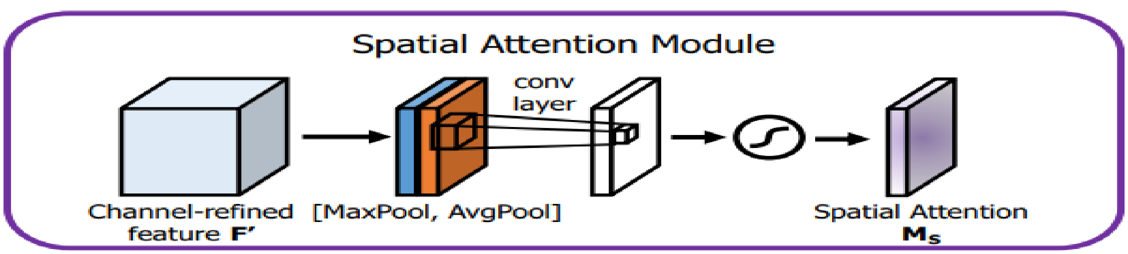

Figure 2. 5: Spatial attention

(S. Woo et al., 2018)

The mixed attention mechanism combines both channel and spatial information to capture multi-scale features, and improve the model's representation and understanding of complex patterns (Y. Wang et al., 2024).

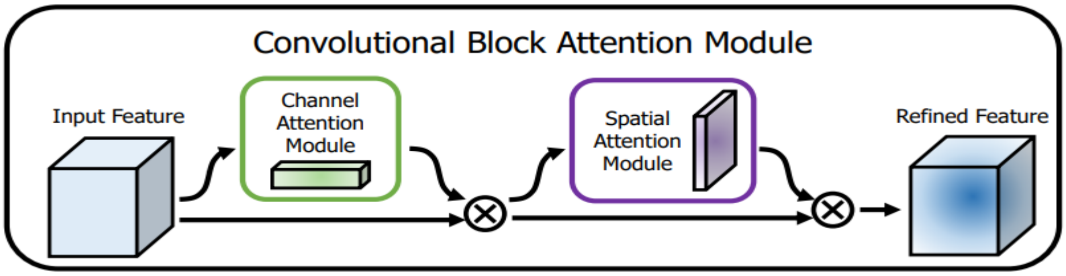

Figure 2.6 : Convolutional block attention module

(S. Woo et al., 2018)

2.10. Residual Neural Network

A residual neural network is a neural network architecture in which layers learn residual mappings with reference to the layer's input. The main concept is the utilization of skip connections, or shortcuts, which skip one or more layers. These are useful in mitigating the problem of vanishing gradients, allowing for easier flow of gradients through the back-propagation algorithm and thereby making it possible to train networks with hundreds or even thousands of layers (He, 2016).

The core idea of ResNet lies in its residual learning formulation. Instead of directly learning the desired mapping$\mathrm{H(x)}$, ResNet learns the residual function $\mathrm{F(x) = H(x) - x}$, so the original function becomes:

$$H(x) = F(x) + x$$

Equation 2.4: Residual Function

Where:

- x: the input to the layer (or block)
- $F(x)$: the residual function (typically a few convolutional layers)
- $H(x)$: the output of the block

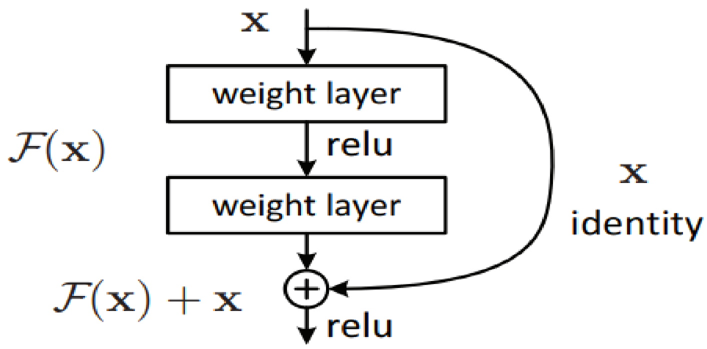

Figure 2. 7: Skip Connection (He, 2016)

2.11. Related Works

Assiri & Hossain (2023) Propose Face emotion recognition based on infrared thermal imagery by the use of machine learning and parallelism. It divides a captured frame into four regions to detect Action Regions. They detect the central point by detecting the tip of the nose, which is further used to differentiate other facial regions. A complex posture estimation method is implemented as facial postures have a great impact on expressions. However, head postures may bring variations in results.Unlike their approach that suffered from pose variations, our approach targets preprocessing and attention mechanisms for improving robustness without needing complicated posture estimation

Chirag Kumar Kyal, (2021) it addresses FER in thermal imaging, focusing on face detection, landmark localization, feature extraction, and parallel deep emotion networks. This study underlines that areas like the forehead, cheeks, nose, lips, and chin are especially relevant for correct emotion recognition in thermal infrared images. However, it does not target any specific area in detail, and the dataset assumes relatively stable head positions. Our approach targets preprocessing and attention mechanisms for improving feature extractions and alleviate small-dataset problems through the use of cGAN for training data expansion.

Aiswarya K. Prabhakaran (2021) Facial expression recognition using thermal images is discussed, and a method is proposed using a pre-trained, modified version of ResNet152. The model relies on a small dataset, and no data augmentation techniques were applied. We overcome this limitation by adding cGAN-based synthetic data augmentation and stochastic augmentation, which increase data diversity and decrease overfitting.

Elbarawy et al. (2019) Facial expressions Recognition in Thermal Images based on Deep Learning Techniques. It takes 90 images with varying rotations, as well as occlusion due to glasses and pose; 60 for training and 30 for testing. Only the poses with less than 45° rotation were selected. The subject has only three different expressions Surprise, Happy, and Angry. We alleviate small-dataset problems through the use of cGAN for training data expansion with five class emotion recognition.

Prasad & Chandana (2023) present a four-stage approach to recognizing human faces in thermal images: pre-processing, feature extraction, detection, and classification. In the pre-processing stage, input thermal face images are cropped by the Difference of the Gaussian filter and normalized by a median filter to handle variations in lighting conditions. Then, facial features position, shape, and object presence will be extracted using the EfficientNet approach. After that, face detection is performed using YOLOv4, followed by the classification of emotions by DenseNet from the thermal images. However, it lacks of preserving fine facial features, relies upon a small subjects and lacking augmentation. We extend this approach by employing Bilateral Filter, Gaussian Blur and CLAHE and integrate EfficientNet-B5 with CBAM for enhanced feature extraction, and cGAN for set expansion for dealing with scalability.

Chaitanya (2020) ultimately, it is used for predicting human emotions using YOLO methodology and a Darknet framework. Background elimination, pre-processing, a training phase, and a test phase are included in the work. The model is good at detecting happiness, sadness, surprise, and disgust. Anger, fear, and neutral are some of those aspects where the model had less accurate precision scores. By incorporating EfficientNet-B5 with CBAM, our approach improves discriminative feature extraction and aids in improving the emotion recognitions.

2.11.1. Summary of Related Work

Table 2.1: Summary of Related Work

Related Papers	**Author**	**Methodology**	**Dataset Used**	**Accuracy**	**Limitation**	**Proposed Solution**	**Problem Solved**
Face Emotion Recognition Based on Infrared Thermal Imagery by Applying Machine Learning and Parallelism	(Assiri & Hossain, 2023)	Image Segmentation, Image Registration and Preprocessing**,** Feature Extraction and Classification using Convolutional Neural Network, a machine learning technique and parallelism.	CK+ dataset, JAFFE databases	96.87 %, 91.8%	Only four active regions are selected and variations in head poses introduce inconsistencies.	Use EfficientNet-B5 with CBAM for robust region-focused feature extraction	Reduces dependency on explicit region segmentation and head pose sensitivity
Human Emotion Recognition from Spontaneous Thermal Image Sequence Using GPU Accelerated	(Chirag Kumar Kyal, 2021)	Thermal domain based on face detection, facial landmark localization, feature extraction, and parallel deep emotion network	USTC-NVIE dataset	94%	It does not target any specific area in detail and the datasets assumes relatively stable head positions.	Use EfficientNet-B5 with CBAM for robust region-focused feature extraction	Reduces dependency on explicit region segmentation and head pose sensitivity

Emotion Landmark Localization and Parallel Deep Emotion Net							
Human Emotions Recognition from Thermal Images Using YOLO Algorithm	(Chaitanya, 2020)	Background elimination, pre-processing, training phase, and test phase using Yolo Algorithm	NVIE Database	65 %	It shows lower precision scores for emotions like anger, fear, and neutral.	Use Bilateral Filter, Gaussian Blur and CLAHE for preprocessing, use cGAN generated data, combine EfficientNet-B5 with CBAM for feature extraction	Enhanced image quality, improves robustness and emotion recognition accuracy
Thermal Facial Expression Recognition Using Modified ResNet152	(Aiswarya K. Prabhakaran, 2021)	Data Preprocessing**,** Training, and Testing using a modified ResNet152 model.	NVIE Database	87.3%	The model relies on a small datasets and no data augmentation techniques were applied.	Incorporate augmentation and cGAN-based synthetic data generation	Mitigate datasets limitations

Facial Expressions Recognition in Thermal Images based on Deep Learning Techniques	(Elbarawy et al., 2019)	Facial Expressions Recognition in Thermal Images Based on Deep Learning Techniques	IRIS dataset	96.7%	The system utilizes 90 images, only poses with rotations of less than 45° were included and the subject displays three expressions: surprise, happiness, and anger.	Incorporate data augmentation and cGAN-based synthetic data generation	Mitigate datasets limitations, improves robustness and emotion coverage
Human Face Emotions Recognition from Thermal Images Using DenseNet	(Prasad & Chandana, 2023)	Preprocessing, feature extraction, detection, and classification of the emotions on faces in thermal images, using the DenseNet technique.	RGB-D-T dataset	95.97%	Only 51 subject are included, use DOG and median filter for preprocessing, and absence of GAN-based or other augmentation methods	Use Bilateral Filter, Gaussian Blur and CLAHE for preprocessing, use cGAN generated data, combine EfficientNet-B5 with CBAM for feature extraction	Enhanced image quality, preserving fine facial feature, improves datasets diversity and enhances discriminative feature extraction

Previous studies, such as those by Assiri & Hossain (2023) and Chirag Kumar Kyal (2021), achieved high precision; however, their requirements for specific face areas and static head positions. Likewise, Prabhakaran (2021) and Elbarawy et al. (2019) illustrated the capability of deep learning employing the use of ResNet and other models; however, these studies were limited by the use of highly limited datasets as well as the lack of data augmentation. Further advanced methods, such as those by Prasad & Chandana (2023), employed the use of EfficientNet with DenseNet classifiers; however, their requirements for limited subjects, lack of preserving fine facial feature, lack of enhanced image quality and lack of GAN-based augmentation limited them for scaling.

Differing from these deficiencies, this paper makes up for them by employing cGAN-based data generation for mitigating the scarcity of datasets, integrating EfficientNet-B5 and Convolutional Block Attention Module for enhancing feature learning, and a ResNet-18 classifier for effective classification.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Overview

This part describes the methods, datasets, software tools, and hardware components used to perform the study. To accomplish the study objectives, the following procedures are carried out:

Domain Knowledge: The initial step involved identifying the specific domain of interest. Through the segregation of relevant areas, a broader knowledge can be obtained to choose the most appropriate domain to proceed with this study.

Literature Review: In this section, numerous publications related to the field of studies are chosen. Those are articles, research journals, or conference papers, all of which are aimed at laying a quite comprehensive ground for existing literature within the field.

Gap identification: We conducted an elaborate review of relevant literature to come up with excellent gaps and specific issues that need attention and further investigation.
Formulating the Proposed Solution: - In view of the shortcomings pointed out, we constructed a solution designed to meet the issues revealed during the literature review.

Implementing the Proposed Solution: - This stage involved the practical implementation of the proposed solution, including model training and testing to ensure the solution works effectively.
The last step states the proposed outcome, followed by an overall evaluation of it in terms of effectiveness and accuracy, among other pertinent measures, to determine overall success.

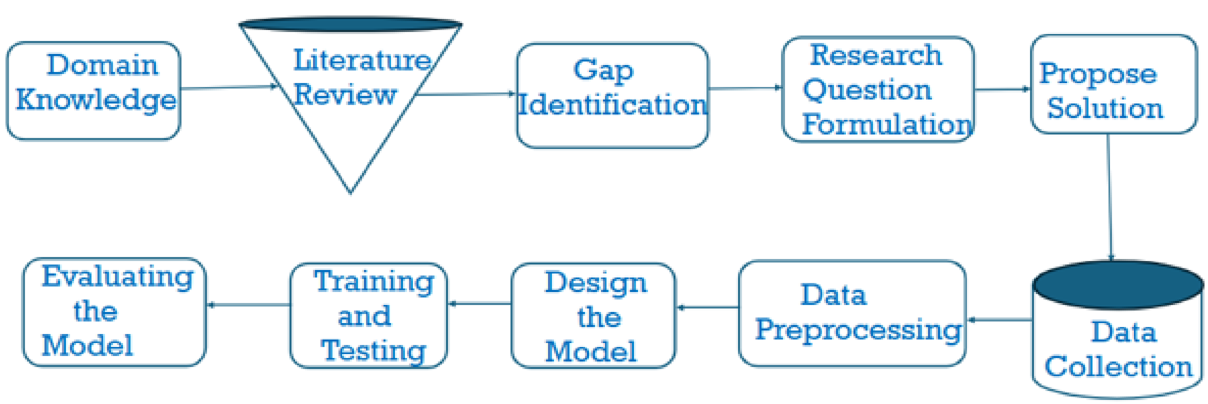

Figure 3.1: Overall process of the study

3.2. Data Collection

This study employed deep learning techniques to improve human face emotion recognition in thermal images, utilizing cGANs and EfficientNet integrated with attention mechanisms. Deep learning models rely heavily on data for training and learning, making data a critical element in the research process. In this study, we utilized the Comprehensive Facial Thermal dataset (Mohamed Fawzi, 2024) for both training and testing. The Comprehensive Facial Thermal Dataset, which was introduced by Mohamed Fawzi in the year 2024, was specifically prepared for research purposes. The dataset was first deposited onto arXiv with the title "Exploring Thermography Technology: A Comprehensive Facial Dataset for Face Detection, Recognition, and Emotion" (Fawzi, 2024). The datasets is publicly accessible on Mendeley Data with a CC BY 4.0 license such that unrestricted academic and research use may be made with the inclusion of proper citation.The authors explicitly state that the primary aim of CFTD is to advance deep learning applications in thermal-based facial analysis.

3.2.1. Comprehensive Facial Thermal Datasets

The datasets was designed with three core protocols for facilitating research into face recognition, face detection, and emotion estimation from thermal images. The data set was collected by a UNI-T UTi165A thermal camera. The whole set of the datasets includes 6,823 images divided into three independent sets: 2,054 images for subjects identification tasks, 2,284 images for face detection, and 2,485 images for emotion recognition.

The datasets for emotion recognition includes images of five different states of emotion, specified by the following distribution: Anger, Happiness, Neutrality, Sadness, and Surprise. The images were taken for a vast number of occasions, utilizing a large number of color schemes, views, and levels of zooming. The images all contain a resolution of 19,200 pixels as well as a temperatures range from −10 °C to 400 °C (Fawzi, 2024).

Table 3.1: Dataset Distribution Per Emotional Class

Emotion	Count	Percentage
Angry	509	20.47%
Happy	538	21.64%

Natural	541	21.76%
Sad	457	18.39%
Surprise	440	17.72%

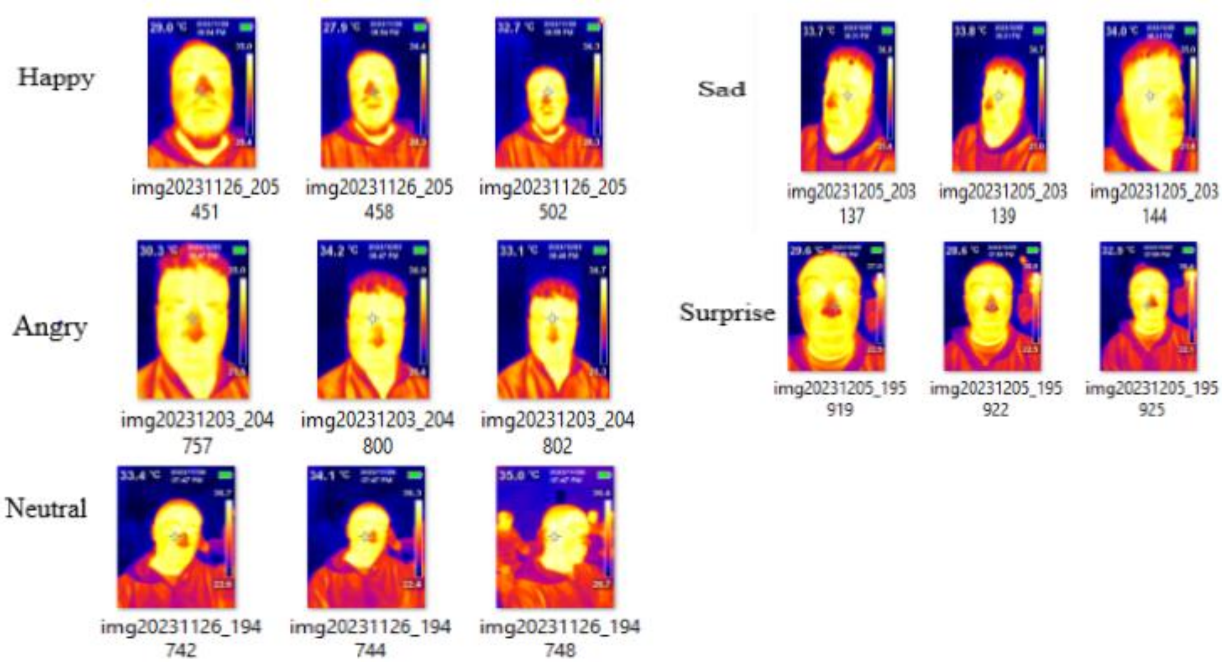

Figure 3.2: Dataset samples

3.3. Data Preprocessing and Augmentation

Thermal images typically suffer from lower contrast and are more susceptible to noise. During preprocessing, the collected images undergo various transformations to prepare them for analysis. These steps include bilateral filtering which helps enhance contrast and reduce noise, ,Gaussian Blur to Smooth’s the image, CLAHE for enhancing contrast in low-light conditions, normalizing the images to standardize intensity levels, resizing them to ensure uniform dimensions, and applying augmentation techniques to mitigate data imbalance. These preprocessing methods are crucial for standardizing the input data and optimizing the model's ability to learn effectively.

We employed data augmentation to expand the training datasets by augmenting thermal images featuring a variety of facial expressions. This approach helps in increasing the diversity of the dataset, addressing potential data limitations and enhancing the model's ability to generalize across different emotional states and conditions.

3.3.1. Data Pre-processing and Augmentation Techniques

The data preprocessing steps ensure thermal image preparation for deep learning. The preprocessing steps are necessary to improve model performance by providing clean, consistent, and diverse input training data.

Thermal images are loaded from subfolders within the dataset folder. The function iterates through all subfolders to find images with supported formats like .bmp, .png, .jpg, and .jpeg. Each image's label is derived from the name of the parent folder, separating the images based on their respective emotion or class.

Bilateral Filtering is nonlinear, noise-reducing, and edge-preserving smoothing filter is applied in images. Enhances the quality of thermal images by removing noise without destroying important edges and details.

$$I'_p = \frac{1}{W_p} \sum_{q \in s} I_q \cdot exp\left(-\frac{\| p - q \|^2}{2\sigma_s^2}\right) \cdot exp\left(-\frac{\left(I_p - I_q\right)^2}{2\sigma_r^2}\right)$$

Equation 3.1: Bilateral Filtering

$$W_p = \sum_{q \in s} exp\left(-\frac{\| p - q \|^2}{2\sigma_s^2}\right) \cdot exp\left(-\frac{\left(I_p - I_q\right)^2}{2\sigma_r^2}\right)$$

Equation 3.2: Normalization Factor

Where:

- I'_p: filtered intensity at pixel p
- I_q: intensity at neighboring pixel q
- $\| p - q \|$: Euclidean distance between pixels (spatial distance)
- $I_p - I_q$: intensity difference (photometric distance)
- σ_s: spatial standard deviation controls how much nearby pixels influence
- σ_r: range (intensity) standard deviation controls how much pixels with similar intensities influence
- W_p: normalization factor to ensure values remain in range

Normalization is the transformation of data into a common and comparable format. It can usually be easily done by either scaling the data to a certain interval or sharing by some predefined statistical property. These are some of the techniques that increase model performance, algorithm optimization, and reduction of training time. The objective is to normalize the pixel values of images to ensure consistent input to the model. Standardize pixel intensity values for consistent representation. Uses cv2.normalize to scale image intensity values to a desired range from 0 to 255. Provides consistency and ease for the model's feature extraction process.

$$I_{normalized} = \frac{(I - I_{min})}{(I_{max} - I_{min})} x \, (range_{max} - range_{min}) + range_{min}$$

Equation 3.3: Image Normalization

Where:

- I = original intensity value of a pixel
- Imin, Imax = minimum and maximum pixel intensities in the image
- $range_{min,}$ $range_{max}$ = desired intensity range (e.g., 0 to 255 or 0 to 1)

Image resizing is an essential step in pre-processing. Image resizing is used to transform the size of the input images to the model's desired input size. Because deep learning needs a huge number of examples, the model does extra computations to accommodate larger images. By reducing the images to a smaller scale, the computational need can be significantly lessened, thereby accelerating the training process. To facilitate easy compatibility with deep learning models such as EfficientNet-B5 and ResNet, which require fixed input sizes all images are resized to a target size of 224x224 pixels.

The objective to transform images into a standard resolution (e.g., 224×224) while keeping the aspect ratio or, if required, introducing distortion.

Formula for Pixel Mapping in Resizing (Using Nearest Neighbor):

$$\mathrm{I}'(x', y') = \mathrm{I}(x, y)\,, \quad x = \frac{x'}{\mathrm{scale}_x}, y = \frac{y'}{\mathrm{scale}_y}\,, \mathrm{scale}_x = \frac{\text{original width}}{\text{new width}}\,, \mathrm{scale}_y = \frac{\text{original height}}{\text{new height}}$$

Equation 3.4: Image Resizing

Where:

- $I'(x',y')$ is the pixel value at the new image at position (x',y')
- $I(x,y)$ is the pixel value in the original image at position (x,y)
- x' and y' are coordinates in the output (resized) image
- x and y are coordinates in the input (original) image

Data augmentation strategies can make a training set bigger and more diverse. The enhanced set can be applied to train model which can further make model stronger and adaptable to diverse environments. This act helps to avoid the over fitting. We expanded the training data set through a set of procedures to increase the diversity it provides'. The task of enrichment was performed to enhance the generalization ability of the model and to achieve better performance. In this work, we utilized various data augmentations to improve the diversity and robustness of our datasets. Brightness and contrast calibration has been applied by changing the brightness between -10 and 10 and the contrast between 0.8 and 1.2. Images were scaled between 0.8 and 1.2 to mimic object size changes. Gaussian blur using a5×5kernel was applied to simulate defocusing effects. Moreover, the images were randomly rotated up to 15 degrees, to introduce orientation variability and improve model generalization.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Equation 3.5: Image Rotation

Where:

- (x, y) are the coordinates of the original pixel in the image.
- (x', y') are the coordinates of the rotated pixel.
- θ is the rotation angle in radians.

Shifts**:** Random shifts in width and height by up to 10%.

$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$, Where, T_x, T_y are translation offsets in the x and y axes.

Equation 3.6: Random shifts

Zoom**:** Applying random scaling in the range of 10%.

$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$, Where, S_x, S_y are scaling factors along the x and y axes.

Equation 3.7: Random Scaling

Flipping**:** Flipping horizontally to increase variety.

x'=W-x-1, y'=y Where W is the width of the images.

Equation 3.8 : Horizontal Flipping

For enhancing the training data set and for making the model stronger towards variations, the random transformations are carried out with the help of an Image Data Generator. Also, every image is modified to include a channel dimension, leading to an arrangement of (224, 224, 1). This arrangement helps in ensuring that augmentation mechanisms can be effectively implemented on grayscale images.

For sample visualization, raw thermal images are displayed using a heatmap-style color map ("gray") to simulate the appearance of thermal imaging. To validate the applied transformations, one augmented sample is also shown. In each processed image, the "gray" color map is used to represent pixel intensities in terms of thermal gradients. This mapping effectively translates intensity values into RGB components based on the selected color map, allowing for a clearer visual interpretation of temperature variations across the thermal images.

3.4. Synthetic Image Data Generation

3.4.1. Generator Model

The generator creates synthetic thermal images from random noise and a given label. This is a neural network trained to "fool" the discriminator. The generator model begins with input layers that accept two types of inputs: a class label representing the emotion category such as happy, angry, sad, neutral, or surprise and a random vector drawn from a latent space, serving as noise input. To incorporate label information effectively, a label embedding layer transforms the categorical label into a dense representation, which is then reshaped to align with the spatial dimensions of the image, enabling the generator to condition the output on specific emotion categories. Simultaneously, the noise vector undergoes transformation through dense layers and is reshaped into an initial 3D tensor (height × width × channels), forming the base of the generated image. These two components the embedded label and processed noise are then concatenated, allowing the generator to access both random variations and class-specific features. The resulting tensor is passed through a series of upsampling layers using transposed convolutions (Conv2DTranspose) that progressively increase the spatial dimensions of the image. Each layer is followed by batch normalization and LeakyReLU activation to stabilize training and ensure smooth gradient flow. Finally, the output layer generates the thermal facial image with dimensions matching those in the real datasets, using a Tanh activation function to constrain pixel values within the range of [-1, 1].

3.4.2. Discriminator Model

The discriminator distinguishes between real and fake images, while also ensuring that fake images are conditionally accurate based on the given label. The discriminator model begins with input layers that accept two inputs: a class label and an image (which can be either real or generated). The label embedding layer transforms the class label into a dense vector and reshapes it to match the spatial dimensions of the input image, ensuring compatibility for further processing. Next, the image and label embedding are concatenated along the channel dimension, allowing the model to evaluate whether the provided image aligns with the corresponding emotion label. This combined input is passed through a series of convolutional layers that extract hierarchical features while progressively reducing the spatial dimensions. Each convolutional block uses LeakyReLU activation to introduce non-linearity and Batch Normalization to stabilize the learning process by normalizing the feature distributions. After the convolutional feature maps are extracted, they are flattened, and a final dense layer with a sigmoid activation function outputs a single value representing the probability that the input image is real (1) or fake (0), based on how well it corresponds to the given class label.

3.4.3. cGAN Model

The cGAN combines the generator and discriminator into one system to train them together. In the training process of the conditional GAN, the generator is responsible for creating fake images by combining random noise vectors with class labels, such as different facial emotions. These synthetic images are then passed to the discriminator, which evaluates them and produces a prediction indicating whether the image is real or fake. During the generator's training phase, the discriminator’s weights are frozen, meaning it is not trainable. This strategy ensures that only the generator is updated during this step, encouraging it to produce more realistic and label-consistent images that can successfully fool the discriminator. This adversarial setup gradually improves both components, leading to higher quality synthetic data generation.

3.4.4. Training the cGAN

This section trains the generator and discriminator iteratively. During training, the model follows a structured process involving both real and generated data. First, real samples are selected by taking a batch of real thermal images along with their corresponding emotion labels. These are fed into the discriminator, which is trained to classify them correctly as real (label = 1). Then, the generator creates a batch of fake images using random noise vectors and class labels. These generated images are also passed to the discriminator, which is trained to identify them as fake (label = 0). The discriminator loss is calculated as the average of the loss incurred on both real and fake samples, guiding its learning to improve discrimination accuracy. In the next step, the generator is trained to produce images that the discriminator mistakenly classifies as real (label = 1), thereby updating its weights to enhance the realism and class consistency of its outputs. This training cycle repeats across multiple epochs, and after each epoch, the performance of both models is monitored using the discriminator loss and generator loss to evaluate progress and model convergence.

3.4.5. Generating New Thermal Images

After training, the generator can be used to create new synthetic thermal images. To generate synthetic thermal images using a trained conditional GAN, the process begins by generating noise, where random vectors are sampled from a normal distribution to serve as inputs for the generator. Next, specific labels corresponding to desired emotion categories such as happy, angry, sad, neutral and surprise are defined, guiding the generator to produce emotion-specific outputs. These noise vectors and labels are then passed to the generator, which creates synthetic thermal images that align with the provided emotional categories. After image generation, post-processing is applied, as the generator typically outputs pixel values in the range [-1, 1]. These values are rescaled to the range [0, 255] to make the images suitable for visualization or saving in standard formats.

3.5. Feature Extraction

Feature extraction plays a pivotal role in the model development process, utilizing advanced techniques to identify and extract key features from the preprocessed images. After preprocessing the images, we then extracted the features by using efficientnet-B5 with attention mechanism that is the backbone of any model development. The effectiveness of the feature extraction stage has a direct effect on the capability of the model to correctly identify emotions. Using EfficientNet with attention mechanisms guarantees that the classifier focuses on emotion-specific portions of the facial area in which the emotion is detected.

3.5.1. Description of Feature Extraction Using EfficientNet-B5 with CBAM

The thermal images are first resized to 456 × 456 pixels, the EfficientNet-B5 default input size, and normalized to [0, 1]. The images are then passed through EfficientNet-B5, initialized with ImageNet pretrained weights to leverage dense feature representations. The network is used without its final classification head so that the output is high-dimensional feature maps rather than class probabilities.

In order to improve the quality of the features extracted, the Convolutional Block Attention Module is incorporated in chosen blocks of EfficientNet-B5. CBAM uses both channel attention (emphasizing the most informative feature channels) and spatial attention (highlighting salient spatial locations), thus sharpening the feature maps and improving the capability of the model to delineate subtle temperature changes and structural patterns in facial thermal images.

The refined feature maps are then pooled or flattened into a dense feature vector, which is passed through fully connected classification layers. These layers finally transform the refined feature representations into their respective emotional states.

3.6. Classification

The classification occurs after the features are extracted using EfficientNet-B5 and the attention mechanism, with ResNet-18 being employed for further processing. In this approach, EfficientNet with the attention mechanism first extracts relevant features from the thermal images, highlighting important regions associated with facial emotions. These extracted features are then passed through ResNet-18 classifier, which assigns one of the predefined emotion labels, such as happy, sad, surprise, angry, or neutral, based on the processed features.

3.7. Baseline Works

After reviewing several studies on facial emotion recognition, especially those using thermal images rather than visual light, due to thermal modalities eliminating problems of varying pose and illumination. Among the reviewed works, the paper 'Human Face Emotions Recognition from Thermal Images Using DenseNet'(Prasad & Chandana, 2023) was selected as the baseline for this study. This paper was chosen because it presents a complete deep learning pipeline for thermal facial emotion recognition and achieved strong results despite using a relatively small dataset. Human face emotion recognition from thermal images using DenseNet(Prasad & Chandana, 2023)was selected as a baseline of this study based the method used and the generated results without requiring the data used like small subjects when compared to others. The proposed model by Prasad & Chandana (2023) comprises of data preprocessing using DOG and mean filtering, feature extraction using efficientnet-B0 and DenseNet for classification. To evaluate the effectiveness of our proposed model, we compared its performance against this baseline for human face emotion recognition from thermal images.

3.8. Model Evaluation

Measuring the performance of a machine learning model is an essential component in determining its accuracy and reliability in practical applications. Procedures for model evaluation provide valuable insight into the model's extrapolation ability to generalize to new, unseen data and its capability to identify patterns in the training data. It is important to use empirical evaluation methods that provide valuable information about the strengths, weaknesses, and efficacy of the developed model in the evaluation process. In this research, 80% of the data is used for training, 10% for validation, and 10% for testing. This approach guarantees that the model is learned over the larger portion of the data, validated to tune its parameters, and tested on a separate, unseen datasets to evaluate its performance.

3.9. Evaluation Metrics

The proposed model undergoes thorough evaluation using a range of key metrics to assess its effectiveness and reliability.

3.9.1. Frechet Inception Distance

FID estimates the similarity between the distribution of generated images and the distribution of real images in a feature space (typically using a pre-trained neural network like Inception). Lower FID values correspond to higher similarity, meaning the GAN-generated images closely match the real images.

$$\text{FID}=\| \boldsymbol{\mu}_{\mathbf{r}} - \boldsymbol{\mu}_{\mathbf{g}} \|^{2} + \mathbf{Tr}\left(\boldsymbol{\Sigma}_{\mathbf{real}} + \boldsymbol{\Sigma}_{\mathbf{gen}} - 2\left(\boldsymbol{\Sigma}_{\mathbf{real}}\boldsymbol{\Sigma}_{\mathbf{gen}}\right)^{\mathbf{1/2}}\right)$$

Equation 3.9: Frechet Inception Distance

Where:

- $\mu_{\text{real}}, \Sigma_{\text{real}}$: mean and covariance of real image features
- $\mu_{\text{gen}}, \Sigma_{\text{gen}}$: mean and covariance of generated image features
- Tr: trace of a matrix (sum of diagonal elements)
- $\left(\Sigma_{\text{real}}\Sigma_{\text{gen}}\right)^{1/2}$: matrix square root of the product of covariance

3.9.2. Evaluation Metrics for Classification

Accuracy, F1-Score, Precision, Recall, and the Confusion Matrix are utilized to evaluate the model's performance across each emotion class.

Accuracy is a measure of the general accuracy of the model by calculating the number of accurately predicted instances out of the total predictions. Accuracy measures the proportion of correctly classified instances out of the total datasets.

$$\text{Accuracy}=\frac{TP + TN}{TP + TN + FP + FN}$$

Equation 3.10: Accuracy Performance Measure

Precision, also referred to as positive predictive value, approximates the ratio of accurately predicted positive cases to all positives predicted. It is calculated as follows:

$$\text{Precision}=\frac{\text{True Positives}}{\text{True Positives+False Positives}}$$

Equation 3.11: Precision Performance Measure

Recall, also known as sensitivity or true positive rate, is the ratio of actual positive cases accurately predicted by the model.

$$\text{Recall}=\frac{\text{True Positives}}{\text{True Positives}+\text{False Negatives}}$$

Equation 3.12: Recall Performance Measure

The F1-score is the harmonic mean of recall and precision and therefore presents a single metric that addresses both issues. The F1-score becomes an assessment metric that incorporates both recall and precision, making a general assessment of their performance. It goes as follows:

$$\text{F1 Score}=2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$$

Equation 3.13: F1-score Performance Measure

A confusion matrix is a binary classification 2×2 matrix summarizing the comparison of true and predicted values. It also has an axis of actual results and one of predicted values. True Positive happens when the model accurately detects instances of the positive class, indicating both prediction and real outcome are favorable. True Negative means the model correctly detecting occurrences of the negative class, where both the predicted and the actual outcomes are negative. A false positive results from the model's classification of a negative instance as positive, indicating a predicted-positive but actual-negative scenario. Similarly, False Negative is occurs when a model erroneously assigns a positive example as a negative one, leading to a predicted-negatively but actually-positive state.

3.9.3. Training Accuracy

Training accuracy reflects the model's performance in terms of correct prediction of the outcomes for the same datasets on which it has been trained. This metric is very important to establish how well the model has learned the task of emotion recognition whether it has learned the features and patterns in the training data that will be used for distinguishing the emotions.

3.9.4. Validation Accuracy

The accuracy of validation is the one that estimates the performance on a datasets that has not been used for training. This metric is important since it assesses the model's ability to generalize its learning and correctly predict on new, unseen data. A high validation accuracy means the model can apply its knowledge on real scenarios well instead of just memorizing from the training data.

3.9.5. Training Loss

Training loss is a good insight into the learning process of a model: it tells how much error the model generates while learning from the training datasets. A lower training loss indicates that the model is making fewer errors as it learns. Monitoring training loss is indispensable for understanding how well the model is improving over time and helps identify when adjustments may be needed, whether in the model's architecture, learning rate, or other aspects of the training process, to enhance its performance.

3.9.6. Validation Loss

The loss on validation represents the model's error rate for data it had not seen during the training phase. This metric plays a vital role in identifying how well the model works on new, unseen data and forms a more realistic measure of its potential usefulness in clinical practice. The low validation loss means that the model is generalizing well and performs better on unseen data, which is desirable in its practical application.

3.10. Development Tools

To design and implement the proposed work, a range of software and hardware tools are employed in this study.

3.10.1. Hardware Tools

The following hardware tools are used for the implementation of the research:

Table 3.2: Hardware Tools

SNo.	Tools	Purpose
1	RAM	To speed up the training process and enhance the performance
2	Hard Disk	To store dataset and models
3	GPU	To perform the computations and speed up the training process
4	Flash disk	To store the file

3.10.2. Software Tools

Software tools are essential in assisting different phases of the research process, including data preprocessing, model development, visualization, and deployment. Here are several software tools that we utilized throughout this research:

Anaconda[1]: A free is an open-source distribution of programming languages used for scientific computing, data visualization, data analysis, and machine learning application. The distribution includes a wide variety of pre-installed packages and libraries, including the well-known ones such as NumPy, pandas, SciPy, Matplotlib, and scikit-learn, among many others.

Jupyter Notebook[2]: An open-source web application that is highly appropriate for developing documented workflows that can be exchanged among users. The integration of code, explanatory text, equations and visualizations allows for comprehension and reproducibility of the analysis by other users.

TensorFlow[3]: An open-source machine learning framework that provides a complete set of tools, libraries, and community resources. TensorFlow is designed to help researchers and developers build, train, and deploy machine learning models efficiently.

Keras: The high-level deep learning library toolkits that running on top of TensorFlow; famous for its friendly interface, it contains a wide variety of layers, activation functions, and also tools for model evaluation, data preprocessing, and visualization.

PyTorch[4]: Another open-source deep learning framework, PyTorch is a popular platform with which to build and train neural network models. The most favored language utilized in machine learning, deep learning, and GANs.

Kaggle[5]: is an online community and data science competition platform for data scientists and machine learning professionals. It is a popular web-based machine learning, deep learning, and data science platform. It provides data enthusiasts, researchers, and developers with tools and resources to address real-world issues.

Colab: is an online platform developed specifically for the purpose of writing and executing python code. It dispenses with local installation and thus is particularly well suited for quick startup, as well as offering free GPU access for a short period of time. Such high-performance graphics processing units can greatly enhance the speed of computation in machine learning tasks that are computationally expensive, particularly those involving deep learning models.

OpenCV: This is used in the preprocessing of images, such as resizing, denoising, and augmentation of thermal images to fit input dimensions.

NumPy and Pandas: Very useful for numerical operations, data manipulation, and preprocessing.

Matplotlib/Seaborn: For attention maps visualization, FID/SSIM metrics and training progress.

Scikit-learn: Provides functions to compute metrics such as accuracy, F1-score, precision, and recall for classification tasks.

---

[1] https://anaconda.org/ [2] https://jupyter.org/ [3] https://www.tensorflow.org/

[4] https://pytorch.org/ [5] https://builtin.com/

CHAPTER FOUR

4. PROPOSED ARCHITECTURE

4.1. Chapter Overview

This chapter provides architecture of proposed method for human face emotion recognition in thermal images based on the utilization of preprocessing, data augmentation, synthetic data generation, feature extraction and classification. It includes graphical illustrations and mathematical formulations. The chapter consists of five sections: the first section provides an overview of model selection and model architecture, followed by the explanation of preprocessing steps. The third section is about generator and discriminator design. The fourth segment describes about EfficientNet-B5 with CBAM and the ResNet-18 classifier. The five segments describe about learning functions of the model, and the concluding part explains the training pseudo code.

4.2. Model Selection

Feature extraction is vital in facial emotion recognition since classification precision relies on learning subtle and discriminative patterns from input images.Traditional CNNs have been promising yet usually constrained by high computational cost, lack of scalability, and overfitting susceptibility when datasets are small or highly imbalanced. For these issues, this research utilized EfficientNet for feature extraction by tapping into its compound scaling approach that balances network depth, width, and resolution to attain strong representation using fewer parameters than standard CNNs. Although EfficientNet-B0 offered lightweight efficiency, it had limitations in depth which restricted it from learning fine-grained thermal cues.Conversely, EfficientNet-B5 had deeper and wider architecture, facilitating more complicated feature learning, which was also boosted by incorporating the Convolutional Block Attention Module. Through CBAM, the model was enabled to focus on emotion-related facial regions and suppress background noises for enhanced robustness and recognition performance.For classification purposes, light-weighted and residual-connected ResNet-18 was chosen instead of deeper variants like ResNet-50 or DenseNet due to lightweight design facilitating decreased computational cost, faster processing time, and less overfitting susceptibility.

4.3. Model Architecture

The proposed approach is human face emotion recognition in thermal images based on utilization of preprocessing, data augmentation, synthetic data generation, feature extraction and classification. The proposed model employed preprocessing the input images using bilateral filter for reducing noise while keeping edges sharp, Gaussian Blur to Smooth's the image, CLAHE for enhancing contrast in low-light conditions and data augmentation such as random rotation to rotates image between -10 to 10 degrees, random horizontal flip to simulates variations in face orientation, brightness adjustment to adjusts brightness for different lighting conditions, generating the synthetic images using cGAN ,feature extraction using EfficientNet-B5 with CBAM ,and ResNet-18 for classification.

The purpose of the CBAM is to focus on the most relevant emotional feature of the thermal images and to pay attention to the most important areas in a feature map.

After the features are extracted it passed to the ResNet-18 classifier for classification into emotional classes.

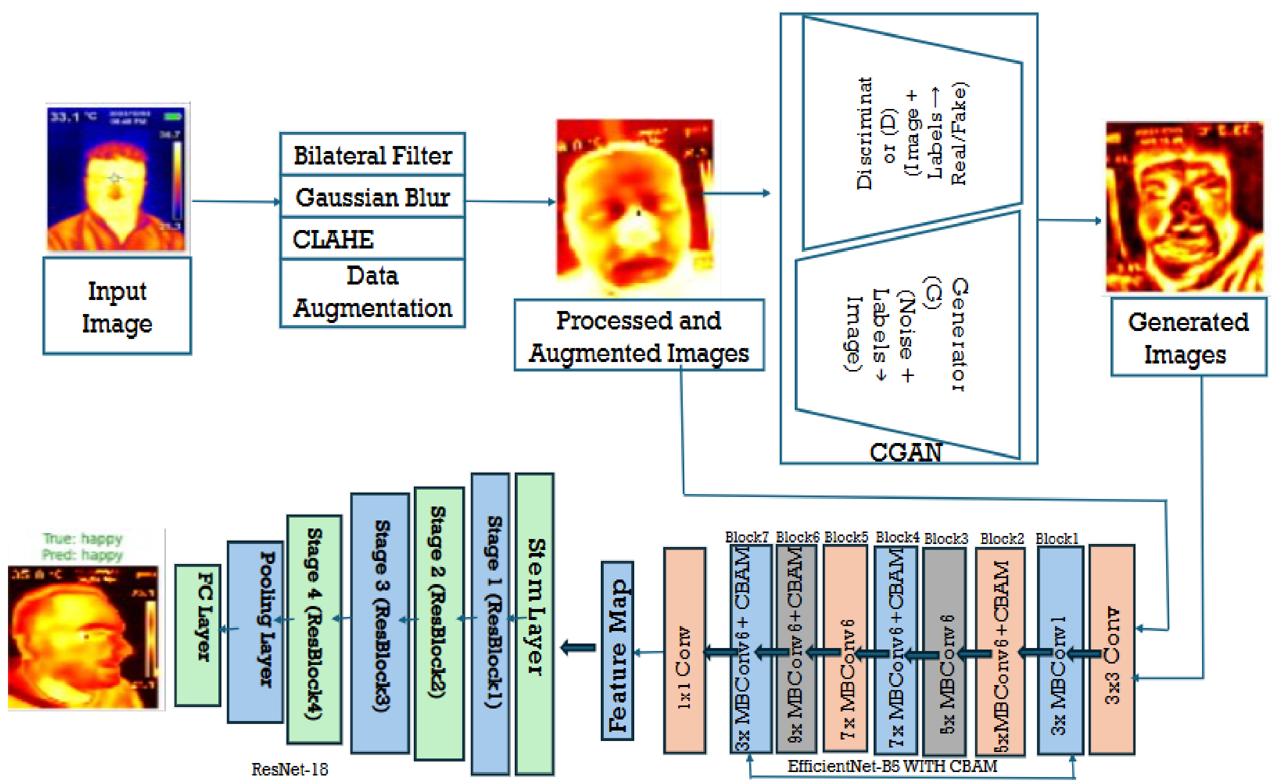

Figure 4.3 : Proposed architecture

4.4. Thermal images preprocessing pipeline

All the thermal images processed by the preprocessing image function go through a sequence of enhancement and transformation procedures. The image is initially taken in grayscale mode to facilitate simple processing and emphasis on intensity distributions. A bilateral filter is then applied to retain edge definition while simultaneously smoothing the image. A Gaussian blur is applied to Smooth's the image. Then, Contrast Limited Adaptive Histogram Equalization is applied to improve the image's contrast, particularly in areas with poor visibility. The image is then sent to the augment image function, where various stochastic transforms like rotation, flipping, and alterations of brightness are implemented. Gaussian noise may also be introduced to replicate imperfections that typically happen in real-world scenarios. Finally, the image is resized to a standardized size of 128 pixels by 128 pixels, thus ensuring uniformity for further processing or model training.

The original thermal images are preprocessed according to the below flowchart.

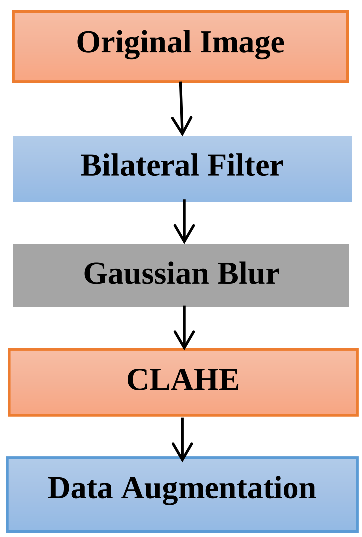

Figure 4.4: Flowchart Illustrating the Images Preprocessing Steps

4.5. Conditional Generator Adversarial Network

4.5.1. Generator Architecture

Generator architecture of the conditional GAN is designed such that high-resolution grayscale thermal images are produced with the use of two kinds of input: a latent vector (z), and respective class labels. First of all, the class labels with their respective emotional categories are processed through an nn.Embedding layer that maps each label into a dense embedding vector (num_classes → embed_dim). Then this embedding undergoes reshaping as per the dimensions of the latent vector and gets concatenated with it and thus allows the generator to incorporate label-specific details during the process of synthesizing images.

The integrated input is subsequently subjected to a succession of five ConvTranspose2D layers, commonly referred to as transposed convolutions. These layers incrementally enhance the spatial dimensions of the data, expanding both the width and height at each stage to achieve the intended final image dimensions. Following each transposed convolution, Batch Normalization is implemented to stabilize the training process, while ReLU activation is employed after every layer except for the final one. The concluding layer utilizes a Tanh activation function, thereby guaranteeing that the output pixel values are confined within the interval of [-1, 1], which is appropriate for representing thermal image intensities.

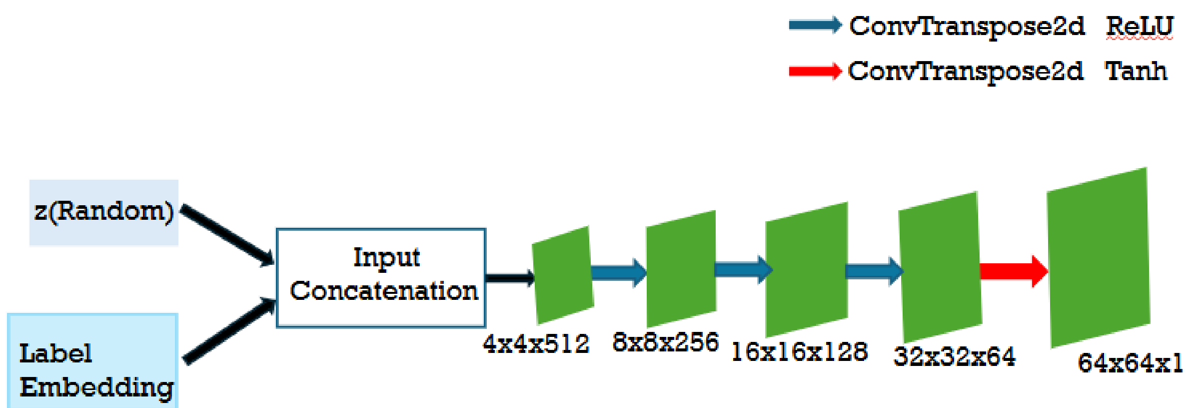

Figure 4.5: Generator architecture

The generator upsamples the noise input to create a 64x64 grayscale thermal image.

4.5.2. Discriminator Architecture

The discriminator within the conditional GAN framework is structured to determine the authenticity of a given image, distinguishing between real and fake instances by considering both the content of the image and its associated class labels. It accepts two forms of input: an image, which may either be real or artificially generated, along with its corresponding class label. Initially, the class label undergoes processing through an embedding layer, resulting in the transformation into a 50-dimensional vector. Subsequently, this embedded vector is reshaped to conform to the dimensions of the input image (for instance, 64×64) and incorporated into the image tensor, thereby enabling the discriminator to evaluate the veracity of the image in relation to its specific label.

The combined input gets processed with a sequence of five Conv2D layers successively decreasing the spatial dimensions of the image while increasing the number of channels to allow extraction of deeper features. A LeakyReLU activation function follows each of the convolutional layers to introduce non-linearity and continue gradient flow except the final layer, which has a sigmoid activation function. The final layer outputs a single probability value indicating the discriminator's opinion of the input image being real (trending towards 1) or false (trending towards 0).

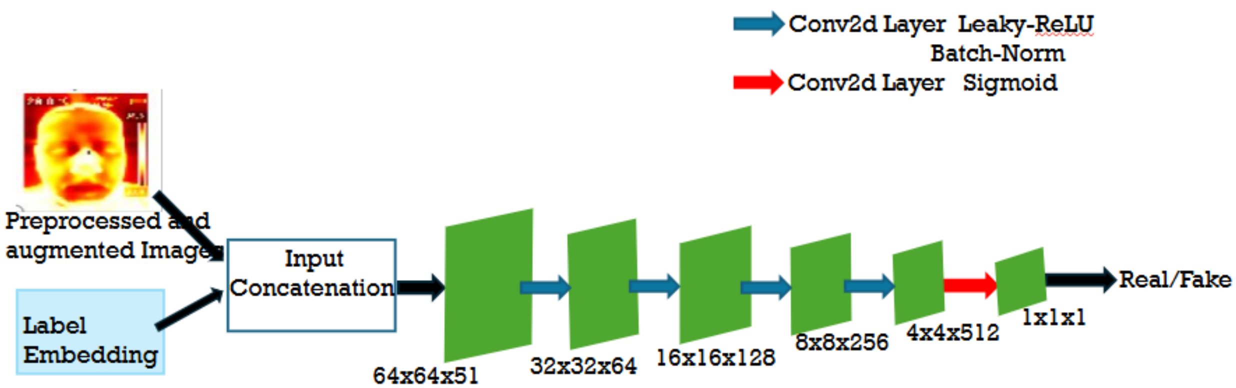

Figure 4.6: Discriminator architecture

The discriminator down samples the image to a single real/fake probability.

4.6. EfficientNet-B5 with CBAM

The architecture utilizes EfficientNet-B5 as the feature extractor backbone, taking advantage of its dense representational capacity. It follows a compound scaling method, which uniformly scales depth, width, and resolution to optimize performance while maintaining efficiency.

Convolutional Layers uses MBConv blocks with depthwise separable convolutions and SE Blocks improves channel-wise feature recalibration.

Swish Activation is a non-linear activation function that improves learning capacity and Adaptive Pooling ensures a flexible input size by reducing spatial dimensions before classification.

To further improve the feature representation learned by EfficientNet-B5, Convolutional Block Attention Modules are used on 2, 4, 6, and 7 block of the network. These attention modules operate on the features both spatially and channel-wise, making the model focus on more informative regions while ignoring less relevant or noisy information. This hybrid model of a strong backbone with attention mechanisms added increases the capability of the model to extract and highlight relevant patterns, particularly helpful in difficult tasks like facial emotion recognition.

The EfficientNet-B5 base model extracts hierarchical features and the classifier is replaced with an identity layer to allow direct feature extraction.

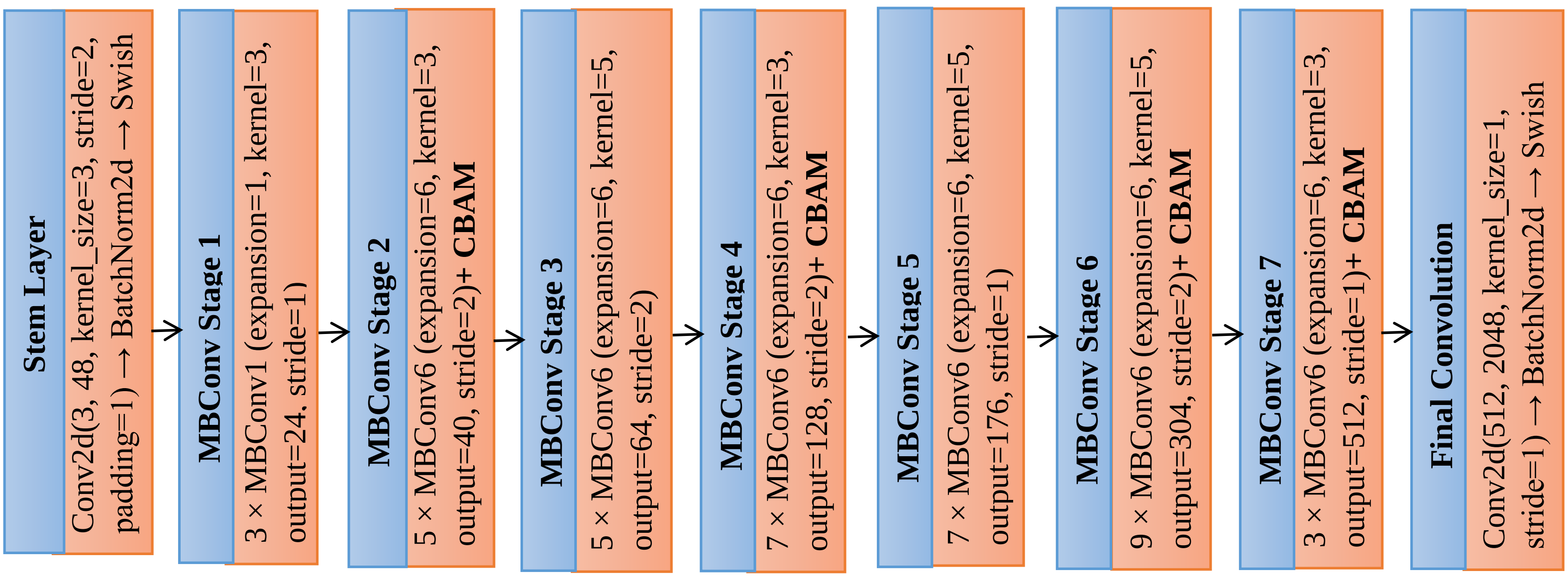

Figure 4.7: EfficientNet B5 with CBAM Architecture

4.7. Attention Mechanisms

Attention mechanisms enhance feature representation by focusing on the most relevant information.

4.7.1. Channel Attention Mechanism

The channel attention mechanism is designed to amplify the most significant feature maps across various channels. It initiates the process by utilizing Global Average Pooling and Global Max Pooling on the input feature map to aggregate the importance of each channel. The resultant pooled outputs are subsequently directed through a common set of Fully Connected layers, beginning with a bottleneck layer that compresses the feature dimensionality from 2048 to 320, followed by an expansion layer that reinstates it to 2048. The output generated by these Fully Connected layers undergoes a sigmoid activation function, which produces channel-specific attention weights. These weights are then multiplied by the original feature map, effectively highlighting the crucial channels while diminishing those of lesser significance, thus enhancing representation learning.

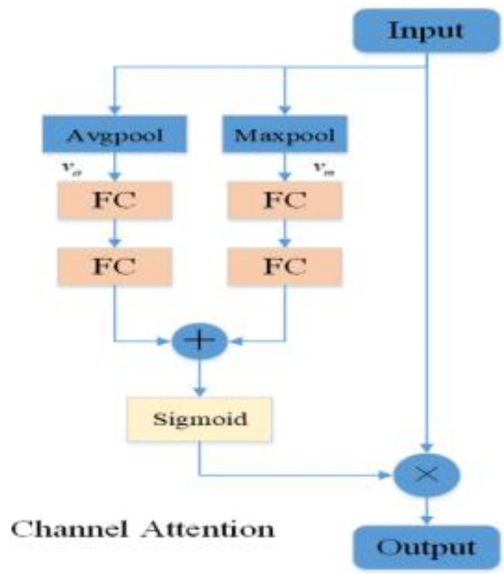

Figure 4.8: Channel Attention Architecture

(Lin et al., 2022)

4.7.2. Spatial Attention Mechanism

Spatial attention mechanism focuses on locating and refining notable sections of the image. It begins with the use of Global Average Pooling and Global Max Pooling to abstract prominent features from various sections of the image. These results are then subjected to a convolutional layer that learns how to produce spatial attention maps. These maps indicate sections of the image that are most relevant for a task. Then a sigmoid activation produces a spatial attention map. These maps assign greater weights to prominent features and less weights to less prominent features. Then the attention map is multiplied with the feature map. This enhances the feature map by selectively focusing attention on the best parts and suppressing less useful sections, and enhances the global representation of features.

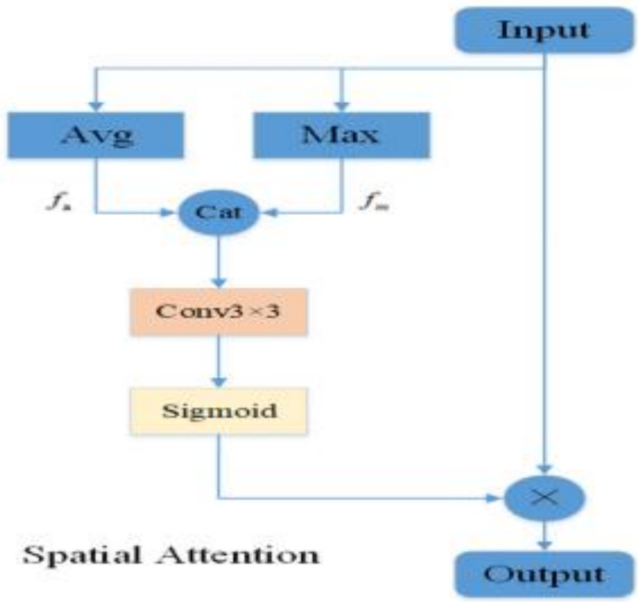

Figure 4.9: Spatial Attention Architecture

(Lin et al., 2022)

4.8. ResNet 18 Classifier

After feature extraction from EfficientNet-B5 and refinement by attention modules, final classification is done by a ResNet-18-based classifier. The 2048-dimensional feature vector of EfficientNet-B5 is first projected to 512 dimensions, which matches ResNet-18's native embedding dimension. The projected features are subjected to residual blocks comprised of linear layers, batch normalization, ReLU activations, and shortcut connections, allowing for efficient and stable learning. The last fully connected layer of ResNet-18 is replaced with five emotion classes rather than its original 1000-class output. This architecture takes advantage of the rich feature representations learned by EfficientNet-B5 along with the residual learning strengths of ResNet-18. The model is trained with CrossEntropyLoss, which transforms logits into probability distributions for efficient multi-class emotion prediction.

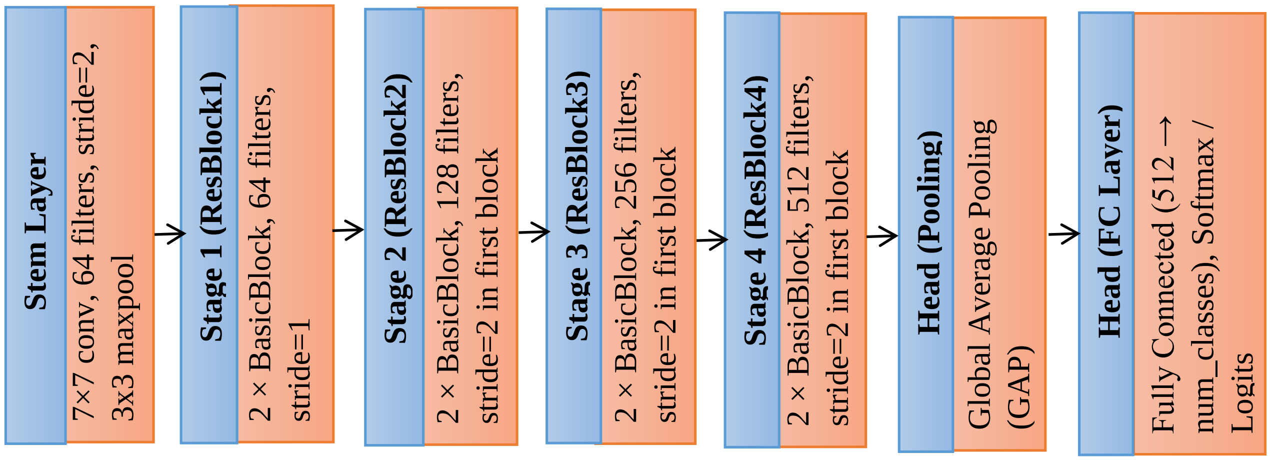

Figure 4.10: Resnet-18 Architecture

4.9. Learning Functions of the Model

The learning process consists of two phases. The training the model and evaluating the model.

Training Function is the function that trains the model using supervised learning with the CrossEntropyLoss function. The function initializes lists to store training and validation loss/accuracy for visualization. It loops through multiple epochs. For each epoch, the model goes through two phases. The training phase updates the model parameters and validation phase evaluates model performance. In training phase the model is set to training mode. It loops through mini-batches from the training dataset and each batch is passed through the model.

In the Forward Pass the model predicts class labels and the error are calculated using CrossEntropyLoss. In the backward propagation gradients are computed and optimizer updates model weights to compares predicted labels with actual labels.

In the validation phase the model is set to the evaluation mode. The validation datasets is passed through the model without updating weights. The loss and accuracy are computed similarly to training but without backward propagation.

The average loss and accuracy for training and validation phases are stored. These values help analyze the model's performance over epochs.

Evaluation Function is the function used after training, to evaluate the model on unseen data. Set the model to evaluation mode to ensure that dropout layers and batch normalization behave correctly. Inputs are passed through the model and predictions are generated and compared with actual labels. The percentage of correct predictions is calculated and displayed.

The model uses CrossEntropyLoss**,** which is standard for classification problems. It measures the difference between predicted probabilities and actual class labels.

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Equation 4.14: Cross Entropy Loss

Where:

- C: number of classes
- y_i: is the actual class label
- $\hat{y}_i$: predicted probability for class i

Optimization Algorithm: Adam is used for updating model weights. Learning rate is set to 1e-4 for stable training. Weight decay (1e-4) prevents overfitting by adding an L2 regularization term.

$$\theta_{t+1} = \theta_t - \alpha \cdot \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon}$$

Equation 4.15: Adaptive Moment Estimation

Where,

- $\alpha = 1 \times 10^{-4}$ Learning rate
- $\lambda = 1 \times 10^{-5}$ Weight decay
- θ_t: model parameters at time step t
- m_t: first moment estimate (mean of gradients)
- v_t: second moment estimate (uncentered variance)
- ϵ: small constant for numerical stability (typically 10^{-8})

4.10. Training Pseudocode

Algorithm 1: Synthetic Data Generation using cGAN

1. **Initialize**:

Generator G with random weights

Discriminator D with random weights

2. **For each epoch** do:

For each batch of real thermal images I_t and labels y:

Update Discriminator D**:**

Sample real images $x \sim pdata(x)$

Impact Sample noise $z \sim pz(z)$

Generate fake images $\hat{x} = G\ (z, y)$

Compute loss: $L_D = E_{x\sim pdata}[\log D(x,y)] + E_{z\sim pz}[\log(1 - D(\hat{x},y))]$
Backpropagate and update D.

Update Generator G**:**

Sample noise $z \sim pz(\ z)$

Generate fake images $\hat{x} = G\ (z, y)$

Compute loss: $L_G = E_{z\sim pz}[\log D(\hat{x},y)] + \lambda \cdot\ L_{recon}(I_t, \hat{x})$
Backpropagate and update G.

3. **End for batches**
4. **End for epochs**
5. **Save** trained G and D weights.
6. **Output:** Synthetic thermal facial images generated by G.

Algorithm 2: Thermal Emotion Classification (EfficientNet-B5 + CBAM + ResNet-18)

1. **Initialize**:

EfficientNet-B5 backbone with pretrained weights

CBAM attention modules (inserted into backbone)

ResNet-18 classification head (modified last FC layer for number of emotion classes)

2. **For each epoch** do:
3. **For each batch** of thermal images I_t and labels y:

Apply data preprocessing

Apply data augmentation

Include synthetic images generated from cGAN.

Feature Extraction: Pass images through EfficientNet-B5 backbone.

Attention Refinement: Apply CBAM to enhance relevant features.

Classification Head: Pass features through ResNet-18

Compute Cross-Entropy Loss: LCE =- $\sum_{i=1}^{C} y_i \log(\hat{y}_i)$

Backpropagate and update parameters of backbone, CBAM, and ResNet-18 head.

4. **End for batches**
5. **End for epochs**
6. **Save** trained EfficientNet-B5 + CBAM + ResNet-18 weights.
7. **Output:** Predicted emotion classes for thermal facial images.

CHAPTER FIVE

5. IMPLEMENTATION DETAILS

5.1. Chapter Overview

This chapter takes into account the implementation of the model as suggested, together with operational context, image pre-processing methods, data augmentation, synthetic data generation, implementation of EfficientNet-B5 with CBAM, ResNet-18 classifier and the experimental protocols adopted.

5.2. Working Environment

This section explains the hardware tools configuration utilized in deploying the model in the stated environment.

Table 5.1: Hardware Tools Specification

Computer Component	Specification
Processor	Intel(R) Core(TM) i7-8700M CPU @ 3.20GHz 3.19 GHz
Installed RAM	8.00 GB
Operating System	Windows 10
System Type	64-bit operating system, x64-based processor
Graphics	Intel ® HD Graphics 520 / NVIDIA Corporation
GPU	NVIDIA GeForce RTX 2070 SUPER
Driver Version	560.94
CUDA Version	12.6
GPU Memory	8192 MB (8 GB)

5.3. Environmental Setup

In this research, we used several software tools and IDEs that are compatible with popular machine learning frameworks like Tensor Board and PyTorch to assist us in properly designing and debugging our models.

In the implementation stage, a range of tools was employed to support the processes of development, experimentation, and deployment. Anaconda served as the main environment for Python library and dependency management, ensuring compatibility and easing the installation of essential packages including NumPy, pandas, SciPy, Matplotlib, and scikit-learn. The solution utilized in this case utilizes Anaconda version 23.7.4 to install all the libraries and modules required for implementation. Jupyter Notebook served as an interactive environment integrating code, documentation, and visualization, supporting experimentation and improving the reproducibility of the workflow. Google Colab provided access to free cloud-based GPUs, significantly accelerating the training of computationally intensive deep learning models, particularly in scenarios where local hardware resources were limited. Finally, Kaggle served as a platform for acquiring a range of datasets, conducting experiments in cloud-based kernels, and comparing findings with existing research, thus supporting data procurement and collaborative research.

5.4. Implementation of Proposed Model

The proposed models utilize Bilateral Filter, resizing, normalization, Gaussian Blur and CLAHE for preprocessing of images, data augmentation like random rotation, Random Horizontal Flip and Brightness Adjustment, cGAN for synthetic data generation, EfficientNet-B5 ,CBAM for feature extraction and then utilize RensNet-18 for classification.

5.4.1. Implementation of Bilateral Filter

Bilateral filter process is carried out by using the function bilateral = cv2.bilateralFilter (img, d=5, sigmaColor=50, sigmaSpace=50).This minimizes noise and keeps edges sharp, which is crucial for keeping facial boundary information in thermal imagery.

5.4.2. Implementation of Gaussian Blur

Implementation of Gaussian Blur is employed using the function Gaussian_blur = cv2.GaussianBlur (bilateral, (5, 5), 0) to smooths the facial emotional thermal images.

5.4.3. Implementation of CLAHE

The CLAHE process is carried out by using the function clahe=cv2.createCLAHE (clipLimit = 2.0, tileGridSize = (8, 8)), and enhanced_img=clahe.apply (gaussian_blur). This enhancesthe contrast of the facial emotional thermal image in low-light conditions,improving local contrast and enhances visibility of fine thermal facial features.

5.4.4. Implementation of Resizing

All thermal facial images are resized to a predetermined resolution of 128 × 128 pixels using the function: resized_img = cv2.resize (image, (128, 128), interpolation=cv2.INTER_CUBIC). This provides consistency throughout the dataset and minimizes computational complexity in training, while bicubic interpolation maintains image quality.

5.4.5. Implementation of Normalization

The Normalization process is carried out with two versatile options: Normalization to [0, 1] range: normalized_img = resized_img.astype(np.float32) / 255.0.This normalizes pixel intensities to a floating-point range of 0 to 1, which is usually desirable for deep learning models to normalize training.

Normalization to [0, 255] range (default): normalized_img = resized_img. Here, the pixel values are kept in their original intensity range, which is appropriate when normalization is done internally by preprocessing layers in the deep learning workflow.

5.4.6. Implementation of Data Augmentation

The random rotation data augmentation of emotional facial thermal images is implemented using function: angle = random.uniform(-10, 10), M = cv2.getRotationMatrix2D((w // 2, h // 2), angle, 1), image = cv2.warpAffine(image, M, (w, h), flags=cv2.INTER_LINEAR, borderMode = cv2.BORDER_REFLECT). This step randomly rotates the image between the ranges of –10° to +10°, thus emulating natural head tilts.

The Random Horizontal Flip data augmentation of emotional facial thermal images is implemented using function: if random.random() > 0.5: image = cv2.flip(image, 1) to simulates variations in face orientation. Horizontal flipping of the image with a probability of 50% is used to simulate variations in facial orientation.

Brightness Adjustment data augmentation of emotional facial thermal images is implemented using function: factor = random.uniform (0.9, 1.), image = np.clip(image * factor, 0, 255).astype(np.uint8) to adjusts brightness for different lighting conditions. In these steps the thermal image brightness is adjusted within a safe range to suit the variations in illumination conditions.

5.5. Implementation of cGAN

5.5.1. Generator Architecture

The Generator architecture formed from the latent vector and class labels as input and generated thermal image (1 channel, grayscale) as output. The embedding layer in the generator architectures are used to converts class labels into embeddings of size embed_dim = 10. The latent vector and label embeddings are concatenated along the channel dimension.

Table 5.2: Generator Architecture (Transposed Convolution Blocks)

Layer Type	Details
Input Layer	ConvTranspose2d(in_channels=z_dim + embed_dim, out_channels=512, kernel_size=4, stride=1, padding=0) Output size: **4×4** BatchNorm2d(512)ReLU(True)
Hidden Layer 1	ConvTranspose2d(512, 256, kernel_size=4, stride=2, padding=1)Output size: **8×8** BatchNorm2d(256)ReLU(True)
Hidden Layer 2	ConvTranspose2d(256, 128, kernel_size=4, stride=2, padding=1)Output size: **16×16** BatchNorm2d(128)ReLU(True)
Hidden Layer 3	ConvTranspose2d(128, 64, kernel_size=4, stride=2, padding=1)Output size: **32×32** BatchNorm2d(64)ReLU(True)
Output Layer	ConvTranspose2d(64, img_channels, kernel_size=4, stride=2, padding=1) Output size: **64×64 grayscale image** Tanh() (scales output to [-1, 1])

The input to the model consists of a latent vector z concatenated with an embedding vector, which can be a class label, resulting in an overall dimensionality of z_{dim} + $embed_{dim}$. This input is fed through the model's generator network to produce the final output, a single-channel grayscale image. The pixel values of the generated image are normalized to the range [−1, 1].

5.5.2. Discriminator Architecture

The discriminator network accepts a gray scale image (1 channel) and class labels and produces a probability score, with 0 being fake and 1 being real. To condition on class information, the labels are first fed through an embedding layer that transforms them into 50-sized embedding's.

The embedding's are then reshaped and expanded using label_emb.view(labels.size(0), -1, 1, 1) to have the same spatial resolution as the image, after which they are concatenated with the image along the channel axis. Therefore, the first convolutional block is presented with a total of 1+50 input channels. The discriminator consists of five convolutional blocks followed by a sigmoid activation, which provides the probability that the input image is real or fake.

Table 5.3: Discriminator Architecture

Layer Type	Details
Input Layer	Conv2d(in_channels=img_channels + 50, out_channels=64, kernel_size=4, stride=2, padding=1)LeakyReLU(negative_slope=0.2)
Hidden Layer 1	Conv2d(64, 128, kernel_size=4, stride=2, padding=1)BatchNorm2d(128)LeakyReLU(0.2)
Hidden Layer 2	Conv2d(128, 256, kernel_size=4, stride=2, padding=1)BatchNorm2d(256)LeakyReLU(0.2)
Hidden Layer 3	Conv2d(256, 512, kernel_size=4, stride=2, padding=1)BatchNorm2d(512)LeakyReLU(0.2)
Output Layer	Conv2d(512, 1, kernel_size=4, stride=1, padding=0)Sigmoid()

The final output is a single scalar value in the range [0, 1], representing the probability that the input image is real. The use of LeakyReLU activation functions and BatchNorm helps stabilize training and improve convergence.

5.6. Data Pipeline

Dataset class reads thermal images and extracts emotion labels from folder names by applying transformations and loads dataset in batches.

5.7. Training Loop

The Discriminator training computes loss on real and fake images to back propagate and update discriminator. The generator training generate fake images to compute loss and wants D to classify fake images as real and back propagate to update generator.

Loss Functions

Binary Cross-Entropy Loss is a loss function used for binary classification tasks. It calculates the difference between actual class labels (0 or 1) and predicted probabilities from the model.

5.8. Implementation of EfficientNet-B5 with CBAM

The input layer of EfficientNet-B5 takes images of size (456 × 456 × 3). EfficientNet-B5 is used in combination with the Convolutional Block Attention Module to extract deep feature representations, while the default classifier is discarded.

Table 5.4 : EfficientNet-B5 with CBAM MBConv Blocks

Layer Type	**Details**
Input Layer	Image Input: **(456 × 456 × 3)**
Stem Convolution	Conv2d(3, 48, kernel_size=3, stride=2, padding=1) → BatchNorm2d → Swish
MBConv Stage 1	3 × MBConv1 (expansion=1, kernel=3, output=24, stride=1)
MBConv Stage 2	5 × MBConv6 (expansion=6, kernel=3, output=40, stride=2)**+ CBAM after block 7**
MBConv Stage 3	5 × MBConv6 (expansion=6, kernel=5, output=64, stride=2)
MBConv Stage 4	7 × MBConv6 (expansion=6, kernel=3, output=128, stride=2)**+ CBAM after block 19**
MBConv Stage 5	7 × MBConv6 (expansion=6, kernel=5, output=176, stride=1)
MBConv Stage 6	9 × MBConv6 (expansion=6, kernel=5, output=304, stride=2)**+ CBAM after block 35**
MBConv Stage 7	3 × MBConv6 (expansion=6, kernel=3, output=512, stride=1)**+ CBAM after block 38**
Final Convolution	Conv2d(512, 2048, kernel_size=1, stride=1) → BatchNorm2d → Swish
Pooling Layer	Global Average Pooling (GAP)
Dropout Layer	Dropout(p=0.4)

5.9. Implementation of channel attention

At each CBAM insertion location namely blocks 7, 19, 35, and 38, the feature maps, which have dimensions C × H × W (with C being different based on the output channel size of the corresponding block, e.g., 40, 128, 304, or 512), undergo a channel-wise attention transformation. First, global average pooling and global max pooling are performed over the spatial dimensions, hence reducing the feature maps to dimensions of $C \times 1 \times 1$ for each pooling output. Then, these pooled features are fed through a shared two-layer MLP: The first fully connected layer shrinks the channel dimension from C to C/r (reduction ratio $r = 16$) with ReLU activation. The second fully connected layer again brings the dimension to C.

The two resulting outputs are subsequently integrated and subjected to a sigmoid activation function, which produces a channel attention vector of dimensions $C \times 1 \times 1$. This vector is then applied element-wise to the original feature maps, thereby adaptively amplifying informative channels while diminishing those that are less beneficial. Such selective weighting serves to enhance feature representation and bolster robustness within the deeper blocks of EfficientNet-B5.

Channel attention adopts adaptive channel-wise weighting, enhancing informative channels and reducing noise from less useful ones.

5.10. Implementation of spatial attention

Following the application of channel attention, the refined feature maps ($C \times H \times W$) are input into the spatial attention module. First, global average pooling and global max pooling are performed along the channel axis, resulting in two spatial maps of size $1 \times H \times W$. These two spatial maps are concatenated along the channel axis, ending up with a tensor of size $2 \times H \times W$.

It is followed by a 2D convolution of kernel size 7×7, stride 1, and padding 3, and sigmoid activation to generate a spatial attention map of size $1 \times H \times W$. The map emphasizes salient spatial areas and depresses less informative regions. The spatial attention map is element-wise multiplied with the channel-refined feature maps; allowing EfficientNet-B5 to concentrate on the most discriminative spatial locations for emotion recognition. This method enhances the feature map by allowing the network to concentrate on informative spatial locations.

5.11. Implementation of classification with ResNet -18

After feature extraction from the EfficientNet-B5 backbone provided with CBAM modules, classification is done by a modified ResNet-18 head. Rather than using a plain dense feedforward network, the model incorporates the residual learning mechanism of ResNet-18 to improve feature representation while preventing vanishing gradients. This 2048-dimensional feature vector from EfficientNet-B5 is the input to the ResNet-18 classifier head. This head is built by taking ResNet-18's convolutional and residual block structure, but adapted to take feature vectors instead of raw images. The input vector is reshaped and projected via an initial fully connected layer to match dimensionality with ResNet-18's expected embedding dimension (usually 512). The features are then sent through a series of residual blocks comprising linear transformations, batch normalization, and ReLU activations, with identity shortcuts to maintain gradient flow. Finally, a fully connected classification layer maps the learned representation to the five intended emotion classes: happiness, sadness, anger, surprise, and neutrality. Thus, the forward propagation follows the scheme: input projection → residual blocks (according to the ResNet-18 architecture) → global average pooling → final fully connected layer → logits. With the use of residual connections, the ResNet-18 head provides stable training and more abundant feature abstraction than a simple multilayer perceptron, with relatively modest computational requirements.

Table 5.5: Resnet-18 architecture

Stage / Layer	Details	Output Shape
Input	Image (224 × 224 × 3)	224 × 224 × 3
Conv1	7×7 conv, 64 filters, stride=2	112 × 112 × 64
MaxPool	3×3 max pool, stride=2	56 × 56 × 64
Stage 1 (ResBlock1)	2 × BasicBlock, 64 filters, stride=1	56 × 56 × 64
Stage 2 (ResBlock2)	2 × BasicBlock, 128 filters, stride=2 in first block	28 × 28 × 128
Stage 3 (ResBlock3)	2 × BasicBlock, 256 filters, stride=2 in first block	14 × 14 × 256
Stage 4 (ResBlock4)	2 × BasicBlock, 512 filters, stride=2 in first block	7 × 7 × 512
Head (Pooling)	Global Average Pooling (GAP)	1 × 1 × 512
Head (FC Layer)	Fully Connected (512 → num_classes), Softmax / Logits	Five_classes

5.12. Hyper parameter Configuration

Hyper parameters are the configuration settings that govern the learning of a machine learning model. They are adjustable and have a direct influence on the learning effectiveness of a model. Optimizers are techniques that regulate a neural network's parameters during training to reduce the error or loss function.

5.12.1. Data Generation Hyper parameters

The input images are also resized to a standard size of (224, 224) to ensure consistency in the dataset. The pixel values are normalized with a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225], which is default for models pre-trained on ImageNet. The model is trained at a batch size of 16 to enable data processing in batches. Shuffling is permitted during training to ensure that each batch is randomly selected from the datasets, which favors better generalization. Shuffling is disabled for validation and test phases to maintain the consistency of data order. Data loading is made efficient by setting the number of workers to 4, permitting multi-threaded data loading for faster batch fetching, which contributes to training efficiency.

Table 5.6 : Image Generation Hyper parameters

Parameter	**Value / Description**
Latent vector dimension (z_dim)	2048
Embedding dimension (embed_dim)	10
Number of classes	5
Image resolution	64×64
Generator architecture	5 ConvTranspose2d layers with BatchNorm + ReLU
Discriminator architecture	5 Conv2d layers with BatchNorm + LeakyReLU
Loss function	Binary Cross-Entropy Loss
Optimizer	Adam
Learning rate	0.00016
Adam betas	(0.5, 0.999)
Batch size	64

Number of epochs	400

Table 5.6 adjusted parameters. In our experimentation, the model was trained under various hyper parameter settings, including different latent vector dimension (z_dim =512, 1024, 2048), training epochs(100 , 200 , 400), batch size (32 and 64), and learning rates. These variations were systematically explored to evaluate the models performance on a variety of configurations. Following repetitive experiments, the optimal set of hyper parameters was selected based on the model's stability during training and its ability to generate high-quality thermal facial emotion images.

5.12.2. Model Training Hyperparameters

For this model CrossEntropyLoss function is acts as the loss function, appropriate for multi-class classification tasks. Optimization for the model is achieved using the Adam optimizer where the learning rate has been set at 0.0001, for providing a balance between convergence rate and stability. Training is performed for 50 epochs, and early stopping is activated to monitor validation performance and terminate training if there is no improvement in 10 consecutive epochs, thus preventing over fitting. For weight initialization, the model utilizes pretrained EfficientNet-B5 weights, thus facilitating transfer learning of a model trained on the large ImageNet dataset.

Table 5.7: Model Hyperparameters

Hyper parameter	Value/Description
Loss Function	CrossEntropyLoss
Optimizer	Adam
Learning Rate	1e-4
Weight Decay	1e-4
Batch Size	16
Epochs	50
Early Stopping Patience	10

Table 5.7 adjusted parameters. In our experiment, the model was trained under various hyper parameter settings, including different training epochs (20, 30, 50), batch size (8 and 16), and learning rates. These variations were systematically explored to evaluate the models performance on a variety of configurations. Following repetitive experiments, the optimal set of hyper parameters was selected based on the model's stability during training and its capacity to classify into different emotional class.

5.13. Experimental Class

Table 5.8 : Experimental Class

SN	Notation	Experiment	Dataset used
1	HFER using thermal images	Baseline Utilizing efficient-net B0	Comprehensive Facial Thermal Dataset
2	HFER using thermal images	Utilizing bilateral filter,CLAHE and efficient-net B5	Comprehensive Facial Thermal Dataset
3	HFER using thermal images	Utilizing bilateral filter, CLAHE ,cGAN and efficient-Net B5 for data diversification and feature extraction	Comprehensive Facial Thermal Dataset
4	HFER using thermal images	Utilizing bilateral filter, CLAHE, cGAN ,efficient-Net B5 and CBAM	Comprehensive Facial Thermal Dataset

In table 5. 8, different experiments which are conducted on this study are summarized. In the first experiment, a baseline was trained using the original datasets to provide a reference for the later comparison with the proposed model. In the second experiment, bilateral filtering and CLAHE preprocessing were applied, with EfficientNet-B5 used for feature extraction. The goal of this experiment is to address the impact of preprocessing on thermal facial emotion recognition. In the third experiment, bilateral filter, CLHE, and cGAN -generated synthetic data were combined with EfficientNet-B5 without attention to investigate the impact of synthetic data on model performance. Finally, the fourth experiment were conducted by utilizing bilateral filter, CLAHE, cGAN and EfficientNet-B5 integrated with CBAM to address the impact of attention based feature extraction on classification performance.

CHAPTER SIX

6. RESULTS AND DISCUSSION

6.1. Overview of the Chapter

In this chapter, the findings of the preliminary work and the diverse experiments conducted, such as the new method proposed by us, are presented. The research findings are organized through the use of charts, graphs, and tables to facilitate understanding and allow for comparative analysis.

6.2. Sample preprocessing Log

The proposed model applied Bilateral Filter, Gaussian Blur and CLAHE in preprocessing of images. Furthermore, images were augmented by using stochastic transformations such as rotation, flipping, and brightness alteration. These processing stages aimed at enhancing quality and variability of learning data. The corresponding experiments compared models trained on raw images alongside with those trained on preprocessed images. The preprocessing outputs are visualized in Figure 6.1, displaying sharper and more informative face features presentations.

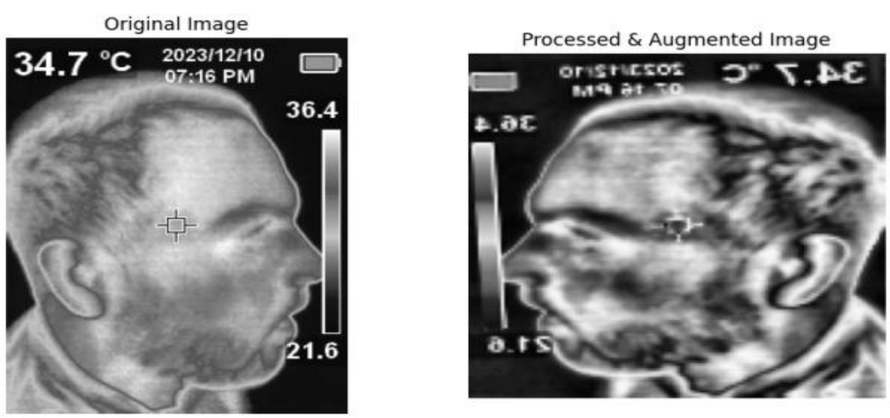

Figure 6.1 : Sample augmented images

6.2. Sample Training Log for cGAN

To measure the performance of the conditional GAN in generating synthetic thermal facial emotion images, we tracked the generator and discriminator loss curves, which is shown in Figure 6.2.

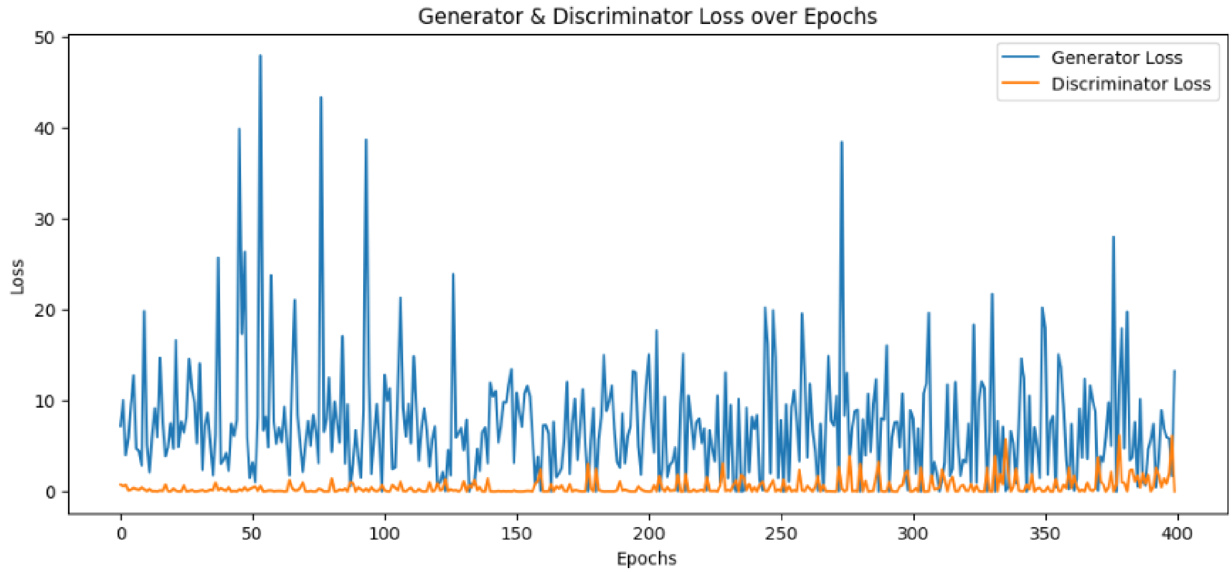

Figure 6.2: cGAN generator and discriminator loss

Figure 6.3: Sample generated images

Realism and generated image quality were measured quantitatively using the Fréchet Inception Distance, which measures the divergence between the feature distributions between fake and real images in terms of Inception-v3 activation. Decreasing values for FID show increasing similarity to actual image distributions. During our experiments, the calculated FID value of 20.7062 confirmed that the generated thermal images closely matched the distribution of real samples while maintaining diversity. In addition to reporting the final FID score, we also tracked the progression of FID values during training, as illustrated in Figure 6.4. This plot indicates how the cGAN, with every successive epoch, enhanced its capacity to generate diverse and realistic thermal facial emotion images by closing the gap of distributions between synthetic and real data. The trend of declining FID value shows that the model converged and produced increasingly better-quality samples with training. This ensured that the cGAN effectively learned structural and statistical characteristics of thermal facial images. By combining loss monitoring and FID assessment, the experiments offered robust evidence that synthetic data creation was effective and sufficient in mitigating datasets shortages in thermal facial emotion recognition. The results confirm that synthetic data generation is a feasible solution for enriching small-scale thermal FER datasets and for reducing overfitting risk in subsequent classification experiments.

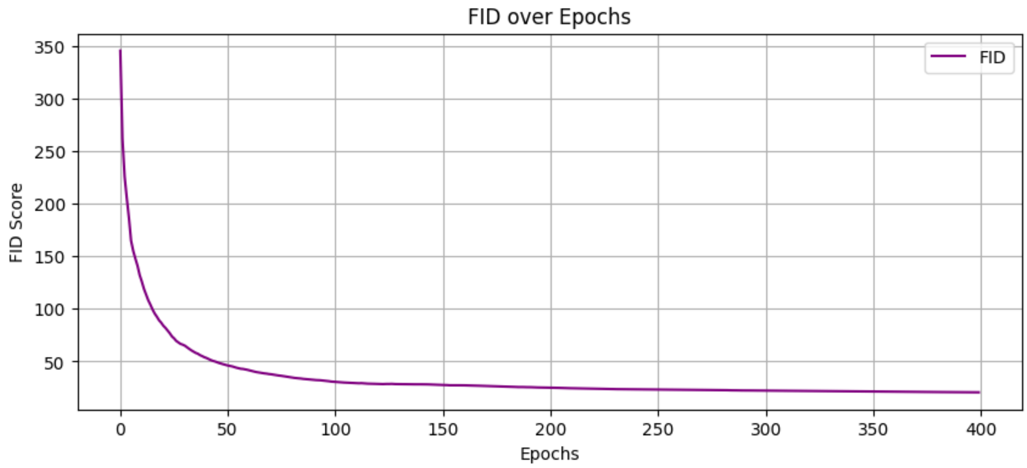

Figure 6.4: FID progress

6.3. Training Log for the model

We used graphs and charts to monitor the performance of the model as it learned knowledge. The data monitored various facets of the training process, such as the model's performance, minimized errors (loss) and provide useful insights regarding the model's effectiveness and training status for thermal human face emotion recognition.

To evaluate the performance of the proposed thermal facial emotion recognition model, we calculated various key classification performance measures, such as accuracy, precision, recall, and F1-score.

In addition to scalar metrics, a confusion matrix was used to provide a clear overview of the classification outcome for each emotion class. The confusion matrix enables visualization of true positives, false positives, false negatives, and true negatives, which can be utilized to spot specific classes where the model may underperform or misclassify. Such analysis is crucial in the evaluation of model robustness, especially in multiclass emotion recognition.

The combination of high precision and a balanced recall, precision, and F1-score performance indicates the success of our thermal imaging-based feature extraction and classification pipeline. Additionally, the confusion matrix ensured that the model had high discriminative power among various facial expressions, with minimal misclassification among similar emotional states.

6.4. Experimental Results

We conducted multiple training iterations using different factors for the proposed model's network. To verify results consistency and maintain a fair comparison in this research four experiments were performed using identical datasets, hardware configurations and software settings. We evaluated our approach's performance against the Comprehensive Facial Thermal Dataset's.

To quantify the performance of the proposed methodology, we used a combination of generative and classification performance metrics. The performance of the cGAN was quantified based on the implementation of FID which is known metrics for assessing the quality and realism of the synthesized images. Within the framework of the classification task, we made use of various evaluation metrics, such as accuracy, confusion matrices, F1 score, precision, and recall, to evaluate systematically the model's efficacy in recognizing and classifying facial emotions. Collectively, these metrics offer a comprehensive evaluation of the model's performance from generative and classification perspectives.

6.4.1. Evaluation of experiments on Comprehensive Facial Thermal datasets

To systematically evaluate the effectiveness of each component in our proposed approach, we conducted a series of experiments with incremental model configurations. This allowed us to isolate the contribution of each element namely, data augmentation using cGAN, feature extraction using EfficientNet-B5, and CBAM. In all experiments, ResNet-18 served as the classification head, leveraging its residual connections to improve training stability and enhance the mapping from high-dimensional features to emotion labels. The performance of each configuration was assessed using accuracy, precision, recall and F1-score metrics.

Baseline Model

The baseline model was trained using the original comprehensive facial thermal dataset without s ynthetically generated data, and attention mechanism. It achieved the test accuracy of 97.04%. This setup was utilized as a baseline to validate the impact of the enhancements made in subsequ ent experiments.

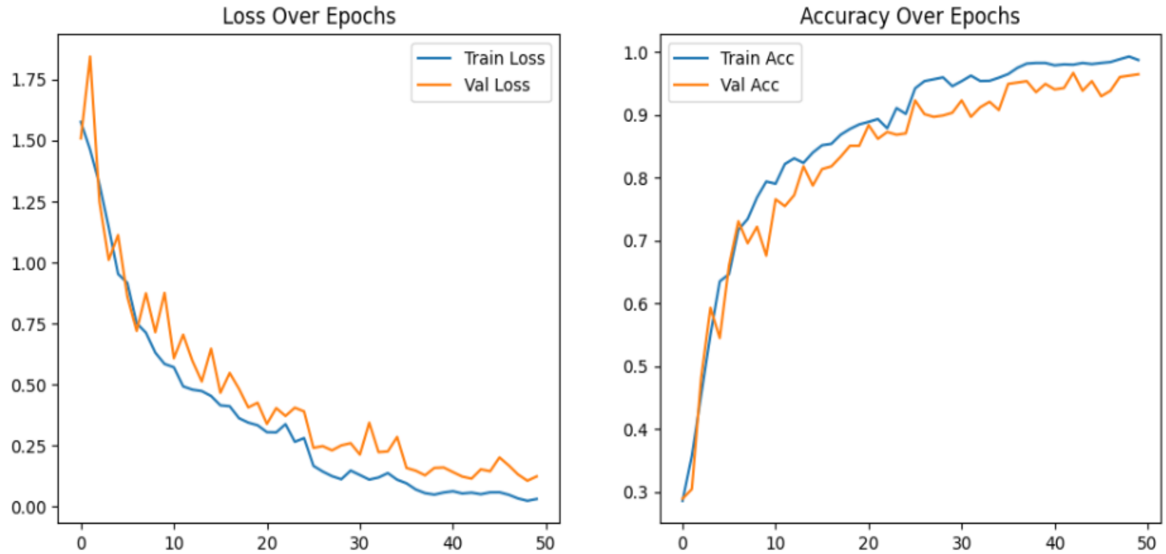

Figure 6.5: Training and Validation graph

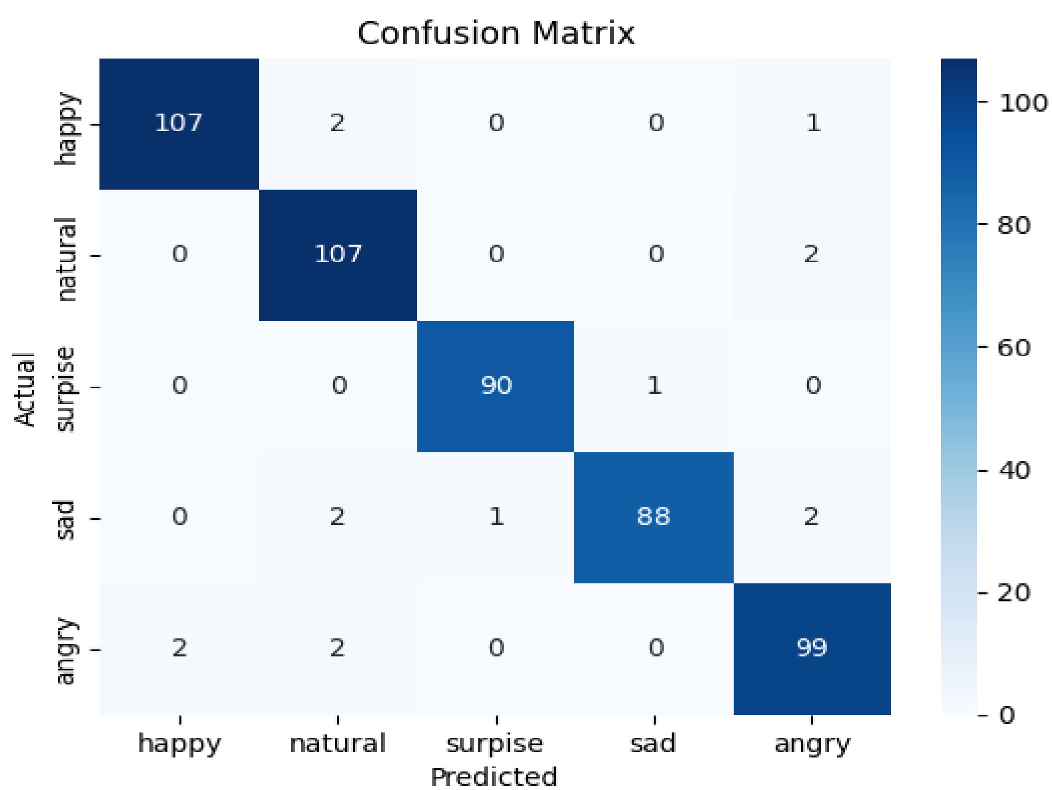

Figure 6.6: Confusion Matrix

Table 6.1 : Performance Metrics per Emotion Class

Face Emotion	**Precision**	**Recall**	**F1-Score**
Happy	0.98	0.97	0.98
Natural	0.95	0.98	0.96
Surprise	0.99	0.99	0.99
Sad	0.99	0.95	0.97
Angry	0.95	0.96	0.96

Table 6.1 shows the class-wise performance measures of the suggested thermal facial emotion recognition system. Precision, recall, and F1-score are provided for all emotion classes, such as happy, natural, surprise, sad, and angry.

Utilizing Bilateral Filter, CLAHE and EfficeintNet B5

In the second configuration, we incorporated the Bilateral Filter and CLAHE for data preprocessi ng and augmentation techniques to increase the data size. A bilateral filter is applied to retain edg e definition while simultaneously smoothing the image. A Gaussian blur is applied to suppress n oise further. Then, Contrast Limited Adaptive Histogram Equalization is applied to improve the i mage's contrast, particularly in areas with poor visibility. The image is then sent to the augmentat ion function, where various stochastic transform such as rotation, flipping, and brightness alterati on are applied. These augmentation techniques increased the original datasets to 7,455 images. T he model achieved a test accuracy of 97.21%, which resulted in a slight improvement over the ba seline. This demonstrated that bilateral filtering effectively preserved key features while reducing noise.

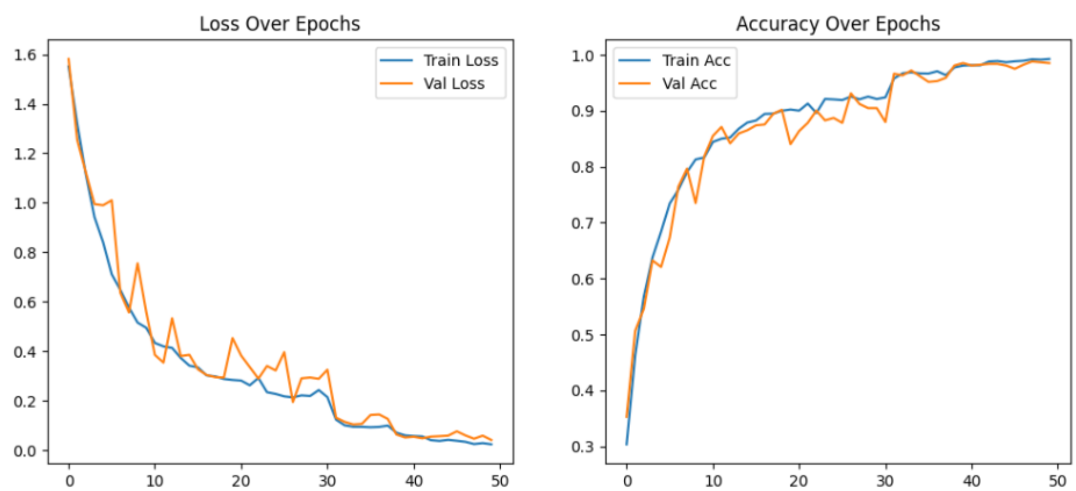

Figure 6.7: Training and Validation graph

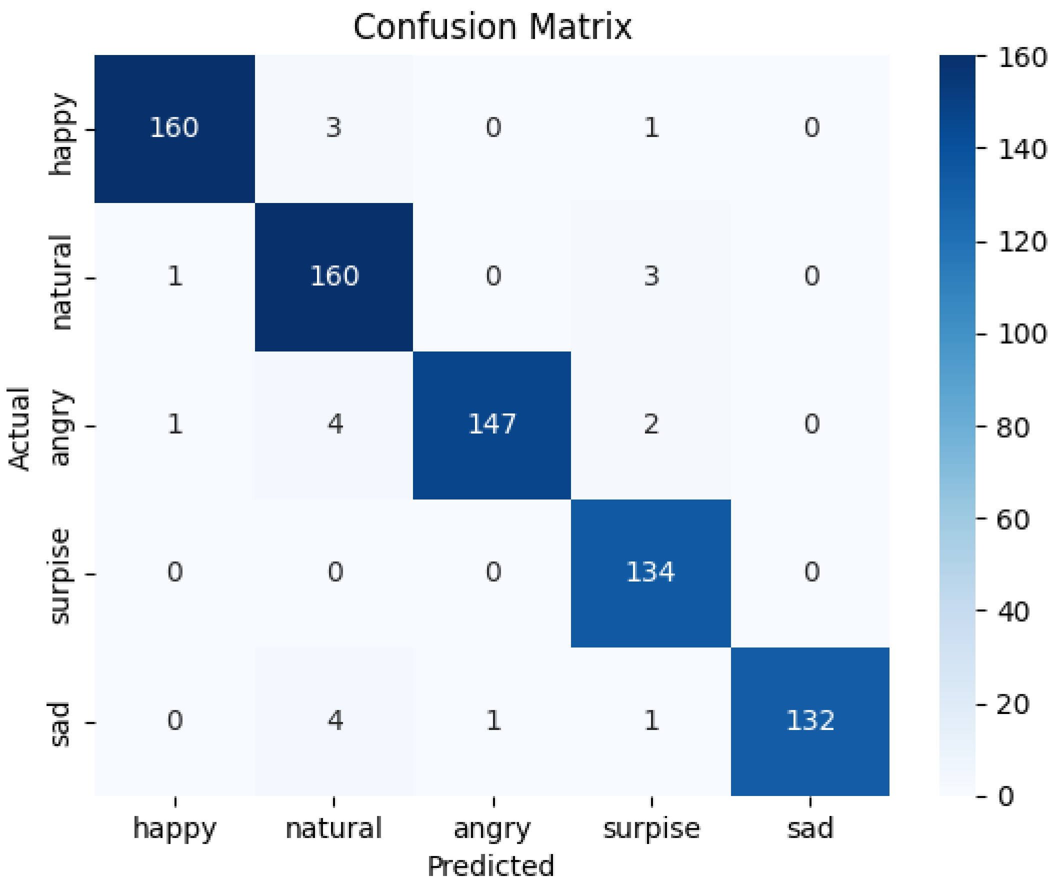

Figure 6.8: Confusion Matrix

Table 6.2 : Performance Metrics per Emotion Class

Face Emotion	Precision	Recall	F1-Score
Happy	0.99	0.98	0.98
Natural	0.94	0.98	0.96
Angry	0.99	0.95	0.97
Surprise	0.95	1.00	0.97
Sad	1.00	0.96	0.98

Table 6.2 shows the class-wise performance measures of the suggested thermal facial emotion recognition system. Precision, recall, and F1-score are provided for all emotion classes, such as happy, natural, surprise, sad, and angry.

Utilizing Bilateral Filter, CLAHE, cGAN and EfficeintNet B5

In the third configuration, data diversification was introduced by utilizing cGAN to generate 1,000 additional thermal facial images expanding the datasets and addressing the challenge of data scarcity. The model was trained on this expanded datasets using EfficientNet as the backbone network. This configuration aimed to evaluate the impact of increased data diversity and the utilization of a stronger backbone network on model performance, with the architecture kept free from channel and spatial attention modules. The model achieved a test accuracy of 98.01% with these diversified datasets, demonstrating that increasing data diversity through synthetic generation enhances generalization and classification performance.

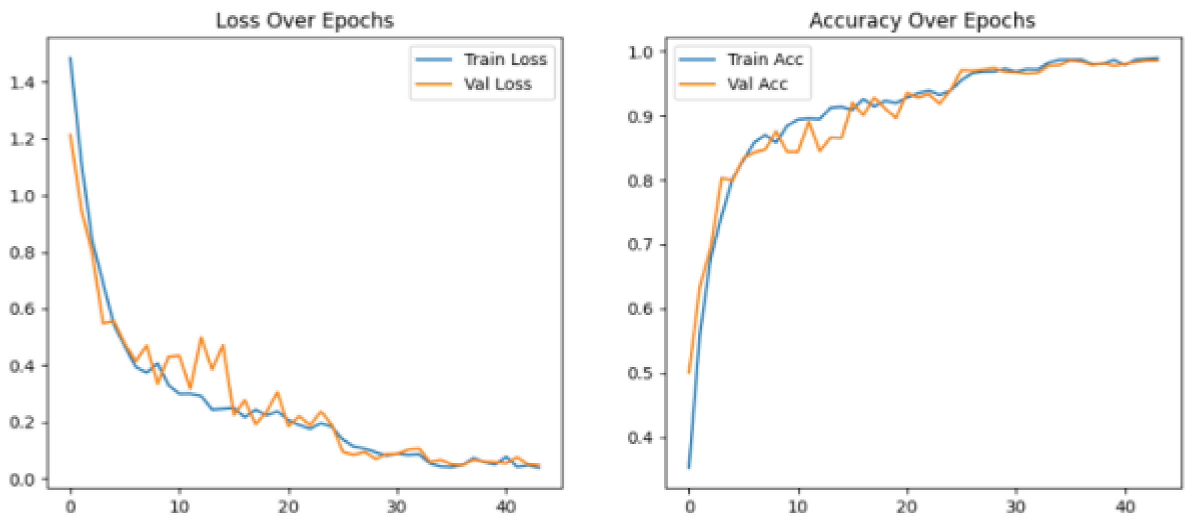

Figure 6.9: Training and Validation graph

Figure 6.10: Confusion Matrix

Table 6.3 : Performance Metrics per Emotion Class

Face Emotion	**Precision**	**Recall**	**F1-Score**
Happy	0.98	1.00	0.99
Natural	0.99	0.94	0.96
Angry	0.97	0.99	0.98
Surprise	0.96	0.99	0.98
Sad	0.99	0.99	0.99

Table 6.3 shows the class-wise performance measures of the suggested thermal facial emotion recognition system. Precision, recall, and F1-score are provided for all emotion classes, such as happy, natural, surprise, sad, and angry.

Utilizing Bilateral Filter, CLAHE, cGAN, EfficeintNet-B5 and CBAM

The fourth configuration employed EfficientNet-B5 was integrated with CBAM to enhance feature extraction. The integration of attention improves the model's representational capacity by allowing it to focus on the feature channels and spacial regions carrying the most vital features. Incorporating CBAM into EfficientNet-B5 markedly improved performance, achieving a test accuracy of 98.81%. This result confirmed the role of selective channel weighting and spacial attention in enhancing feature representation and enabling the model to focus on the most informative region for emotion classification.

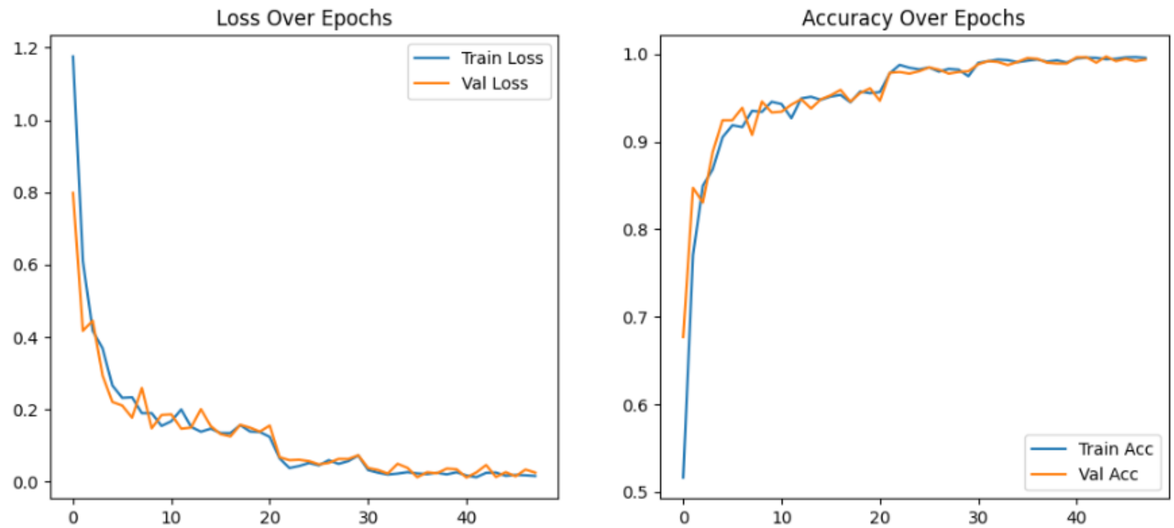

Figure 6.11: Training and Validation graph

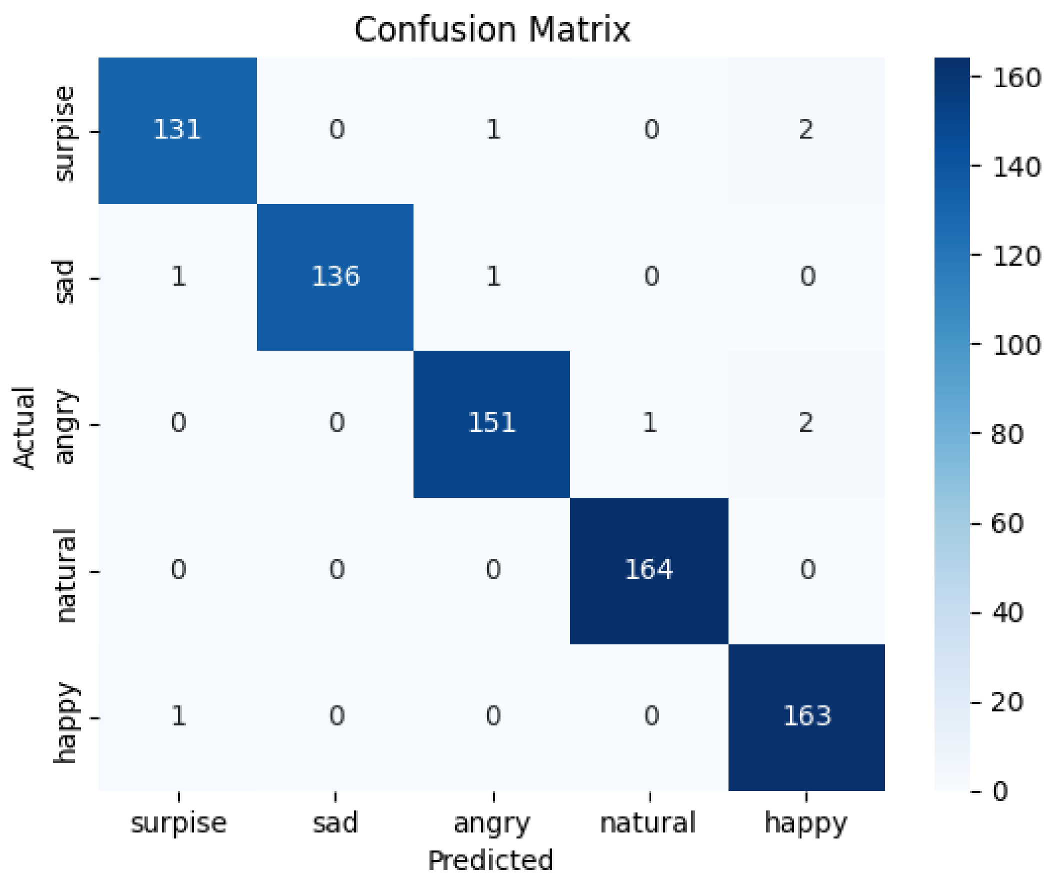

Figure 6.12: Confusion Matrix

Table 6.4 : Performance Metrics per Emotion Class

Face Emotion	Precision	Recall	F1-Score
Surprise	0.98	0.98	0.98
Sad	1.00	0.99	0.99
Angry	0.99	0.98	0.98
Natural	0.99	1.00	1.00
Happy	0.98	0.99	0.98

Table 6.4 shows the class-wise performance measures of the suggested thermal facial emotion recognition system. Precision, recall, and F1-score are provided for all emotion classes, such as happy, natural, surprise, sad, and angry. The step-by-step experimental results validated the role of each component to the proposed thermal facial emotion recognition. The use of cGAN significantly enhanced diversity in the datasets, which facilitated generalization. Attention mechanisms, particularly the combination of channel and spatial attention, further focused the

model's attention, resulting in the highest classification accuracy. Overall, the progressive improvements validated the contribution of each enhancement.

Table 6.5 : Experimental comparison on the Evaluation metrics

SN	Experiment	Accuracy	Precision	Recall	F1 score
1	Baseline	97.04%	0.97	0.97	0.97
2	Utilizing Bilateral Filter ,CLAHE and EfficeintNet-B5	97.21%	0.97	0.97	0.97
3	Utilizing Bilateral Filter, CLAHE ,cGAN and EfficeintNet-B5	98.01%	0.98	0.98	0.98
4	Utilizing Bilateral Filter, CLAHE , cGAN , and EfficeintNet-B5 with CBAM	**98.81%.**	**0.99**	**0.99**	**0.99**

The results presented in the table 6.5 reveal that the experimental results clearly demonstrate the step-by-step improvements achieved by the integration of different components in the thermal facial emotion recognition. The last experimental configuration, which incorporated bilateral filtering, cGAN-based synthetic data generation, and EfficientNet-B5 integrated with CBAM, enabling the model to focus on the most discriminative features. This configuration achieved the highest overall performance of 98.81% accuracy with precision, recall, and F1-scores of 0.99 across all emotional classes. This outcome proves that the combined effect of state-of-the-art preprocessing, data augmentation, synthetic image generation and attention-augmented deep learning architecture yields a very robust and trustworthy thermal facial emotion recognition.

6.4.3. Experimental Results of the predicted Thermal Facial Emotion Images

The proposed model takes a preprocessed thermal facial image as input and predicts emotional class among the predefined categories such as happy, sad, angry, surprise and neutral. Features are first extracted by the EfficientNet backbone enhanced with CBAM. These features are then passed through a ResNet-18 classification head, which maps the high-dimensional feature representations to emotion classes. Finally, a softmax activation generates a probability distribution across all possible emotional classes, and the class having the maximum probability is identified as the predicted emotion.

Formally, given an input image x, the model outputs a vector $\hat{y} = [p_1, p_2, \ldots, p_n]$, where, pi is the predicted probability for class i, and n is the total number of emotion classes. The predicted emotional label $\hat{c}$ is then defined as: $\hat{c} = \arg\max_{i \in \{1,\ldots,n\}} p_i$: This predicted class is then compared against the ground truth label to compute performance metrics such as accuracy, precision, recall, and F1-score.

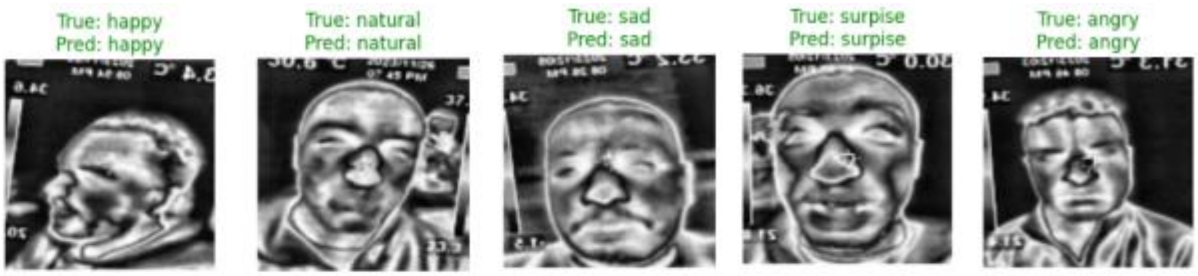

Figure 6.13: Predicted facial emotion sample images

6.5. Research Question and Answer Discussion

RQ1: What is the impact of applying Bilateral Filter and CLAHE on enhancing the quality of thermal images and their role in improving thermal facial emotion recognition accuracy?

To evaluate if bilateral filter and CLAHE preprocessing can enhance the quality of thermal images and improve the accuracy of emotion recognition the experiments were carried out training models with and without these preprocessing methods. Performance of the model was compared with accuracy, Precision, recall, F1-score, and confusion matrices. Results confirmed that bilateral filter was able to effectively remove noise from thermal facial images while still preserving informative edge information critical in perceiving subtle facial features of emotional expressions. Compared to previous methods that utilize the Difference of Gaussian filter, the bilateral filter produced clearer and more structurally consistent images, thereby improving the quality of features input to the deep learning model. Conversely, CLAHE enhanced local contrast by redistributing pixel intensity values, which served to bring out finer texture patterns in thermal images without over-amplifying noise. Collectively, these preprocessing methods improved the discriminative capability of the input data, enabling the model to learn richer features and achieve improved classification performance. This illustrates that the selection of preprocessing is an important factor in preparing thermal images for emotion recognition analysis.

RQ2: What is the impact of generated synthetic thermal facial images in mitigating data scarcity and improving the classification performance of the thermal facial emotion recognition?

To evaluate whether or not thermal facial data generated by cGAN could alleviate the insufficient data and upgrade accuracy of emotion recognition, experiments were conducted. The model was trained from original images as well as a combination of original and simulated images. The performance of both configurations were evaluated with accuracy, precision, recall, F1-score, and confusion matrices. Experimental results demonstrated that incorporating simulated thermal images assisted greatly with the issues of datasets and enhanced classification accuracy. Empirical results confirmed that the addition of synthetic data not only improved both training and validation accuracy but also reduced over fitting and improved classification performance across most evaluation metrics. These findings highlight the crucial role of synthetic data strategies in thermal-based emotion recognition, particularly when large-scale real-world datasets are unavailable.

RQ3: What is the impact of integrating EfficientNet with attention mechanisms in enhancing discriminative feature extraction and improving classification accuracy in thermal facial emotion recognition?

To evaluate the possibility of attention-based feature extraction enhancing classification accuracy relative to baseline models, experiments were conducted with a comparison of EfficientNet-B0 and proposed EfficientNet-B5 combined with the CBAM module. Accuracy, precision, recall, F1-score, and confusion matrices were used to evaluate the performance of both setting. Results showed that combining EfficientNet with CBAM improved feature extraction with a focus on informative face areas, resulting in higher recognition accuracy and higher precision, recall, and F1-scores of all emotion classes. The combination of EfficientNet-B5 with the Convolutional Block Attention Module demonstrated a clear enhancement in feature extraction and classification accuracy compared to the standalone use of EfficientNet. EfficientNet-B5 offered a powerful baseline owing to its compound scaling method, which achieves high performance using fewer parameters by balancing depth, width, and resolution. These results prove that attention mechanisms greatly enhance the discriminative power of deep learning models of thermal FER.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORKS

7.1. Conclusion

The research was conducted to overcome the shortcomings associated with conventional facial emotion recognition techniques. Such conventional techniques are commonly plagued by the variability of lighting conditions and the ease with which human subjects can hide or fake facial expressions. To circumvent these shortcomings, this research explored the utilization of thermal imaging a modality that is less susceptible to lighting variability coupled with state-of-the-art deep learning algorithms.

This study emphasized the benefit of using thermal images to recognize physiological patterns of emotion based on heat patterns. Thermal images are less susceptible to varying light conditions compared to visual images and provide a less obtrusive way of monitoring subtle changes in facial thermal distribution that are commonly linked to emotional states. The application of thermal images in emotional recognition is, however, presently quite limited primarily due to a lack of large-scale and varied datasets.

To mitigate datasets limitations, cGANs were employed to create synthetic thermal images conditioned on emotion labels. The synthetic data were instrumental in training a stable model, avoiding over fitting, and improving generalization on unseen data. The use of synthetic data not only improved model training but also provided robustness and reliability to emotion recognition. EfficientNet-B5 was fused with CBAM so that the model could attend to the most emotion-relevant areas of the face, while the ResNet-based classifier further purified these features using residual learning for robust classification.

The attention mechanism enabled the network to concentrate on facial regions of importance to emotion, resulting in a high recognition rate and better performance compared to contemporary models. This compound attention mechanism greatly improved the model's ability to distinguish between subtle thermal cues associated with different emotions.

Experimental findings verified that this combined strategy outperformed the baseline models substantially, with improved recognition accuracy and robustness. The research is significant in the domain of HCI since the technology promotes affective computing through improved ability of machines to recognize emotions in a broader range of situations.

7.2. Future Works

Although the present research has shown potential in advancing human facial emotion detection through thermal imaging, a lot is still available for research and development. Picking up where this research has left off, some potential areas of focus for future research are listed below.

The proposed framework introduced the primary emotions such as happiness, sadness, anger, neutrality, and surprise. Future research would involve incorporating secondary emotions such as fear, disgust, contempt, and confusion.

Although thermal imaging is good at recognizing emotions, its combination with other data like visible images, speech, physiological signals, or facial landmarks could enhance the performance.

This study employed one thermal facial datasets. For better generalization, future studies would be better off performing cross-datasets testing on varying conditions and populations.

As thermal FER systems are being used in sensitive domains like health and safety, interpretability is very crucial. The future research needs to employ XAI techniques to interpret the contribution of thermal features for emotion classification, enhancing trust and transparency.

SPECIAL ACKNOWLEDGMENT

This research work was funded by grant number ASTU/SM-R/1118/25 from Adama Science and Technology University.

REFERENCES

Agung, E. S., Rifai, A. P., & Wijayanto, T. (2024). Image-based facial emotion recognition using convolutional neural network on emognition dataset. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-65276-x

Aiswarya K. Prabhakaran, J. J. N. and S. S. (2021). *Thermal Facial Expression Recognition Using Modified ResNet152*. https://doi.org/https://doi.org/10.1007/978-981-33-6987-0_32

Assiri, B., & Hossain, M. A. (2023). Face emotion recognition based on infrared thermal imagery by applying machine learning and parallelism. *Mathematical Biosciences and Engineering*, *20*(1), 913–929. https://doi.org/10.3934/mbe.2023042

Bharat. (2023). *What is Computer Vision in 2024?* https://opencv.org/blog/what-is-computer-vision/

Bharat. (2024). *Deep Learning vs Machine Learning*. https://opencv.org/blog/deep-learning-vs-machine-learning/

Bhattacharyya, A., Chatterjee, S., Sen, S., Sinitca, A., Kaplun, D., & Sarkar, R. (2021). A deep learning model for classifying human facial expressions from infrared thermal images. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-99998-z

Chaitanya, S. S. M. P. and K. (2020). *Proceedings of the 2020 IEEE International Conference on Communication and Signal Processing (ICCSP) : 28th - 30th July 2020, Melmaruvathur, India*. IEEE.

Chirag Kumar Kyal, H. P. and M. R. (2021). *Human Emotion Recognition from Spontaneous Thermal Image Sequence Using GPU Accelerated Emotion Landmark Localization and Parallel Deep Emotion Net*. https://doi.org/https://doi.org/10.1007/978-981-15-5113-0_78

Dada, E. G., Oyewola, D. O., Joseph, S. B., Emebo, O., & Oluwagbemi, O. O. (2023a). Facial Emotion Recognition and Classification Using the Convolutional Neural Network-10 (CNN-10). *Applied Computational Intelligence and Soft Computing*, *2023*. https://doi.org/10.1155/2023/2457898

Dada, E. G., Oyewola, D. O., Joseph, S. B., Emebo, O., & Oluwagbemi, O. O. (2023b). Facial Emotion Recognition and Classification Using the Convolutional Neural Network-10 (CNN-10). *Applied Computational Intelligence and Soft Computing*, *2023*. https://doi.org/10.1155/2023/2457898

Elbarawy, Y. M., Ghali, N. I., & El-Sayed, R. S. (2019). Facial Expressions Recognition in Thermal Images based on Deep Learning Techniques. *International Journal of Image, Graphics and Signal Processing*, *11*(10), 1–7. https://doi.org/10.5815/ijigsp.2019.10.01

He, K. , Z. X. , R. S. , & S. J. (2016). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2016). *Densely Connected Convolutional Networks*. http://arxiv.org/abs/1608.06993

Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2017). *Squeeze-and-Excitation Networks*. http://arxiv.org/abs/1709.01507

Jacintha, V., Simon, J., Tamilarasu, S., Thamizhmani, R., Thanga yogesh, K., & Nagarajan, J. (2019). A Review on Facial Emotion Recognition Techniques. *2019 International Conference on Communication and Signal Processing (ICCSP)*, 517–521. https://doi.org/10.1109/ICCSP.2019.8698067

Krishna, A. H., Sri, A. B., Priyanka, K. Y. V. S., Taran, S., & Bajaj, V. (2019). Emotion classification using EEG signals based on tunable-Q wavelet transform. *IET Science, Measurement and Technology*, *13*(3), 375–380. https://doi.org/10.1049/iet-smt.2018.5237

Kundur, N. C., & Mallikarjuna, P. B. (n.d.). Insect Pest Image Detection and Classification using Deep Learning. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 13, Issue 9). www.ijacsa.thesai.org

Lin, Y., Sun, H., Liu, N., Bian, Y., Cen, J., & Zhou, H. (2022). *A lightweight multi-scale context network for salient object detection in optical remote sensing images*. http://arxiv.org/abs/2205.08959

Mirza, M., & Osindero, S. (2014). *Conditional Generative Adversarial Nets*.

Mohamed Fawzi, A. A. A. D. A. ella H. (2024). *Comprehensive Facial Thermal Dataset*.

Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing*, *452*, 48–62. https://doi.org/10.1016/j.neucom.2021.03.091

Prasad, S. B. R., & Chandana, B. S. (2023). Human Face Emotions Recognition from Thermal Images Using DenseNet. *International Journal of Electrical and Computer Engineering Systems*, *14*(2), 155–167. https://doi.org/10.32985/ijeces.14.2.5

Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. http://arxiv.org/abs/1905.11946

Thampi, S. M., Gelenbe, E., Atiquzzaman, M., Chaudhary, V., & Li Editors, K.-C. (2021). *Lecture Notes in Electrical Engineering 736* (Vol. 2). http://www.springer.com/series/7818

Wang, S., He, M., Gao, Z., He, S., & Ji, Q. (2014). Emotion recognition from thermal infrared images using deep Boltzmann machine. *Frontiers of Computer Science*, *8*(4), 609–618. https://doi.org/10.1007/s11704-014-3295-3

Wang, Y., Wang, W., Li, Y., Jia, Y., Xu, Y., Ling, Y., & Ma, J. (2024). An attention mechanism module with spatial perception and channel information interaction. *Complex and Intelligent Systems*, *10*(4), 5427–5444. https://doi.org/10.1007/s40747-024-01445-9

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). *CBAM: Convolutional Block Attention Module*. http://arxiv.org/abs/1807.06521

Woo, S., P. J. , L. J.-Y. , & K. I. S. (2018). CBAM: Convolutional Block Attention Module. *In Proceedings of the European Conference on Computer Vision (ECCV), 3–19.*

Wu, M., & Zhong, Q. (2024). Image Enhancement Algorithm Combining Histogram Equalization and Bilateral Filtering. *Systems and Soft Computing*, 200169. https://doi.org/https://doi.org/10.1016/j.sasc.2024.200169

Achour, G., Sung, W., Pinon Fischer, O., & Mavris, D. (2020, January 6). Development of a conditional generative adversarial network for airfoil shape optimization. *American Institute of Aeronautics and Astronautics*. https://doi.org/10.2514/6.2020-2261

APPENDIXES

Appendix 1: sample code

Sample code for importing necessary library

```
import cv2
import os
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import transforms
from torch.utils.data import DataLoader, Dataset
from torchvision.utils import save_image
import numpy as np
import matplotlib.pyplot as plt
import shutil
from sklearn.model_selection import train_test_split
from PIL import Image
from torchmetrics.image.fid import FrechetInceptionDistance
from skimage.metrics import structural_similarity as ssim
from torchvision.models.efficientnet import EfficientNet_B7_Weights
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, classification_report
import seaborn as sns
from torchvision import datasets, transforms, models
```

Sample code to check if GPU available

```
# Check if GPU is available
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")
```

Sample code for data split

```
def split_data_into_folders(input_dir, output_dir, test_size=0.1, val_size=0.1, random_state=42):
    """
    Splits the dataset into train, validation, and test sets while preserving class and subfolder structure.
    """
    # Define output folders
    train_dir = os.path.join(output_dir, "train")
    val_dir = os.path.join(output_dir, "val")
    test_dir = os.path.join(output_dir, "test")
    os.makedirs(train_dir, exist_ok=True)
    os.makedirs(val_dir, exist_ok=True)
    os.makedirs(test_dir, exist_ok=True)

    split_counts = {"train": {}, "val": {}, "test": {}}

    # Process each class and subfolder
    for root, _, files in os.walk(input_dir):
        # Get the relative path of the subfolder
        relative_class_path = os.path.relpath(root, input_dir)
        if len(files) == 0:
            continue  # Skip empty folders
```

```
    # Collect images
    images = [os.path.join(root, file) for file in files
              if file.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp'))]

    if not images:
        continue  # Skip if no images found

    # Split data
    train_images, test_images = train_test_split(images, test_size=test_size, random_state=random_state)
    train_images, val_images = train_test_split(train_images, test_size=val_size, random_state=random_state)

    # Store counts for logging
    split_counts["train"][relative_class_path] = len(train_images)
    split_counts["val"][relative_class_path] = len(val_images)
    split_counts["test"][relative_class_path] = len(test_images)

    # Copy images to the respective subfolders
    for image_path in train_images:
        save_image_to_folder(image_path, train_dir, relative_class_path)
```

```
        for image_path in val_images:
            save_image_to_folder(image_path, val_dir, relative_class_path)

        for image_path in test_images:
            save_image_to_folder(image_path, test_dir, relative_class_path)

    # Display split counts
    print("\nSplit counts:")
    for split, counts in split_counts.items():
        print(f"\n{split.capitalize()} Set:")
        for class_name, count in counts.items():
            print(f"  {class_name}: {count} images")
```

```
def save_image_to_folder(image_path, destination_dir, relative_path):
    """
    Copies an image to the destination folder while preserving the original subfolder structure.
    """
    target_dir = os.path.join(destination_dir, relative_path)  # Preserve subfolder path
    os.makedirs(target_dir, exist_ok=True)  # Ensure subfolder exists
    shutil.copy(image_path, os.path.join(target_dir, os.path.basename(image_path)))

# Example usage:
input_dataset_dir = '/kaggle/working/proutput3'  # Replace with your dataset path
output_dataset_dir = '/kaggle/working/prop3split'   # Replace with your output path

split_data_into_folders(input_dataset_dir, output_dataset_dir, test_size=0.1, val_size=0.1)
print("\nData split into train, val, and test folders successfully!")
```

Sample code for data transformation

```
data_transforms = {
    "train": transforms.Compose([
        transforms.RandomRotation(10),
        transforms.RandomHorizontalFlip(),
        transforms.RandomResizedCrop(224, scale=(0.8, 1.0)),  #
        transforms.ColorJitter(brightness=0.2, contrast=0.2),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.5], std=[0.5]),
    ]),
    "val": transforms.Compose([
        transforms.Resize((224, 224)),  # Match train size
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.5], std=[0.5]),
    ]),
    "test": transforms.Compose([
        transforms.Resize((224, 224)),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.5], std=[0.5]),
    ]),
}
```

Sample code for plot training curve

```
plt.plot(g_losses, label='Generator')
plt.plot(d_losses, label='Discriminator')
plt.title("Loss Over Epochs")
plt.legend()
plt.show()

plt.plot(ssim_values, label='SSIM', color='green')
plt.title("SSIM Over Epochs")
plt.grid()
plt.legend()
plt.show()

plt.plot(fid_values, label='FID', color='purple')
plt.title("FID Over Epochs")
plt.grid()
plt.legend()
plt.show()
```

```python
# === Plot Training History ===
def plot_training_history(train_loss, val_loss, train_acc, val_acc):
    epochs = range(1, len(train_loss) + 1)
    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    plt.plot(epochs, train_loss, label="Train Loss")
    plt.plot(epochs, val_loss, label="Val Loss")
    plt.title("Loss")
    plt.legend()
    plt.subplot(1, 2, 2)
    plt.plot(epochs, train_acc, label="Train Acc")
    plt.plot(epochs, val_acc, label="Val Acc")
    plt.title("Accuracy")
    plt.legend()
    plt.show()
```

```python
# Plot training & validation loss and accuracy
plot_training_history(train_loss, val_loss, train_acc, val_acc)
# === Confusion Matrix and Classification Report ===
class_names = original_train_dataset.classes  # Use ImageFolder's built-in class names
def plot_confusion_matrix(y_true, y_pred, class_names):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=class_names, yticklabels=class_names)
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title('Confusion Matrix')
    plt.show()
plot_confusion_matrix(y_true, y_pred, class_names)
print("\nClassification Report:\n", classification_report(y_true, y_pred, target_names=class_names))
```

Summarized model parameters

Layer Name	Shape	Params	Trainable
feature_extractor.efficientnet._conv_stem.weight	[48, 3, 3, 3]	1296	True
feature_extractor.efficientnet._bn0.weight	[48]	48	True
feature_extractor.efficientnet._bn0.bias	[48]	48	True
feature_extractor.efficientnet._blocks.0._depthwise_conv.weight	[48, 1, 3, 3]	432	True
feature_extractor.efficientnet._blocks.0._bn1.weight	[48]	48	True
feature_extractor.efficientnet._blocks.0._bn1.bias	[48]	48	True
feature_extractor.efficientnet._blocks.0._se_reduce.weight	[12, 48, 1, 1]	576	True
feature_extractor.efficientnet._blocks.0._se_reduce.bias	[12]	12	True
feature_extractor.efficientnet._blocks.0._se_expand.weight	[48, 12, 1, 1]	576	True
feature_extractor.efficientnet._blocks.0._se_expand.bias	[48]	48	True
feature_extractor.efficientnet._blocks.0._project_conv.weight	[24, 48, 1, 1]	1152	True
feature_extractor.efficientnet._blocks.0._bn2.weight	[24]	24	True
feature_extractor.efficientnet._blocks.0._bn2.bias	[24]	24	True

feature_extractor.cbam_7.channel_attention.1.weight	[2, 40, 1, 1]	80	True
feature_extractor.cbam_7.channel_attention.3.weight	[40, 2, 1, 1]	80	True
feature_extractor.cbam_7.spatial_attention.0.weight	[1, 2, 7, 7]	98	True
feature_extractor.cbam_19.channel_attention.1.weight	[8, 128, 1, 1]	1024	True
feature_extractor.cbam_19.channel_attention.3.weight	[128, 8, 1, 1]	1024	True
feature_extractor.cbam_19.spatial_attention.0.weight	[1, 2, 7, 7]	98	True
feature_extractor.cbam_35.channel_attention.1.weight	[19, 304, 1, 1]	5776	True
feature_extractor.cbam_35.channel_attention.3.weight	[304, 19, 1, 1]	5776	True
feature_extractor.cbam_35.spatial_attention.0.weight	[1, 2, 7, 7]	98	True
feature_extractor.cbam_38.channel_attention.1.weight	[32, 512, 1, 1]	16384	True
feature_extractor.cbam_38.channel_attention.3.weight	[512, 32, 1, 1]	16384	True
feature_extractor.cbam_38.spatial_attention.0.weight	[1, 2, 7, 7]	98	True

classifier.resnet18.layer4.0.bn2.weight	[512]	512	True
classifier.resnet18.layer4.0.bn2.bias	[512]	512	True
classifier.resnet18.layer4.0.downsample.0.weight	[512, 256, 1, 1]	131072	True
classifier.resnet18.layer4.0.downsample.1.weight	[512]	512	True
classifier.resnet18.layer4.0.downsample.1.bias	[512]	512	True
classifier.resnet18.layer4.1.conv1.weight	[512, 512, 3, 3]	2359296	True
classifier.resnet18.layer4.1.bn1.weight	[512]	512	True
classifier.resnet18.layer4.1.bn1.bias	[512]	512	True
classifier.resnet18.layer4.1.conv2.weight	[512, 512, 3, 3]	2359296	True
classifier.resnet18.layer4.1.bn2.weight	[512]	512	True
classifier.resnet18.layer4.1.bn2.bias	[512]	512	True
classifier.resnet18.fc.weight	[5, 512]	2560	True
classifier.resnet18.fc.bias	[5]	5	True

```
Total Parameters: 45,939,965
Trainable Parameters: 45,939,965
Non-trainable Parameters: 0
```

Sintayehu Hirpassa

finaleditedpl.pdf

Lung

Lung

Adama Science and Technology University

Document Details

Submission ID

trn:oid:::1:3352080578

Submission Date

Sep 26, 2025, 4:36 AM GMT

Download Date

Sep 26, 2025, 4:47 AM GMT

File Name

finaleditedpl.pdf

File Size

3.4 MB

85 Pages

19,996 Words

113,960 Characters

19% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **310 Not Cited or Quoted 17%**
  Matches with neither in-text citation nor quotation marks
- **30 Missing Quotations 2%**
  Matches that are still very similar to source material
- **0 Missing Citation 0%**
  Matches that have quotation marks, but no in-text citation
- **1 Cited and Quoted 0%**
  Matches with in-text citation present, but no quotation marks

Top Sources

- 11% Internet sources
- 14% Publications
- 7% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.