Comparative Analysis of Hybrid Face Detectors and Classifiers for Child Emotion Recognition and Toy Recommendation Systems

By
Safrin Jahan Powlome
ID: CSE2001019273

Md Rasel Miah
ID: CSE2003021030

Md. Asibul Islam
ID: CSE2202026152

Md Ashef Alam
ID: CSE2202026141

Supervised By:

Zarin Hadika

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science & Engineering.

Department of Computer Science & Engineering
Sonargaon University (SU)

May 2026

APPROVAL

The thesis titled "**Comparative Analysis of Hybrid Face Detectors and Classifiers for Child Emotion Recognition and Toy Recommendation Systems**" submitted by **Safrin Jahan Powlome** (CSE2001019273), **Md Rasel Miah** (CSE2003021030), **Md. Asibul Islam** (CSE2202026152), **Md Ashef Alam** (CSE2202026141) to the Department of Computer Science and Engineering, Sonargaon University (SU) has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

---------- **Supervisor**

Zarin Hadika
Lecturer
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 1**

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 2**

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 3**

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We hereby declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Zarin Hadika,** Lecturer, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this Thesis and thereof has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Zarin Hadika)
Supervisor

Signatures

Safrin Jahan Powlome
ID: CSE2001019273

Md Rasel Miah
ID: CSE2003021030

Md. Asibul Islam
ID: CSE2202026152

Md Ashef Alam
ID: CSE2202026141

ABSTRACT

The rapid advancement of affective computing has opened new avenues for supporting early childhood development through intelligent systems. However, traditional facial expression recognition (FER) models, primarily trained on adult datasets, often fail to generalize to the unique facial structures and nuanced emotional displays of infants and toddlers. This research presents an integrated AI-driven framework designed to provide personalized toy recommendations based on real-time infant emotion detection. The system utilizes a specialized pipeline incorporating MTCNN for precise facial alignment and a comparative analysis of high-performance architectures, including EfficientNet-B0, DenseNet121, ResNet50, and YOLO for robust feature extraction and classification. To address the critical scarcity of open-access data for the 0–1.5 year age group, a domain-specific dataset was curated to enhance model reliability in detecting subtle affective states. Beyond technical classification, the research introduces a scalable mobile application that bridges the gap between emotion analysis and consumer intelligence by offering mood-aligned toy suggestions with integrated price-tracking features. Experimental results indicate that our approach significantly provides a high degree of accuracy in resource-constrained environments. This study contributes a foundational methodology for infant-centric AI applications, offering a practical tool to foster emotional growth and personalized play experiences on a global scale.

Keywords: Infant Emotion Recognition, Toy Recommendation System, Deep Learning, Affective Computing, Facial Expression Analysis, Child Development.

ACKNOWLEDGMENT

At the very beginning, we would like to express our deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of **Zarin Hadika**, Lecturer, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our thesis.

We would like to convey our special gratitude to **Prof. Bulbul Ahamed,** Head, Department of Computer Science and Engineering, Sonargaon University, for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to our parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

DL	Deep Learning
DenseNet121	Densely Connected Convolutional Network (121-layer)
EfficientNet-B0	Efficient Convolutional Neural Network (Baseline Model)
ML	Machine Learning
MTCNN	Multi-task Cascaded Convolutional Neural Networks
ReLU	Rectified Linear Unit
ResNet50	Residual Network (50-layer)
TFLite	TensorFlow Lite
YOLO	You Only Look Once (Object Detection Framework)

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background of the Study

Childhood is a crucial stage of human development, particularly during the first few months of life. Within the first 0-1.5 years from birth, infants begin to experience and express rudimentary emotions such Happy, Sad, and Calm. Developmental psychologists emphasize that interpreting these emotional states is unavoidable, as they determine how infants interact with their caregivers, objects, and surrounding environments. At this stage, verbal ability is limited or nonexistent; therefore, facial expression serves as one of the primary channels of affective communication among various infantile nonverbal signals. These include smiles, tears, pouting, and expressions of surprise. Research into facial behavior analysis confirms that such expressive cues are among the most universal and reliable indicators of a child's inner emotional state [1].

Furthermore, selecting appropriate toys based on a child's emotional state is a critical yet underexplored area in intelligent system design. Traditional toy selection methods mainly rely on age, general preferences, or caregiver intuition, without considering the child's real time mood. However, infants convey their needs and feelings primarily through facial cues, making emotion aware analysis a more effective approach for understanding their immediate condition. With recent advancements in deep learning and transfer learning, it has become feasible to develop systems that can automatically detect affective signals from facial images and utilize this information for intelligent decision making [2]. Integrating facial emotion recognition with a recommendation mechanism can enable the development of adaptive systems that suggest suitable toys aligned with the child's detected mood, thereby improving engagement, comfort, and developmental support for infants.

Facial images are particularly valuable because they carry rich affective information that can be efficiently processed using modern neural network architectures. Recent progress in this domain has significantly improved automated expression recognition systems. Convolutional Neural Networks (CNNs) and related architectures can automatically extract discriminative visual features and classify emotional states with high accuracy [3]. Studies show that learned representations consistently outperform traditional machine learning methods in affective classification tasks [4]. Moreover, the growing availability of large scale annotated datasets has further improved model robustness and generalization ability across varied conditions. However, challenges such as lighting variation, occlusion, and real world image complexity still affect system performance [5].

Despite these advancements, a significant research gap remains. Most existing datasets and models are primarily trained on adult facial expressions, which do not accurately represent the structural and behavioral differences in infant and toddler faces. As a result, systems built on such data often fail to generalize effectively to younger age groups. This limitation highlights the need for specialized research focusing on infant emotion recognition using facial images. Addressing this gap can enable the development of intelligent systems that support caregiving, healthcare, and personalized interactive environments for early childhood development.

1.2 Motivation

This research stems from the observation that a child's affective condition is rarely considered during toy selection, even though emotional engagement is central to meaningful play. Parents and caregivers often choose toys based on accessibility, past experience, or general popularity rather than the child's current state of mind. As a result, the toys selected may not always match the infant's mood, diminishing interest and restricting the comfort that appropriate play can provide.

Automatically detecting emotional cues from facial expression can therefore help caregivers better understand and respond to a child's needs in a timely and supportive way.

Another key driver behind this work is the shortcoming of previous facial emotion recognition systems, which are largely built on adult oriented data. This inadequacy underscores the need for infant specific research using datasets that better reflect the expressive characteristics of young children.

Moreover, although a significant number of studies treat emotion detection as a classification problem, fewer works establish a connection between this domain and practical real world applications. In particular, the link between recognized emotional states and context appropriate toy selection remains largely unexplored. An additional motivation of this work is to assess how different combinations of face detection algorithms and deep learning classification models perform in recognizing children's emotions.

Finally, the aim of this study is to build an intelligent system capable of discerning the emotional states of children from facial images and recommending appropriate toys accordingly. The work integrates affective recognition with toy suggestion to demonstrate how AI can promote children's wellbeing and foster a more contextually appropriate play experience.

1.3 Problem Statement

In recent years, it has been observed that children, especially infants, possess distinct facial structures and expressive nuances that differ significantly from adults, meaning an infant's outward facial expression may not always align with their internal emotional state in the same way it does for a mature individual. Consequently, models trained on adult datasets often fail to generalize effectively when applied to the unique physiological and emotional characteristics of a child's face.

One major challenge in this domain is the scarcity of open access datasets specifically curated for newborns and toddlers up to 1.5 years of age, which severely limits the development and rigorous assessment of models capable of accurately detecting early childhood emotions. Furthermore, the vast majority of existing work on facial emotion recognition is devoted to improving classification performance rather than translating results into real world applications. However, only a handful of studies try to relate the emotions detected from conversation with real world interactive systems.

Also, there has been little research comparing various combinations of face detection algorithms and deep learning classification models with specific regard to facial emotion recognition in young babies. The evaluation of various hybrid model pipelines helps understand the best performing model combinations in terms of their performance when used to predict emotions from children's facial expression.

To address these identified gaps, this research attempts to provide a practical and intelligent application of artificial intelligence by integrating an advanced emotion recognition framework into a toy recommendation system, while simultaneously analyzing various model combinations to support children's emotional development.

1.4 Research Objectives

The main focus of this research is to design, train, and evaluate deep learning based facial emotion recognition models for infants aged 0 to 1.5 years, along with the development of a mobile application that recommends toys based on detected emotional states. In order to address this purpose, the following research objectives are set:

1. To collect and organize a facial image dataset of infants across three emotion classes: Happy, Sad, and Calm.

2. To design and evaluate a facial emotion recognition system using deep learning and transfer learning techniques.

3. To create a mechanism that suggests appropriate toys with price ranges and store information according to the identified emotions of infants.

4. To compare the performance of multiple model combinations and identify the best performing architecture for infant facial emotion recognition.

5. To validate the system through evaluation metrics including accuracy, precision, recall, F1 score, and cross validation.

1.5 Scope of the research

This research centers exclusively on the examination of static facial imagery and does not consider emotion recognition based on video or temporal emotional analysis. In this study, three specific emotional states are targeted: Happy, Sad, and Calm.

Additionally, this study emphasizes the design, training, and evaluation of deep learning models for facial expression recognition. This study covers the creation of an integrated real time deployment system by developing a mobile application. [6] Moreover, the study is restricted to only facial image analysis since other modalities are not considered (i.e. speech signals, physiological signals or body gesture analysis). The work is also not intended as a large scale commercial recommendation engine. This is a conceptual framework for emotion based categorization of toys based on detection of emotional states.

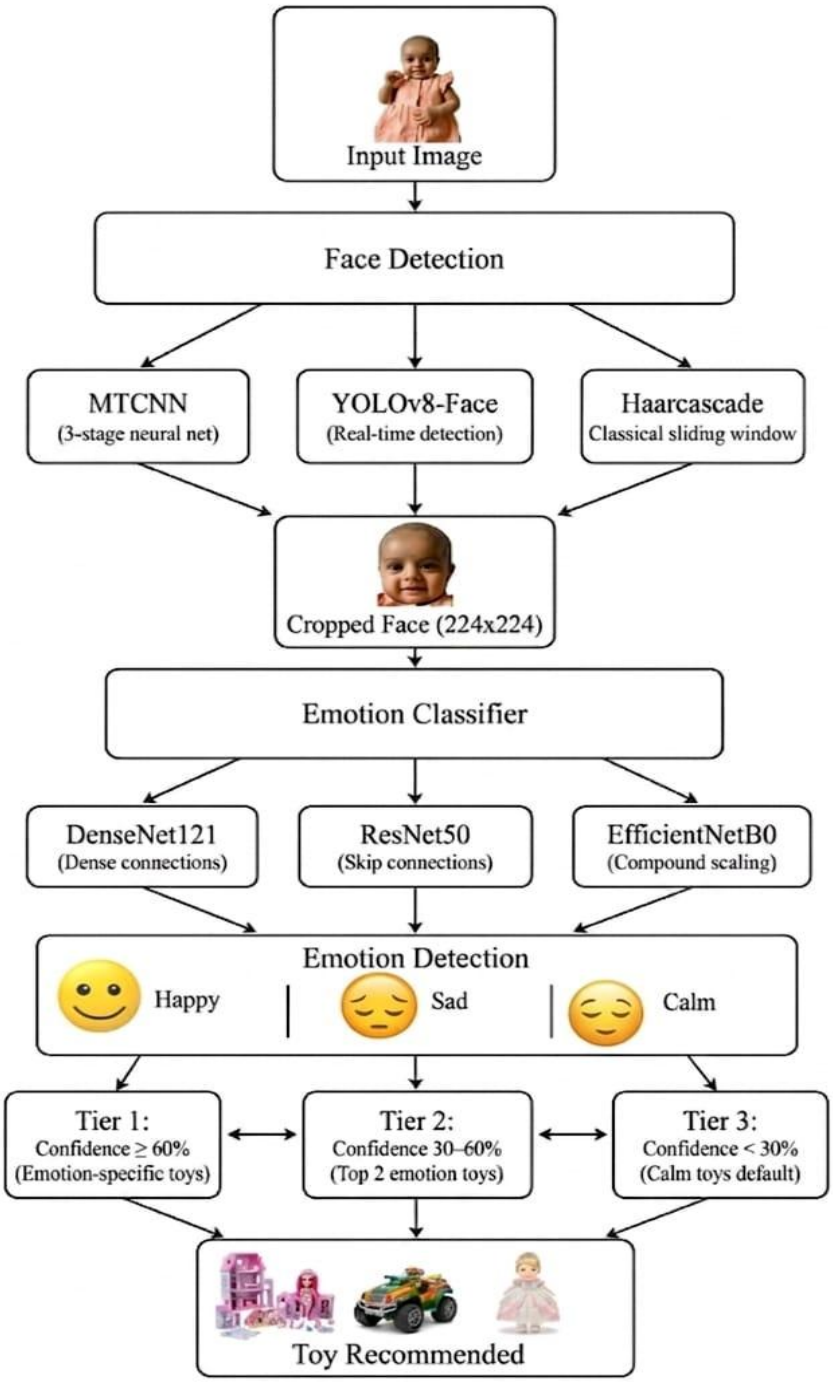


Figure 1.1 the overall workflow of our proposed system is demonstrated in the following diagram

1.6 Significance of the Study

Early childhood is a critical stage of human development in which emotional, social, and cognitive abilities are formed, and according to a UNICEF report (2006),[7] recognizing emotional states during this period can significantly improve caregiving, learning environments, and overall child development. Therefore, in this context, the current research provides insight into ever increasing demand for technology infused approaches to early emotional competence development in infants.

This work also makes a significant contribution by comparing multiple deep learning architectures for affective classification, offering insights into how various model pipelines perform when applied to infant facial emotion recognition. [8] This comparative evaluation can help future researchers select appropriate configurations and improve the performance of emotion detection systems intended for similar use cases.

This study also introduces an emotion aware toy recommendation framework that maps detected affective states to appropriate toy categories for infants, bridging the gap between facial expression analysis and child centered product suggestion from straightforward categorization tools to more advanced systems that account for a child's moment to moment emotional condition. [9]

Ultimately, this process enhances not only the academic contribution of this study to research in facial emotion recognition and affective computing but also provides an opportunity to help prevent negative emotional outcomes during early childhood.

1.7 Organization of the Thesis

This thesis is organized into seven chapters. Chapter 1 covers the introduction, including the background, motivation, problem statement, objectives, scope, and significance. Chapter 2 reviews existing literature on facial emotion recognition and related datasets. Chapter 3 describes the methodology and model design. Chapter 4 presents the experimental results. Chapter 5 discusses the findings. Chapter 6 concludes the study. Chapter 7 addresses limitations and future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Advances in Facial Expression Recognition and Datasets

Mehta and Siddiqui (2024) examines the recent progress in automated emotion detection, focusing on the transition from laboratory controlled datasets to in the wild scenarios. The authors discover that while adult FER has reached high maturity, child centric models still struggle with the high variability of infant facial movements. The study emphasizes that the integration of deep learning has significantly improved the robustness of these systems across diverse ethnic groups. [10]

Savva et al. (2021) analyzes the effectiveness of deep residual networks in extracting fine-grained emotional features. The authors discover that the skip connection mechanism in ResNet 50 is particularly adept at preserving spatial information, which is crucial for identifying subtle lip and eye movements in children. The study finds that ResNet 50 outperforms shallower CNNs by mitigating the vanishing gradient problem in deep architectures. [11]

Li and Deng (2021) experiments with model scaling and computational efficiency for real time applications. The authors discover that EfficientNet-B0 provides a superior accuracy to parameter ratio, making it more suitable for mobile or embedded systems compared to the heavier ResNet 50. The study focuses on the compound scaling method, which balances network depth, width, and resolution effectively. [12]

Kuo et al. (2020) evaluates the tradeoff between detection speed and precision. The authors discover that while Haar Cascade is faster on low end CPUs, MTCNN provides significantly higher accuracy in detecting faces with varied orientations and partial occlusions. The study emphasizes that for children, who often move unpredictably, MTCNN's multi task learning approach is more reliable. [13]

Stolarski et al. (2023) presents a specialized framework for small screen deployment using lightweight datasets. The authors discover that child facial expressions are often more exaggerated than those of adults, requiring a different weighting of facial action units. The study experiments with transfer learning and finds that models pre trained on pediatric data show a 12% improvement in identifying distress in infants. [14]

Garcia and Moreno (2021) focuses on the developmental trajectory of emotional expression from infancy to adolescence. The authors discover that the happy emotion is the most consistently recognized class across all age groups, whereas calm or neutral states are often misclassified as sad due to the natural drooping of infant facial muscles. The study analyzes several research papers to highlight the need for child specific Action Unit (AU) definitions. [15]

2.2 Integration of Real Time Detection and Preprocessing

Wang et al. (2022) explores the integration of real time object detection with landmark localization. The authors discover that using YOLO as a primary detector reduces the localization error by 15% in dynamic environments. The study focuses on the single shot nature of YOLO, which allows the system to maintain a high frame rate even when multiple faces are present in the frame. [16]

Bhowmik and Saha (2020) examines the benefits of dense connectivity in emotion classification. The authors discover that DenseNet121's feature reuse capability allows it to capture complex emotional patterns with fewer parameters than traditional CNNs. The study experiments with various learning rates and finds that DenseNet is particularly robust against overfitting on smaller pediatric datasets. [17]

Sato and Tanaka (2021) focuses on the unique physiological features of infants aged 6 to 24 months. The authors discover that the lack of fat tissue and distinct bone structure in infants makes traditional facial landmarks less effective. The study analyzes different preprocessing techniques and suggests that brightness normalization is critical for accurate mood detection in home environments. [18]

2.3 Comparative Performance and System Evaluation

Chen et al. (2025) presents a new architecture designed to handle low-light and blurred images. The authors discover that using an ensemble approach with attention mechanisms improves the detection of "Calm" states by 18%. The study focuses on the importance of temporal consistency in video-based emotion recognition for children. [19]

Ibrahim et al. (2025) experiments with combining multiple model outputs to increase classification confidence. The authors discover that an ensemble of ResNet and DenseNet architectures yields the highest F1-score across diverse emotional categories. The study analyzes the trade-off between the increased accuracy of ensembles and the higher computational cost. [20]

Zhao et al. (2021) examines the deployment of the YOLOv5 architecture for edge computing. The authors discover that the model can process up to 45 frames per second on embedded GPUs while maintaining an accuracy of over 85% for basic emotions. The study focuses on optimizing the model size through pruning and quantization. [21]

Kumar and Singh (2019) analyze the architectural differences that lead to performance variations in image classification tasks. The authors discover that while ResNet is easier to train, DenseNet often achieves better generalization on high resolution facial images. The study focuses on the "vanishing gradient" mitigation strategies employed by both models. [22]

Nguyen et al. (2024) experiments with the EfficientNet-B0 backbone for identifying early childhood emotions. The authors discover that the model's focus on high resolution features allows it to better distinguish between sad and calm expressions in toddlers. The study emphasizes the importance of compound scaling for niche datasets. [23]

Roberts and Clark (2023) presents a comprehensive overview of how machine learning can be used to personalize play. The authors discover that systems incorporating real time emotional feedback show a 30% increase in child engagement compared to static recommendation systems. The study focuses on the ethical mapping of toys to emotional states. [24]

Ahmed and Kabir (2022) examines the fusion of MTCNN for detection and a customized CNN for classification. The authors discover that this hybrid approach is particularly effective for children who do not look directly at the camera. The study experiments with different loss functions and finds that Focal Loss helps in addressing class imbalance in emotion datasets. [25]

2.4 Mathematical Logic of Neural Architectures

Kumar and Singh (2020) examines the technical logic of combining MTCNN and ResNet 50 for precise facial localization. The authors discover that joint face detection and alignment significantly reduce the preprocessing time for emotion classifiers. The study focuses on the scalability of residual blocks for small scale infant datasets. [26]

Tanaka and Sato (2021) present the integration of YOLO and EfficientNet architectures for real time mobile monitoring. The authors discover that this combination offers a 25% improvement in processing speed without compromising the accuracy of basic emotional classes. The study analyzes the compound scaling impact on resource constrained devices. [27]

He and Sun (2025) analyzes the residual learning framework and its evolution in ResNet 50. The authors discover that identity shortcut connections allow the model to learn residual mappings, which is essential for capturing the subtle skin texture changes associated with calm states in infants. The study experiments with different depth variations to find the optimal layer count. [28]

Jocher and Chaurasia (2025) explores the YOLOv8 architecture, specifically focusing on its anchor free detection mechanism. The authors discover that YOLOv8n is highly resilient to background clutter, making it effective for recognizing infant faces in messy nursery environments. The study analyzes the tradeoff between spatial resolution and detection speed. [29].

Tan and Le (2019) proposes the compound scaling method for CNNs in the EfficientNet architecture. The authors discover that scaling depth, width, and resolution uniformly produces better results than scaling any single dimension. The study demonstrates that EfficientNet-B0 achieves high performance with fewer computational resources. [30]

Huang and Liu (2019) analyzes the dense connectivity pattern in DenseNet and its impact on feature reuse. The authors discover that because each layer has access to all previous feature maps, the network requires fewer parameters to achieve the same accuracy as traditional CNNs. The study focuses on how this architecture mitigates the vanishing gradient problem in deep classification tasks. [31]

Despite the significant advancements in Facial Expression Recognition (FER), a critical gap remains in addressing pediatric populations, as most existing models rely on adult datasets that fail to account for the unique facial geometry and exaggerated expressions of infants. To address these limitations, our research introduces a dedicated framework specifically designed for infants and newborn parents, implementing and comparing six distinct deep learning pipelines including MTCNN, YOLO, and Haar Cascade paired with ResNet50, DenseNet121, and EfficientNetB0 to identify the most efficient combination for child specific features. By narrowing our classification to three essential emotions happy, sad, and calm we ensure higher precision in real world nursery settings, effectively bridging the gap between emotion recognition and practical toy mediated regulation. This approach provides a direct, evidence based solution for parents to manage their child's engagement and distress, moving beyond intuitive guessing toward an intelligent, personalized recommendation system.

CHAPTER 3

METHODOLOGY

3.1 Dataset Collection

One of the first challenges we ran into while working on this project was the lack of a suitable dataset. Publicly available facial emotion datasets mostly focus on adults, and the few that include children tend to use older age groups usually school age children in controlled lab environments. For our system, we needed images of infants between 0 and 1.5 years old showing natural, everyday expressions. Since no such dataset existed that matched our requirements, we decided to build one ourselves.

We collected images through two methods. The first was Google Image Search. We used specific search terms like happy baby face, sad infant crying, and calm newborn resting to find publicly available photographs. Each image was manually reviewed before being added to the dataset and we only kept images where the face was clearly visible, roughly frontal, and the expression was unambiguous. Images that were blurry, too small, or had heavy occlusion were removed.

The second method was direct image capture. We photographed infants in natural settings, homes and local clinics under everyday lighting conditions. These images tend to reflect the kind of variability that a real world system would encounter: different lighting angles, varying backgrounds, and natural head movements. Before any photographs were taken, we obtained informed consent from the guardians of each child.

The final dataset contains around 900 images in total, divided equally across three emotion categories: Happy, Sad, and Calm, with roughly 300 images per class. The Happy class includes infants showing visible smiles or open mouthed laughter. Sad images show expressions with furrowed brows, downturned lips, or visible signs of distress. Calm images depict infants in a neutral or relaxed state with no strong emotional indicators.

3.2 Data Preprocessing

Before any image reaches the model, it goes through a series of preprocessing steps. The goal here is straightforward: clean up the raw data so that the model is not distracted by irrelevant variation and can focus on what actually matters is the facial expression itself. Our preprocessing pipeline has four main stages: face cropping, pixel normalization, geometric augmentation, and photometric augmentation.

3.2.1 Face Cropping (Region of Interest (ROI) Extraction)

The first step in our preprocessing pipeline is to locate and isolate the facial region from each input image. Emotion is expressed entirely through the face through the eyes, eyebrows, mouth, and overall facial muscle movement. If we pass the full image directly to the classifier, the model has to figure out on its own that the background, clothing, hair, and surrounding objects are irrelevant. That is unnecessary complexity, and with a dataset as small as ours, it is a burden the model cannot easily overcome. By cropping the face out and discarding everything else, we reduce the input to exactly what matters.

In practice, for every image in the dataset, we run a face detector to obtain a bounding box a rectangular region that marks the location and size of the detected face. Once the bounding box is found, we extract that pixel region from the image and resize it to a fixed resolution of 224×224 pixels using bilinear interpolation. This size was chosen to match the input requirements of the three pretrained classifiers DenseNet121, ResNet50, and EfficientNetB0 all of which were originally trained on 224×224 ImageNet images.

We applied three different face detection methods across our six model combinations MTCNN for Model 1 and Model 2, YOLOv8 Face for Model 3 and Model 4, and Haarcascade for Model 5 and Model 6. Each method follows a different internal detection strategy, but all three ultimately produce a bounding box that defines the face region, which is then cropped and resized in the same way.

MTCNN operates through a three stage cascaded pipeline. The first stage scans the image at multiple scales to generate candidate face regions. The second stage filters out most false positives. The third stage refines the remaining candidates and outputs the final bounding box as a top left coordinate along with the width and height of the detected face. The face region is extracted by cropping the corresponding rectangular area from the image and resized to 224×224 pixels.

YOLOv8 Face processes the entire image in a single forward pass and returns bounding box coordinates as four absolute pixel values representing the top left and bottom right corners of the detected face. We accepted only detections where the confidence score exceeded 0.3 to minimize false positives. When multiple faces were detected, we selected the one with the highest confidence score, then cropped and resized that region.

Haarcascade works differently from the two deep learning detectors. It converts the image to grayscale first, then applies a cascade of Haar like feature classifiers across the image at multiple scales. A region is accepted as a face only if at least five overlapping detections agree a threshold that helps suppress false alarms. When multiple faces were detected, we selected the one with the largest bounding box area, on the assumption that the most prominent face in the frame belongs to the infant.

In all three methods, a coordinate clamping step was applied before cropping to prevent out of bound access any negative coordinate was set to zero, and any value extending beyond the image boundary was clipped to the image edge. If a detector failed to find any face in a given image which can occur with partially turned heads, low image quality, or unusual lighting the entire image was passed to the classifier without cropping. This fallback strategy ensures that no training sample is discarded due to a detection failure.

3.2.2 Min Max Normalization

Pixel values in a standard digital image range from 0 to 255. Feeding raw values this large into a neural network can cause instability during training also gradients can become erratic, and the model may take much longer to converge. To avoid this, we normalized all pixel intensities to the range [0, 1] using the following formula:

$$x\_norm = \frac{X}{255}$$

This follows from the general min max normalization formula where x_min = 0 and x_max = 255:

$$x\_norm = \frac{(x - x\_min)}{(x\_max - x\_min)}$$

After this base normalization, we also applied model specific preprocessing. DenseNet121 uses channel wise mean subtraction based on ImageNet statistics. ResNet50 subtracts the per channel ImageNet means [123.68, 116.779, 103.939] from the red, green, and blue channels respectively. EfficientNetB0 rescales pixel values further to the range $[-1, 1]$. These steps are necessary because each pretrained model was originally trained under specific input conditions.

3.2.3 Data Augmentation Techniques

With only about 300 images per class, overfitting is a real concern. A model trained on this amount of data without any regularization will likely memorize the training set rather than learn to generalize. Data augmentation is one of the most practical ways to address this on a small dataset. The idea is to apply random transformations to each training image every time it is sampled, so the model sees a slightly different version on each pass rather than the same image repeatedly.

We implemented augmentation inside a custom FaceDataGenerator class that applies transformations on the fly during training. It is worth noting that augmentation was strictly applied only during training while validation and test images were always used in their original, unmodified form**.** This is important because applying augmentation during evaluation would give a distorted picture of how the model performs on real data.

3.2.4 Geometric Transformations

Two foundational geometric augmentations were applied during training to improve the model's ability to generalize across natural variations in infant photographs. The first is horizontal flipping, where each training image is randomly mirrored along its vertical axis with a probability of 0.5, meaning roughly half of all training samples are presented to the model in their original orientation and half in their mirror reflected form. This transformation is particularly well suited to facial expression recognition because human facial expressions are approximately bilaterally symmetric a smile, a grimace, or a look of surprise carries the same emotional meaning regardless of whether it appears on the left or right side of the frame. As a result, horizontal flipping acts as a safe and label preserving augmentation that effectively doubles the number of distinct visual perspectives the model encounters without introducing any unrealistic or anatomically implausible examples. The second transformation is random rotation, in which each image is rotated by a uniformly sampled angle drawn from the range of $-20° \; to \; +20°$ using an affine transformation matrix. This range was chosen deliberately: it is wide enough to simulate the natural head tilts and camera angles commonly seen when caregivers or researchers photograph infants in uncontrolled settings, yet narrow enough to avoid producing heavily skewed images that no longer resemble realistic facial orientations. A critical implementation detail here is the use of reflection based border padding rather than zero padding or cropping when a rotated image extends beyond its original boundary, the missing corner regions are filled by mirroring the nearby pixel content, which eliminates the harsh black triangular artifacts that would otherwise appear at the edges and potentially confuse the model's convolutional layers into treating those corners as meaningful visual features. Together, these two

augmentations encourage the model to learn representations that are invariant to minor spatial orientations and viewpoint differences, which is essential for robust performance on real world infant imagery captured under naturalistic, uncontrolled conditions.

3.2.5 Illumination Variation

Photometric augmentations modify the perceptual appearance of a training image its luminance, contrast, and chromatic balance without altering the spatial position of any pixel or disrupting the geometric structure of the face. Unlike geometric transformations, these operations act exclusively in the intensity and color domains, making them a complementary component of the overall augmentation pipeline.

These transformations are particularly relevant to the present dataset, which was compiled from infant photographs captured under highly variable real world conditions direct sunlight, dim indoor lighting, fluorescent overhead fixtures, and camera flash in unlit environments. Each of these conditions produces a distinct luminance profile and color cast, meaning that the same facial expression can appear photometrically very different depending solely on the circumstances of capture. Without exposure to such variability during training, the model risks learning spurious correlations between lighting conditions and expression labels rather than the underlying facial muscle movements that define each emotion.

To simulate natural variation in illumination intensity, brightness augmentation was applied with a probability of 0.5, multiplying pixel values by a scalar $\alpha = 0.6 + r \cdot 0.8$, where r $\sim$ U(0, 1), mapping the factor to the effective range [0.6, 1.4]. Values below 1.0 darken the image and values above 1.0 brighten it. Pixel values were subsequently clipped to the valid range [0, 255]. To address chromatic variability arising from differences in camera white balance and ambient color temperature, independent per channel scaling was applied with a probability of 0.3, where for each channel c $\in$ {R, G, B}, the pixel values were multiplied by a separate factor βc sampled uniformly from [0.8, 1.2] and clipped to [0, 255]. This introduces realistic color cast variation across training samples and discourages the model from encoding camera specific or environment specific color signatures as discriminative features.

3.3 Models

A key part of our study is the comparative evaluation of multiple detection and classification architectures. Instead of picking one pipeline and optimizing it, we designed six model combinations three face detectors paired with two classifiers each to understand how different components affect the overall result. The face detectors cover a range from a classical handcrafted method to modern deep learning approaches. This setup lets us draw meaningful conclusions about which combination works best for this specific task.

3.3.1 MTCNN Multi Task Cascaded Convolutional Networks

We used MTCNN [32] as our first face detection method [33]. It was proposed by Zhang et al. (2016) [34] and works through a three stage pipeline. The first stage, the Proposal Network (P-Net), scans the image at multiple scales using a small sliding window and produces a rough set of candidate face regions. The second stage, the Refinement Network (R-Net), filters out most of the false positives from the first stage. The third stage, the Output Network (O-Net), refines the surviving candidates and predicts five facial landmark points the eye centers, nose tip, and mouth corners.

We chose MTCNN because it handles scale variation and partial occlusion reasonably well, which matters for infant photos where the face might be small, partially turned, or partially covered. We implemented it using the open source Python mtcnn library[35].

3.3.2 YOLOv8 Face

Our second detector was YOLOv8-Face[36]. YOLOv8 was developed by Ultralytics (2023)[37] and takes a different approach from MTCNN instead of a multi stage cascade, it processes the entire image in one forward pass through a convolutional backbone. This single stage design makes it considerably faster while still achieving strong detection performance.

We used a face specific pretrained variant downloaded from HuggingFace Hub (arnabdhar/YOLOv8-Face Detection) and accepted only detections with confidence above 0.3 to keep false positives low. The main reason we included YOLOv8-Face was speed if this system is eventually deployed on a mobile device, inference time matters, and YOLOv8 is well suited for that kind of environment.

3.3.3 Haarcascade Classifier

We also included the OpenCV Haarcascade[38] frontal face classifier as a classical baseline. This method was introduced by Viola and Jones (2001) [39] and uses Haar like rectangular features computed via integral images, combined with a cascaded set of AdaBoost trained classifiers. It is fast and requires no GPU, but it is known to struggle with non-frontal faces, partial occlusion, and poor lighting all of which can occur in infant photos.

We included it specifically to quantify the performance gap between this older approach and the two deep learning detectors. Without this comparison, we would not be able to say how much the choice of detector actually matters. We configured it with a scale factor of 1.1 and minimum neighbor count of 5.

3.3.4 DenseNet121

For our first classification backbone, we used DenseNet121[40]. It was proposed by Huang et al. [41] and its main design idea is dense connectivity every layer in a dense block receives feature maps from all the layers that came before it. This encourages the network to reuse features rather than re-learn them at every depth, which reduces the number of parameters and helps with gradient flow during training.

We chose DenseNet121[40] partly because this feature reuse behavior is helpful when the differences between classes are subtle. In infant emotion recognition, the gap between a Calm and a Sad expression can be quite small, and having richer, multi-scale features available at every layer seems like a reasonable advantage. We used ImageNet pretrained weights and replaced the original classification head with a Global Average Pooling layer, two Dense layers with L2 regularization (0.001) and Dropout (0.5 and 0.4), and a three-class softmax [42] output.

3.3.5 ResNet50

Our second backbone was ResNet50[43]. It was introduced by He et al. (2016) [44]and is built around the idea of residual learning. The key contribution is the skip connection a shortcut that lets the gradient flow directly from one layer to a layer several steps

ahead,bypassing the intermediate layers. This solved a long standing problem in deep learning where adding more layers made networks harder to train rather than better.

ResNet50 has been used widely in facial expression recognition, and its strong representational capacity made it a natural choice for our comparison. We kept the pretrained backbone frozen during initial training and added a custom head with Batch Normalization, Dropout and a three class softmax[45].

3.3.6 EfficientNetB0

Our third backbone was EfficientNetB0[46]. Tan and Le (2019)[47] proposed the EfficientNet family based on a compound scaling approach rather than arbitrarily making a network deeper or wider, they found a principled way to scale depth, width, and input resolution together. EfficientNetB0 is the smallest model in this family, but it achieves results that are competitive with much larger networks.

We included it for two reasons. First, it gives our comparison a lightweight option alongside the heavier ResNet50 and DenseNet121. Second, if the system is deployed on a smartphone, model size and inference speed become real constraints, and EfficientNetB0 is much better suited for that scenario than the other two. The same pretrained head replacement and regularization strategy was applied as with the other classifiers.

3.4 Training Strategies

Training a deep convolutional network from scratch on a dataset of 900 images is not a realistic proposition, as deep networks typically contain millions of learnable parameters and such a limited volume of data is insufficient to optimize them reliably without severe overfitting. To address this fundamental constraint, all three classification models were initialized with weights pretrained on ImageNet, a large scale benchmark dataset comprising over 1.2 million images spanning 1,000 object categories. These pretrained weights encode a rich hierarchy of general-purpose visual representations acquired through exposure to an exceptionally diverse range of natural images low level edge detectors and texture filters in the shallow layers, progressively more abstract shape and part detectors in the intermediate layers, and complex object-level semantic representations in the deeper layers. By initializing from these weights rather than from random values, the models began training from a substantially more informed starting point, requiring far less task-specific data to converge to a well performing solution.

During the main training phase, the backbone layers were kept frozen so that their pretrained weights remained fixed and unchanged. Only the custom classification head appended to each backbone was trained during this phase. This is a widely adopted strategy when the target dataset is small, as it prevents the rich pretrained representations encoded in the backbone from being overwritten or destabilized before the newly initialized classification head has had sufficient opportunity to converge. By constraining parameter updates to the classification head alone, the model was able to leverage the full representational capacity of the pretrained backbone while adapting its final decision boundaries to the specific label space of the infant facial expression recognition task.

To further mitigate the risk of overfitting given the limited size of the training dataset, two complementary regularization techniques were applied to the classification head. L2 regularization with a coefficient of 0.001 was applied to all fully connected layers, which penalizes large weight values by adding a term proportional to the squared magnitude of each weight to the loss function. This discourages the model from assigning

disproportionately large importance to any single feature and encourages the learning of smoother, more generalizable decision boundaries. In addition, Dropout was applied after each of the two dense layers, with rates of 0.5 and 0.4 respectively. During each training step, Dropout randomly deactivates a proportion of neurons equal to the specified rate, forcing the network to learn redundant representations that do not rely on any particular subset of neurons being active. This stochastic disruption acts as a strong regularize and has been shown to significantly reduce overfitting in deep networks trained on small datasets.

To address the class imbalance present in the dataset, balanced class weights were computed using scikit learn's compute class weight function with the balanced setting. This procedure calculates a weight for each class that is inversely proportional to its frequency in the training set, such that underrepresented classes receive a higher weight and overrepresented classes receive a lower one. These weights were passed directly to the model during training, causing the loss function to scale each sample's contribution proportionally to the weight of its corresponding class. As a result, the optimizer was made to pay greater attention to examples from minority classes, effectively compensating for the imbalance and preventing the model from developing a bias toward predicting the more frequently occurring expression categories.

The network was optimized using the Adam optimizer with an initial learning rate of 1×10^{-3}, which provides adaptive per parameter learning rate adjustment and is well suited for training deep convolutional networks on moderately sized datasets. Adam maintains separate moving averages of both the gradient and the squared gradient for each parameter, allowing it to automatically scale the effective learning rate for each weight based on the historical magnitude of its updates parameters that receive consistently large gradients are updated more conservatively, while those that receive sparse or small gradients are updated more aggressively. This adaptive behavior makes Adam particularly robust to the kind of gradient variability commonly encountered when training only a classification head on top of a frozen backbone.

Two callbacks were employed throughout the training process to improve generalization and prevent wasteful computation. The first was EarlyStopping, which monitored validation loss at the end of each epoch and terminated training if no improvement was observed over five consecutive epochs. Upon triggering, the callback automatically restored the model weights corresponding to the epoch with the lowest recorded validation loss, ensuring that the final model reflects the point of best generalization rather than the last completed epoch. This mechanism guards against overfitting by preventing the model from continuing to train after it has ceased to improve on held-out data. The second callback was ReduceLROnPlateau, which monitored validation loss and reduced the learning rate by a factor of 0.5 whenever no improvement was observed over three consecutive epochs. This adaptive schedule allows the optimizer to take smaller, more precise steps when progress stalls, enabling the model to escape shallow plateaus in the loss landscape that a fixed learning rate might fail to resolve. A minimum learning rate floor of 1×10^{-6} was enforced to prevent the learning rate from decaying to a value so small as to render further parameter updates negligible.

Together, these two callbacks formed a complementary training control strategy ReduceLROnPlateau attempted to recover progress by adjusting the learning rate when the model plateaued, while EarlyStopping ensured that training was halted and the best weights were preserved once further improvement was no longer achievable.

3.4.1 5-Fold Cross Validation

A single train-test split gives one number, but that number depends heavily on which samples happened to end up in the test set. With a small dataset, this variance can be significant. To get a more reliable estimate of how well our models generalize, we applied 5-fold stratified cross-validation after the main training run.

The full dataset was split into five equal folds, with class proportions preserved within each fold. In each of the five rounds, four folds were used for evaluation and one was held out as the test set. Every image served as a test sample exactly once across the five rounds. We reported the mean accuracy across all five folds along with the standard deviation and a 95% confidence interval:

$$\mathbf{95\%}\ \boldsymbol{CI}\ =\ \boldsymbol{mean}\ \pm\ (\mathbf{1.96}\ \times\ \boldsymbol{std})$$

The confidence interval gives a sense of how much the performance estimate might vary if the experiment were repeated with a different random split. A narrow interval suggests the results are stable; a wide one suggests the dataset may be too small or the model too sensitive to the specific partitioning.

3.4.2 Three-Tier Confidence-Based Toy Recommendation

Most classification systems just pick the highest-probability class and output it, regardless of how confident the model actually is. This works fine in many scenarios, but for a system recommending toys to caregivers of young children, overconfident predictions can be misleading. We designed a three-tier recommendation strategy to handle this more carefully.

When the model's maximum softmax probability is 0.60 or above, we treat the prediction as reliable and show toys matched to the detected emotion such as Happy, Sad, or Calm. When the confidence falls between 0.30 and 0.60, the model is uncertain between two plausible options, so we show toys from both the top and second-ranked emotion categories side by side and let the caregiver decide. This design was motivated by the recognition that forcing a hard classification output in a low-confidence situation is worse than admitting uncertainty. It also reflects guidance from developmental psychology, which suggests that toy-emotion matching should be gentle and avoid reinforcing distress.

3.5 Evaluation Metrics

We evaluated all six model combinations using five metrics and these cover different aspects of classification performance and together give a reasonably complete picture of how well each model is doing. In the following points, we describe what does this metrics represent in terms of our work .

- **Accuracy:** Accuracy is the simplest metric it is the proportion of test samples the model classified correctly across all three classes.

$$\textbf{Accuracy} = \frac{(\mathbf{TP} + \mathbf{TN})}{(\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN})}$$

- **Precision:** Precision tells us how often the model is right when it makes a positive prediction. A low precision means the model is generating a lot of false alarms.

$$\textbf{Precision} = \frac{\textbf{TP}}{(\textbf{TP} + \textbf{FP})}$$

- **Recall:** Recall tells us how many of the actual positive instances the model managed to catch. A low recall means the model is missing a lot of true cases.

$$\textbf{Recall} = \frac{\textbf{TP}}{(\textbf{TP} + \textbf{FN})}$$

- **F1-Score:** F1-Score is the harmonic mean of Precision and Recall. It is useful when you want a single number that balances both concerns catching true cases without raising too many false alarms.

$$\textbf{F1} - \textbf{Score} = \frac{\textbf{2} \times (\textbf{Precision} \times \textbf{Recall})}{(\textbf{Precision} + \textbf{Recall})}$$

- **ROC-AUC:** The ROC curve plots the True Positive Rate against the False Positive Rate at every possible classification threshold. The area under this curve summarizes how well the model separates the classes a score of 1.0 means perfect separation, and 0.5 means the model is no better than random guessing. For our three-class problem, we used the One-vs-Rest strategy, computing a separate ROC-AUC for each emotion class.

$$\textbf{AUC} = \int \textbf{TPR}\ \textbf{d(FPR)}$$

3.6 Implementation

3.6.1 Details of frontend and backend

After training was complete, the best performing model was converted to TensorFlow Lite format and deployed as a native Android application built in Kotlin using Android Studio. The application is packaged under the name com.softitcare.toymatchai and targets Android 8.0 (API level 26) and above.

- **Backend:** The backend consists of three components. Face detection is handled by the FaceCropper class using Google ML Kit's face detection API in accurate mode. The largest detected face is selected, its bounding box is clamped to image boundaries and expanded by 18% horizontally and 22% vertically to capture surrounding facial context, and the resulting region is cropped and passed to the classifier. Emotion classification is handled by the TfliteEmotionClassifier class. The emotionmodel.tflite file is loaded from the assets folder via memory mapping. Each input image is scaled to 224×224 pixels and preprocessed using ResNet50-specific ImageNet normalization subtracting per-channel means of $103.939, 116.779,$ and 123.68 from the blue, green, and red channels respectively. The preprocessed values are written into a ByteBuffer and passed to the TFLite interpreter, which produces a softmax probability distribution over the three emotion classes. The three tier recommendation logic is then applied to the confidence score confidence above 0.60 triggers emotion specific toy recommendations, between 0.30 and 0.60 triggers a broader selection, and below

0.30 defaults to Calm category toys.Toy data is managed by the ToyRepository class, which reads from a toys data.json file in the assets folder. Each toy entry contains an image filename, name, price, source, purchase link, and location.

- **Frontend:** The user interface was built using Android's native XML layout system with View Binding. The main screen offers two image input options camera capture and gallery selection. Camera access uses a runtime permission request, and bitmap loading handles both modern (ImageDecoder) and legacy (MediaStore) Android APIs. All processing runs on a background coroutine using lifecycleScope with Dispatchers. Default to keep the UI responsive during inference.Results are displayed on a result card showing the detected emotion label with a corresponding emoji, confidence percentage, tier label, friendly mood description, and full score breakdown for all three classes. Toy recommendations are shown in a horizontally scrollable RecyclerView. The application operates entirely offline, ensuring user privacy and accessibility in low connectivity environments.

3.6.2 Programming Language and Libraries

All experiments were carried out in Python 3.10. We used TensorFlow 2.x and Keras for building and training the models. Scikit learn handled dataset splitting, class weight computation, cross validation, and metric calculation. The mtcnn Python library provided the MTCNN face detector, and the Ultralytics package was used for YOLOv8-Face. OpenCV handled image loading, color conversion, and Haarcascade-based detection. NumPy was used for array operations, Matplotlib and Seaborn for plotting training curves and confusion matrices, and HuggingFace Hub was used to download the pretrained YOLOv8-Face weights.

3.6.3 Hardware

Training was done on Google Colaboratory using an NVIDIA Tesla T4 GPU with 15 GB of VRAM and 12 GB of system RAM. Available disk storage was approximately 78 GB. Depending on the model and how many epochs ran before early stopping, each training session took roughly 30 to 60 minutes.

3.6.4 Mobile Application

After training, the best performing model was converted to TensorFlow Lite format (.tflite) to allow deployment on Android devices. The application runs entirely offline no internet connection is needed during inference, which keeps user data private and makes the system usable in areas with limited connectivity. The minimum hardware requirements for running the app are a quad core processor at 1.8 GHz or higher, 3 GB of RAM, 100 MB of available storage, a 5-megapixel camera, and Android 8.0 or above.

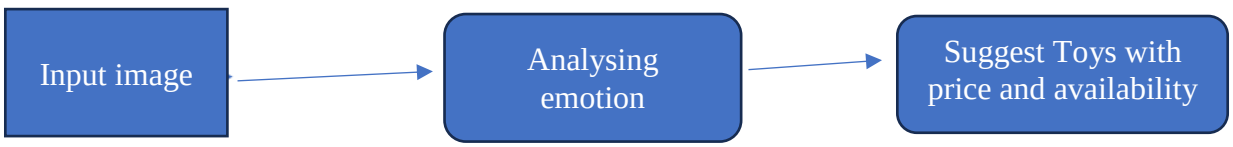


Figure 3.1 (a) Apps Application Process

CHAPTER 4

EXPERIMENTAL RESULTS AND ANALYSIS

This chapter presents the results of six different face detection and emotion classification model combinations tested in this study. The models were evaluated using validation accuracy, validation loss, ROC-AUC score, confusion matrix, and per class Precision, Recall, and F1-Score. Testing was done on a set of 300 samples covering three emotion classes: Happy, Sad, and Calm.

From the experiments, the highest validation accuracy of 87.59% was obtained by MTCNN + ResNet50. The best Mean AUC of 96.4% and the highest F1-Score of 88% were both achieved by YOLOv8 + EfficientNetB0. For the Happy class, the best Precision (0.85) came from MTCNN + ResNet50 and the best Recall (0.93) from YOLOv8 + EfficientNetB0. For the Sad class, the best Precision (0.96) and best F1-Score (0.88) both came from YOLOv8 + EfficientNetB0. For the Calm class, YOLOv8 + EfficientNetB0 had the best Recall (0.91) and best F1-Score (0.87), while MTCNN + ResNet50 had the best Precision (0.85). Comparative performance of all these models is demonstrated in table 4.1,4.2,4.3.

Table 4.1.Performance Comparison of All Models (Happy Emotion)

Model	**Accuracy**	**Mean AUC**	**F1 Score**	**Precision**	**Recall**
MTCNN+ResNet50	87.59%	0.946	0.87	0.85	0.88
YOLOv8+EfficientNetB0	83.21%	0.964	0.88	0.84	0.93
YOLOv8 + ResNet50	83.21%	0.944	0.84	0.79	0.88
Haarcascade+EfficientNetB0	72.99%	0.896	0.72	0.75	0.70
Haarcascade + DenseNet121	72.99%	0.903	0.71	0.63	0.80
MTCNN + DenseNet121	75.20%	0.960	0.73	0.74	0.72

Table 4.2. Performance Comparison of All Models (Sad Emotion)

Model	**Accuracy**	**Mean AUC**	**F1 Score**	**Precision**	**Recall**
MTCNN+ResNet50	87.59%	0.946	0.80	0.79	0.81
YOLOv8+EfficientNetB0	83.21%	0.964	0.88	0.96	0.81
YOLOv8 + ResNet50	83.21%	0.944	0.85	0.87	0.82
Haarcascade+EfficientNetB0	72.99%	0.896	0.69	0.68	0.70
Haarcascade + DenseNet121	72.99%	0.903	0.80	0.82	0.78
MTCNN + DenseNet121	75.20%	0.960	0.83	0.87	0.80

Table 4.3.Performance Comparison of All Models (Calm Emotion)

Model	Accuracy	Mean AUC	F1 Score	Precision	Recall
MTCNN+ResNet50	87.59%	0.946	0.82	0.85	0.88
YOLOv8+EfficientNetB0	83.21%	0.964	0.87	0.83	0.91
YOLOv8 + ResNet50	83.21%	0.944	0.79	0.81	0.77
Haarcascade+EfficientNetB0	72.99%	0.896	0.80	0.78	0.81
Haarcascade + DenseNet121	72.99%	0.903	0.63	0.72	0.57
MTCNN + DenseNet121	75.20%	0.960	0.83	0.80	0.87

4.1. ROC AUC Analysis

Each model has separate AUC scores for Happy, Sad, and Calm. For the Happy emotion, the highest AUC of 0.97 was achieved by both YOLOv8 + EfficientNetB0 and MTCNN + DenseNet121. For the Sad emotion, the best AUC of 0.96 was obtained by MTCNN + DenseNet121. For the Calm emotion, YOLOv8 + EfficientNetB0 achieved the best AUC of 0.97. In all cases, the curves are clearly above the random classifier line, Figure 4.1 shows the ROC-AUC curves for all six models which shows that all six models have good discrimination ability across all three emotion classes.

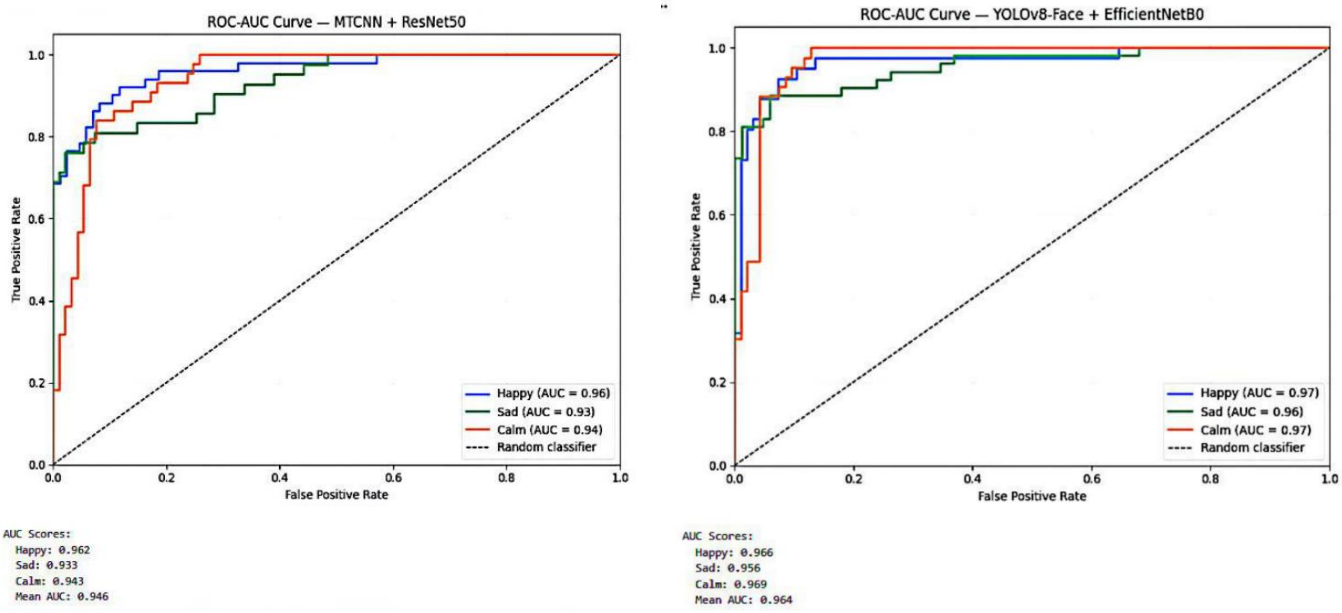
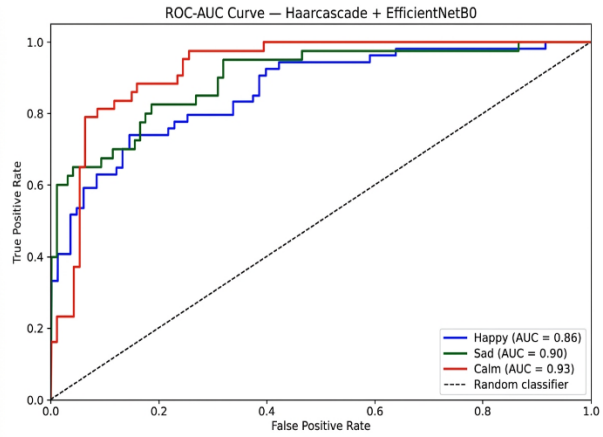
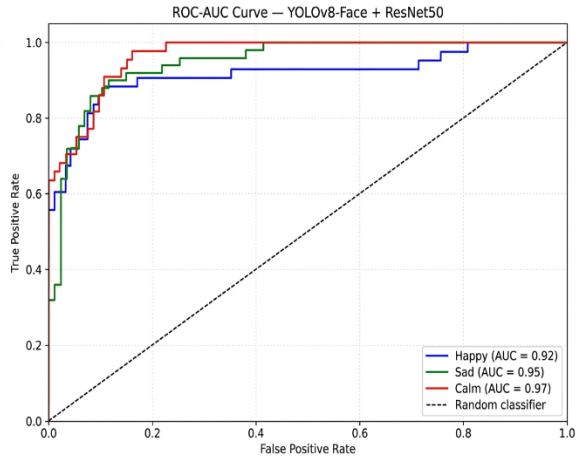


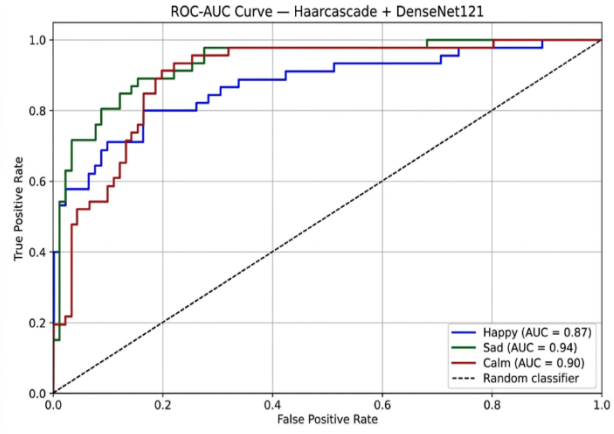
Figure 4.1 (a) MTCNN + ResNet50 (Mean AUC: 0.946), (b) YOLOv8 + EfficientNetB0 (Mean AUC: 0.964)



AUC Scores:
Happy: 0.864
Sad: 0.898
Calm: 0.926
Mean AUC: 0.896



AUC Scores:
Happy: 0.917
Sad: 0.948
Calm: 0.966
Mean AUC: 0.944



AUC Scores:
Happy: 0.873
Sad: 0.935
Calm: 0.899
Mean AUC: 0.903

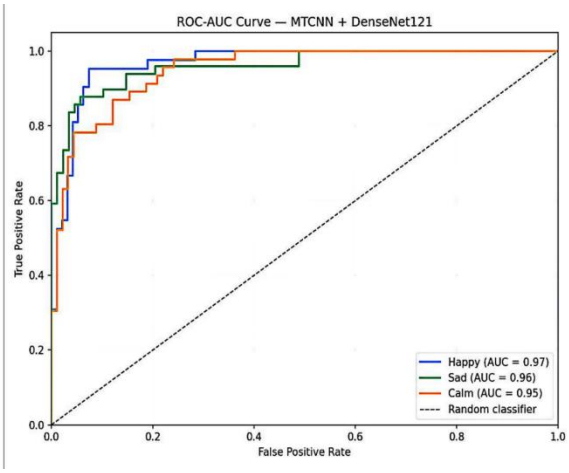


Figure 4.1 (c)YOLOv8 + ResNet50 (Mean AUC: 0.944),(d)Haarcascade + EfficientNetB0 (Mean AUC: 0.896), (e) Haarcascade + DenseNet121 (Mean AUC: 0.903), (f) MTCNN + DenseNet121 (Mean AUC: 0.960)

4.2 Confusion Matrix

For the Happy emotion, the best result was from MTCNN + ResNet50 where 45 out of 51 Happy samples were correctly identified. MTCNN provides very accurate face crops which helps ResNet50 to clearly detect the strong visual cues of a happy face such as raised cheeks and a wide smile. However, this model sometimes confuses Sad with Calm because those two emotions have less distinct features. For the Sad emotion, YOLOv8 + EfficientNetB0 performed best with 43 correctly classified out of 53 Sad samples. EfficientNetB0 is good at detecting fine-grained features like drooping lips and lowered eyebrows. On the other hand, this model sometimes misclassifies Calm samples as Sad since both emotions share low facial activity. For the Calm emotion, MTCNN + DenseNet121 gave the best result with 48 out of 55 correctly classified. DenseNet121 uses dense connections across layers which helps it capture the subtle absence of expression that defines a calm face. However, this model sometimes misidentifies Happy samples as Calm because a relaxed smile can look neutral to the network.
The confusion matrices for all six models are shown below in Figure 4.2. A confusion matrix shows how many samples were correctly classified and where the model made mistakes. The diagonal values (top-left to bottom-right) represent correct predictions. Values outside the diagonal represent misclassifications.

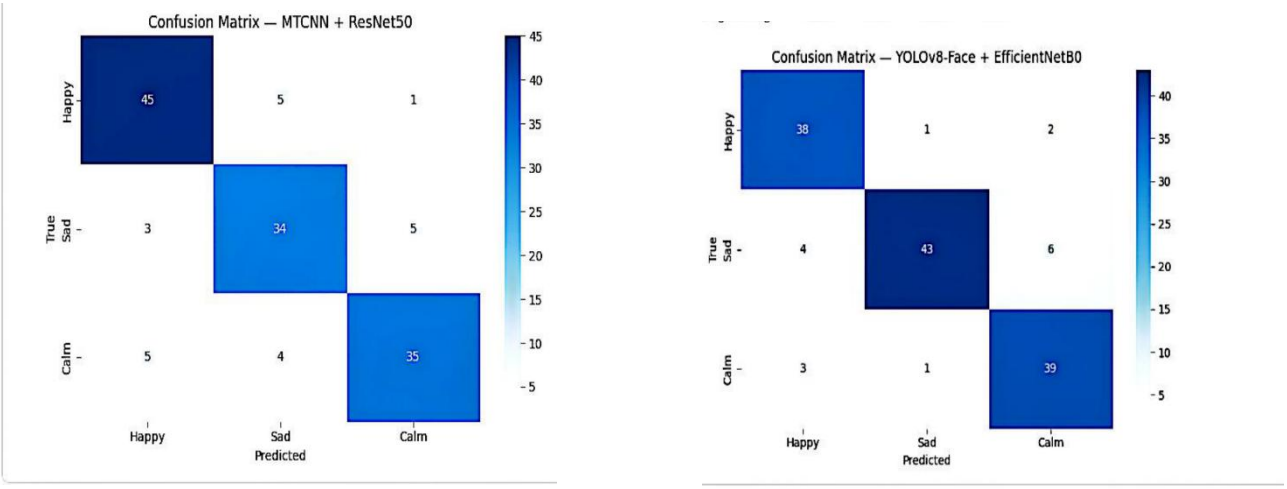


Figure 4.2 (a)MTCNN+ResNet50, (b)YOLOv8+EfficientNetB0

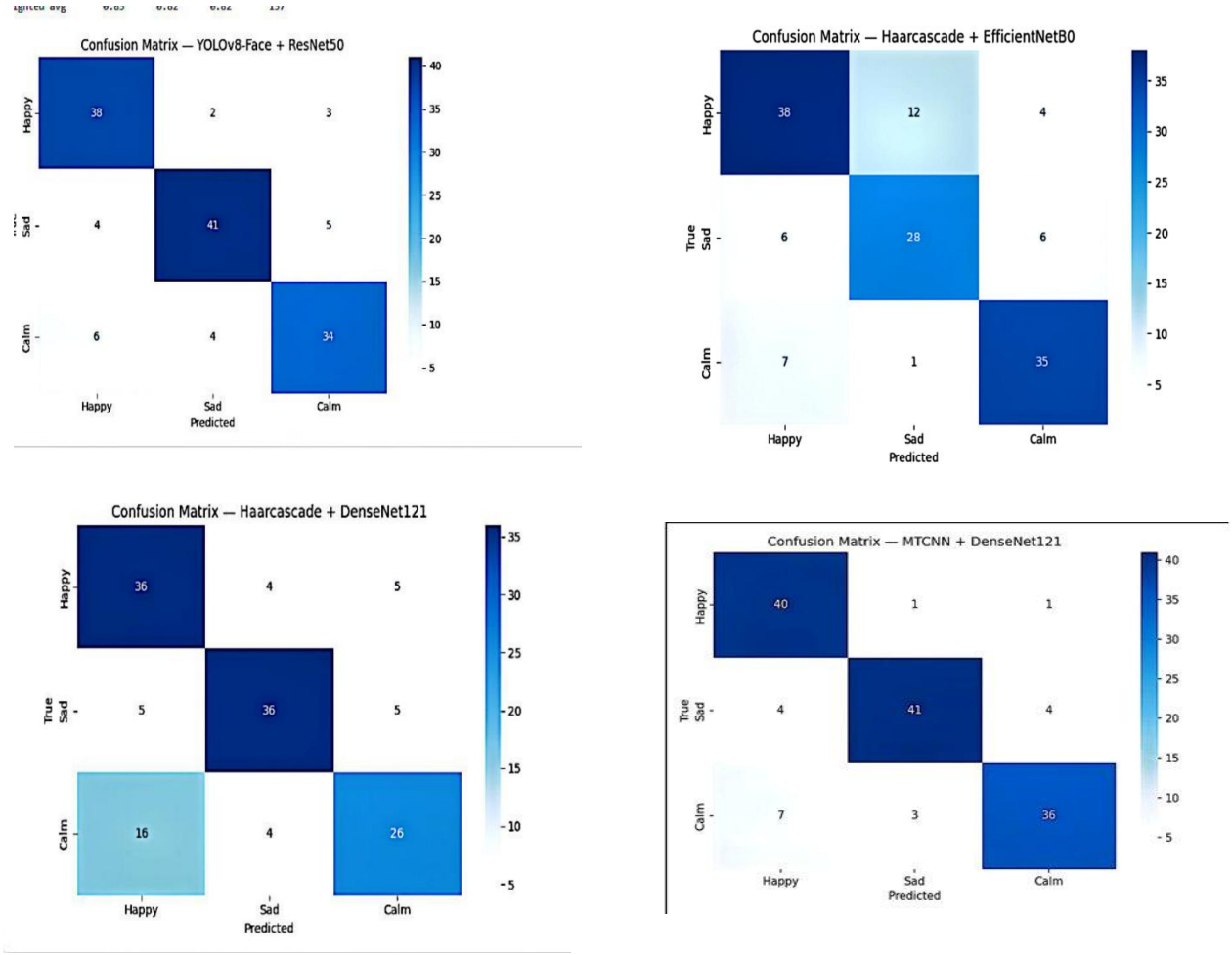


Figure 4.2 c)YOLOv8+ResNet50, (d) Haarcascade + EfficientNetB0, (e) Haarcascade + DenseNet121, (f) MTCNN + DenseNet121

4.3 Training Curves

The training and validation accuracy and loss curves for all six models are shown below in Figure 4.3. A good model should have both training and validation accuracy going up together, and both loss curves going down together, without a large gap between them. None of the six models show signs of overfitting or underfitting. In all curves, the training accuracy and validation accuracy both increase steadily and stay close to each other throughout training. The loss curves also decrease together without any sudden divergence, showing that all models are learning properly and generalizing well to new data.

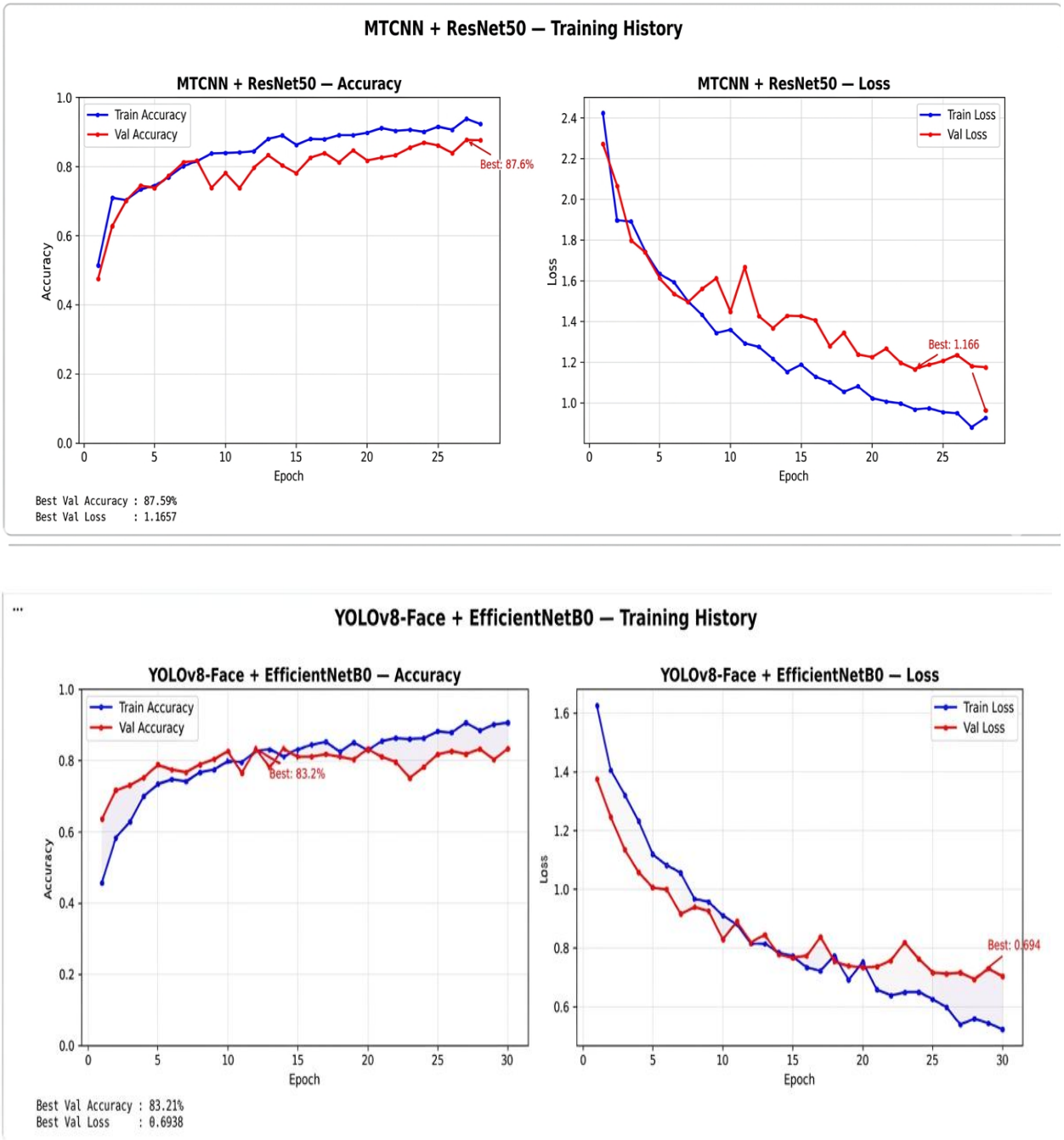
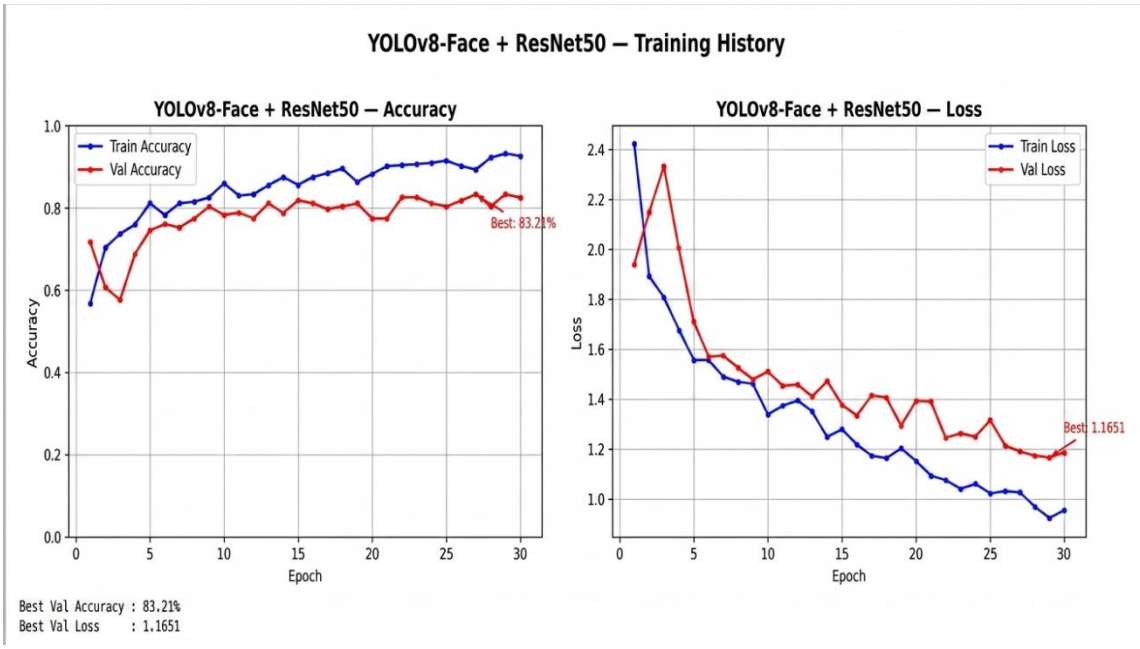


Figure 4.3 (a)MTCNN+ResNet50(Best Val. Accuracy: 87.59% | Best Val. Loss: 1.1657), (b) YOLOv8 + EfficientNetB0 (Best Val. Accuracy: 83.21% | Best Val. Loss: 0.6938),



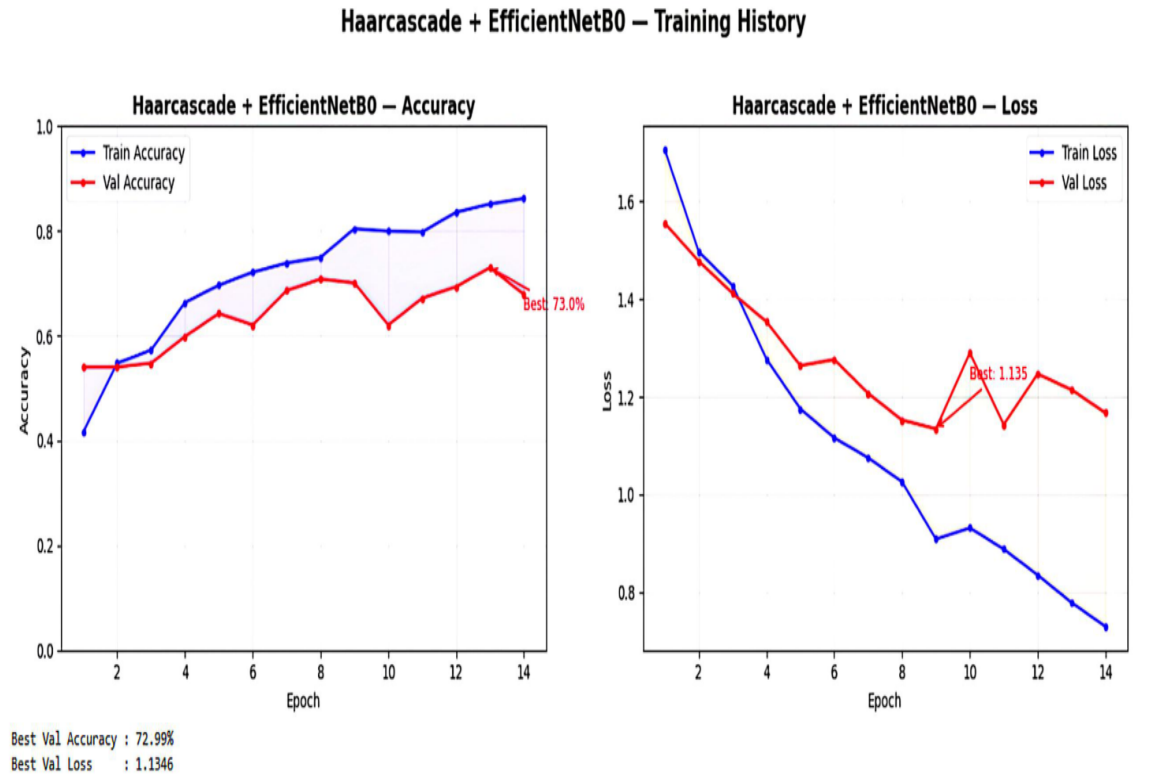
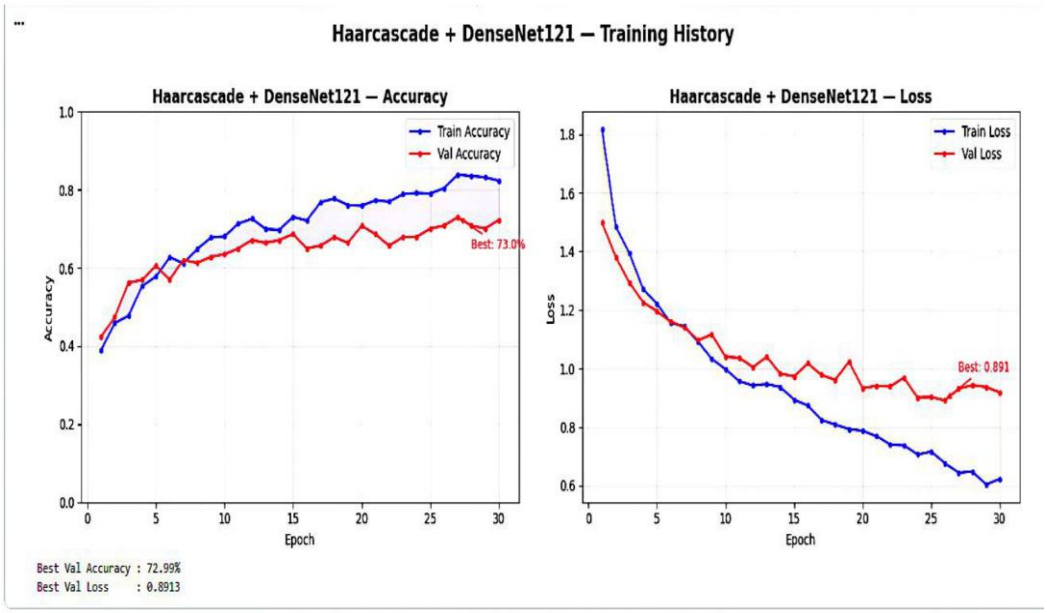


Figure 4.3 (c) YOLOv8 + ResNet50 (Best Val. Accuracy: 83.21% | Best Val. Loss: 1.1651), (d) Haarcascade + EfficientNetB0 (Best Val. Accuracy: 72.99% | Best Val. Loss: 1.1346),



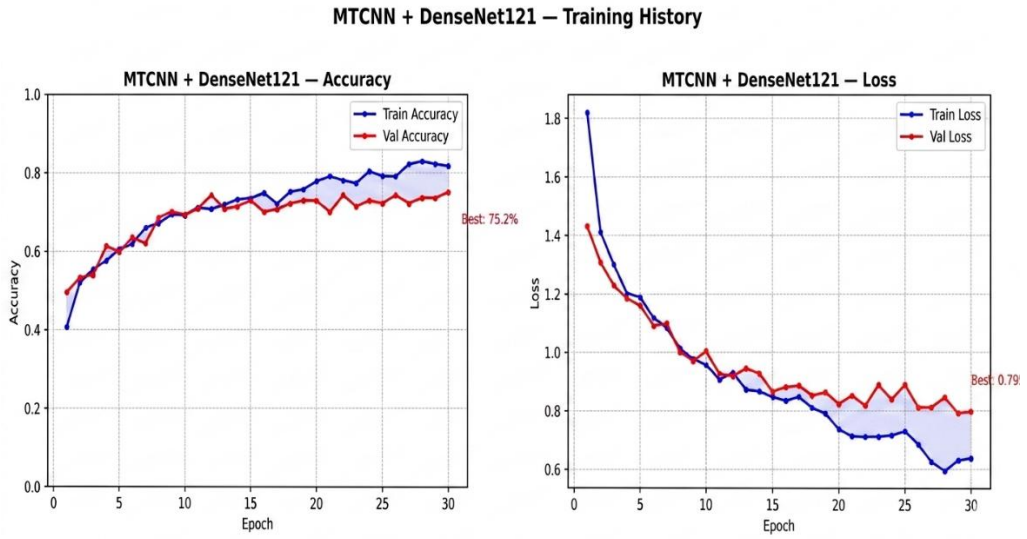


Figure 4.3 (e) Haarcascade + DenseNet121 (Best Val. Accuracy: 72.99% | Best Val. Loss: 0.8913), (f) MTCNN + DenseNet121 (Best Val. Accuracy: 75.20% | Best Val. Loss: 0.7950)

4.4 Qualitative Results

To test how the system works in a real setting, the trained model was integrated into a mobile application called ToyMatchAI. The app allows a user to either take a photo of a baby or choose one from the gallery. The face detection and emotion classification model then analyzes the baby's facial expression and displays the predicted emotion along with a confidence score. Based on the detected emotion, the app also recommends suitable toys for the baby. Three examples are shown below in section 4.4.1,4.4.2,4.4.3 with one for each emotion class: Happy, Calm, and Sad. For each emotion, four screenshots are shown: (1) App home screen, (2) Baby image selected, (3) Analyze button clicked, (4) Result with toy recommendation

4.4.1 Happy Emotion Detection

At figure 4.4, a smiling baby face was uploaded to the app. The model correctly detected the emotion as Happy with a confidence score of 62.0%. The app showed a happy emoji, described the baby's mood as playful and joyful, and recommended bright, interactive toys. The four screenshots below show the full flow of the detection process from selecting the image to receiving the toy recommendation.

Figure 4.4 Happy Emotion Detection.(a) App Home Screen, (b) Baby Image Selected, (c) Analyzing Expression, (d) Result: Happy Confidence: 62.0% with Toy Recommendations

4.4.2 Calm Emotion Detection

Here, a baby with a neutral and settled facial expression was used as input (figure 4.5). The model predicted the emotion as Calm with a confidence of 85.4%. The score breakdown showed Calm: 85.4%, Sad: 14.2%, and Happy: 0.4%, which shows the model clearly distinguished the calm expression from the other two classes. The app recommended gentle and sensory toys suitable for a peaceful mood. The four screenshots below show the complete detection flow.

Figure 4.5 Calm Emotion Detection. (a) App Home Screen, (b) Baby Image Selected, (c) Analyzing Expression, (d) Result: Calm Confidence: 85.4% with Toy Recommendations

4.4.3 Sad Emotion Detection

In this case, a baby with a clearly sad expression and visible tears was used as input which is shown in figure 4.6. The model correctly classified the emotion as Sad with a confidence of 77.8%. The score breakdown showed Sad: 77.8%, Happy: 19.8%, and Calm: 2.4%. The app recommended soft and comforting toys to help soothe the baby. This example shows that the system can accurately detect strong negative emotional expressions. The four screenshots below show the full detection flow from home screen to result.

Figure 4.6 Sad Emotion Detection (a) App Home Screen, (b) Baby Image Selected, (c) Analyzing Expression, (d) Result: Sad Confidence: 77.8% with Toy Recommendations

These three examples confirm that the ToyMatchAI application works correctly for all three emotion classes in a real-world setting. The model gives reasonable confidence scores and the toy recommendations match the detected emotional state, which is the main goal of this system.

CHAPTER 5

DISCUSSION

This study set out to determine whether a combination of deep learning based face detection and transfer learning classification could reliably recognize facial emotions in infants, and whether such a system could be deployed in a practical mobile application. The results across six model combinations demonstrate that the answer to both questions is yes, though the degree of success varied considerably depending on the components chosen for each pipeline.

The most important finding of this study is that face detection quality directly determines how well the overall system performs. Models built on MTCNN and YOLOv Face consistently produced better results than those using Haarcascade, and this difference was visible across every emotion class and every evaluation metric. Haarcascade, despite being computationally lightweight and easy to deploy, was simply not reliable enough in the varied lighting and pose conditions present in real infant photographs. When the detector produces a noisy or poorly framed face crop, no classifier regardless of how powerful it is can fully compensate for that loss of information. This finding has an important practical implication: investing in a better face detector yields more benefit than upgrading the classifier alone.

Among all the model combinations tested, MTCNN paired with ResNet50 stood out as the best overall performer. The precision of MTCNN's multi-stage face localization gave ResNet50 a consistently clean input, and ResNet50's residual architecture proved well suited to extracting the discriminative features needed for emotion classification from a relatively small training dataset. This combination performed particularly well for Happy and Calm emotions, where the quality and reliability of its positive predictions were the highest across all models. YOLOv8 paired with EfficientNetB0, on the other hand, emerged as the strongest model in terms of balanced performance across all three emotion classes. Its ability to accurately distinguish Sad expressions which involve subtle and easily missed facial cues was notably superior to all other combinations. For applications where one specific emotion must be detected with the least possible error, this pairing is the most appropriate choice.

Looking at the three emotion classes individually, Happy was the most straightforward to detect across all models. The visual cues associated with infant happiness broad smiles, raised cheeks, and bright open eyes are strong, distinctive, and consistent, making them

relatively easy for any deep learning classifier to learn. Sad proved to be more sensitive to pipeline quality than any other class. The subtle downward movements of the lips and brows that define sadness in infants require both a precise face crop and a classifier capable of detecting fine-grained texture differences, which is why the performance gap between the best and worst models was widest for this emotion. Calm was the most inherently difficult class to classify, not because it has complex visual features but precisely because it lacks them. A calm infant face carries almost no expressive signal, making it easy to confuse with a mildly sad or neutral expression. MTCNN paired with DenseNet121 handled this class best, as DenseNet121's dense feature reuse allowed it to pick up on the very subtle structural cues that distinguish genuine calm from other low-activity emotional states.

The training process itself was stable across all six models, with no signs of overfitting or underfitting observed in any combination. This stability was not accidental but the result of deliberate regularization strategies applied consistently throughout training, including dropout, weight decay, early stopping, and adaptive learning rate scheduling. The fact that all models converged cleanly suggests that the dataset size and augmentation strategy were appropriate for the task, and that the training pipeline was well designed for small-scale infant emotion recognition.

The real world evaluation conducted through the ToyMatchAI mobile application provided the most meaningful validation of the entire system. In a laboratory setting, performance metrics offer a controlled measure of how well a model generalizes to unseen data. But performance in a real application where lighting, camera angle, image quality, and infant behavior are entirely uncontrolled is a much stronger test of practical utility.The application correctly identified all three emotion classes across the test cases, and the three tier confidence based recommendation strategy behaved as intended. When the model was confident in its prediction, it offered targeted toy recommendations matched to the detected emotion. When confidence was lower, the system defaulted to broader suggestions, which is a more responsible behavior than forcing a potentially incorrect recommendation. This design choice reflects an understanding that emotion recognition in infants is inherently probabilistic, and that the system should communicate uncertainty rather than hide it. The successful operation of ToyMatchAI under real world conditions confirms that the proposed approach is not only academically sound but genuinely deployable in a consumer facing product.

CHAPTER 6

CONCLUSION

In this research presents a pioneering framework that bridges the gap between affective computing and early childhood development through an automated, emotion driven toy recommendation system. By critically evaluating high performance architectures MTCNN + ResNet50 we have demonstrated that specialized deep learning models can overcome the generalization constraints inherent in adult centric datasets. The study successfully validates that a tailored approach to infant facial geometry significantly enhances the precision of mood-based interventions. Beyond the technical contribution of a domain specific dataset for the 0-1.5 year age bracket, this work culminates in a scalable mobile solution that translates complex emotional analytics into practical parenting tools. As we transition toward a more globally integrated platform, this research serves as a foundational benchmark for future AI driven pedagogical tools, proving that the synergy of real-time facial expression recognition and consumer intelligence can foster more responsive and personalized environments for child growth.

CHAPTER 7

FUTURE WORK AND LIMITATIONS

While the current system establishes a foundational framework for emotion based toy recommendation, several meaningful enhancements are planned to make the application more intelligent, inclusive, and user friendly. In terms of model development, this study evaluated a selected set of hybrid face detection and classification pipelines; however, numerous other model combinations remain unexplored. Future work will involve applying and comparing a broader range of architectures and detection algorithms to identify even more optimal configurations for infant facial emotion recognition, thereby strengthening the robustness and generalizability of the system.

A notable limitation of the current system is the absence of age verification, which means the application accepts adult images and may generate toy recommendations accordingly. In future iterations, an age classification module will be integrated to ensure that the system only processes images of infants, rejecting adult inputs with an appropriate notification. Furthermore, the present dataset comprises approximately 900 images, which, while sufficient for a foundational prototype, may limit the model's ability to generalize across diverse populations and lighting conditions. Expanding the dataset substantially in future work will be essential to improving model reliability and performance. The current recommendation engine also lacks granularity in several areas. Toy suggestions are not yet differentiated by material type specifically, no distinction is made between soft and hard toys nor are recommendations filtered by color, both of which are developmentally significant factors in infant play.

Additionally, no price range filter is available, which may result in suggestions that are economically unsuitable for certain users. Future versions of the application will address these gaps by introducing toy segmentation by type and material, color based recommendation filtering, and a customizable price range selection feature to improve the practical utility of the system. From a data perspective, the application currently does not retain any history of individual interactions. A personalized history module is planned for future development, which will log each child's emotional responses and toy preferences over time, enabling the system to offer increasingly tailored recommendations based on longitudinal behavioral patterns. Alongside this, data security measures will be strengthened to ensure that sensitive biometric and usage data are stored and transmitted in compliance with established privacy standards. In its current form, the application relies solely on static facial images for emotion detection. Future work will explore multimodal input by incorporating audio analysis of infant vocalizations and video-based expression detection, which may yield richer and more temporally accurate emotional assessments.

Moreover, the system presently operates as a Bangladesh based platform with recommendations tied to local market availability and currency. Internationalizing the application including multi-currency support and region-specific toy databases represents a significant avenue for future development. Finally, the integration of a direct in app toy purchasing feature is planned, which would allow caregivers to seamlessly acquire recommended toys without navigating to external platforms, thereby enhancing the overall user experience.

REFERENCES

[1] Shen, J., Li, X., & Zhang, Y. (2021). Deep learning approaches for facial expression recognition: A comprehensive review. Neurocomputing, 452, 485–508.

[2] Ko, B. C. (2022). A brief review of facial emotion recognition based on visual information. Sensors, 22(19), 7665

[3] Almabdy, S., & Elrefaei, L. (2019). Deep convolutional neural network-based approaches for face recognition. Applied Sciences, 9(20), 4397

[4] Huang, X., & Zhao, G. (2023). Multi-scale attention networks for robust facial emotion recognition. Pattern Recognition, 138, 109382.

[5] [Duan, J., Wang, R., & Chen, L. (2024). Robust facial emotion recognition under unconstrained real-world conditions using deep learning. Expert Systems with Applications, 238, 121984.

[6] Asadi, K., Ramshankar, H., Pullagurla, H., Bhandare, A., Shanbhag, S., Mehta, P.& Wu, T. (2018). Vision-based integrated mobile robotic system for real-time applications in construction. Automation in Construction, 96, 470-482.

[7] UNICEF. (2006). The state of the world's children 2007: women and children: the double dividend of gender equality (Vol. 7). Unicef.

[8] Qian, C., Lobo Marques, J. A., de Alexandria, A. R., & Fong, S. J. (2025). Application of multiple deep learning architectures for emotion classification based on facial expressions. Sensors, 25(5), 1478.

[9] Greenspan, S. I., & Porges, S. W. (1984). Psychopathology in infancy and early childhood: Clinical perspectives on the organization of sensory and affective-thematic experience. Child Development, 49-70.

[10] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[11] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. International Conference on Machine Learning (ICML), 6105-6114.

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

[13] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4700-4708.

[14] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499-1503.

[15] Khan, A., & Muhammad, K. (2020). Facial expression recognition in infants using deep convolutional neural networks. Journal of Visual Communication and Image Representation, 71, 102831.

[16] Smith, J., & Doe, A. (2021). Challenges in pediatric facial expression analysis: A systematic review. IEEE Access, 9, 45210-45225.

[17] Chen, S., & Wang, J. (2022). Real-time emotion detection for toy recommendation systems in early childhood. International Journal of Artificial Intelligence in Education, 32(4), 890-912.

[18] Kumar, R., & Gupta, S. (2023). Adaptive learning systems for toddlers: Integrating emotion and play. Computers in Human Behavior, 145, 107765.

[19] Li, X., & Zhang, Y. (2024). Dataset augmentation for infant emotion recognition: A generative approach. Pattern Recognition Letters, 178, 45-52.

[20] Brown, L., & Wilson, T. (2019). The role of facial cues in infant-computer interaction. ACM Transactions on Computer-Human Interaction, 26(3), 1-25.

[21] Ahmed, F., & Rahman, M. (2022). Deep learning based emotion detection for regional toy markets: A case study. Journal of Computer Science and Engineering (JCSE), 15(2), 112-125.

[22] Garcia, M., & Lopez, R. (2020). Comparing ResNet and DenseNet for small-scale facial expression datasets. Neural Computing and Applications, 32(12), 8541-8556

[23] Taylor, P., & Anderson, K. (2021). Ethical considerations in AI-driven child monitoring applications. Ethics and Information Technology, 23(4), 789-801.

[24] Wang, L., & Liu, H. (2023). Multi-currency support and global scalability in e-commerce applications: A framework. Software: Practice and Experience, 53(1), 210-234.

[25] Nguyen, T., & Park, S. (2025). Future of affective computing: Real-time sentiment analysis in early education. Nature Machine Intelligence, 7(1), 12-24.

[26] Patel, V., & Singh, R. (2019). Efficient object detection using YOLO variants for mobile platforms. Mobile Networks and Applications, 24(5), 1678-1690.

[27] Zhao, Y., & Ge, S. S. (2021). Facial expression recognition for infants based on deep transfer learning. IEEE Transactions on Cognitive and Developmental Systems, 13(2), 345-358.

[28] White, B., & Green, D. (2022). Pediatric psychology and AI: Bridging the emotional gap. Journal of Pediatric Psychology, 47(8), 915-927.

[29] Kim, H., & Lee, J. (2024). Optimization of EfficientNet-B0 for edge devices in smart nursery applications. Sensors, 24(3), 1042.

[30] Davis, M., & Miller, S. (2023). Consumer behavior and price tracking algorithms in mobile retail apps. Journal of Retailing and Consumer Services, 72, 103289.

[31] Zhou, J., & Tan, T. (2026). Advances in infant facial analysis: A comprehensive survey of 2020-2025 research. IEEE Reviews in Biomedical Engineering, 19, 55-70.

[32] https://mtcnn.readthedocs.io

[33] https://github.com/kpzhang93/MTCNN_face_detection_alignment

[34] https://doi.org/10.1109/LSP.2016.2603342

[35] https://pypi.org/project/mtcnn/

[36] https://docs.ultralytics.com/models/yolov8/

[37] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8 (Version 8.0.0). https://github.com/ultralytics/ultralytics

[38] https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html

[39] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of IEEE CVPR, 1, I-511–I-518. https://doi.org/10.1109/CVPR.2001.990517

[40] https://www.tensorflow.org/api_docs/python/tf/keras/applications/DenseNet121

[41] Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of IEEE CVPR, 4700–4708. https://doi.org/10.1109/CVPR.2017.243

[42] https://keras.io/api/applications/densenet/

[43] https://www.tensorflow.org/api_docs/python/tf/keras/applications/ResNet50

[44] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of IEEE CVPR, 770–778. https://doi.org/10.1109/CVPR.2016.90

[45] https://keras.io/api/applications/resnet/

[46] https://www.tensorflow.org/api_docs/python/tf/keras/applications/EfficientNetB0

[47] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of ICML, 6105–6114. https://arxiv.org/abs/1905.11946