Faculty of Media Engineering and Technology
German University in Cairo

Facial Micro-Expression Analysis Using Deep Learning

Bachelor Thesis

by

Mohamed Shady Marzban

Supervised by

Prof. Dr. Gamal Abdel Shafy

May 2025

Declaration

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor of Science (B.Sc.) at the German University in Cairo (GUC),

(ii) due acknowledgement has been made in the text to all other material used

---

Mohamed Shady Marzban

May 2025

Acknowledgement

I would like to sincerely thank my supervisor, Dr. Gamal A. Ebrahim, for his continuous guidance, support, and insightful feedback throughout the course of this research. His expertise and commitment have been fundamental to the development of this work. I am also grateful to my colleagues and friends for their assistance, encouragement, and moral support, which made this journey both enriching and memorable. A special thanks goes to my family for their unwavering love, encouragement, and belief in my potential; their support has been the cornerstone of my academic achievements. This research would not have been possible without the collective efforts and contributions of everyone mentioned above. Thank you all for being part of this journey.

Contents

List of Figures

List of Tables

Abstract

A micro-expression is a subtle, brief, and involuntary facial movement that reveals a person's true emotions, often occurring when one attempts to conceal their feelings. These expressions are considered valuable cues in applications such as lie detection, security, and mental health assessment. Despite the growing interest in micro-expression recognition, the task remains highly challenging due to the fleeting nature and low intensity of these expressions.

Previous research conducted on micro-expression recognition using deep learning has employed a variety of models, such as Convolutional Neural Networks (CNNs), 3D-CNNs, and Long Short-Term Memory (LSTM) networks. However, a common limitation across these studies is the use of small-scale datasets, which are often insufficient for training models to effectively capture the brief and subtle characteristics of micro-expressions. Furthermore, the limited availability of these datasets continues to hinder progress in developing accurate and generalizable recognition models.

In this study, two deep learning models are introduced: an Auto-Keras based model and a custom CNN + Bidirectional LSTM model (BiLSTM). Two benchmark datasets are utilized, which are SAMM and CASME II datasets. The CNN+BiLSTM model is applied to the SAMM dataset, while the Auto-Keras model is applied to the CASME II dataset. Due to limitations in the CASME II dataset, specifically the lack of continuous frame sequences, the Auto-Keras model treats each frame as a static image and performs image classification rather than sequence learning. As a result, the model achieved a low overall accuracy of 36.49%, which is significantly lower than other models trained on the same dataset. In contrast, the custom CNN+BiLSTM model was designed to learn from the full temporal sequence, capturing the brief and subtle features of micro-expressions. This model outperformed both the AutoKeras model and several models trained and evaluated on the SAMM dataset, achieving an overall accuracy of 65.22%.

Chapter 1

Introduction

Facial micro-expressions are rapid and involuntary facial movements that can reveal a person's true emotions, often occurring in high-stakes situations where individuals attempt to conceal their feelings. Despite their significance in fields such as security, psychology, and human-computer interaction, the detection and classification of micro-expressions remains a challenging task due to their subtle intensity and extremely short duration. Traditional approaches struggle to achieve high accuracy due to the complexity of capturing these fleeting expressions. This project aims to explore deep learning methods to improve the automatic recognition of facial micro-expressions, contributing to the development of more effective and reliable emotion analysis systems.

1.1 Problem Statement

Understanding human emotions plays a crucial role in many areas, such as security (e.g., lie detection), healthcare (e.g., mental health monitoring), and technology (e.g., emotion-aware applications). While macro-expressions are relatively easy to detect and interpret, micro-expressions are brief, involuntary facial movements that last less than half a second and often reveal concealed emotions. These subtle expressions are difficult to identify manually, requiring trained experts and frame-by-frame analysis, which is both time-consuming and prone to human error. However, micro-expressions often provide deeper insights into a person's true emotions than macro-expressions, making them especially valuable for emotion analysis.

Despite the advancements in deep learning, recognizing micro-expressions remains a highly challenging task. The main obstacles include the limited availability and size of annotated micro-expression datasets and the subtlety and variability of the expressions themselves which require capturing both spatial (facial appearance) and temporal (motion over time) features together.

1.2 Motivation

Recognizing genuine human emotions through facial analysis plays an important role in advancing fields such as security, psychological assessment, healthcare, and human-computer interaction. Micro-expressions, despite their brief and subtle nature, provide critical insights into emotions that individuals may attempt to conceal. Accurately detecting these expressions can enhance the reliability of emotion-based applications and decision-support systems by offering deeper emotional understanding that supports better and more informed decisions.

The motivation behind this project stems from the growing demand for intelligent systems capable of interpreting human emotions more effectively. Traditional methods have struggled with the detection of micro-expressions due to the limited availability of training data and the inherent complexity of capturing such fleeting facial movements. By exploring deep learning techniques, this project aims to address these challenges and contribute to the development of more accurate, reliable, and adaptive emotion recognition technologies.

1.3 Objectives

The goal of this project is to develop a method for the automatic recognition of facial micro-expressions using deep learning techniques. The project aims to improve the accuracy and reliability of emotion classification by addressing challenges related to the

subtlety and short duration of micro-expressions, as well as the limited size of available datasets.

The specific objectives of this thesis are:

- Gain a better understanding of facial micro-expressions and their significance in emotion analysis.
- Collect, preprocess, and augment micro-expression data to improve training diversity.
- Explore and implement deep learning techniques for recognizing micro-expressions.
- Evaluate the model's performance using standard classification metrics such as accuracy, precision, recall, and F1-score.
- Compare the proposed models performance against each other and against existing models in the literature.

1.4 Thesis Organization

This thesis is organized into five chapters, each addressing a critical aspect of the study.

- Chapter 1 provides an introduction to the research topic, outlining the problem statement, motivation, project objectives, and the overall organization of the thesis.
- Chapter 2 presents the literature review. It introduces the key terminologies related to micro-expression recognition and deep learning, reviews traditional methods, and discusses recent methods using deep learning techniques.
- Chapter 3 describes the methodology adopted in this project. It discusses the datasets utilized, the frameworks and tools employed, and details of the proposed models architecture.

- Chapter 4 discusses the experimental results. It highlights challenges encountered in the datasets, describes the preprocessing techniques applied, presents the training and evaluation results, and provides a comparative analysis against existing methods and different models used in the paper.

- Chapter 5 concludes the thesis by summarizing the key findings and contributions, and suggesting directions for future research in micro-expression recognition.

Chapter 2

Background

Facial micro-expressions have gained significant attention as a subtle yet powerful means of understanding human emotions. These brief, involuntary facial movements often reveal concealed emotions, making them invaluable in fields such as psychology, security, and human-computer interaction. While traditional methods for detecting micro-expressions have relied on manual observation or algorithms to extract facial features, they face limitations in terms of accuracy and scalability. With the advent of deep learning, more sophisticated and automated approaches for micro-expression recognition have emerged, offering promising improvements in detection and classification.

Before exploring existing studies that have contributed to the detection of facial micro-expressions, it is essential to define and explain some key terms and concepts related to micro-expressions and deep learning techniques.

2.1 Facial Micro-Expressions

Facial expressions, including micro-expressions, are essential for humans to convey emotions and are closely tied to mental health and social behaviors [1]. Micro-expressions (MEs) are involuntary, fleeting, and subtle facial expressions that occur when individuals attempt to conceal or suppress their true feelings. These expressions often appear in high-stakes situations and are challenging to detect due to their brief duration and subtle nature. The distinction between facial expressions and micro-

expressions is illustrated in the figures below, Figure 2.1 and Figure 2.2 . Despite this, micro-expressions are a crucial source of information, as they provide essential clues about a person's genuine emotions [2]. Micro-expressions are based on specific facial muscle movements, which serve as observable cues for underlying emotions. To objectively analyze these movements, researchers have developed tools like the Facial Action Coding System (FACS), which provides a structured way to study and categorize facial expressions. Due to their subtle and fleeting nature, detecting micro-expressions requires specialized training or the use of advanced technological tools [1]. Their significance is evident in various potential applications, such as national security, clinical diagnosis, and interrogations, where understanding concealed emotions is critical [2].

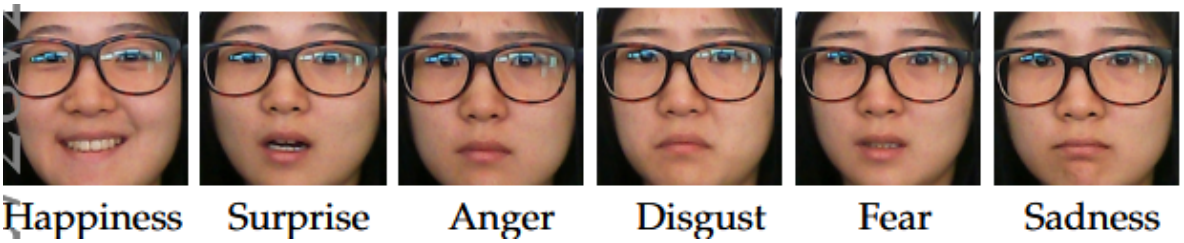

Figure 2.1: Facial-expressions [3]

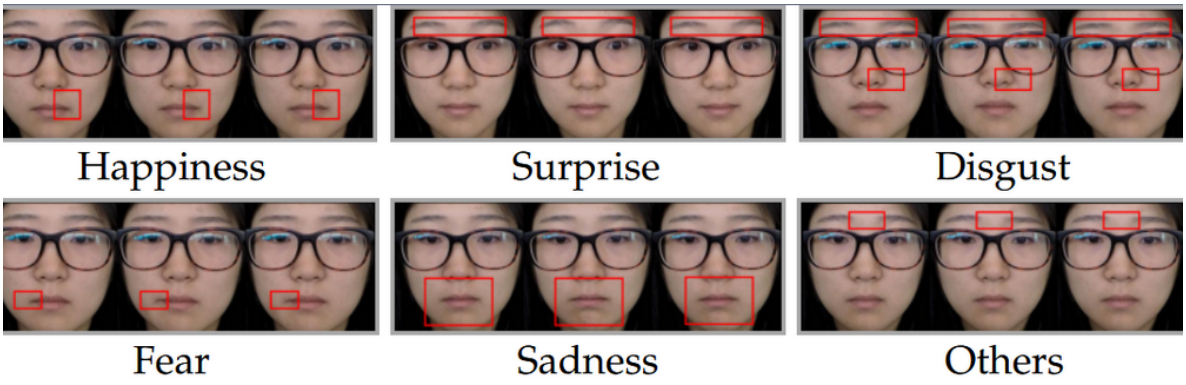

Figure 2.2: Micro-expressions [3]

2.1.1 Importance in Human Communication

Micro-expressions play a significant role in human communication, as they provide vital clues to an individual's emotional state. Their ability to uncover concealed emotions

has made them particularly useful in high-stakes situations, such as interrogations, clinical diagnoses, and security assessments. Understanding and analyzing MEs can help people communicate better by allowing them to recognize and respond to emotions that might not be openly expressed [2].

2.1.2 Challenges in Detecting Micro-Expressions

Detecting micro-expressions is inherently challenging due to their fleeting nature and subtle intensity. These expressions often last less than half a second, requiring advanced tools or trained professionals for accurate detection. Trained professionals must be familiar with facial muscle annotations, particularly the Action Unit (AU) coding system illustrated in Figure 2.3, which objectively categorizes facial muscle movements [1]. Additionally, factors such as cultural differences, facial variations, and the context in which the expressions occur add complexity to their analysis, emphasizing the need for automated approaches and robust detection techniques [2].

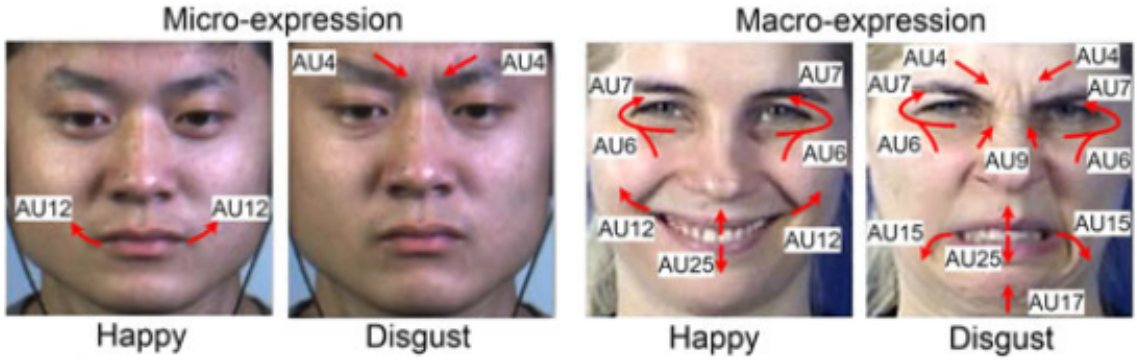

Figure 2.3: Action-Unit for labelling Facial Muscle movements [4]

2.2 Machine Learning

Machine learning is a branch of artificial intelligence that focuses on enabling computers to learn from data without being explicitly programmed. Instead of following predefined instructions, machine learning systems analyze patterns in data and improve

their performance over time. The primary goal is to build models that can make predictions or decisions when exposed to new information. This ability to generalize from past experiences makes machine learning valuable in various fields. There are three main methods of learning in machine learning: supervised learning, unsupervised learning, and reinforcement learning. Moreover , there is new method introduced which is Ensemble learning. Each method has a unique approach to training models, depending on the type of data available and the problem being solved [5].

2.2.1 Supervised Learning

Supervised learning is a type of machine learning where the system is trained using labeled data. Labeled data means that each data point in the training set consists of input features along with the correct output, serving as an example for the model to learn from. The model learns by identifying patterns in the data and mapping inputs to the correct outputs. Once trained, the model can make predictions on new, unseen data by applying the patterns it has learned [5].

2.2.2 Unsupervised Learning

Machine learning type where the system receives data without labels or categories. The goal is to find patterns in the data on its own. Unlike supervised learning, there is no correct answer provided, so the model does not get direct feedback. This method is used in tasks like grouping similar customers, detecting unusual activities, and recommending products. Common techniques include clustering, which groups similar items together, and association rule learning, which finds relationships between items [5].

2.2.3 Reinforcement Learning

Reinforcement learning is a way for a system (agent) to learn by trying different actions in an environment. It gets rewards for good actions and penalties for bad ones. Over time, it learns the best strategy to get the most rewards. [5].

2.2.4 Ensemble learrning

Ensemble learning is a machine learning technique that combines multiple models to improve overall accuracy and robustness. Instead of relying on a single model, ensemble methods aggregate predictions from multiple models (called base learners) to produce a more reliable outcome. This helps reduce variance, bias, and overfitting, leading to better generalization on unseen data. One popular ensemble method is stacking (stacked generalization), which takes a unique approach by combining different models using a meta-learner. In stacking, multiple base models (such as decision trees, or neural network) are trained independently on the same dataset. Their predictions are then used as input for a higher-level model (meta-learner), which learns how to best combine them for the final prediction. This technique is useful when different models capture different patterns in the data, making it a powerful tool for improving performance in complex machine learning tasks.

2.3 Neural Networks

Neural networks are computer models inspired by the human brain. They are made up of connected units, like brain cells (neurons), that process information. In the brain, neurons send signals through connections called synapses as shown in Figure 2.5. Similarly, artificial neurons take inputs, apply weights, sum them up, and decide an output as shown in Figure 2.4. Weights are numbers that determine how important an input is, much like how some brain connections are stronger than others. These networks consist of multiple layers where data is processed and passed forward, allowing them to recognize patterns, make predictions, and generalize from incomplete or noisy information. The network adjusts these weights during learning, improving its ability to recognize patterns and make decisions. This learning process is similar to how the brain adapts to new experiences, making neural networks useful for tasks like recognizing images, understanding language, and solving problems [6].

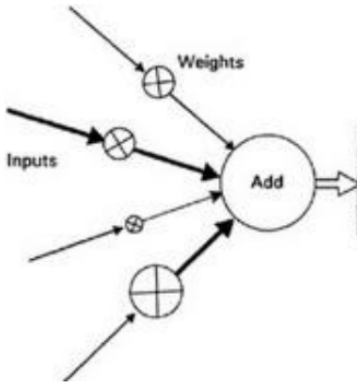

Figure 2.4: Artificial neuron [6]

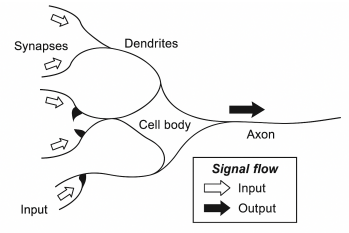

Figure 2.5: Brain neuron [6]

2.3.1 Neural-Network Layers

A neural network consists of three main layers: the input layer, the hidden layer, and the output layer, as illustrated in Figure 2.6. The input layer contains neurons that receive raw data and pass it to the next stage. This data is then processed in the hidden layer, where synapses with assigned weights manipulate the information before passing it forward. The hidden layer plays a crucial role in recognizing patterns and extracting features from the input. Finally, the processed data reaches the output layer, which generates the final result or prediction. The weights stored in the synapses determine how the input is transformed at each stage, allowing the network to learn and improve its accuracy [7].

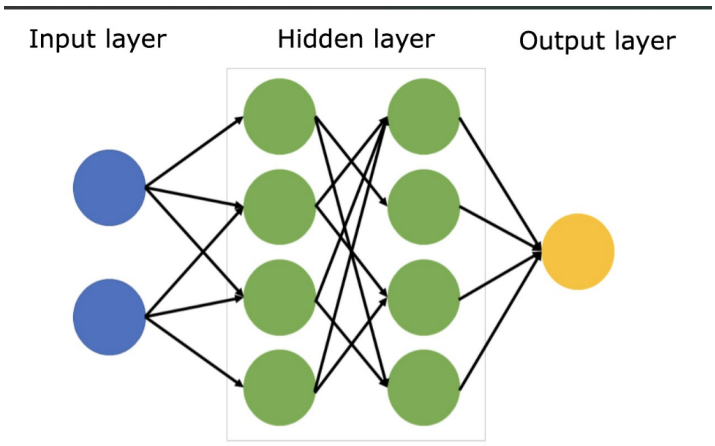

Figure 2.6: Whole Neural network

2.4 Deep Learning

Deep learning is a subset of machine learning that is inspired by how the human brain processes information. It uses artificial neural networks to analyze large amounts of data and learn patterns automatically. Deep learning models improve by stacking multiple layers of processing, where each layer extracts more complex features from the data. This process, called hierarchical feature learning, allows systems to recognize patterns with minimal human intervention. Unlike traditional machine learning models, deep learning performs better as more data is provided as shown in Figure 2.7. This scalability makes it highly effective in tasks like image recognition, sequence classification, natural language processing, and autonomous systems, often surpassing human abilities in certain areas [5].

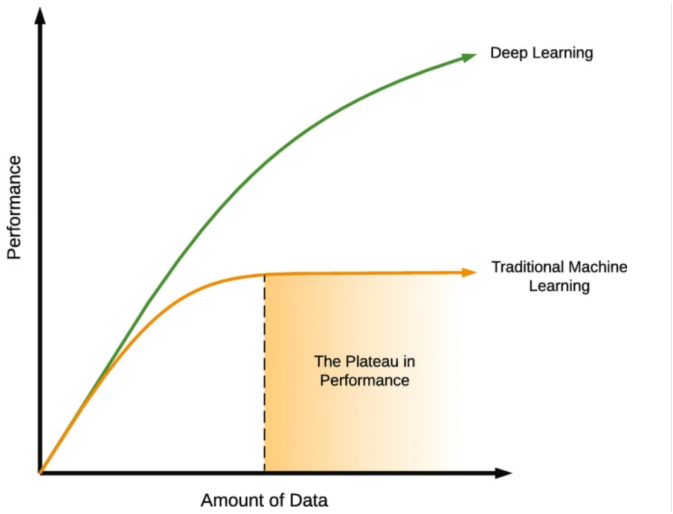

Figure 2.7: Machine learning vs Deep Learning [5]

2.4.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a deep learning model designed to analyze images by automatically detecting patterns and features. It consists of several key layers that work together to process visual data , as illustrated in Figure 2.8. First, the convolutional layer extracts features by applying small filters (small grids of numbers aka Kernal) that slide over the image, detecting edges and textures in early layers and complex patterns in deeper layers. Next, the ReLU activation function removes negative values with zero, improving efficiency. The pooling layer (downsampling) reduces the size of the image while keeping important information, with max pooling being the most common method, selecting the highest value in a region to make computations more efficient and less sensitive to slight changes. The fully connected layer flattens the data into a single vector and connects it to a traditional neural network, making final predictions such as classifying whether an image is a dog or a cat. Finally, the softmax or sigmoid activation function converts the output into probabilities to determine the final classification. CNNs are powerful because they learn features automatically, reduce computational complexity, and generalize well across different images, making them widely used in fields like facial recognition, medical imaging, and self-driving cars.

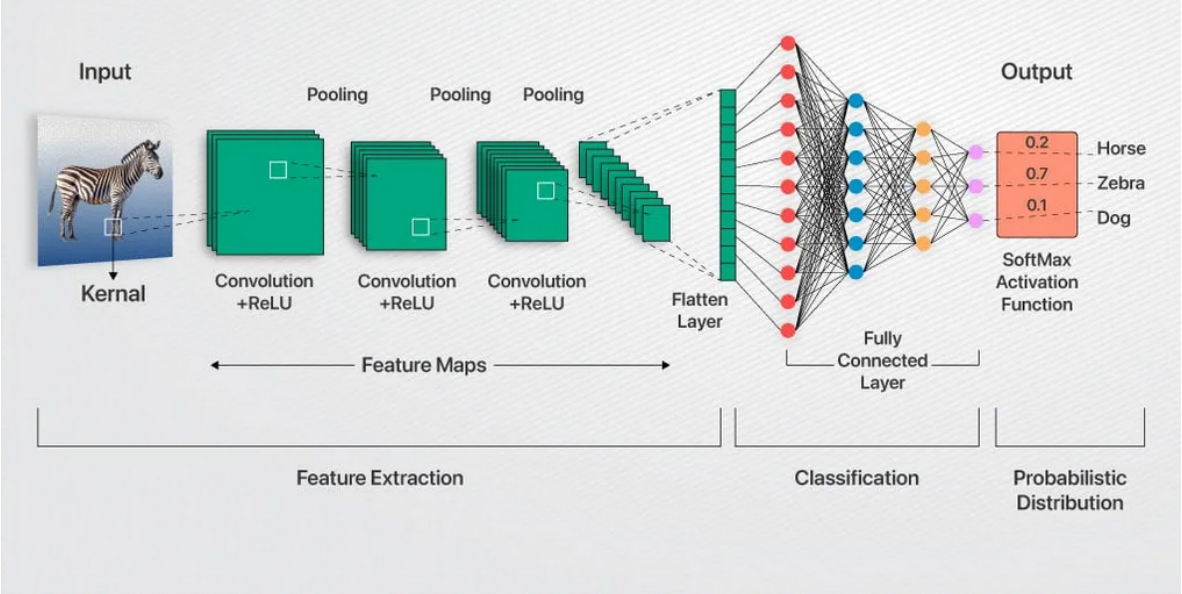

Figure 2.8: CNN (Convolutional Neural Network)

2.4.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of Deep Learning (DL) model designed for processing sequential data, such as time series, speech, and text. Unlike traditional feedforward neural networks, RNNs have connections that allow information to persist, enabling them to maintain a form of memory across time steps , as shown in Figure 2.9. This is achieved through recurrent connections, where the output of a previous step is fed as input into the next step, allowing the network to capture temporal dependencies. However, standard RNNs suffer from issues like vanishing and exploding gradients, making it difficult to learn long-term dependencies [8]. To address this, advanced variants like Long Short-Term Memory (LSTM) have been developed to solve this problem [9].

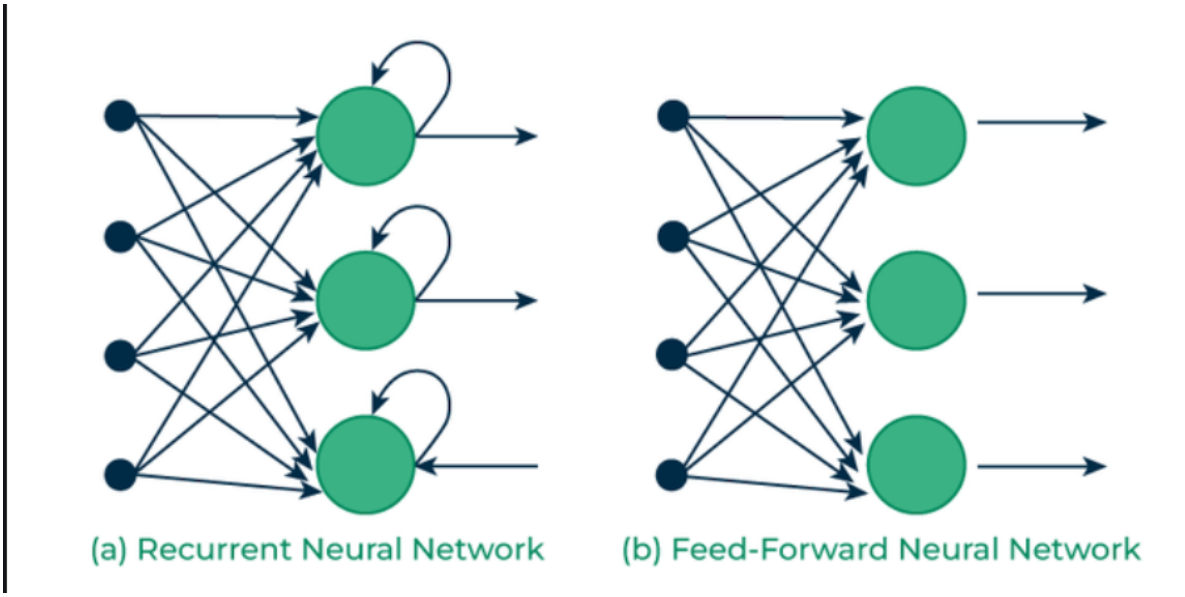

Figure 2.9: Recurrent Vs Feedforward networks

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN) designed to effectively model sequential data by capturing long-term dependencies. Traditional RNNs often suffer from the vanishing gradient problem, which limits their ability to learn patterns over extended sequences. LSTMs address this limitation through a gated architecture composed of forget, input, and output gates, which act as decision-making units that control the flow of information. Specifically, the forget gate decides which past information to discard, the input gate determines what new information to store, and the output gate regulates what information is passed to the next time step. These mechanisms allow LSTMs to maintain a stable memory state over time, enabling them to retain relevant context from earlier in the sequence. As a result, LSTMs are particularly well-suited for tasks involving temporal dynamics, such as speech recognition, language modeling, and facial micro-expression recognition, where contextual information across time steps is critical for accurate prediction.

Figure 2.10 shows a LSTM network applied to a named entity recognition (NER) task. Each word in the input sequence is processed sequentially from left to right, allowing

the model to learn dependencies based only on past context. The LSTM outputs are then used to assign a label to each word, such as `B-ORG` (beginning of an organization) for "EU" and `B-MISC` (beginning of a miscellaneous entity) for "German," while other words like "rejects" and "call" are tagged as `O` (outside any entity).

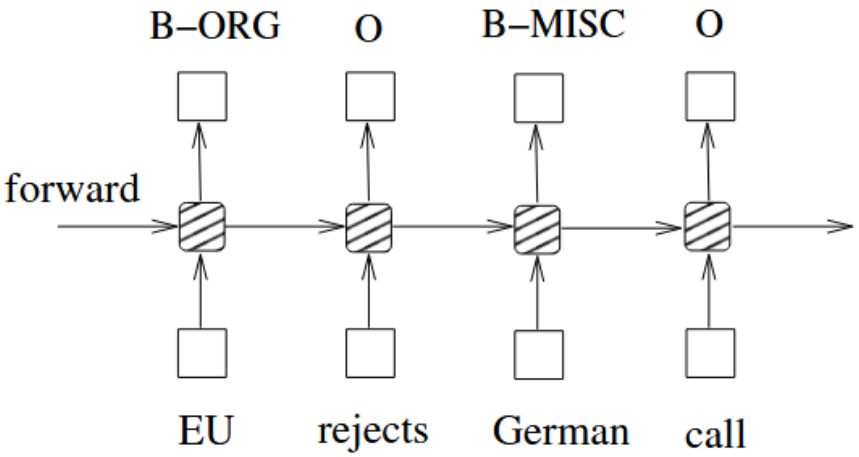

Figure 2.10: LSTM network [10].

Bidirectional LSTM

Bidirectional Long Short-Term Memory (BiLSTM) networks extend the capabilities of standard LSTMs by processing sequence data in both forward and backward directions. While traditional LSTMs capture dependencies from past to future, BiLSTMs utilize two parallel LSTM layers: one that reads the sequence from start to end, and another that reads it from end to start. This dual perspective allows the model to access both preceding and succeeding context at each time step, enabling more comprehensive understanding of the input sequence. In tasks such as facial micro-expression recognition, where subtle cues may depend on both prior and upcoming frames, BiLSTMs offer a significant performance advantage by modeling the full temporal context more effectively.

Figure 2.11 illustrates a Bidirectional LSTM (BiLSTM) network applied to a named entity recognition (NER) task. Each word in the input sequence is processed in both

forward and backward directions using two LSTM layers. This bidirectional processing allows the model to incorporate both past and future context when predicting the tag for each word. For example, "EU" is tagged as `B-ORG` (beginning of an organization), and "German" as `B-MISC` (beginning of a miscellaneous entity), while non-entity words like "rejects" and "call" are tagged as `O` (outside any entity).

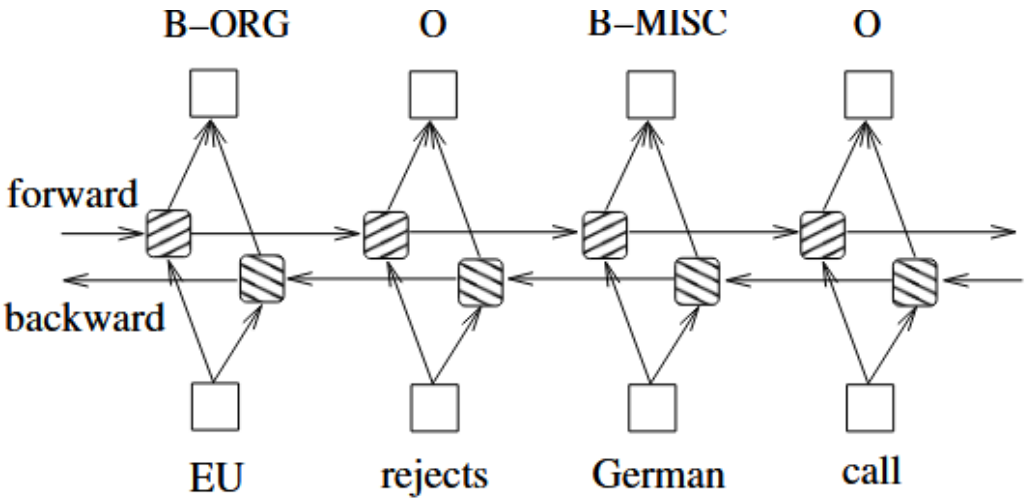

Figure 2.11: Bidirectional LSTM network [10].

2.5 Literature Survey

In this literature survey, we will explore the evolution of facial micro-expression recognition, focusing on both traditional and modern approaches. Initially, traditional methods relied on handcrafted feature extraction and classical machine learning techniques, which had limitations in accuracy and efficiency. However, with the rise of deep learning, more advanced techniques have emerged, enabling automated and more precise micro-expression detection.

2.5.1 Traditional Methods

Micro-Expression Recognition (MER) has historically relied on traditional methods that emphasized manual analysis and human expertise. These approaches laid the foundation for understanding subtle facial expressions by introducing structured frameworks

and training tools. Despite their contributions to advancing emotion recognition, these methods come with significant limitations, especially in terms of scalability and automation, which have driven the need for more sophisticated techniques.

Facial Action Coding System (FACS)

Early methods for Micro-Expression Recognition (MER) primarily relied on handcrafted features and manual analysis techniques. One of the foundational approaches was the Facial Action Coding System (FACS), developed by Ekman et al., which decomposed facial expressions into individual muscle movements known as Action Units (AUs). This system helps resolve expression ambiguity and improves Facial Expression Recognition (FER) performance. FACS plays a crucial role in Micro-Expression Recognition (MER) by providing a structured way to represent facial expressions [4]. This system allows professionals to analyze facial movements manually and has been widely used in emotion recognition research. However, recognizing micro-expressions through FACS requires trained professionals, making it challenging for large-scale automatic recognition [11].

Micro Expression Training Tool (METT)

The Micro Expression Training Tool (METT) helps people become more aware of emotions by improving their ability to recognize micro-expressions. By training individuals to spot subtle facial movements, METT enhances manual detection of micro-expressions, which is useful in areas like psychology, security, and human-computer interaction. This improved detection also helps researchers build better micro-expression datasets, leading to more accurate labeling and advancements in Micro-Expression Recognition (MER) systems [4].

Drawbacks of Traditional Methods

FACS is slow and labor-intensive, as micro-expressions are hard to detect, even for trained professionals. It is also prone to human error, as these fleeting expressions (1/25s to 1/2s) are difficult to capture consistently. Moreover, FACS struggles with recognition, unlike deep learning methods that excel at detecting subtle facial movements. Similarly, METT, while useful for training, relies on human interpretation, making it subjective, less reliable for real-time analysis and more prone to human error. Deep learning overcomes these limitations by automating detection and improving accuracy in micro-expression recognition.

2.5.2 Deep Learning Methods

In this section, several deep learning approaches for micro-expression recognition will be discussed and reviewed, highlighting the importance of advanced techniques in improving recognition performance and potentially replacing traditional methods.

Deep Local-Holistic Network

Facial micro-expression recognition is challenging since deep learning models struggle to learn important features from small datasets. To address this, researchers introduced the Deep Local-Holistic Network (DLHN), inspired by the human cognitive process of recognizing micro-expressions starting with a broad perception, shifting to detailed analysis, and concluding with a final decision. This process is illustrated in Figure 2.12. DLHN consists of two key components: the Hierarchical Convolutional Recurrent Network (HCRNN) and the Robust Principal Component Analysis Recurrent Network (RPRNN).

- Hierarchical Convolutional Recurrent Network (HCRNN)
  - Captures fine-grained facial movements.
  - Uses a convolutional neural network (CNN) to extract spatial features.

  - Employs a recurrent neural network (RNN) to analyze temporal changes in micro-expression video frames.

- Robust Principal Component Analysis Recurrent Network (RPRNN)
  - Enhances recognition by applying Robust Principal Component Analysis (RPCA) which is a technique that separates essential features from noise by decomposing the data into:
    * A clean, structured component.
    * A sparse, noisy component.
  - This process removes irrelevant variations, such as lighting changes or background distractions, allowing the model to focus on key micro-expression patterns.

The two networks are trained separately, allowing them to specialize in different aspects of facial feature learning, and their outputs are then fused to improve recognition accuracy. By learning both fine details and overall patterns, DLHN enhances micro-expression recognition, making it more effective even with limited training data [11].

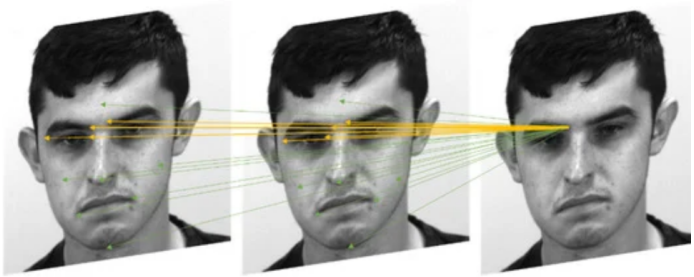

Figure 2.12: Global (green clipping head) and local area of interest (yellow arrow) tracking of micro-expression action [11].

3D CNN

A 3D CNN model is a model which directly recognize micro-expressions from video sequences while capturing both spatial details and motion over time. Unlike traditional

CNNs, which only extract spatial features, this model enables spatio-temporal feature extraction, allowing it to analyze both facial expressions and their changes across frames. Existing methods face challenges, as standard CNNs lack temporal awareness. Additionally, combining CNNs with regular LSTMs for spatial and temporal analysis separately does not allow for simultaneous feature extraction. To overcome these limitations, the 3D CNN model integrates spatial and temporal information effectively and employs an advanced pre-processing technique that selects the Apex frame sequence, ensuring the most expressive moments are analyzed, leading to improved classification accuracy [12].

Multi-Stream Deep Convolution Neural Network With Ensemble Learning

Overfitting is a major challenge in facial micro-expression recognition, especially due to the limited size of available datasets and the complexity of deep learning models. When a model is trained on a small dataset, it may memorize specific patterns instead of learning generalizable features, leading to poor performance on new data. To address this issue, a multi-stream(process data through more than one path simultaneously) deep learning model with ensemble classification has been introduced. The model layers and overall architecture are shown in Figure 2.13. The approach consists of three main steps:

- Feature Extraction: Features are extracted from multiple deep learning models to capture diverse and informative representations of micro-expressions.
  - ResNet: A deep learning model that addresses the vanishing gradient problem, where extremely small gradient values make it difficult for deep networks to learn. The gradient indicates how much the network should adjust its weights to improve predictions. ResNet introduces skip connections, which allow information to bypass certain layers, enabling effective training even in very deep networks.

  - DenseNet: Connects each layer to every other layer to improve gradient flow and feature reuse, enhancing learning efficiency.
  - VGG: A deep learning model that uses many simple layers with small filters to capture fine details in facial expressions. Its uniform design helps the model learn important features effectively.

- Dimensionality Reduction: High-dimensional deep features are computationally expensive, so a reduction technique is applied.
  - Principal Component Analysis (PCA): Reduces feature dimensions while preserving essential information, making the model more efficient.

- Ensemble Classification: A stacking ensemble technique is employed to improve classification accuracy.
  - Base Learners:
    * Random Tree: Constructs a decision tree using a subset of features, enabling fast and efficient predictions.
    * J48: A decision tree algorithm that selects key features to improve classification accuracy.
    * Random Forest: Aggregates multiple decision trees to enhance robustness and reduce overfitting.
  - Meta Learner:
    * A Random Forest model aggregates predictions from the base learners, enhancing accuracy and robustness.

This approach effectively mitigates overfitting, improves generalization, and enhances computational efficiency, making it well-suited for real-world facial micro-expression recognition applications [13].

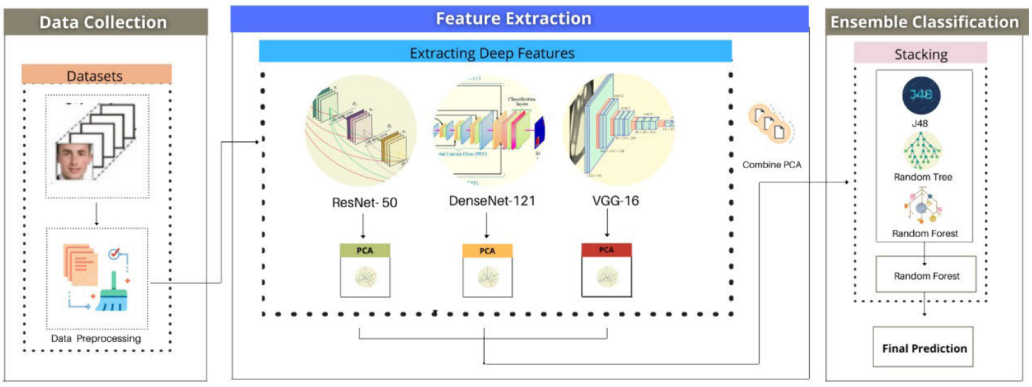

Figure 2.13: Multi- Stream Deep CNN Ensemble Model Architecture [13]

Modified Multimodal Ensemble Learning Approach

The Facial Micro-Expression Detection and Classification using Multi-Stream Deep Convolutional Neural Network with Ensemble Learning (FMEDC-MMEL) approach is designed to address the challenges of detecting and classifying subtle and brief facial micro-expressions, a task where most existing micro-expression recognition (MER) algorithms fall short. Traditional methods rely on shallow learning techniques, which use simpler machine learning models heavily dependent on manually extracted features. These methods struggle to capture complex patterns, particularly in high-dimensional data, leading to poor performance when dealing with the subtle and fleeting nature of micro-expressions under real-world conditions.

To overcome these limitations, the proposed FMEDC-MMEL approach uses a Multimodal Ensemble Learning Method. By combining data from multiple sources, such as visual, thermal, and physiological signals, and leveraging the powerful feature-learning abilities of deep neural networks, FMEDC-MMEL is expected to significantly outperform existing methods in accuracy and robustness.

The proposed approach consists of three main steps: pre-processing, feature extraction, and classification.

- Pre-Processing: The process begins with a pre-processing phase where Histogram Equalization (HE) is applied to enhance the contrast of facial images. This step ensures that the subtle micro-expressions are more distinct and easier to analyze in subsequent stages.

- Feature Extraction: Improved Densely Connected Networks (DenseNet) are employed for extracting meaningful features from the pre-processed images. DenseNet connects each layer to every other layer to promote feature reuse and efficient gradient flow, making it particularly effective for capturing the subtle and detailed patterns of micro-expressions. Additionally, the model's performance is optimized using the Stochastic Gradient Descent (SGD) algorithm(an optimization algorithm), which fine-tunes hyperparameters(settings of a modal) for enhanced learning.

- Classification: The final step involves a robust ensemble learning framework comprising three classifiers: Long Short-Term Memory (LSTM), Bi-Directional Gated Recurrent Unit (Bi-GRU), and Extreme Learning Machine (ELM). This ensemble method combines the strengths of each classifier to achieve higher accuracy and robustness in detecting and classifying subtle micro-expressions.

  - Long Short-Term Memory (LSTM): LSTM is a type of recurrent neural network (RNN) designed to remember important information over long sequences, making it effective for capturing temporal dependencies in micro-expression sequences.

  - Bi-Directional Gated Recurrent Unit (Bi-GRU): Bi-GRU is an advanced RNN variant that processes data in both forward and backward directions, improving the model's ability to understand context in sequential data like micro-expressions.

  - Extreme Learning Machine (ELM): ELM is a simple and fast machine learning algorithm that excels in handling small datasets by learning quickly with

minimal tuning, making it a good fit for micro-expression classification tasks.

By integrating these steps, the FMEDC-MMEL approach effectively overcomes the limitations of traditional methods, providing a comprehensive solution for micro-expression recognition with improved accuracy and reliability [14].

Color-Based Micro-Expression Recognition Using LSTM

In contrast to traditional approaches that focus on detecting subtle facial muscle movements, this method leverages facial skin color changes caused by variations in blood flow during emotional responses as the primary cue for recognizing micro-expressions. Since these color changes are less voluntary and more difficult to suppress, they offer a potentially more reliable signal for detecting genuine emotions.

The process begins by extracting a sequence of facial frames from micro-expression video clips. Each frame is then decomposed into three perceptual color channels: luminance (grayscale), red–green, and yellow–blue. This breakdown helps reveal very small changes in skin color that are hard to notice using regular RGB images.

Next, facial landmarks are detected to identify and consistently track key facial regions across the entire video sequence. Color values from these localized regions are extracted and assembled into temporal sequences that capture how skin tone evolves over time.

These sequences are then used as input to a Long Short-Term Memory (LSTM) network . The LSTM processes the color change dynamics across frames, learning the temporal dependencies associated with different emotion types. A dense layer followed by a softmax activation is applied to the LSTM output to produce the final emotion classification.

The following approach is clearly illustrated in Figure 2.14. By relying solely on facial color variations and discarding motion-related features, this approach demonstrates competitive and in some cases superior performance compared to movement-based methods, particularly in scenarios where facial motion is minimal or intentionally suppressed [15].

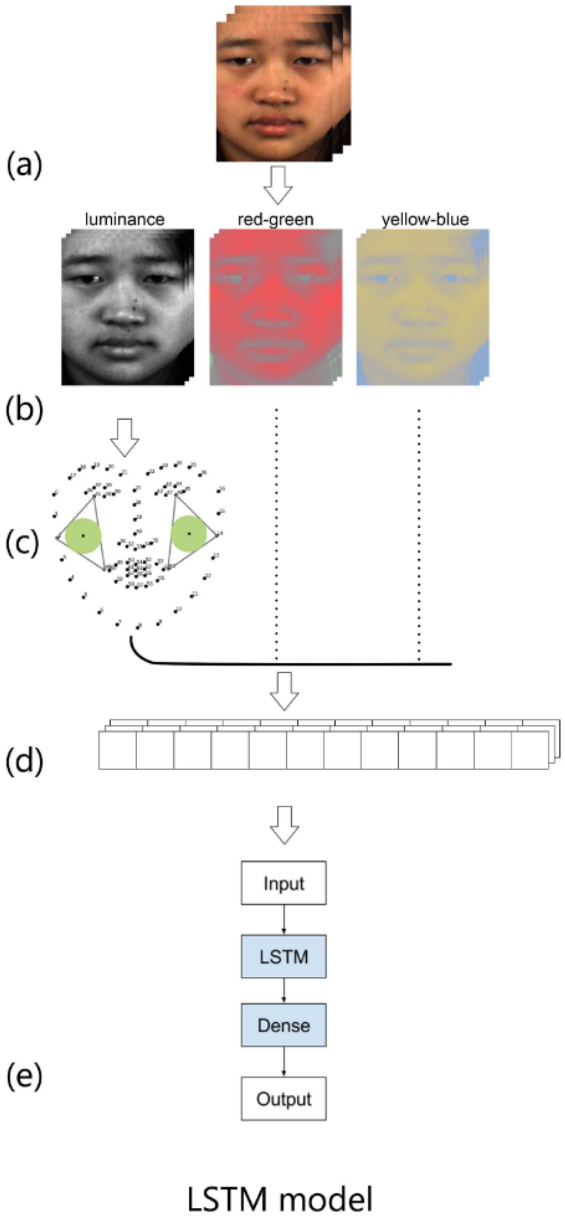

Figure 2.14: Color-Based Micro-Expression Recognition diagram [15]

Gender-aware Micro-Expression recognition

GEME (Gender-aware Micro-Expression) is a deep learning method that introduces a dual-stream multi-task learning (MTL) framework for facial micro-expression recognition. The model is designed to capture both micro-expression features and gender-specific features by training on two related tasks simultaneously: emotion classification and gender recognition.

The architecture consists of two separate convolutional streams, as Figure 2.15 shows:

- The first stream learns features specific to gender.
- The second stream learns features related to micro-expressions.

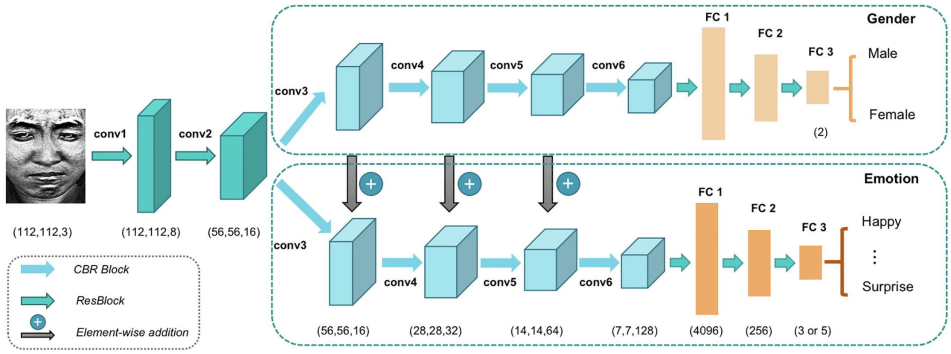

Figure 2.15: Gender-aware Micro-Expression recognition diagram [16]

The outputs of both streams are then fused to form a joint representation that captures both emotional content and gender-based expressiveness. This fusion enables the model to better account for interpersonal differences in emotional expression, particularly across gender.

To address the issue of class imbalance, GEME employs the Focal Loss function, which down-weights well-classified examples and emphasizes minority-class samples, helping the model to focus on more challenging cases.

The GEME framework was evaluated under two different training settings:

- In the single-task setting, the model was trained to perform only micro-expression recognition.
- In the multi-task setting, the model was trained to jointly recognize both gender and micro-expressions.

Experimental results showed that the multi-task version outperformed the single-task model, highlighting the effectiveness of incorporating gender-awareness as an auxiliary task to improve micro-expression recognition performance especially on datasets with imbalanced emotion distributions [16].

CapsuleNet for Micro-Expression Recognition

Traditional Convolutional Neural Networks (CNNs) are widely used in image classification tasks, including facial expression recognition. However, they have a key limitation: they focus on detecting features (like eyes or mouth), but often lose information about how these features are arranged relative to one another. This is especially problematic for micro-expression recognition, where understanding the exact positioning and movement of facial muscles is critical. Capsule Networks (CapsuleNet) were introduced to solve this problem. Instead of just detecting features, CapsuleNet also captures how these features are positioned and connected, helping the model understand the full structure of a facial expression. This makes CapsuleNet more capable of recognizing subtle and brief expressions, even when the data is limited or when there are slight changes in pose or orientation.

In the method proposed by [17], CapsuleNet is applied in a simple and effective pipeline for classifying micro-expressions. The approach includes the following steps, as shown in Figure 2.16:

- Apex Frame Extraction: From each micro-expression video, only the apex frame, the moment showing the peak of the expression, is selected to reduce complexity and focus on the most informative content.

- Face Cropping: The facial region is detected and cropped from the apex frame, removing background elements and emphasizing the expressive area of the face.
- CapsuleNet Classification: The cropped face image is passed into the Capsule Network, which analyzes both the presence of facial features and their spatial arrangement. This allows the model to better interpret subtle facial movements characteristic of micro-expressions.
- Emotion Prediction: Finally, the model classifies the expression category.

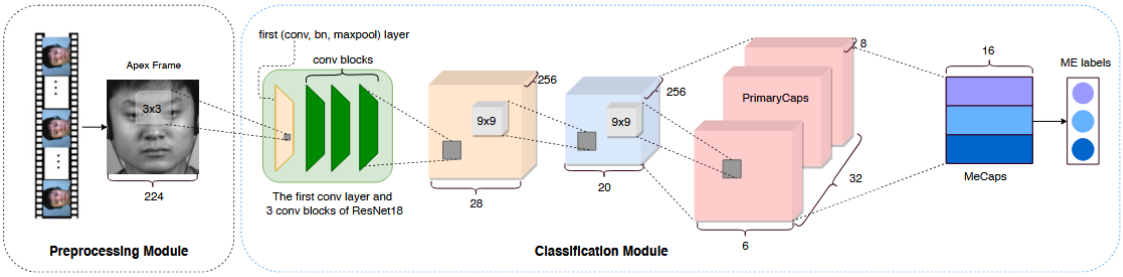

Figure 2.16: Complete framework for micro-expression recognition using CapsuleNet architecture [17]

Chapter 3

Methodology

This chapter outlines the methodology followed in the study, focusing on three core components. First, the benchmark datasets and the specific datasets utilized in the experiment are introduced. Second, the development frameworks and tools used for model training and evaluation are presented. Third, a comprehensive, layer-by-layer description of the two proposed models is provided.

3.1 Datasets

The primary challenge in facial micro-expression recognition is the limited availability of diverse and comprehensive datasets. To address this, several datasets have been developed, including CASME, CASME II, CAS(ME)^2, and SAMM. These datasets capture high-speed recordings of spontaneous micro-expressions, providing valuable resources for research. Their contributions have significantly advanced micro-expression analysis and improved recognition systems.

3.1.1 CASME

The CASME dataset consists of 1500 facial movements recorded at 60 frames per second (fps), from which 195 genuine micro-expressions were carefully selected. Figure 3.1 shows a sample of the dataset. Each micro-expression was annotated by tagging three key frames: the first frame marking the onset, the peak frame indicating the highest intensity, and the last frame capturing the offset of the expression [18].

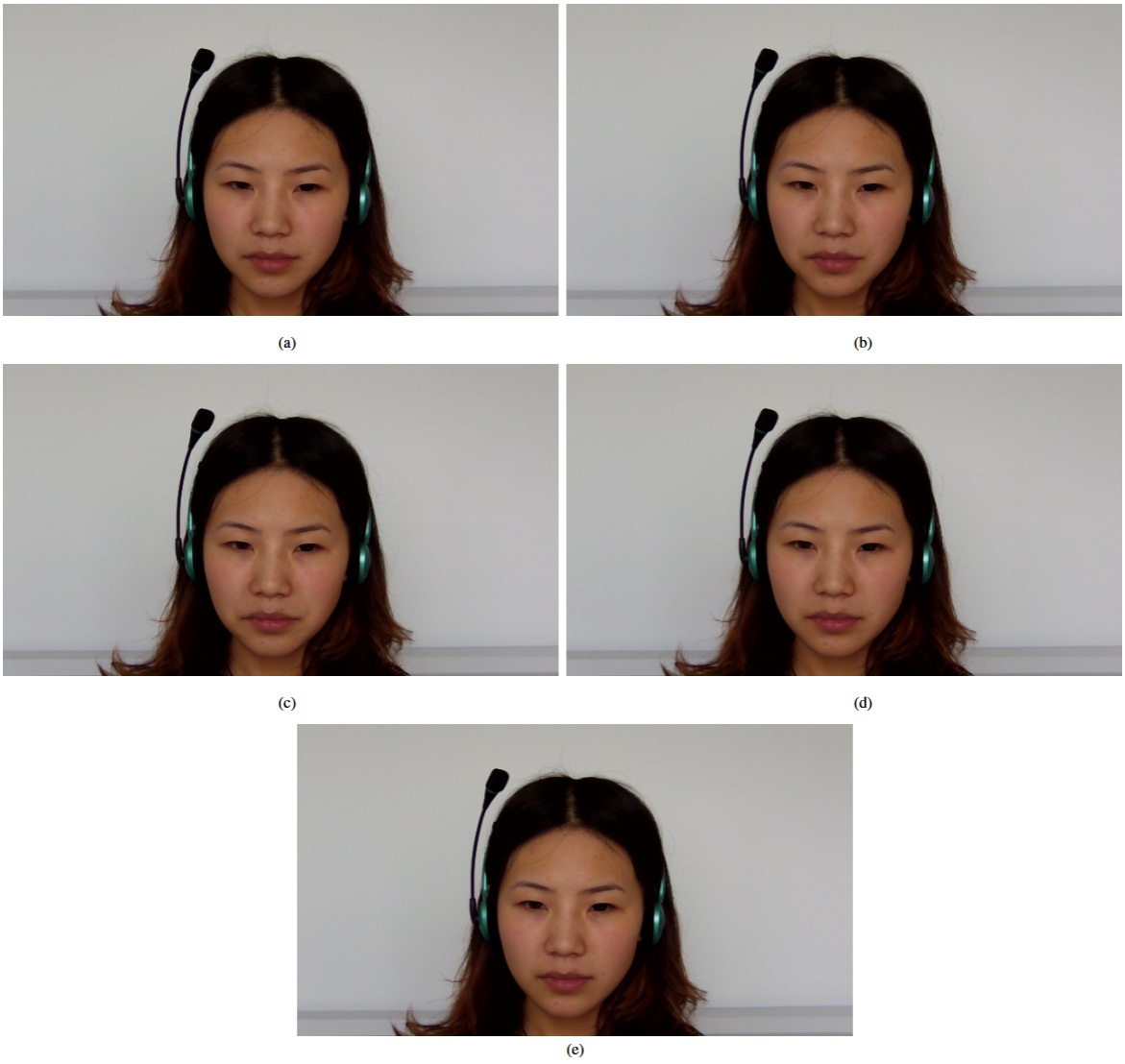

Figure 3.1: Sample from CASME dataset [18]

3.1.2 CASME II

CASME II was created to enhance micro-expression recognition for research in security, psychology, and medical fields. It was recorded in a controlled environment with consistent lighting and conditions to ensure accuracy, as illustrated in Figure 3.2. The dataset includes 247 spontaneous micro-expressions, carefully selected from nearly 3000 facial movements. To capture fine details, videos were recorded at 200 frames per second (fps) with high resolution [19].

Figure 3.2: Acquisition setup for elicitation and recording of micro-expressions.[19]

3.1.3 CAS(ME)²

The CAS(ME)² database is the first publicly available dataset that includes both macro-expressions and micro-expressions in long video recordings. Figure 3.3 shows a sample of both macro-expressions and micro-expressions from the dataset. All expressions were collected from the same participants under identical experimental conditions, allowing researchers to develop more effective feature extraction techniques to distinguish between macro-expressions and micro-expressions. The database integrates AU annotations, emotional classifications from elicitation videos, and participants' self-reported emotions for each expression sample. By having participants review their recorded expressions and provide self-reports, the dataset ensures high-quality emotion labeling while filtering out irrelevant facial movements, making it a valuable resource for emotion recognition research [20].

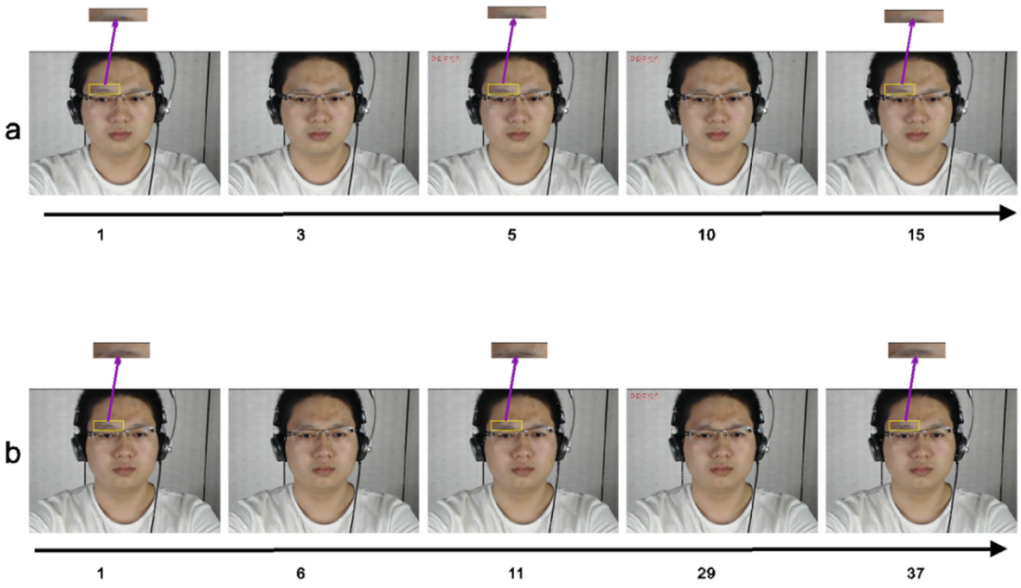

Figure 3.3: Micro(a) and Macro(b) Expression samples from the CAS(ME)^2 dataset [20]

3.1.4 SAMM

The SAMM dataset is a high-resolution micro-expression dataset designed for spontaneous facial movement analysis. It includes a diverse range of participants from different backgrounds and age groups, making it more representative of real-world scenarios. Figure 3.4 shows a sample from the dataset. The dataset is recorded at 200 frames per second (fps) and is coded using the Facial Action Coding System (FACS) to provide detailed annotations of facial muscle movements. Unlike previous datasets, where every participant watched the same videos or images to trigger emotions, this dataset takes a different approach. Instead of using uniform stimuli for all, it selects videos or images based on what genuinely triggers an emotional response in each individual. This results in more natural and intense emotional reactions, leading to higher-quality and more accurate micro-expression data [21].

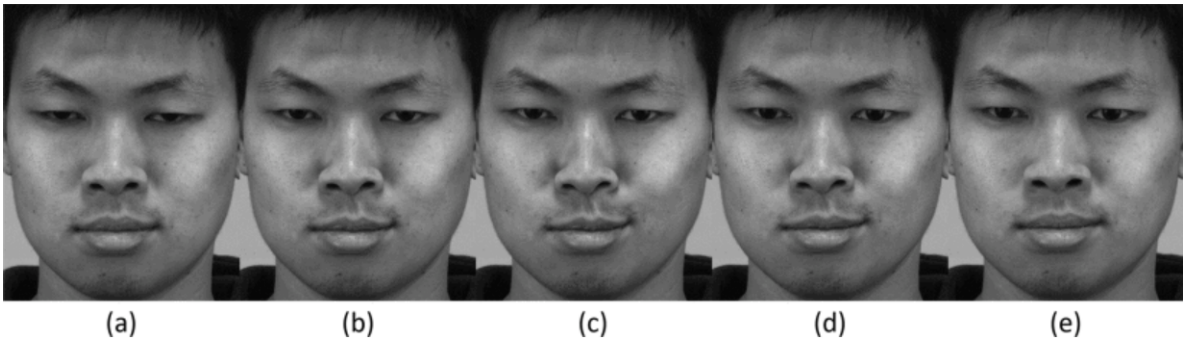

Figure 3.4: sample from the SAMM dataset [21]

3.1.5 Utilized Datasets for Micro-Expression Analysis

In this study, the SAMM and a pre-processed version of CASME II dataset were utilized, both obtained from Kaggle. These two datasets were selected because they are among the most commonly used benchmarks in micro-expression recognition research. While CASME II served as a strong starting point due to its popularity, the specific version used in this work presented two challenges: class imbalance and the absence of frame-level annotations for expression phases. To address the major issue of frame anotations, the SAMM dataset, featuring frame level labels for onset, apex, and offset, was introduced as a complementary source to improve temporal representation and support a more robust analysis.

3.2 Proposed Models

AutoKeras will be used to build the first model for micro-expression recognition. This model, which utilizes the image classification module, is applied to the CASME II dataset to analyze individual facial micro-expressions and categorize them based on predefined emotional labels, by treating frames as static images. For the second model TensorFlow and Keras are used to build a customized architecture capable of capturing the spatial and temporal dynamics of micro-expressions by analyzing the sequence of facial movements across time. This model is applied to the SAMM dataset, which includes time-series data of facial movements.

3.2.1 Utilized Development Frameworks

This section presents and explains the three primary frameworks employed in the training and evaluation of the proposed models. These frameworks played a central role in facilitating the development process, offering essential tools and functionalities required for implementing, optimizing, and assessing model performance.

Introduction to TensorFLow

TensorFlow is a widely used open-source library developed by Google for constructing and training both machine learning and deep learning models. It is built to handle data in the form of tensors(multi-dimensional arrays)and allows this data to flow through a series of computational operations to produce predictions or extract insights. By abstracting much of the underlying mathematical complexity, such as gradient calculations and weight updates during training (where gradients indicate how much a model's weights should change to reduce prediction error), TensorFlow enables researchers and developers to concentrate on designing and evaluating model architectures rather than manually implementing low-level operations [22].

Introduction to Keras

Keras is a high-level deep learning library that provides a simplified interface for building and training neural networks. It is designed to make model development more accessible by abstracting the complexity of lower-level frameworks like TensorFlow, allowing users to define and train models with minimal code [23].

Introduction to AutoKeras

AutoKeras is an open-source Python library designed to simplify machine learning (ML) development for beginners and non-experts. It automates complex tasks such as model architecture selection, hyperparameter tuning, and performance optimization, allowing users to focus on their data without needing extensive technical expertise.

By automating the selection of optimal architectures and hyperparameters, AutoKeras makes the process of creating machine learning models faster, more efficient, and accessible to users with varying levels of experience [24].

Built on top of Keras and TensorFlow, AutoKeras benefits from Keras's intuitive API and TensorFlow's robust computational capabilities, enabling scalable and efficient model training. At the core of AutoKeras is the concept of Automated Machine Learning (AutoML), which aims to eliminate the need for manual intervention in model development. AutoML automates key aspects of the machine learning workflow, including model selection, architecture design, and hyperparameter tuning. This is achieved through a process known as Neural Architecture Search (NAS), where the system intelligently explores a wide range of model configurations to identify those that yield the best performance for a specific task.

To guide this search efficiently, AutoKeras employs a combination of advanced optimization strategies. Bayesian optimization is used to predict which model architectures are most promising based on prior performance, allowing the system to focus on high-potential regions of the search space. Greedy search complements this by selecting the best available option at each step, enabling rapid convergence toward a strong model. Additionally, network morphism enables the system to incrementally improve already effective models by making small modifications, such as adding layers or adjusting parameters, without starting training from scratch. These strategies work together to reduce computational overhead and accelerate the discovery of optimal model architectures, making deep learning more accessible, efficient, and practical for a wide range of users.

AutoKeras supports various learning tasks through specialized APIs such as ImageClassifier and TextClassifier offering flexibility across visual and textual data. Support for time series data isn't official yet, but the modular design allows future extensibility.

3.2.2 Auto-Keras Model

The proposed model, as seen in Figure 3.5, was constructed using the `ImageClassifier` module from AutoKeras, which automatically searches for the best performing convolutional neural network architecture for image classification tasks. The model was trained on the CASME II dataset, and after identifying the optimal architecture, it was exported to the standard Keras format for further evaluation and interpretation. The following section presents a detailed explanation of each layer within the proposed architecture, outlining its purpose, function, and role in the overall recognition process.

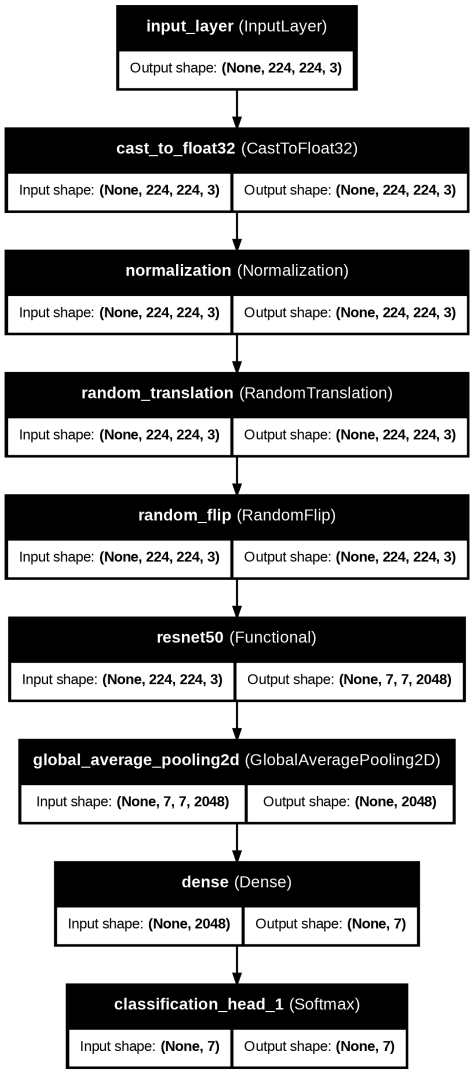

Figure 3.5: Auto-Keras Model Architecture

Input and Preprocessing

The architecture begins with an input layer designed to accept RGB images with dimensions $224 \times 224 \times 3$, where 224 represents the image width and height, and 3 corresponds to the three color channels (red, green, and blue). This is standard for image-based deep learning tasks. It is followed by a `CastToFloat32` layer that ensures all pixel values are represented in `float32` format for computational efficiency. A `Normalization` layer then standardizes the input by adjusting the mean and variance, effectively transforming the pixel values into a more centered range. This process helps improve training performance by maintaining a consistent input distribution across network layers and accelerating model performance.

Data Augmentation

To enhance the model's robustness and generalization ability, two data augmentation techniques are applied during training: a `RandomTranslation` layer that randomly shifts the image horizontally and vertically, and a `RandomFlip` layer that performs horizontal mirroring. These augmentations simulate real-world variations and reduce overfitting by increasing the diversity of training samples.

Feature Extraction

At the core of the model lies the `ResNet50` architecture, a deep convolutional neural network composed of 50 layers and pretrained on the ImageNet dataset. ResNet50 is well known for its use of residual connections, which allow gradients to pass through the network more effectively, mitigating the vanishing gradient problem common in deep learning. In this model, `ResNet50` serves as a feature extractor, transforming the input image into a high-dimensional feature map of shape $7 \times 7 \times 2048$. Each of the 2048 channels captures different learned abstractions from the image, such as textures, edges, or shapes. The 7×7 part represents a compressed spatial grid of the original image, while the 2048 refers to the number of distinct feature detectors that ResNet50

has learned to recognize. This allows the network to preserve critical information from the image while reducing its size, enabling more efficient classification.

Classification Head

The resulting feature map is passed through a `GlobalAveragePooling2D` layer, which compresses each 7×7 channel into a single scalar value by computing the average, resulting in a 2048-dimensional feature vector. This compact vector is then fed into a `Dense` layer consisting of 7 units, which is responsible for producing the classification output. Each unit corresponds to one of the facial expression categories in the CASME II dataset. A `Softmax` activation function is applied to these outputs, converting them into a probability distribution that reflects the model's confidence across all classes.

Model Overview

The image classification model is built on a ResNet50-based architecture designed to extract rich spatial features from RGB facial images. The input is standardized and augmented through normalization, random translation, and horizontal flipping to enhance generalization. ResNet50 serves as the core feature extractor, outputting a high-dimensional representation that is compressed via global average pooling. The resulting feature vector is passed to a fully connected output layer with softmax activation to produce class probabilities. The final model consists of approximately 70.5 million parameters, including over 23.5 million trainable parameters and 47 million optimization-related parameters, reflecting its depth and capacity for complex feature learning.

3.2.3 CNN+BILSTM Model

To effectively recognize subtle and short-lived facial micro-expressions, a deep learning architecture that combines spatial and temporal learning capabilities was developed, as seen in Figure 3.6. The model was designed to process sequences of grayscale facial

frames and extract meaningful patterns of motion across time. It integrates convolutional layers to learn spatial features from each frame, followed by a Bidirectional Long Short-Term Memory (BiLSTM) layer that captures temporal dynamics within the sequence. This hybrid approach enables the system to detect micro-expressions by analyzing both the appearance and evolution of facial movements. The model was trained and evaluated on the SAMM dataset, which provides annotated facial video sequences for micro-expressions. The following section presents a detailed explanation of each layer within the proposed architecture.

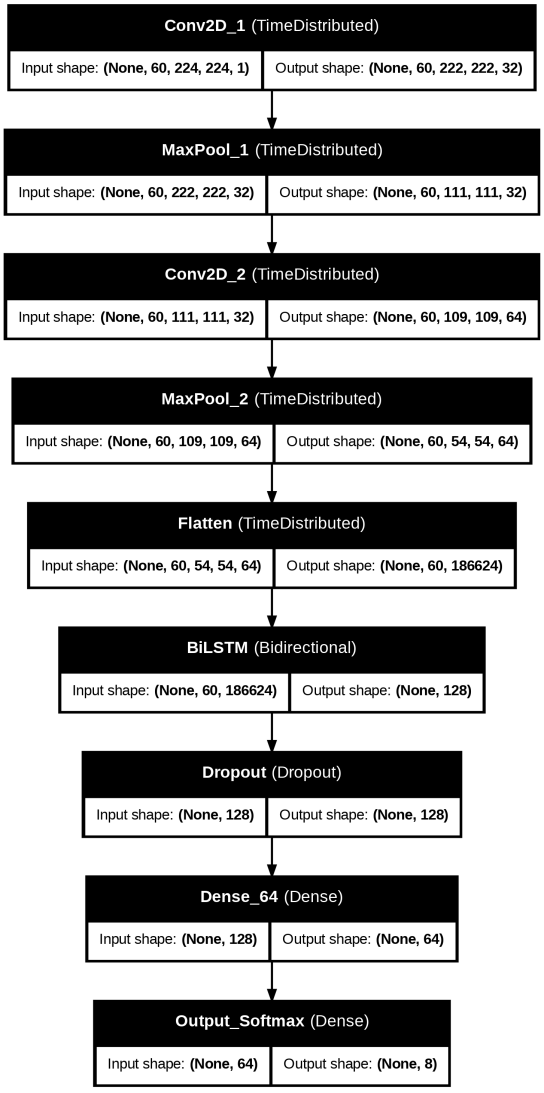

Figure 3.6: Custom CNN+BILSTM Model Architecture

Input Layer

The model begins with an input layer that accepts a sequence of grayscale facial frames. Each input sample consists of 60 frames, where each frame is a 224×224 pixel grayscale image. This layer is defined with the shape (60, 224, 224, 1), representing the number of frames in the sequence, the height and width of each frame, and the number of channels respectively. By structuring the input in this way, the model is able to receive and process entire video sequences rather than individual frames. This setup is essential for micro-expression recognition, as it allows the model to learn both the spatial structure of each frame and the subtle transitions between them over time.

First Convolutional Layer

The first convolutional layer is responsible for extracting low-level spatial features from each frame in the input sequence. It is implemented using a `TimeDistributed` wrapper to apply the same convolutional operation independently to all 60 frames. This layer uses 32 filters of size 3×3, each designed to detect simple visual patterns such as edges, corners, and textures within each frame. The activation function used is ReLU (Rectified Linear Unit), which adds non-linearity to the model, allowing it to learn more complex and realistic facial patterns beyond simple straight-line relationships. ReLU also removes negative values by setting them to zero, which helps the model focus on strongly activated features and ignore weak or irrelevant patterns. As a result of this layer, each frame is transformed from a 224×224×1 grayscale image into 32 feature maps of size 222×222, producing an output shape of (60, 222, 222, 32). This transformation enables the model to begin highlighting essential facial features that may contribute to the expression.

First MaxPooling Layer

Following the initial convolution, a max pooling layer is applied to reduce the spatial dimensions of the feature maps while preserving the most significant information. This

is done using a `TimeDistributed` wrapper to ensure the pooling operation is applied independently to each of the 60 frames in the sequence. The layer uses a 2×2 pooling window, which selects the maximum value from each 2×2 region in the feature map. As a result, each 222×222×32 output from the previous layer is downsampled to 111×111×32, producing an output shape of (60, 111, 111, 32). This reduction in size decreases computational complexity and helps the model focus on the most dominant features in each frame by filtering out less relevant details.

Second Convolutional Layer

The second convolutional layer continues the spatial feature extraction process by applying 64 filters of size 3×3 to the downsampled output of the previous pooling layer. Like the first, this layer is wrapped in a `TimeDistributed` layer to ensure that each frame in the sequence is processed independently. At this stage, the model is capable of detecting more abstract and complex patterns by building on the features learned in the previous convolution. While the first convolutional layer focused on basic visual structures like edges, this layer can recognize more detailed combinations such as localized movements or facial regions associated with specific expressions. ReLU is again used as the activation function to retain only the most relevant positive signals. The output of this layer has a shape of (60, 109, 109, 64), reflecting the increase in feature depth and allowing the model to develop a richer understanding of the spatial characteristics present in each frame.

Second MaxPooling Layer

After the second convolutional layer, a second max pooling layer is applied to further reduce the spatial dimensions of the feature maps. This layer also uses a TimeDistributed wrapper to apply the pooling operation independently to each of the 60 frames. Using a 2×2 window, it selects the maximum value from each small region of the feature map, effectively downsampling each frame while preserving the most significant

features. As a result, the output shape changes from (60, 109, 109, 64) to (60, 54, 54, 64). This step helps reduce computational load, removes noise, and ensures that the model retains only the strongest and most important visual patterns from each frame.

Flatten Layer

After the second max pooling layer, the output from each frame consists of 64 feature maps with a spatial size of 54×54. Before passing this information to the sequence model, it needs to be converted into a format that a recurrent layer like LSTM can understand. This is done using a `Flatten` layer, which transforms each 3D frame representation into a 1D vector by unrolling all the values. Specifically, each frame becomes a vector of size 186,624 (since $54 \times 54 \times 64 = 186{,}624$). The output of this layer has a shape of (60, 186624), representing 60 frames, each now described by a single feature vector. This step is essential to prepare the spatial information for temporal learning in the next layer.

Bidirectional LSTM Layer

After flattening the spatial features of each frame, the model applies a Bidirectional Long Short-Term Memory (BiLSTM) layer to learn the temporal relationships between frames. This layer treats the sequence of 60 frame-level vectors as a time series and analyzes how the facial features evolve across time. The bidirectional structure allows the model to process the sequence both forward (from frame 1 to 60) and backward (from frame 60 to 1), which improves the ability to recognize subtle changes regardless of when they appear in the sequence. The BiLSTM consists of 64 units in each direction, resulting in an output vector of length 128. This vector serves as a compact summary of the entire sequence, capturing motion dynamics that are important for distinguishing micro-expressions.

Dropout and Dense Layers

To prevent overfitting and improve the model's ability to generalize, a Dropout layer is applied after the BiLSTM layer. This layer randomly sets 50% of the values in the output vector to zero during training, forcing the model to learn more robust and widely applicable patterns instead of relying too heavily on specific features.

Following this, the first Dense layer with 64 units and ReLU activation acts as a fully connected transformation layer. It takes the 128-length summary vector from the BiLSTM and learns new combinations of the temporal features, helping the model to focus on the most relevant patterns and reduce noise. ReLU activation ensures that only positive signals are passed forward, enhancing the interpretability and non-linearity of the model's decision-making process.

Finally, a second Dense layer with 8 units and softmax activation is used to generate the final prediction. This layer outputs a probability distribution across the eight emotion classes, allowing the model to select the most likely micro-expression based on the features it has learned from the entire sequence.

Model Architecture Overview

In summary, the proposed model architecture effectively combines convolutional and recurrent layers to capture both spatial and temporal features necessary for facial micro-expression recognition. The convolutional layers extract meaningful visual patterns from each frame, while the max pooling layers reduce complexity and emphasize the most significant features. These spatial features are then flattened and passed to a Bidirectional LSTM layer, which analyzes how facial patterns evolve over time. Finally, fully connected Dense layers refine the extracted information and output a probability distribution over the emotion classes. This step-by-step design allows the model to learn both what facial features are important and how they change throughout the video sequence, enabling accurate classification of micro-expressions. The model consists

of 286,837,082 total parameters, with 95,612,360 trainable parameters, reflecting its complexity and ability to learn subtle features. The remaining parameters, totaling 191,224,722, are optimization parameters related to the training process, such as weights for the optimizer.

3.3 Evaluation Criteria

Evaluating the performance of a deep learning model is essential to understanding its accuracy and generalization ability. In facial micro-expression recognition, where the task is framed as a classification problem, evaluation techniques help assess how well the model distinguishes between expression categories. This involves the use of both numerical metrics and visual tools, which together provide a clear picture of the model's predictive strengths and weaknesses across different classes.

3.3.1 Confusion Matrix

A confusion matrix is a table that summarizes the performance of a classification model by comparing predicted labels to actual labels, as shown in Figure 3.7. It consists of four key components:

- True Positive (TP): The model correctly predicts a positive class (e.g., correctly identifying a micro-expression).
- False Positive (FP): The model incorrectly predicts a positive class when the actual label is negative (e.g., detecting an expression that isn't there).
- True Negative (TN): The model correctly predicts a negative class (e.g., correctly identifying that no micro-expression is present).
- False Negative (FN): The model fails to detect a positive class and incorrectly predicts it as negative (e.g., missing an actual micro-expression).

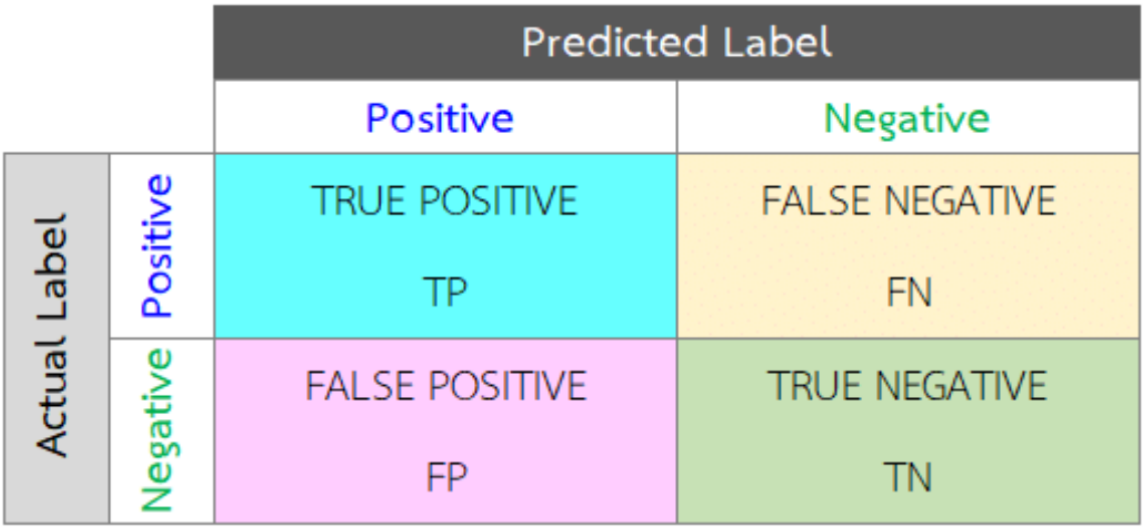

Figure 3.7: Example of confusion matrix

3.3.2 Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

where:

- TP (True Positives): Number of correctly predicted positive cases.
- TN (True Negatives): Number of correctly predicted negative cases.
- FP (False Positives): Number of negative cases incorrectly predicted as positive.
- FN (False Negatives): Number of positive cases incorrectly predicted as negative.

Accuracy measures the overall correctness of the model, counting both true positives (TP) and true negatives (TN) as correct predictions. However, if one class is much more common than the other, accuracy alone may give a misleading impression of the model's performance.

3.3.3 Precision

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

where:

- TP (True Positives): Number of correctly predicted positive cases.
- FP (False Positives): Number of negative cases incorrectly predicted as positive.

Precision measures how many of the predicted positive cases are actually correct, considering only true positives (TP) while ignoring true negatives (TN).

3.3.4 Recall

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

where:

- TP (True Positives): Number of correctly predicted positive cases.
- FN (False Negatives): Number of positive cases incorrectly predicted as negative.

Recall measures how many actual positive cases were correctly identified by the model.

3.3.5 F1-Score

$$F1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

where:

- Precision: The proportion of correctly predicted positive cases among all predicted positives.
- Recall: The proportion of correctly predicted positive cases among all actual positives.

The F1-score is a number that combines precision and recall to measure a model's overall performance.

3.3.6 Unweighted F1 (UF1)

$$UF1 = \frac{1}{N} \sum_{i=1}^{N} F1_i \tag{3.5}$$

where:

- N: Total number of classes.
- $F1_i$: The F1-Score calculated for class i.

UF1 is the average F1-score computed across all N classes, giving equal weight to each class.

3.3.7 Unweighted Average Recall (UAR)

$$UAR = \frac{1}{N} \sum_{i=1}^{N} Recall_i \tag{3.6}$$

where:

- N: Total number of classes.
- $Recall_i$: The Recall value for class i.

UAR is the average recall across all N classes, treating each class equally regardless of class distribution.

Chapter 4

Experimental Results

This chapter focuses on evaluating the performance of the proposed models for micro-expression recognition. The evaluation is based entirely on the confusion matrix, from which all performance metrics are calculated: accuracy, precision, recall, F1-score, unweighted F1-score (UF1), and unweighted average recall (UAR). All metrics are used to compare the two proposed models with each other. Accuracy is used to benchmark Model 1 against existing state-of-the-art approaches trained on the same dataset, as it reflects the overall proportion of correct predictions. Moreover, most existing works using the CASME II dataset do not report UF1 or UAR, which is why accuracy is used as the main benchmark for Model 1. However, accuracy can be misleading in the presence of class imbalance, which is common in micro-expression datasets. Therefore, for Model 2, trained on the SAMM dataset, a more detailed comparison is presented, including UF1 and UAR. These metrics provide a fairer evaluation by treating all classes equally, regardless of their frequency, and are particularly important for assessing performance on minority classes. The chapter is organized into three main sections. The first presents the results of Model 1, trained on the CASME II dataset. The second details the results of Model 2, trained on the SAMM dataset. The third section provides a comparative analysis, examining the performance differences between the two proposed models and comparing their results to existing models in the literature that have been trained on the same datasets.

4.1 Performance Evaluation of Auto-Keras Model

This section presents the performance results of the deep learning model trained using the AutoKeras ImageClassifier on the CASME II dataset. The goal of this experiment was to evaluate the model's ability to recognize facial micro-expressions from RGB images, leveraging automated neural architecture search for optimal performance. CASME II was selected due to its wide usage in the micro-expression recognition community and its high-frame-rate video samples.

4.1.1 Casme II Dataset Summary

The CASME II dataset used in this experiment consists of a total of 16,019 preprocessed RGB frames. The dataset includes seven emotion categories: disgust (4,159 images), fear (127), happiness (2,360), others (6,344), repression (2,178), sadness (274), and surprise (1,577). This version of the dataset, sourced from Kaggle, is characterized by a significant class imbalance, with certain emotions such as "others" and "disgust" dominating the sample distribution, while underrepresented classes like "fear" and "sadness" appear far less frequently.

In addition to class imbalance, this version of CASME II lacks frame-level annotations for the onset, apex, and offset phases of expressions. Furthermore, individual frames are scattered across emotion categories without being grouped into coherent sequences. These factors limit the model's ability to capture the temporal dynamics of micro-expressions, resulting in each frame being treated as a static, independent image.

4.1.2 Auto-Keras Model Experimental Setup

All training experiments were conducted on Google Colab, utilizing NVIDIA T4 GPU to accelerate deep learning computations. The dataset was loaded in batches of 32 and resized to 224×224 pixels, a common CNN standard size for image classification tasks.

The preprocessed dataset was split using an 80:10:10 ratio, corresponding to 12,815 images for training, 1,602 for validation, and 1,602 for testing. The model architecture was determined using the AutoKeras ImageClassifier, which applies neural architecture search to identify the most suitable model for the task. A total of five model architectures were explored, with each model being trained for 20 epochs.

To ensure efficiency during architecture search, early stopping was applied: if the validation accuracy did not improve over five consecutive epochs, the training of the current model was halted and the search moved on to the next trial.

4.1.3 Auto-Keras Model Performance Results

The image classifier trained on the CASME II dataset achieved a final test accuracy of 36.49%, calculated directly from the confusion matrix (Figure 4.1). To gain deeper insight into the model's performance across individual micro-expression categories, precision, recall, and F1-score were computed for each class based on the confusion matrix. The results showed that for the classes disgust, fear, happiness, repression, sadness, and surprise, the precision, recall, and F1-scores were all 0.00%, indicating that the model failed to correctly identify any instances belonging to these categories. In contrast, for the others class, the precision was 36.49%, the recall reached 100.00%, and the F1-score was 53.46%. The complete results are illustrated in Table 4.1. As shown in the confusion matrix, Figure 4.1, the model predominantly predicted all inputs as belonging to the others class, regardless of the actual label.

Table 4.1: Performance of the Auto-Keras model on CASME II by Class

Class	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Disgust	0.00	0.00	0.00
Fear	0.00	0.00	0.00
Happiness	0.00	0.00	0.00
Repression	0.00	0.00	0.00
Sadness	0.00	0.00	0.00
Surprise	0.00	0.00	0.00
Others	36.49	100.00	53.46
Unweighted Average (UAR / UF1)	–	**14.29**	**7.64**

Other class received predictions across all categories, including disgust, happiness, repression, and surprise, while all other classes received no correct predictions. This behavior reflects the model's strong bias toward the majority class, which is consistent with the observed class imbalance in the dataset. These outcomes highlight a major limitation in the model's ability to generalize beyond the dominant class.

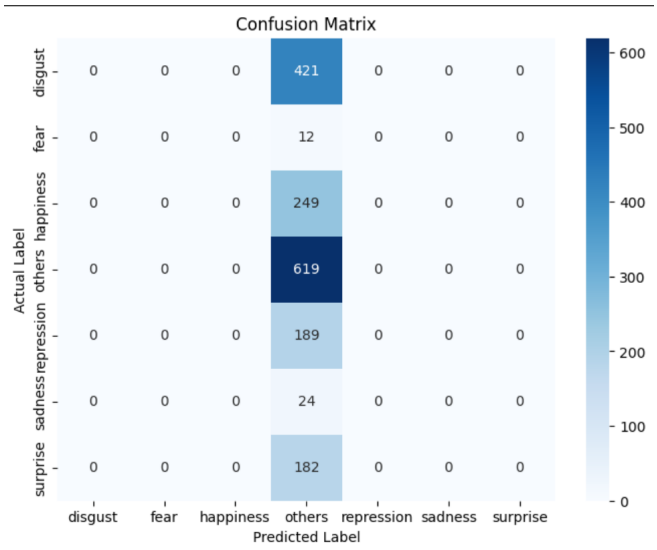

Figure 4.1: Confusion matrix of the Auto-Keras model on the CASME II test set.

Two main factors contribute to these performance limitations. First, the dataset used is heavily imbalanced, with the others class comprising a disproportionately large portion of the training data. This imbalance leads the model to favor the dominant class in order to minimize loss, even at the expense of generalization.

Second, the inherent nature of micro-expressions poses an additional challenge. Micro-expressions are characterized by their extremely short duration and subtle muscle movements, which often span only a few frames. Since the CASME II dataset in this version lacks frame-level annotations (onset, apex, and offset), and frames were provided as isolated, unordered images, the model is unable to capture the temporal progression that is critical for accurate recognition. Without temporal context, many subtle visual cues are lost, making it difficult for a frame-based image classifier to distinguish between classes.

These results highlight the limitations of using static image-based classification for micro-expression recognition and emphasize the need for sequence-based models or temporal analysis techniques in future experiments.

4.2 Performance Evaluation of CNN+BiLSTM Model

This section presents the performance results of the deep learning model trained using a custom CNN+BiLSTM architecture on the SAMM dataset. The objective of this experiment was to assess the model's ability to recognize subtle facial micro-expressions from grayscale image sequences by combining spatial and temporal features. The SAMM dataset was chosen due to its high temporal resolution and detailed frame-level annotations, which make it particularly well-suited for capturing the brief and involuntary nature of micro-expressions. By leveraging convolutional layers for spatial feature extraction and a bidirectional LSTM to analyze temporal dependencies across frame sequences, the model aims to accurately classify micro-expressions into their respective emotional categories.

4.2.1 SAMM Dataset Summary

The dataset used in this experiment is SAMM, which contains a total of 158 micro-expression sequences categorized into eight emotion classes: Anger (57), Contempt (12), Disgust (9), Fear (8), Happiness (26), Other (25), Sadness (6), and Surprise (15). Each sequence represents a short video clip of a micro-expression, extracted as a series of consecutive image frames. The dataset is accompanied by a highly detailed CSV annotation file that records essential metadata, including the subject's folder number, onset frame, apex frame ,offset frame, and the associated emotion label. One of SAMM's main advantages is its structured and organized format, each sequence is stored in its own directory, and frame-level annotations are both clear and consistent making it particularly well-suited for temporal analysis of micro-expressions. The primary

challenge in using SAMM, however, lies in the significant class imbalance as shown in Figure 4.2. Emotions like Anger are highly represented, while others such as Fear and Sadness appear much less frequently. This imbalance, however, is addressed during the preprocessing phase through techniques designed to ensure more balanced training input.

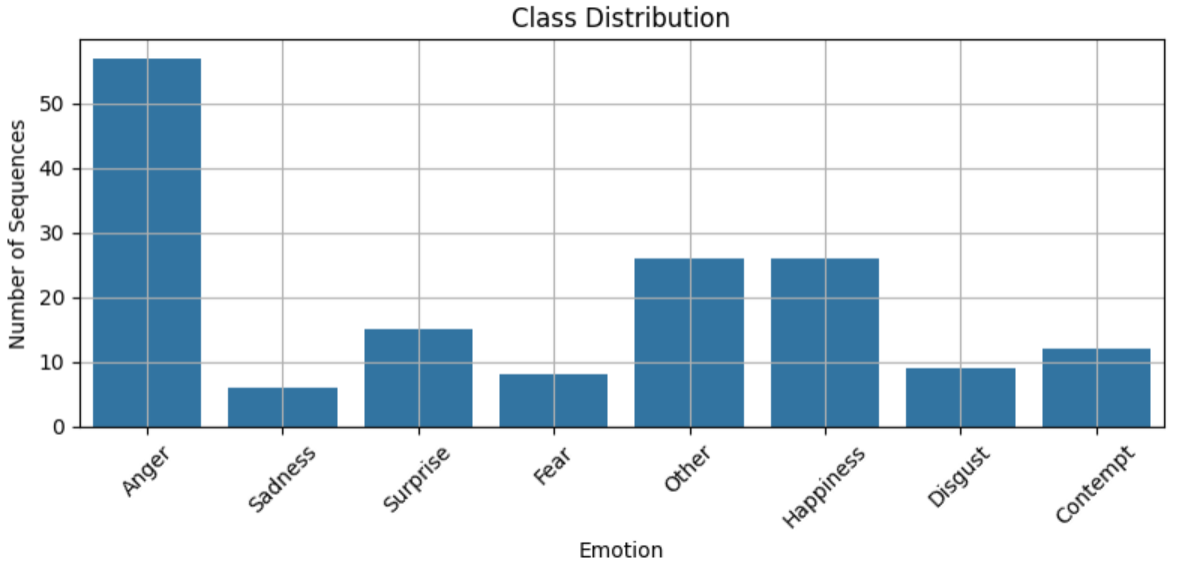

Figure 4.2: Distribution of micro-expression classes in the SAMM dataset originally.

4.2.2 CNN+BILSTM Experimental Setup

The experimental setup was carried out using Google Colab with access to a high-performance NVIDIA A100 GPU, providing the necessary computational resources for training deep learning models efficiently. Data loading was based on the accompanying annotation CSV file, which contained key metadata such as the subject's folder name, onset and offset frame indices, and the corresponding emotion label. This information enabled accurate extraction of image sequences representing each micro-expression instance. During preprocessing, each extracted frame was resized to 224×224 pixels to ensure consistent input dimensions, converted to grayscale to reduce complexity and focus on facial muscle movements, and normalized to the [0, 1] range to standardize pixel values and enhance the model's ability to learn effectively. Only sequences with complete frame coverage between onset and offset were included in the training set.

During sequence loading, all micro-expression samples were padded or truncated to a fixed length of 60 frames to ensure consistent input shape, with padding applied using zero-valued (black) frames. Moreover, to address the issue of class imbalance within the dataset, data augmentation was selectively applied to underrepresented emotion classes. Frame-level transformations included random horizontal flipping, brightness adjustments, and contrast variations. Figure 4.3 shows a few sample frames after applying data augmentation. This augmentation strategy produced a balanced training set, as seen in Figure 4.4, which helped the model generalize more effectively across all emotion categories.

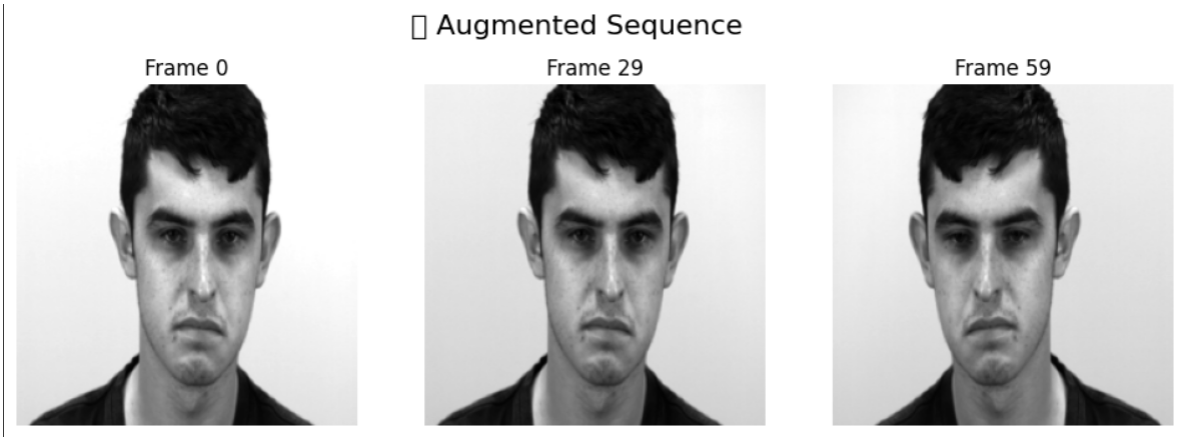

Figure 4.3: Augmented Frames example

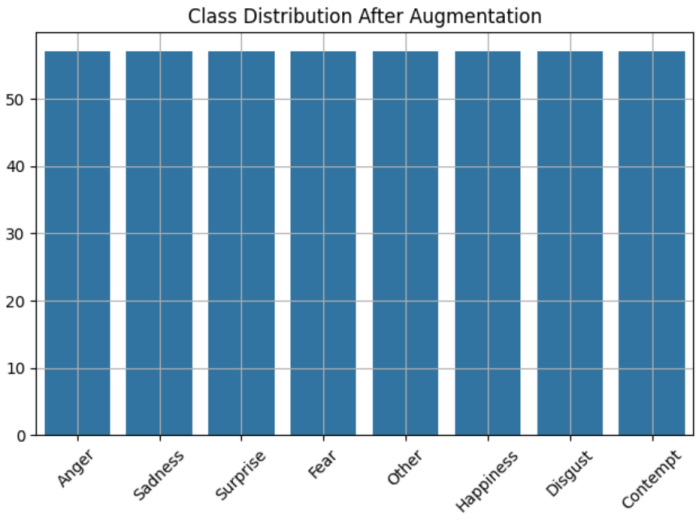

Figure 4.4: Distribution of micro-expression classes in the SAMM dataset after data-augmentation.

The preprocessed dataset was split in two stages using stratified sampling, which ensures that each class is present in approximately the same quantity across all subsets. First, 20% of the data was reserved as the test set. The remaining 80% was further divided into training sets (64%) and validation sets (16%). Although data augmentation techniques were employed to increase the diversity of training samples and reduce overfitting, class weights were also applied as a reassurance step to support balanced learning across all classes. The model was trained for up to 50 epochs with a batch size of 1, using EarlyStopping with a patience of 10 epochs (monitoring validation accuracy) to prevent overfitting.

4.2.3 CNN+BILSTM Performance Results

The model's performance over 50 epochs was tracked through both accuracy and loss curves, as shown in Figure 4.5. Training accuracy increased steadily, stabilizing above 90%, while training loss showed a consistent decrease throughout the epochs. The validation accuracy also improved, reaching an average of approximately 80%. However, both validation accuracy and validation loss plateaued after a certain point, indicating that the model's improvement on the training set did not fully translate to the validation set, which can be attributed to the challenging nature of recognizing subtle facial expressions in micro-expression recognition tasks.

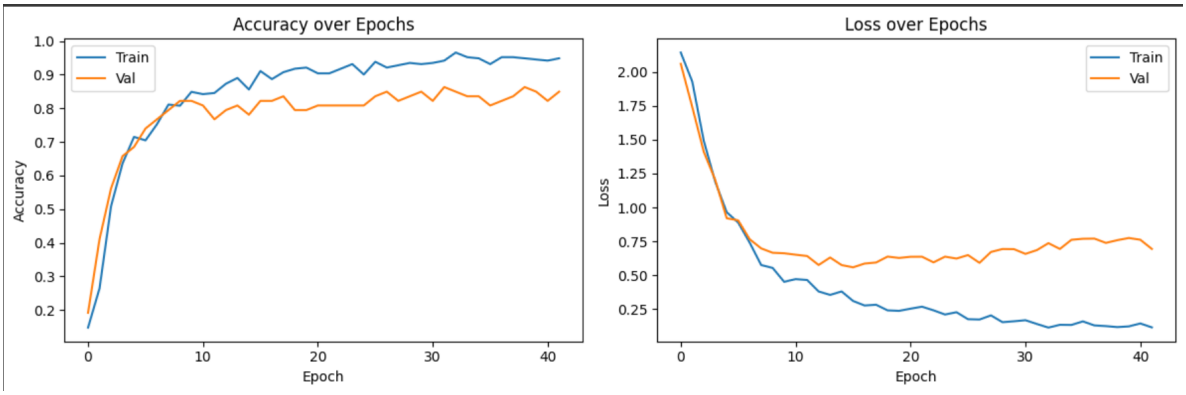

Figure 4.5: Training and validation accuracy and loss over 50 epochs

Below is the confusion matrix, Figure 4.6, which illustrates the distribution of correct classifications and misclassifications across the different expression categories. From the confusion matrix, the overall accuracy was calculated to be approximately 65.22%. Given the subtle and transient nature of micro-expressions, achieving this level of accuracy indicates that the model was able to capture important discriminative features.

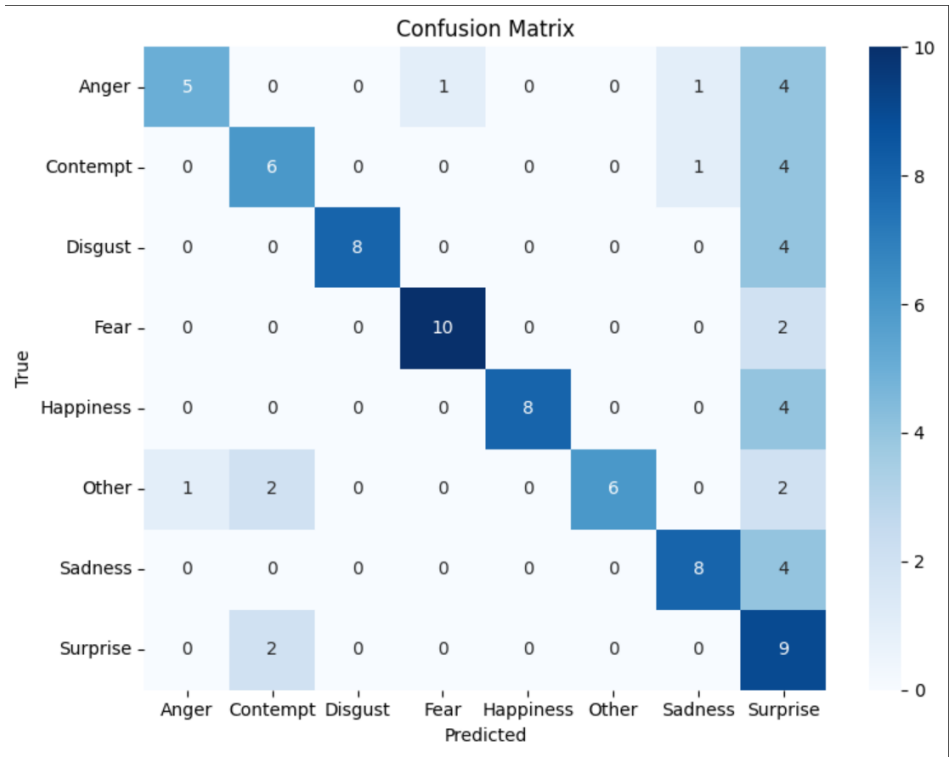

Figure 4.6: Confusion matrix for the Custom CNN+BILSTM model

Furthermore, precision, recall, and F1-score were computed for each class to provide a more detailed evaluation. These metrics offer insights into the model's ability to correctly classify subtle facial micro-expressions. Table 4.2 summarizes the results.

Table 4.2: Performance of the CNN+BILSTM model on SAMM by Class

Class	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Anger	83.33	45.45	58.85
Contempt	60.00	54.55	57.14
Disgust	100.00	66.67	80.00
Fear	90.90	83.33	86.90
Happiness	100.00	66.67	80.00
Other	100.00	54.55	70.60
Sadness	80.00	66.67	72.70
Surprise	27.27	81.82	40.90
Unweighted Average (UAR / UF1)	–	**64.96**	**68.39**

Overall, the results show that the model was able to recognize several micro-expressions effectively, particularly Fear, Happiness, and Disgust, which achieved the highest F1-scores. Meanwhile, recognizing Surprise was more challenging, as reflected by its lower precision and F1-score. These outcomes reflect the difficulty of distinguishing between subtle and rapidly occurring facial expressions, even after balancing the dataset through data augmentation.

4.3 Comparative Analysis

In this section, we present a comparative analysis between the proposed models and other state-of-the-art approaches discussed in the literature. To ensure a fair evaluation, we focus on models that were trained and tested on the same dataset, allowing for consistent performance comparison. Following this, we provide a direct comparison between our two proposed models to assess their relative strengths, weaknesses, and suitability for micro-expression recognition.

4.3.1 Auto-Keras Model on CASME II

To evaluate the performance of the AutoKeras image classification model on the CASME II dataset, we compare its results against two state-of-the-art models.

Table 4.3 summarizes the accuracy achieved by our model in comparison to existing state-of-the-art models on the CASME II dataset.

Table 4.3: Comparison of Auto-Keras model with State-of-the-Art on CASME II Dataset

Model	**Accuracy (%)**
LSTM with Facial Color Changes [15]	66.66
3D CNN Model (Best run) [12]	85.20
3D CNN Model (Worst run) [12]	45.00
AutoKeras Image Classifier (Ours)	**36.49**

The first model, proposed by [15], relies on facial color changes and uses an LSTM to capture temporal dynamics. This approach achieved an accuracy of 66.66% on the CASME II dataset. The second, a 3D CNN model described in [12], used a train–test split strategy and reported a peak accuracy of 85.2%. However, it is important to note that the 3D CNN model's results varied significantly across different random splits dropping as low as 45% raising concerns about its reproducibility and stability.

In contrast, our AutoKeras image classification model achieved an accuracy of only 36.49%. This significantly lower performance can be attributed to two key factors. First, micro-expression recognition is inherently a sequence-based task, as these expressions evolve over time. Treating it as a static image classification problem ignores the temporal dependencies that are critical for accurate recognition. Second, the CASME II dataset , version obtained from kaggle , itself presents several challenges: the absence of frame-level annotations, scattered and misaligned frames across samples, and a substantial class imbalance all contribute to limiting model performance.

These findings highlight the importance of treating micro-expression recognition as a spatio-temporal learning problem rather than a static image classification task.

4.3.2 CNN + BiLSTM Model on SAMM Dataset

To evaluate the performance of the proposed CNN + BiLSTM model on the SAMM dataset, we compare its results against two existing state-of-the-art models reported in the literature.

Table 4.4 presents a comparison of accuracy, unweighted F1-score (UF1), and unweighted average recall (UAR) across different models evaluated on the SAMM dataset.

Table 4.4: Comparison of CNN+BILSTM model with State-of-the-Art on SAMM Dataset

Model	**Accuracy (%)**	**UF1 (%)**	**UAR (%)**
Gender-Aware MER (Single-task) [16]	51.47	39.62	39.07
Gender-Aware MER (Multi-task) [16]	55.88	45.38	46.35
CapsuleNet Model [17]	–	62.09	59.89
CNN + BiLSTM Model (Ours)	**65.22**	**68.39**	**64.96**

The first comparison model, proposed in [16], is a gender-aware micro-expression recognition framework. In its single-task version, which focuses only on recognizing micro-expressions, it achieved a UF1 of 39.62% and a UAR of 39.07%. In the multi-task setting, combining micro-expression recognition with gender classification, the model improved to an accuracy of 55.88%, UF1 of 45.38%, and UAR of 46.35%.

The second model, based on CapsuleNet and reported in [17], focuses on modeling spatial relationships in facial features. It achieved a UF1 score of 62.09% and a UAR of 59.89% on the SAMM dataset.

In contrast, our proposed CNN + BiLSTM model achieved an accuracy of 65.22%, UF1 of 68.39%, and UAR of 64.96%. These results demonstrate strong overall performance and improved generalization across classes, particularly in a dataset like SAMM that suffers from class imbalance and limited sample size. To address these challenges,

data augmentation techniques were applied during training, which helped mitigate overfitting and improve the model's ability to recognize less represented classes.

These findings highlight the effectiveness of combining convolutional feature extraction with temporal modeling using BiLSTM layers, especially when supported by data augmentation in low-resource micro-expression recognition tasks.

4.3.3 Comparative Analysis of Proposed Models

This section compares the two models developed in this study: the AutoKeras image classifier trained on CASME II and the CNN + BiLSTM model trained on SAMM. Both aim to recognize micro-expressions, but differ significantly in approach and effectiveness.

CASME II and SAMM differ not only in structure but also in the emotion classes they include. CASME II features seven categories: other, disgust, fear, happiness, sadness, surprise, and repression, while SAMM includes anger, contempt, disgust, fear, happiness, sadness, surprise, and other. This variation in class definitions contributes to differences in evaluation focus across the two models. Although both datasets have challenges such as imbalance and small sample size, data augmentation was used in SAMM to improve model robustness.

The AutoKeras model treated micro-expression recognition as a static image classification task using CASME II. As shown in Table 4.5 and Table 4.6, this led to extremely poor performance, with all classes except others receiving zero precision, recall, and F1-score. The model predicted nearly all inputs as belonging to the majority class (others), reflecting a strong bias due to class imbalance.

On the other hand, the CNN + BiLSTM model, trained on SAMM, approached the task as a sequential learning problem more appropriate for capturing the dynamic and temporal nature of micro-expressions. Supported by data augmentation, this model

achieved strong per-class and overall performance, demonstrating improved generalization and robustness.

Table 4.5: Per-Class Metrics Comparison Between Auto-Keras and CNN+BILSTM models(%)

Class	**AutoKeras (CASME II)**			**CNN + BiLSTM (SAMM)**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Anger	–	–	–	83.33	45.45	58.85
Contempt	–	–	–	60.00	54.55	57.14
Disgust	0.00	0.00	0.00	100.00	66.67	80.00
Fear	0.00	0.00	0.00	90.90	83.33	86.90
Happiness	0.00	0.00	0.00	100.00	66.67	80.00
Other	36.49	100.00	53.46	100.00	54.55	70.60
Sadness	0.00	0.00	0.00	80.00	66.67	72.70
Surprise	0.00	0.00	0.00	27.27	81.82	40.90
Repression	0.00	0.00	0.00	–	–	–

Table 4.6: Overall Performance Comparison Between Auto-Keras and CNN+BILSTM Models

Model	**Accuracy (%)**	**UF1 (%)**	**UAR (%)**
AutoKeras Image Classifier (CASME II)	36.49	7.64	14.29
CNN + BiLSTM (SAMM)	**65.22**	**68.39**	**64.96**

As seen in the results, the CNN + BiLSTM model provides much more reliable and balanced performance across emotion classes. The AutoKeras model, while simpler, failed to generalize due to both its static design and the structural limitations of the CASME II dataset. This comparison highlights the importance of treating micro-expression recognition as a temporal problem and leveraging models that can capture spatio-temporal dynamics.

Chapter 5

Conclusion and Future Work

In conclusion , this thesis focused on the development of a deep learning-based model for automatic micro-expression recognition. The main objectives of the project were to address the challenges posed by the subtlety and rapid nature of micro-expressions, to overcome the limitations of available data, and to design a model capable of capturing the unique features of micro-expressions. Through careful dataset selection, data preprocessing, augmentation strategies, and model design, the project aimed to improve the accuracy and reliability of recognizing concealed emotions from facial micro-expressions.

5.1 Main Contributions

This section highlights the main contributions achieved throughout the thesis. The work involved developing and evaluating different deep learning models for micro-expression recognition and addressing the limitations of available datasets The key contributions are summarized as follows:

1. Developed and evaluated two different models for micro-expression recognition: an image classification model based on AutoKeras and a custom sequence-based CNN-BiLSTM model.

2. Identified the limitations of static image classification for micro-expression recognition. Since micro-expressions are extremely fast and require tracking subtle facial changes over time, treating them as static images fails to capture the necessary temporal dynamics. This limitation was particularly evident when working

with the CASME II dataset, where the lack of detailed frame labeling and the scattered organization of frames made it difficult to model sequences effectively. This issue was addressed by using the SAMM dataset, which provided clear frame labeling (onset, apex, offset) and enabled building meaningful temporal sequences for more accurate recognition.

3. Conducted preprocessing and data augmentation procedures for both models to enhance training data quality and diversity. All frames were resized to 224x224 pixels, which is a common standard size for deep learning image models, ensuring uniform input dimensions across the datasets. For the CNN-BiLSTM model, frames were also converted into grayscale to focus on essential facial features while reducing computational complexity. Regarding data augmentation, AutoKeras automatically applied random transformations (such as rotations or flips) during training to prevent overfitting without increasing the number of training samples. In contrast, for the CNN-BiLSTM model, manual data augmentation was performed to balance the number of sequences across classes, involving controlled changes to brightness, contrast, and horizontal flipping of frames.

4. Implemented a structured experimental setup involving train, validation, and test splits, and systematically evaluated model performance using accuracy, precision, recall, and F1-score metrics.

5. Performed a comparative analysis between the two developed models, demonstrating the superiority of sequence-based approaches over static image-based classification, particularly in recognizing subtle emotional expressions.

6. Compared the results of the developed models with those reported in the literature review.

Overall, this study demonstrates the potential of the proposed custom deep learning model in improving the automatic recognition of facial micro-expressions. The devel-

oped CNN-BiLSTM model provides a promising solution for capturing subtle emotional cues, addressing dataset limitations, and enhancing the reliability and performance of micro-expression analysis systems. The experimental results highlight that micro-expressions should not be treated as static image classification tasks, but rather require modeling the temporal changes between frames, as effectively captured by the proposed sequence-based model. Furthermore, the study emphasizes the importance of using cleanly organized datasets that provide clear frame labeling and accurate emotion annotations to enable effective sequence modeling and improve recognition performance.

5.2 Future Work

While this study successfully demonstrated the potential of deep learning models, particularly the custom CNN-BiLSTM model, in recognizing facial micro-expressions, still remains several opportunities for further research and improvement in this domain. The following points outline potential directions that could be explored in future work to build upon the findings of this thesis.

1. Dataset Combination

   In the current work, experiments were conducted on a single dataset. A promising direction for future research is to combine multiple micro-expression datasets (such as CASME II, SAMM, and SMIC) to increase sample diversity and enable the model to learn more generalized facial movement patterns.

2. Cross-Dataset Evaluation

   Cross-dataset testing where a model is trained on one dataset and tested on a different one can be employed to better assess the model's generalization ability across variations in recording settings, subject demographics, and environmental conditions. Such evaluation would provide a stronger indication of the model's robustness and its potential for real-world deployment, beyond the constraints of a single dataset.

3. Dataset Size and Balance Improvement

   Another important future direction is to increase the number of video samples within datasets and ensure a balanced distribution across classes. Expanding the dataset size and addressing class imbalance are crucial to prevent model bias, enhance learning stability, and improve the reliability of micro-expression recognition performance.

4. Exploring Multi-Stream Deep Learning Architectures

   The current use of AutoKeras was limited to treating micro-expression recognition as a static image classification task, which may overlook subtle facial motion cues. A valuable improvement would be to explore a multi-stream deep convolutional neural network that integrates multiple architectures (e.g., ResNet, DenseNet, VGG) to extract more discriminative and complementary features from static images. Such an approach is more suitable for extracting rich representations from apex frames and can potentially outperform black-box AutoML models like AutoKeras, especially when dealing with low-intensity micro-expressions. It would also allow the integration of advanced post-processing steps like ensemble classification, resulting in higher accuracy and better class separation.

5. Explainability and Model Interpretation

   A valuable future direction is to enhance micro-expression recognition systems with explainable artificial intelligence (XAI) techniques. Deep learning models often behave like black boxes they produce a prediction (e.g., "disgust" or "surprise") without revealing how or why the decision was made. This lack of transparency is a concern in micro-expression recognition, where expressions are subtle, brief, and difficult even for humans to detect. To improve understanding and trust in the model's behavior, explainability methods such as Grad-CAM (Gradient-weighted Class Activation Mapping) can be applied. Grad-CAM generates a heatmap over the input face image, visually highlighting the regions that influenced the model's decision the most. For instance, if the model predicts

"fear," Grad-CAM might show bright regions around the eyes and mouth, facial areas typically involved in that emotion. These visual cues help verify whether the model is learning meaningful patterns (such as facial muscle movements) instead of unrelated features (like background or lighting). This not only builds trust and accountability but also assists researchers in debugging misclassifications and improving model design. Incorporating such explainability tools makes the system more transparent, interpretable, and suitable for real-world deployment in domains like psychological assessment, security, and human-computer interaction.

In conclusion, future research should explore and address these directions to further advance the field of automated micro-expression recognition. By combining multiple datasets, evaluating models across different domains, improving dataset size and class balance, incorporating Multi-Stream architectures, and integrating explainability techniques, we can continue to enhance the accuracy, robustness, and practical value of deep learning models in this domain.

References

[1] Z. Dong, G. Wang, S. Lu, J. Li, W. Yan, and S.-J. Wang, "Spontaneous facial expressions and micro-expressions coding: From brain to face," *Frontiers in Psychology*, vol. 12, p. 784 834, 2022.

[2] G. Zhao, X. Li, Y. Li, and M. Pietikäinen, "Facial micro-expressions: An overview," *Proceedings of the IEEE*, vol. 111, no. 10, pp. 1215–1235, 2023.

[3] X. Ben, Y. Ren, J. Zhang, *et al.*, "Video-based facial micro-expression analysis: A survey of datasets, features and algorithms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5826–5846, 2021.

[4] Y. Li, J. Wei, Y. Liu, J. Kauttonen, and G. Zhao, "Deep learning for micro-expression recognition: A survey," *IEEE Transactions on Affective Computing*, vol. 13, no. 4, pp. 2028–2046, 2022.

[5] T. Amaratunga and T. Amaratunga, "What is deep learning?" *Deep Learning on Windows: Building Deep Learning Computer Vision Systems on Microsoft Windows*, pp. 1–14, 2021.

[6] K. Gurney, *An introduction to neural networks*. CRC press, 2018.

[7] M. Zakaria, A. Mabrouka, and S. Sarhan, "Artificial neural network: A brief overview," *neural networks*, vol. 1, p. 2, 2014.

[8] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132 306, 2020.

[9] R. C. Staudemeyer and E. R. Morris, "Understanding lstm–a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.

[10] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[11] J. Li, T. Wang, and S.-J. Wang, "Facial micro-expression recognition based on deep local-holistic network," *Applied Sciences*, vol. 12, no. 9, p. 4643, 2022.

[12] K. K. Talluri, M.-A. Fiedler, and A. Al-Hamadi, "Deep 3d convolutional neural network for facial micro-expression analysis from video images," *Applied Sciences*, vol. 12, no. 21, p. 11 078, 2022.

[13] G. Perveen, S. F. Ali, J. Ahmad, *et al.*, "Multi-stream deep convolution neural network with ensemble learning for facial micro-expression recognition," *IEEE Access*, vol. 11, pp. 118 474–118 489, 2023.

[14] F. Zhang, Y. Liu, X. Yu, *et al.*, "Towards facial micro-expression detection and classification using modified multimodal ensemble learning approach," *Information Fusion*, vol. 115, p. 102 735, 2025.

[15] H. Shahar and H. Hel-Or, "Micro expression classification using facial color and deep learning methods," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[16] X. Nie, M. A. Takalkar, M. Duan, H. Zhang, and M. Xu, "Geme: Dual-stream multi-task gender-based micro-expression recognition," *Neurocomputing*, vol. 427, pp. 13–28, 2021.

[17] N. Van Quang, J. Chun, and T. Tokuyama, "Capsulenet for micro-expression recognition," in *2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019)*, IEEE, 2019, pp. 1–7.

[18] W.-J. Yan, Q. Wu, Y.-J. Liu, S.-J. Wang, and X. Fu, "Casme database: A dataset of spontaneous micro-expressions collected from neutralized faces," in *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, IEEE, 2013, pp. 1–7.

[19] W.-J. Yan, X. Li, S.-J. Wang, *et al.*, "Casme ii: An improved spontaneous micro-expression database and the baseline evaluation," *PloS one*, vol. 9, no. 1, e86041, 2014.

[20] F. Qu, S.-J. Wang, W.-J. Yan, and X. Fu, "Cas (me) 2: A database of spontaneous macro-expressions and micro-expressions," in *Human-Computer Interaction. Novel User Experiences: 18th International Conference, HCI International 2016, Toronto, ON, Canada, July 17-22, 2016. Proceedings, Part III 18*, Springer, 2016, pp. 48–59.

[21] A. K. Davison, C. Lansley, N. Costen, K. Tan, and M. H. Yap, "Samm: A spontaneous micro-facial movement dataset," *IEEE transactions on affective computing*, vol. 9, no. 1, pp. 116–129, 2016.

[22] G. Zaccone, *Getting started with TensorFlow*. Packt Publishing ISBN, 2016.

[23] N. Ketkar, "Introduction to keras," in *Deep learning with python: a hands-on introduction*, Springer, 2017, pp. 97–111.

[24] H. Jin, F. Chollet, Q. Song, and X. Hu, "Autokeras: An automl library for deep learning," *Journal of machine Learning research*, vol. 24, no. 6, pp. 1–6, 2023.