CSY3025

Artificial Intelligence Techniques

Assignment II

Project Report

Submitted By

Student Name	Aadarsha Shahi
University ID	23834806
Date of Submission	**1st September 2025**

Table of Contents

1 Introduction 5

1.1 Project Background 5

1.2 Project Rational 6

1.3 Aim and Objectives 7

2 Problem Analysis and Background Study 9

2.1 Problem Definition and Scope 9

2.2 Operational Context and Stakeholders 10

2.3 Assumptions and Constraints 10

2.4 Data Landscape and Curation 10

2.5 Technical Background 11

2.6 Evaluation Methodology 11

2.7 Literature Review 12

2.7.1 Emotion Recognition from Masked Faces using Inception-v3, Ashi Agarwal, Seba Susan, 2023 12

2.7.2 Multimodal Emotion Recognition with Surgical and Fabric Masks — Ziqing Yang; Katherine Nayan; Zehao Fan; Houwei Cao — 2022 (ICASSP) 13

2.7.3 Emotion Detection of People Wearing Face Mask, Naga Sai Tejo Pratardanudu Pedaprolu, D. Sai Sumanth, Y. Jayanth, G. Sri Vishruth, S. Madhur, P. Chandra Sekhar Reddy, 2023 14

2.7.4 EmoNet: Exploring Facial Emotion Recognition Through Machine Learning, Manaswi Karadbhajne, Mayuri Gomkale, Nekita A. Chavhan, Rahul Agrawal, Chetan Dhule, 2023 15

3. Architectural Design 17

3.1 Proposed Architecture 17

3.1.1 Data ingestion and preprocessing 17

3.1.2 Feature-extraction backbone 18
3.1.3 Classification head 19
3.1.4 Optimization, metrics, and regularization 20
3.1.5 Model selection and artefacts 22
3.1.7 Configuration used in the submitted experiments 23
3.3 Training and Evaluation 23
3.3.1 Dataset and split 23
3.3.2 CSV Data Path 27
3.3.3 Training configuration. 28
3.3.4 Quantitative results 29
3.3.5 Interpretation and limitations 30
4 Testing 31
4.1 Test Environment and Reproducibility 31
4.2 Test Data 31
4.3 Test Plan and Protocol 31
4.4 Results 33
4.5 Error Analysis 33
5 Conclusion 35
References 37

List of Figures

Figure 1: Sample Code of Dataset.py 18
Figure 2: Sample code of networks.py 19
Figure 3: network.py Sample code of Mobilenet v2 20
Figure 4: Train.py Sample code 21
Figure 5: Sample code for fine-tune.py 21
Figure 6: Sample code of app_streamlit.py - View Class 22
Figure 7: File size Adjustment 23
Figure 8: Dataset.py Use of Tensorflow 24
Figure 9: Network.py - Sample code of baseline and mobilenet v2 25
Figure 10: Sample code of Evaluate.py 26
Figure 11: Sample code of Predict.py 26
Figure 12: Sample code of plots.py 27
Figure 13: Creating a CSV file from dataset images 28
Figure 14: Head-only Stage of Training 29
Figure 15: Fine Tuning Stage of Training 29

1 Introduction

The reliable interpretation of human emotion from facial cues remains a foundational problem in affective computing and human-computer interaction. The widespread adoption of face masks in healthcare, education, and public-facing environments has introduced systematic occlusion of the lower face, thereby reducing the salience of features traditionally associated with emotion recognition e.g., lip curvature and nasolabial activity. This constraint increases the likelihood of misclassification in both human judgement and machine inference, particularly under realistic variability in illumination, pose, demographics, and mask types. Consequently, the research challenge is not merely to adapt existing pipelines but to re-specify data, models, and evaluation protocols to emphasize peri-ocular patterns and other robust signals while maintaining computational efficiency suitable for deployment on commodity hardware.

This project addresses that challenge by developing and empirically evaluating a deep-learning classifier for recognizing at least five discrete emotions from masked facial images. The work is structured around a reproducible experimental pipeline comprising task definition and success criteria; curated data acquisition with occlusion-aware augmentation; model selection favouring compact convolutional or transfer-learning architectures; and systematic training with regularization and early-stopping strategies. Evaluation extends beyond aggregate accuracy to include macro-averaged F1, per-class metrics, and confusion matrices, enabling analysis of typical failure modes e.g., neutral-happy confusion and robustness considerations. Ethical and methodological rigour are ensured through attention to data provenance, demographic representation, and transparent reporting of limitations and risks. The resulting artefacts a trained model, documented code, and a concise demonstration, aim to provide evidence of feasibility and a basis for future refinement and responsible application.

1.1 Project Background

Facial expression recognition (FER) has evolved from rule-based and feature-engineered pipelines grounded in descriptive systems such as the Facial Action Coding System and handcrafted descriptors e.g., LBP/HOG with SVM classifiers to end-to-end deep architectures

that learn hierarchical representations directly from images. Convolutional and transfer-learning models have delivered substantial gains on widely used benchmarks; however, most canonical datasets assume unobstructed faces and controlled capture conditions. The widespread use of face masks introduces structured occlusion of the lower face, suppressing features that are highly informative for several emotions and thereby increasing class ambiguity for example, neutral versus happy. This shift exposes domain-generalization weaknesses: performance degrades under variations in illumination, pose, camera quality, demographics, and mask types, highlighting the gap between laboratory accuracy and reliable deployment in applied settings such as classrooms, clinics, and customer-service contexts.

To address these constraints, recent practice in masked-face FER emphasizes three complementary directions. First, data curation and augmentation strategies explicitly model occlusion through acquisition of masked images, synthetic mask overlays, random erasing/cutout, and balanced class distributions to improve robustness while mitigating leakage between splits. Second, architectural choices favour compact backbones e.g., MobileNet-family or similarly lightweight CNNs with attention mechanisms or peri-ocular feature emphasis to retain discriminative power under occlusion, while remaining feasible for resource-constrained devices. Third, evaluation protocols move beyond aggregate accuracy to macro-averaged F1 and per-class analyses, supported by confusion matrices and stress tests that probe sensitivity to mask style, lighting, and demographic variation. Alongside technical considerations, contemporary work foregrounds ethical and legal concerns data provenance, consent and licensing, bias detection and reporting, and privacy-preserving handling of facial data, so that empirical improvements translate into responsible, trustworthy applications.

1.2 Project Rational

The proliferation of face coverings in educational, clinical, and customer-facing environments has rendered conventional facial expression recognition pipelines less reliable by occluding salient lower-face features. There is thus a clear practical need for methods that can infer affect from masked faces with acceptable accuracy and computational efficiency. Beyond immediate utility, such capability supports inclusive communication, assisting instructors in gauging learner

engagement, clinicians in assessing patient affect, and service providers in adapting interactions, while foregrounding responsible AI practices through explicit consideration of data provenance, demographic representation, privacy, and bias. Addressing these considerations within a single, well-scoped study offers pedagogical value and societal relevance: it demonstrates how technical design choices intertwine with ethical safeguards to yield trustworthy outcomes suitable for deployment on commodity hardware.

From an academic perspective, the project provides a focused vehicle for demonstrating mastery across the software development lifecycle in a machine-learning context. It motivates clear requirements and a rigorous analysis of constraints, leading to a principled design that favours lightweight, transfer-learning architectures and occlusion-aware augmentation. Implementation then operationalizes these choices in reproducible code with check-pointing and experiment control; evaluation moves beyond aggregate accuracy to macro-F1, per-class metrics, and confusion analyses that surface failure modes; and documentation synthesizes findings, limitations, and future work. In aggregate, the project functions as an evidence-based response to an authentic problem, while providing a traceable, standards-aligned artefact set model, experiments, and report, that satisfies module learning outcomes and supports future research or applied adoption.

1.3 Aim and Objectives

To develop and validate a lightweight deep-learning classifier that recognizes at least five emotions from masked facial images, using reproducible methods and explicit ethical safeguards.

To achieve the aim of this study, the following objectives are undertaken: -

- Define the project scope and acceptance thresholds (≥5 emotions; macro-F1 target; model size and latency limits).
- Establish ethical governance, including data provenance, consent/licensing, privacy safeguards, and bias mitigation.

- Conduct a focused literature review on FER under occlusion and lightweight architectures to justify design choices.
- Acquire and curate a masked-face dataset with leakage-safe splits and class-balance controls.
- Design a deterministic pre-processing and occlusion-aware augmentation pipeline.
- Select and configure compact CNN or transfer-learning backbones with appropriate regularization and learning-rate scheduling.
- Pre-register an experiment matrix and baselines to enable reproducible comparison.
- Implement modular, configuration-driven training and evaluation with seed control and check-pointing.
- Track experiments and artefacts (metrics, logs, models, and plots) to ensure auditability.
- Evaluate using macro-F1, per-class precision/recall, accuracy, confusion matrices, and confidence intervals.
- Perform robustness and fairness diagnostics across mask style, illumination, and demographic strata; document failure modes.
- Package an inference-ready artefact and minimal demonstration; measure CPU-only latency and memory footprint.
- Provide complete reproducibility materials, including environment specification, scripts, and data instructions.
- Produce a formal report detailing methods, results, limitations, threats to validity, and avenues for future work.

2 Problem Analysis and Background Study

The research problem concerns reliable recognition of discrete emotions from masked facial images under realistic variability (illumination, pose, camera quality, demographics, and mask types). Conventional Facial Expression Recognition (FER) pipelines, trained on unobstructed faces, lose performance when the lower face is occluded, as salient cues (e.g., lip curvature) are suppressed. This work re-specifies the task for masked contexts, emphasizing peri-ocular and other robust features, and targets compact models suitable for deployment on commodity hardware while maintaining ethical and methodological rigour.

The background study situates the problem within affective computing and human–computer interaction, reviews technical approaches to FER under occlusion, and derives explicit requirements, constraints, and evaluation criteria. The section concludes with a gap analysis and risk assessment that inform design and experimentation choices in subsequent phases.

2.1 Problem Definition and Scope

This study addresses the reliable recognition of discrete emotions from masked facial images using deep learning. The task is formulated as single-label, multi-class classification over a minimum of five emotions, with explicit emphasis on robustness to lower-face occlusion introduced by masks. The scope encompasses end-to-end development from data acquisition and curation through model design, training, and evaluation under practical constraints of compute, memory, and latency suitable for commodity hardware. The project foregrounds methodological transparency and ethical governance, including data provenance, licensing, fairness analysis, and privacy safeguards. Success is defined by reproducible performance on leakage-safe test sets, macro-averaged metrics that reflect class balance, and documented limitations that delineate appropriate use.

2.2 Operational Context and Stakeholders

The envisaged operational contexts include classrooms, clinical and counselling settings and customer-facing services, all of which increasingly involve face coverings. Primary stakeholders comprise end users affected by model outputs, practitioners who interpret or act on those outputs, and technical maintainers responsible for deployment and monitoring. Secondary stakeholders include institutional governance bodies and data subjects whose images contribute to training datasets. Each stakeholder group has distinct risk tolerances and interpretability needs, motivating conservative claims, clear documentation of limitations, and guardrails against high-stakes autonomous decision-making.

2.3 Assumptions and Constraints

The analysis assumes images contain a single, near-frontal masked face with labels corresponding to mutually exclusive, culturally recognizable emotions. Resource constraints prioritize lightweight architectures and predictable CPU-only inference, while dataset availability may impose class imbalance and domain shift between curated and real-world imagery. Ethical and legal constraints require strict attention to consent, licensing, and privacy, potentially limiting dataset scope and dissemination. The project further assumes that annotations for demographic attributes and mask types may be incomplete; where such metadata exist, evaluations will be stratified, and where they do not, claims will be appropriately bounded. All experiments are conducted under fixed-seed, configuration-driven protocols to mitigate randomness and enhance reproducibility.

2.4 Data Landscape and Curation

The dataset strategy combines genuinely masked-face images, where available and lawfully licensable, with carefully controlled synthetic augmentation applied to unmasked sets when permitted. Curation enforces leakage-safe splits, addresses class imbalance through sampling or

weighting, and stratifies examples across illumination conditions, capture devices, and mask styles to promote generalization. Pre-processing comprises deterministic resizing and normalization, while augmentation includes occlusion-aware techniques, alongside restrained geometric and photometric transforms to avoid distributional artefacts. A data register records sources, licenses, and known limitations, and all decisions impacting representativeness or bias are documented to support ethical review and reproducibility.

2.5 Technical Background

The technical approach builds on modern Facial Expression Recognition methods that replace handcrafted features with convolutional and transfer-learning models capable of learning hierarchical visual representations. Given deployment constraints and occlusion, the design favours compact backbones, potentially augmented by attention or region-emphasis mechanisms that prioritize peri-ocular cues. Regularization. scheduled optimizers, and early stopping guard against overfitting in limited data regimes, while check-pointing and configuration management enable controlled ablations and comparative baselines. The inference package integrates a fixed pre-processing pipeline with frozen weights to ensure portability, determinism, and ease of evaluation on standard hardware.

2.6 Evaluation Methodology

Evaluation is anchored in macro-averaged F1 to reflect class balance, complemented by per-class precision/recall and confusion matrices to surface characteristic error modes, Secondary measures include overall accuracy, calibration or confidence summaries where appropriate, and efficiency metrics such as model size, CPU-only latency, and memory footprint. Robustness is probed via stratified analyses across mask styles, illumination conditions, and demographic strata when annotations permit; absent such metadata, claims are limited and clearly qualified. All experiments employ fixed seeds, pre-registered baselines, and leakage-safe splits, with full

reporting of data provenance, hyper-parameters, and threats to validity to facilitate replication and critical appraisal.

2.7 Literature Review

Research on Facial Expression Recognition has shifted from handcrafted descriptors to deep models, with mask-induced occlusion consistently depressing accuracy. Mitigations emphasize peri-ocular features, attention mechanisms, occlusion-aware augmentation, and lightweight backbones; evaluation prioritizes macro-F1 and per-class analyses with transparent reporting of data provenance, bias, and limitations.

2.7.1 Emotion Recognition from Masked Faces using Inception-v3, Ashi Agarwal, Seba Susan, 2023

This conference paper investigates emotion recognition from masked faces using transfer learning on several deep backbones, EfficientNet-B0, ResNet-50, Inception-v3, Xception, and AlexNet, fine-tuned on the FER-2013 dataset. The authors motivate the problem by noting that masks occlude lower-face cues essential for accurate recognition, and they frame a comparative study to identify a model resilient to such occlusion. They report that Inception-v3 outperforms the alternatives on the masked-face

On FER-2013, Inception-v3 attains the strongest results among the tested models for both masked and unmasked faces; performance degrades with masks, but remains comparatively superior to other baselines. The authors provide a metric table (F1-score, MCC, precision, test accuracy) evidencing the gap between masked and unmasked evaluation and ranking Inception-v3 above ResNet-50 and EfficientNet-B0, with traditional ML baselines and some deep models trailing. They also report an overall ROC-AUC of 0.8638 for Inception-v3, with surprise the best (AUC = 0.95) and fear the worst (AUC = 0.76), and note confusion patterns: better detection of happy and surprise and poorer performance on angry and fear.

Although the study is a useful baseline, several limitations suggest avenues for extension. The dataset is low-resolution (48×48), class-imbalanced, and partly constructed via web search; moreover, the masked set is programmatically generated and smaller than the unmasked set, which may limit ecological validity and contribute to distribution shift. The pipeline converts images to grayscale and trains for a modest number of epochs, with no subgroup analysis or identity-disjoint split audit reported, leaving fairness, robustness, and potential leakage risks underexplored. Future work should incorporate real-world masked datasets, stratified/fairness evaluations, identity-aware partitioning, stronger occlusion-aware augmentation, and calibration analyses alongside macro-F1 and confusion metrics.

2.7.2 Multimodal Emotion Recognition with Surgical and Fabric Masks — Ziqing Yang; Katherine Nayan; Zehao Fan; Houwei Cao — 2022 (ICASSP)

This paper investigates how face masks affect automatic emotion classification across audio, visual, and audio-visual modalities using the CREMA-D dataset. The authors emulate mask effects by (i) re-generating speech through surgical and fabric masks to obtain "muffled" audio, and (ii) restricting visual features to upper-face Action Units to model lower-face occlusion. They extract ComParE acoustic features and Bag-of-AUs visual features, train linear-kernel SVMs per modality, and conduct subject-independent 5-fold cross-validation, including cross-corpus tests between clean and masked speech.

Masks substantially degrade recognition. In audio, unweighted average recall drops from ~0.59 (no mask) to ~0.47 (surgical) and ~0.46 (fabric); in vision, using only upper-face AUs reduces UAR from ~0.63 to ~0.55. Multimodal fusion remains superior to uni-modal masked conditions (UAR ≈0.66 with masks vs ≈0.76 without), indicating complementary information between muffled speech and occluded facial cues. Emotion-wise, fear is most affected by masking, whereas anger and sad are least affected; happy remains comparatively robust in visual features. Contribution analyses show reduced redundancy and altered dominance patterns between audio and visual streams under masking. Cross-corpus experiments reveal poor transfer between clean

and masked speech, while training on combined clean+mask data yields the most robust performance across conditions.

The study uses engineered features with linear SVMs rather than end-to-end deep multimodal architectures, potentially limiting representational capacity under occlusion. Masked speech is "re-generated" via playback through masks rather than collected from masked talkers, which may under-represent natural co-articulation and behavioral adaptations; visual masking is approximated by upper-face AUs rather than real masked-face videos. The work focuses on acted emotions (CREMA-D), does not report latency/footprint for deployment, and offers limited demographic fairness analyses. Future research should evaluate on real masked-face corpora, adopt end-to-end lightweight audio-visual models with attention to peri-ocular cues, report macro-F1 and calibration alongside UAR, include subgroup robustness, and measure efficiency for practical on-device use.

2.7.3 Emotion Detection of People Wearing Face Mask, Naga Sai Tejo Pratardanudu Pedaprolu, D. Sai Sumanth, Y. Jayanth, G. Sri Vishruth, S. Madhur, P. Chandra Sekhar Reddy, 2023

This conference paper (ICSCDS-2023) proposes a convolutional-neural-network approach to classify emotions from images of masked faces. Using FER-2013 as the base corpus, the authors clean the data and synthetically overlay masks via a Python script, then reduce the original seven emotions to five by merging visually confusable pairs (anger+disgust; fear+surprise) before training a CNN on 48×48 grayscale images. The work positions mask occlusion as the primary challenge and motivates class merging as a pragmatic response to reduced lower-face cues.

Across several attempted models (including VGG and ResNet variants noted by the authors), the final, comparatively simple CNN reports a headline test accuracy of 94.31% on the five-class, mask-augmented dataset. The study presents qualitative justifications for merging specific emotion pairs, argues that occlusion-aware preprocessing improves separability, and includes a brief method comparison table indicating the proposed pipeline outperforming selected baselines

under the paper's setup. A small real-time demonstration is described to suggest practical viability.

The evaluation relies primarily on overall accuracy without reporting macro-F1, per-class metrics, calibration, or confidence intervals; identity-disjoint splits, demographic analysis, and fairness reporting are not discussed. The masked data are synthetically generated rather than collected from real masked subjects, and images are low-resolution (48×48), which may limit ecological validity and generalization. Furthermore, collapsing seven emotions to five improves scores but reduces label granularity and may obscure persistent confusions; robustness to illumination, device variation, and mask styles is not systematically probed. These gaps motivate future work using real masked-face corpora, identity-safe partitions, stratified/fairness metrics, and lightweight yet stronger backbones with occlusion-aware augmentation.

2.7.4 EmoNet: Exploring Facial Emotion Recognition Through Machine Learning, Manaswi Karadbhajne, Mayuri Gomkale, Nekita A. Chavhan, Rahul Agrawal, Chetan Dhule, 2023

This IEEE conference paper (ICAICCIT 2023) presents "EmoNet," a convolutional-neural-network (CNN) approach for facial emotion recognition with an emphasis on masked facial emotion recognition (MFER). The authors motivate FER's relevance, describe the use of a Kaggle dataset that includes masked images, and focus on classifying five emotions (happy, anger, neutral, surprise, sad). They outline a standard vision pipeline data pre-processing, CNN feature extraction, training, and testing and supply an architectural summary of the proposed network.

The paper reports that the CNN "properly categorized" the dataset and cites headline performance figures: an overall accuracy of approximately 89.91% on the broader dataset and a lower accuracy of about 65.99% specifically on masked-face recognition. The manuscript includes qualitative demonstrations (sample predictions), a training-versus-validation loss plot,

and a tabulated model summary indicating a compact architecture with on the order of a few million parameters, positioning the model as feasible for practical deployment.

Methodologically, the study relies on a single Kaggle source without detailing identity-disjoint splitting, per-class metrics, or macro-averaged measures; demographic fairness analysis and robustness probes (illumination, device, mask style) are absent. The evaluation emphasizes overall accuracy rather than confusion patterns, calibration, or confidence intervals, and there is no comparison to transfer-learning baselines or occlusion-aware augmentation strategies. Reporting of runtime, memory footprint, and reproducibility controls is limited, leaving external validity and deployment readiness under-specified and suggesting the need for stronger experimental controls and stratified evaluations in future work.

3. Architectural Design

This project implements a complete, reproducible pipeline for masked facial-emotion recognition that is deliberately engineered for robustness to partial facial occlusion, tolerance to imperfect, low-contrast images, and deployability on modest hardware. The system follows a classical deep-learning pattern: images are ingested from class-labelled folders, pre-processed and augmented, passed through a convolutional feature-extraction backbone initialized from large-scale pre-training, and mapped to seven emotion categories by a compact classification head. In order to adapt the pre-trained representation to the masked-face domain without overfitting, fine-tuning proceeds in two stages: a head-only warm-up to align the final layer with the target label space, followed by partial unfreezing of the backbone with a low learning rate. Model selection is driven by validation accuracy with early stopping, and the training process writes artefacts (learning curves, confusion matrix, and a saved checkpoint) that support transparent reporting and replication. For demonstration, a lightweight Streamlit application reproduces the inference pre-processing and exposes top-k predictions alongside an explanation heatmap.

3.1 Proposed Architecture

3.1.1 Data ingestion and preprocessing

Images are stored in a hierarchical directory structure, with one folder per emotion class for the training and validation splits. At load time, each image is resized to 224×224, converted to RGB, and normalised to [0,1] for numerical stability. Because MobileNetV2 expects inputs scaled to [−1,1], the inference graph performs this mapping immediately before the backbone, thereby keeping the dataset and model code decoupled. During training, we apply light geometric and photometric augmentations-horizontal flips, small rotations and zooms, and mild brightness/contrast jitter-to improve invariance to pose and illumination. To make the model specifically robust to occlusion by face masks, we include a synthetic mask overlay that programmatically covers the lower portion of the image with a softly blended colour patch. This augmentation encourages the network to attend to peri-ocular cues that remain visible when the mouth and chin are obscured.

A separate, optional quality-control stage assesses sharpness and contrast and can either reject sub-threshold images or attempt simple auto-enhancement before resizing. Although, in our most recent run, the heuristic thresholds admitted the available files without rejection, the presence of this stage is important for reproducibility and provides a principled path to curate future data without altering the training code.

src > model > dataset.py > _gaussian_kernel

```python
import tensorflow as tf

AUTOTUNE = tf.data.AUTOTUNE

def _mask_augment(image, frac=0.55, strength=0.85):
    image = tf.convert_to_tensor(image, dtype=tf.float32)
    rank = len(image.shape)
    if rank == 3:
        image = tf.expand_dims(image, 0)
    b, h, w, _ = tf.unstack(tf.shape(image))
    y = tf.range(h, dtype=tf.int32)[tf.newaxis, :, tf.newaxis, tf.newaxis]
    cutoff = tf.cast(tf.round(frac * tf.cast(h, tf.float32)), tf.int32)
    binary = tf.cast(y >= cutoff, image.dtype)
    mask = tf.image.resize(binary, (h, w), method='nearest')
    mask = tf.broadcast_to(mask, [b, h, w, 3])
    color = tf.random.uniform([b, 1, 1, 3], 0.0, 1.0, dtype=image.dtype)
    out = image * (1.0 - strength * mask) + color * (strength * mask)
    return tf.squeeze(out, 0) if rank == 3 else out

def _gaussian_kernel(ks: int, sigma: float):
    ax = tf.range(ks, dtype=tf.float32) - (ks - 1.) / 2.
    gauss = tf.exp(-0.5 * tf.square(ax / sigma))
    kernel = tf.tensordot(gauss, gauss, axes=0)
    kernel = kernel / tf.reduce_sum(kernel)
    kernel = kernel[:, :, tf.newaxis, tf.newaxis]
    kernel = tf.repeat(kernel, repeats=3, axis=2)
    return kernel

def _random_gaussian_blur(image, p=0.0, ks=3, sigma=0.8):
```

Figure 1: Sample Code of Dataset.py

3.1.2 Feature-extraction backbone

The network uses MobileNetV2 (width multiplier 1.0, input size 224) as the convolutional backbone. MobileNetV2 is composed of depth wise separable convolutions with inverted residual blocks, which provide a favorable accuracy-to-latency ratio. We initialize the backbone

with ImageNet weights to benefit from general-purpose visual features, and keep Batch-Normalization layers frozen during fine-tuning to avoid destabilizing learned statistics on relatively small batches.

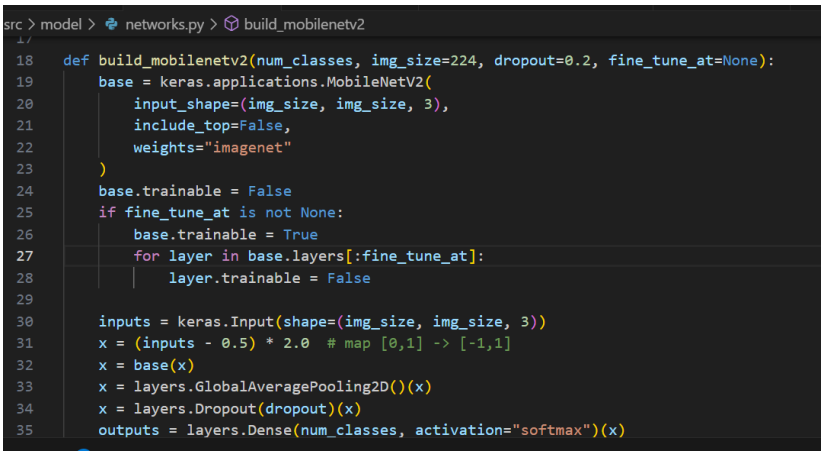


Figure 2: Sample code of networks.py

3.1.3 Classification head

Backbone feature maps are aggregated by Global Average Pooling, followed by a Dropout layer (rate 0.2–0.3) to mitigate co-adaptation, and a fully connected Softmax layer with seven outputs corresponding to angry, disgust, fear, happy, neutral, sad, and surprise. The training objective is Categorical Cross-Entropy with label smoothing ($\varepsilon = 0.05$) to reduce over-confidence and improve calibration, which is particularly relevant in the presence of label noise and visually ambiguous expressions.

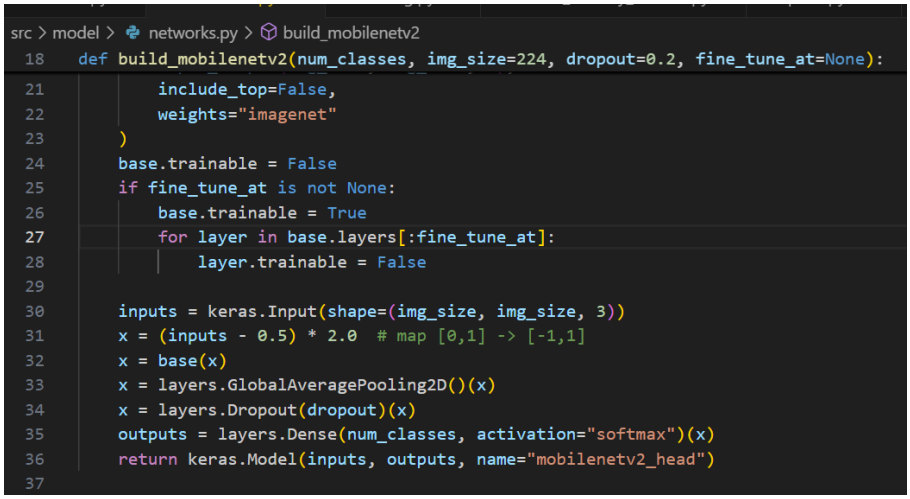


Figure 3: network.py Sample code of Mobilenet v2

3.1.4 Optimization, metrics, and regularization

Optimization is performed with Adam and a cosine learning-rate schedule. In the head-only stage, a relatively larger initial learning rate ($\approx 3\times10^{-3}$) enables rapid alignment of the new head to the dataset. In the fine-tuning stage, we reduce the initial rate ($\approx 1\times10^{-4}$ to 5×10^{-5}) and unfreeze a tail of the backbone while keeping early layers fixed. We monitor top-1 accuracy and Top-3 accuracy-the latter reflects whether the correct class is among the three most likely predictions and is informative when classes are confusable under occlusion. Where the class distribution is imbalanced, we compute class weights from folder counts and pass them to the loss, thereby compensating for under-represented emotions. We also experimented with Mix-up ($\alpha\approx0.2–0.3$) to improve calibration and robustness; this is enabled or disabled depending on observed stability.

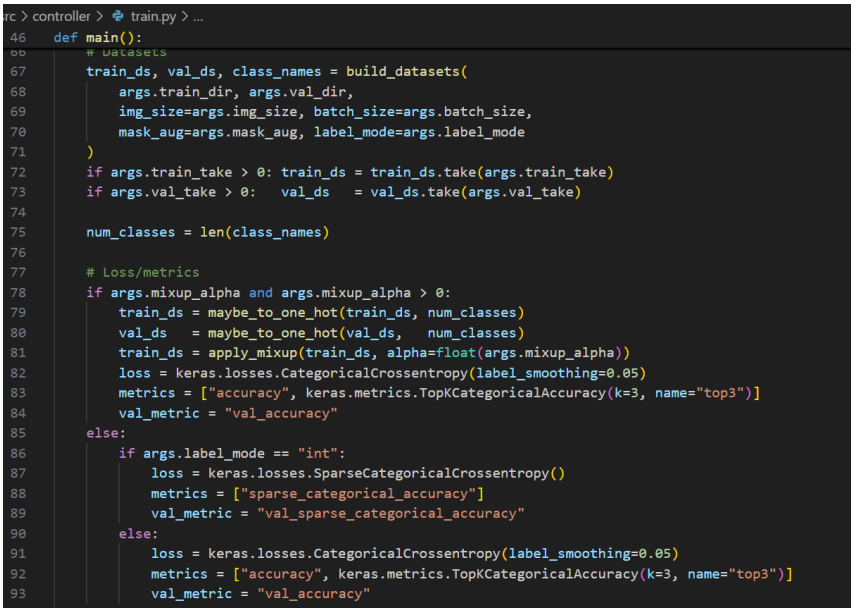


Figure 4: Train.py Sample code

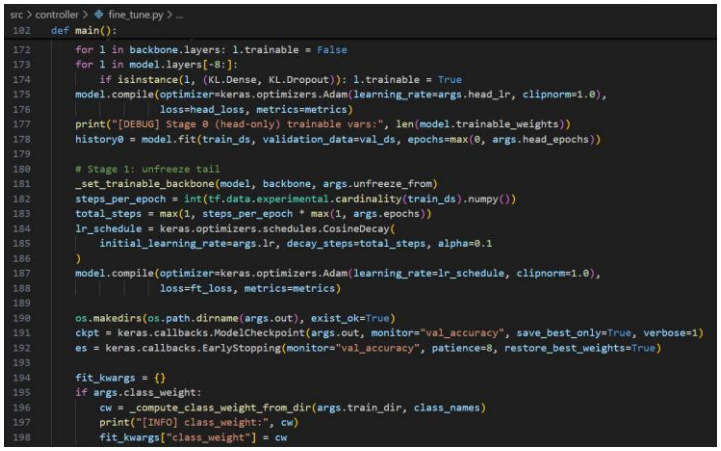


Figure 5: Sample code for fine-tune.py

3.1.5 Model selection and artefacts

Training uses early stopping on validation accuracy with a patience window and best-checkpoint saving. Each run writes diagnostic artefacts, accuracy and loss curves and, post-training, a confusion matrix to facilitate inclusion of figures in the report and to support subsequent error analysis.

3.1.6 Inference application

A Streamlit interface reproduces the pre-processing pipeline, loads the saved `.keras` model, and displays top-k probabilities. An explanation view computes Grad-CAM on the last convolutional feature map; if gradients are unavailable due to execution mode constraints, a saliency-based fallback is provided. The interface also highlights low-confidence cases explicitly, which promotes honest communication of model uncertainty in the demo.

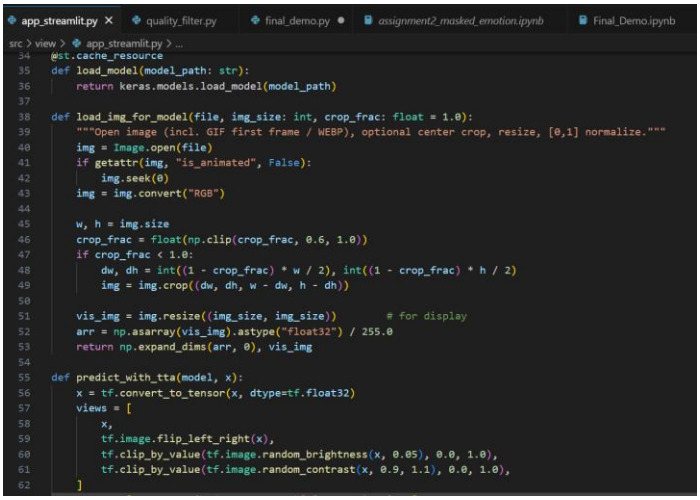


Figure 6: Sample code of app_streamlit.py - View Class

3.1.7 Configuration used in the submitted experiments

Images are 224×224, batch size is 32, and label smoothing is set to 0.05. Dropout in the head is between 0.2 and 0.3. When enabled, Mix-up uses α≈0.3. Class weights are computed from per-class image counts in the training directory. We use Adam with gradient-norm clipping (clip-norm = 1.0) for numerical stability. Random seeds are fixed across Python, NumPy, and TensorFlow to promote reproducibility.

```python
def main():
    ap = argparse.ArgumentParser("Filter or lightly fix blurry/low-contrast images.")
    ap.add_argument("--src", required=True, help="Source root with class subfolders.")
    ap.add_argument("--dst", required=True, help="Destination root for cleaned copies.")
    ap.add_argument("--reject_dir", default="", help="If set, rejected images are copied here.")
    ap.add_argument("--min_sharpness", type=float, default=55.0)
    ap.add_argument("--min_contrast", type=float, default=0.06)
    ap.add_argument("--try_fix", action="store_true")
    ap.add_argument("--resize", type=int, default=224, help="Resize outputs square (0=keep).")
    ap.add_argument("--report_csv", default="", help="Optional CSV path.")
    ap.add_argument("--dry_run", action="store_true")
    args = ap.parse_args()

    src_root = Path(args.src)
    dst_root = Path(args.dst)
    rej_root = Path(args.reject_dir) if args.reject_dir else None

    if not src_root.is_dir():
        sys.exit(f"[ERROR] Source not found: {src_root}")

    files = [p for p in src_root.rglob("*") if p.is_file() and p.suffix.lower() in IMG_EXTS]
    if not files:
```

Figure 7: File size Adjustment

3.3 Training and Evaluation

3.3.1 Dataset and split

The training set comprises 15,531 images, and the validation set comprises 4,953 images, each distributed across the seven emotion classes. The data were already partitioned into training and validation directories; no images were shared across splits. Although a quality-filter pipeline is

available, the current thresholds did not reject images in this cohort; nonetheless, exploratory analysis indicates that many samples are noticeably blurred or low-contrast, which likely constrains achievable accuracy, especially for emotions whose discriminative cues lie around the mouth and chin areas frequently occluded by masks.



Figure 8: Dataset.py Use of Tensorflow

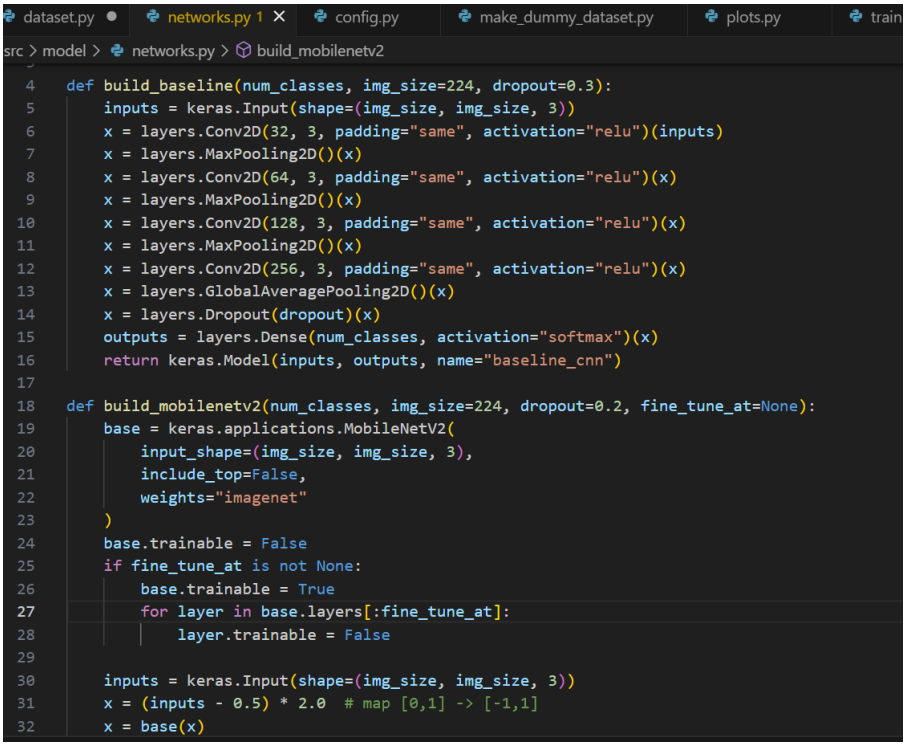


Figure 9: Network.py - Sample code of baseline and mobilenet v2

src > controller > evaluate.py > main

```python
def main():
    p = argparse.ArgumentParser("Evaluate Masked Emotion Classifier")
    p.add_argument("--val_dir", type=str, required=True)
    p.add_argument("--model_path", type=str, required=True)
    p.add_argument("--img_size", type=int, default=224)
    p.add_argument("--batch_size", type=int, default=32)
    p.add_argument("--label_mode", type=str, default="int", choices=["int","categorical"])
    args = p.parse_args()

    if not os.path.isdir(args.val_dir):
        raise SystemExit(f"[ERROR] Val directory not found: {args.val_dir}")

    val_ds_raw = tf.keras.utils.image_dataset_from_directory(
        args.val_dir,
        label_mode=args.label_mode,
        image_size=(args.img_size, args.img_size),
        batch_size=args.batch_size,
        shuffle=False,
    )
    class_names = val_ds_raw.class_names
    norm = tf.keras.layers.Rescaling(1.0 / 255.0)

    def preprocess(image, label):
        return norm(image), label

    val_ds = val_ds_raw.map(preprocess)
    model = keras.models.load_model(args.model_path)
```

Figure 10: Sample code of Evaluate.py

c > controller > predict.py > main

```python
import argparse, json
import numpy as np
from PIL import Image
from tensorflow import keras

def load_image(path, img_size):
    img = Image.open(path).convert("RGB").resize((img_size, img_size))
    arr = np.asarray(img).astype("float32") / 255.0
    return np.expand_dims(arr, 0)

def main():
    p = argparse.ArgumentParser("Predict one image")
    p.add_argument("image_path", type=str)
    p.add_argument("--model_path", type=str, required=True)
    p.add_argument("--class_names", type=str, default="experiments/model/artifacts/class_names.json")
    p.add_argument("--img_size", type=int, default=224)
    p.add_argument("--topk", type=int, default=3)
    args = p.parse_args()

    model = keras.models.load_model(args.model_path)
    with open(args.class_names, "r", encoding="utf-8") as f:
        class_names = json.load(f)

    x = load_image(args.image_path, args.img_size)
    probs = model.predict(x, verbose=0)[0]
    idxs = np.argsort(probs)[::-1][:args.topk]
    for i in idxs:
        print(f"{class_names[i]}: {probs[i]:.4f}")
```

Figure 11: Sample code of Predict.py

src > utils > plots.py > ...

```python
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

def save_history_plots(history, out_dir):
    os.makedirs(out_dir, exist_ok=True)
    hist = history.history

    plt.figure()
    if "accuracy" in hist: plt.plot(hist.get("accuracy", []), label="train_acc")
    if "val_accuracy" in hist: plt.plot(hist.get("val_accuracy", []), label="val_acc")
    if "top3" in hist: plt.plot(hist.get("top3", []), label="train_top3")
    if "val_top3" in hist: plt.plot(hist.get("val_top3", []), label="val_top3")
    plt.xlabel("Epoch"); plt.ylabel("Accuracy"); plt.legend(); plt.title("Accuracy / Top-3")
    plt.savefig(os.path.join(out_dir, "accuracy.png"), bbox_inches="tight"); plt.close()

    plt.figure()
    if "loss" in hist: plt.plot(hist.get("loss", []), label="train_loss")
    if "val_loss" in hist: plt.plot(hist.get("val_loss", []), label="val_loss")
    plt.xlabel("Epoch"); plt.ylabel("Loss"); plt.legend(); plt.title("Loss")
    plt.savefig(os.path.join(out_dir, "loss.png"), bbox_inches="tight"); plt.close()

def save_confusion_matrix(y_true, y_pred, class_names, out_path):
    cm = confusion_matrix(y_true, y_pred)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=class_names)
    disp.plot(xticks_rotation=45)
    plt.title("Confusion Matrix")
    plt.savefig(out_path, bbox_inches="tight")
    plt.close()
```

Figure 12: Sample code of plots.py

3.3.2 CSV Data Path

In addition to the directory-based image loader used for all operational experiments, we implement an auxiliary CSV representation of the dataset. Each image is resized to 224×224, converted to RGB, normalized to [0,1][0,1][0,1], and flattened into a single row of 150,528 pixel values preceded by an integer class label. This encoding fixes the exact preprocessing and label order at export time, which aids reproducibility and independent verification. At training time, the CSV is streamed via a TensorFlow `tf.data` pipeline, where rows are parsed, reshaped back to (224,224,3)(224,224,3)(224,224,3), and mapped to one-hot labels before entering the MobileNetV2 backbone. Although CSVs are substantially larger and slower to parse than image files, this path aligns with the coursework requirement to “accept pixel-based CSV data” and demonstrates that the proposed architecture is agnostic to the storage format of the inputs.

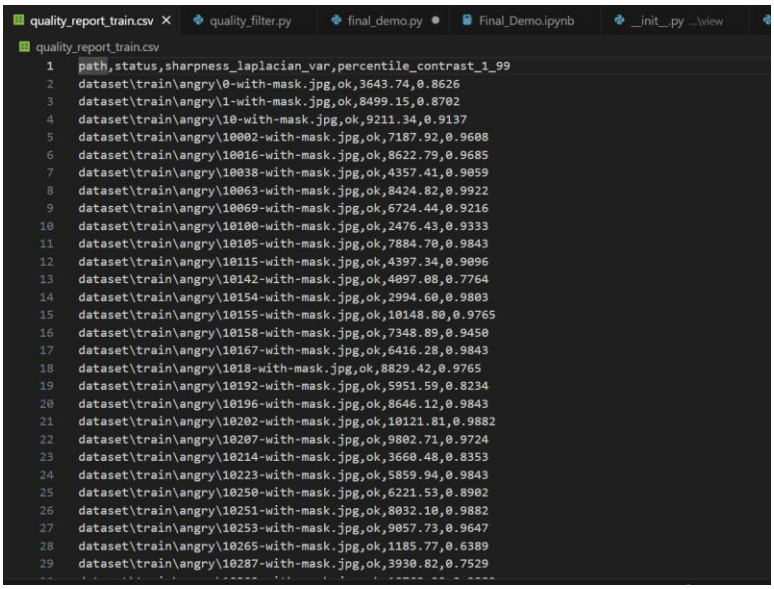


Figure 13: Creating a CSV file from dataset images

3.3.3 Training configuration.

We followed the two-stage protocol described above. In the head-only stage, the model trained for 2-3 epochs and achieved progressive improvements in validation accuracy as the head adapted. In the fine-tuning stage, we unfroze the last portion of MobileNetV2 and continued for up to 30 epochs with early stopping and best-checkpoint saving. Label smoothing, dropout, and (optionally) class weights and Mix-up were applied as regularizes.

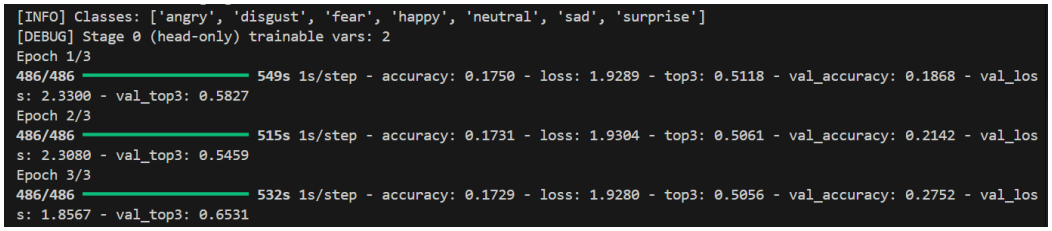


Figure 14: Head-only Stage of Training

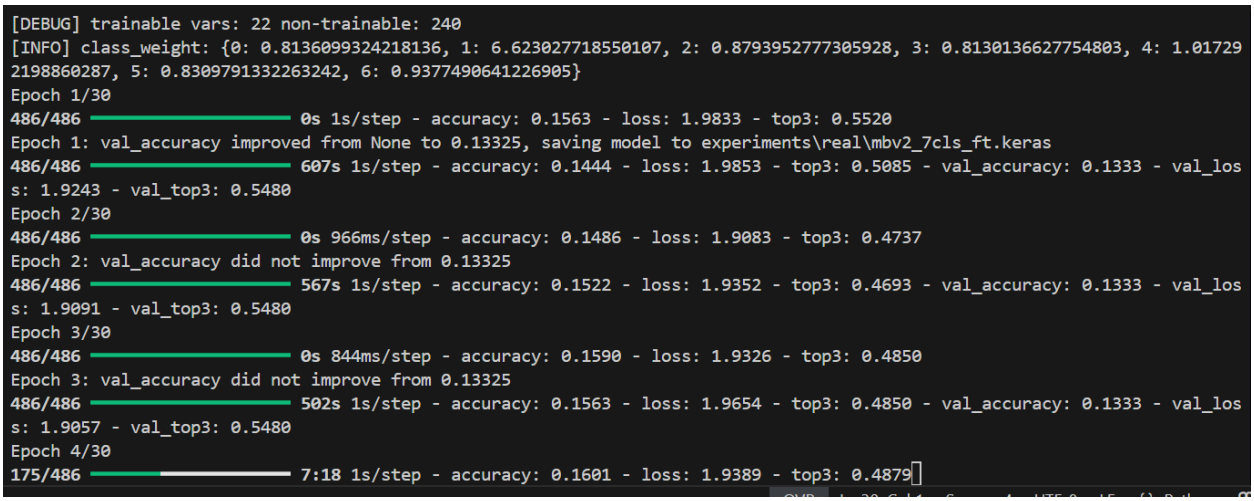


Figure 15: Fine Tuning Stage of Training

3.3.4 Quantitative results

The best checkpoint achieved a validation top-1 accuracy of 0.287 and a Top-3 accuracy of 0.610. Per-class F1 scores reveal that performance concentrates on the “happy” class, while “angry, disgust, fear, neutral, sad,” and “surprise” show weak discrimination under the current data conditions. The confusion matrix indicates frequent misclassification among negative or

subtle emotions an outcome consistent with occluded mouth cues, motion blur, and the limited resolution of salient features in many images.

3.3.5 Interpretation and limitations

Architecturally, the system behaves as expected: the top-3 metric substantially exceeds top-1, suggesting that the network often ranks the correct class among its leading hypotheses even when forced to make a single choice. The bottleneck is dominated by data quality and signal visibility rather than by architectural capacity. In particular, mask occlusion removes high-information regions for affect recognition, and blur degrades edge and texture cues that MobileNetV2 relies on. While stronger augmentation, heavier backbones, or longer training might produce incremental gains, meaningful improvements will most likely require clearer, better-lit images, more balanced class distributions, and possibly label curation.

4 Testing

This section documents how the system was tested, the datasets and protocols used, the quantitative outcomes obtained, and the principal failure modes observed. Our goals were to verify functional correctness of the end-to-end pipeline (data loading - preprocessing - inference/UI), quantify predictive performance with standard metrics, and understand where and why the model under-performs so that limitations are transparent during demonstration.

4.1 Test Environment and Reproducibility

All experiments were executed on the same workstation and code base as training. To promote reproducibility, we (a) fixed Python/TensorFlow random seeds in the training and evaluation controllers, (b) used the validation split only for all reported metrics (no training images were used), and (c) disabled data augmentation at evaluation time. The exact commands used to train, fine-tune and evaluate are preserved in the terminal logs and can be rerun to regenerate the same artefacts saved model, confusion matrix, and classification report JSON.

4.2 Test Data

Testing was performed on the prepared validation set at clean_dataset/val, comprising **4,953 images** across **7 classes** (angry, disgust, fear, happy, neutral, sad, surprise). Images were normalised to [0,1][0,1][0,1] and resized to 224×224224\times224224×224 pixels. No geometric or photometric augmentation was applied in evaluation. The class order during evaluation is identical to training, ensuring label consistency.

4.3 Test Plan and Protocol

Testing is structured into three layers:

- Data & pipeline sanity tests
    - Every file in clean_dataset/val decodes to RGB and maps to a valid class index.
    - Model forward pass returns a length-7 probability vector; probabilities are non-negative and sum to 1 (softmax).
    - Top-K extraction returns strictly ordered class indices without ties.
- Functional/UI tests
    - Batch and single-image prediction routes (controller and Streamlit app) produce identical class probabilities for the same input.
    - Low-confidence warnings trigger when the top-1 probability falls below a user-set threshold.
    - Grad-CAM generation is attempted; if gradients are unavailable (a known Keras edge case), the app falls back to an input-gradient saliency map without crashing.
- Quantitative evaluation
    - We report top-1 accuracy, top-3 accuracy, per-class precision/recall/F1, and the confusion matrix. These are computed by src/controller/evaluate.py and saved under experiments/real/eval/ (PNG + JSON).

To contextualize results, comparison done against a naïve baseline: random guessing over 7 classes yields an expected top-1 accuracy of 14.29% and top-3 of 42.86%.

4.4 Results

Using the final fine-tuned MobileNetV2 model (experiments/real/mbv2_7cls_ft.keras), evaluation on the held-out validation set produced:

- Top-1 accuracy: 28.73%
- Top-3 accuracy: 60.97%
- Per-class F1: markedly uneven; the happy class achieved the only substantial F1 (≈0.446). Other classes were frequently confused and, in several cases, received near-zero F1 given the model's tendency to default to a few dominant classes under uncertainty.
- Confusion matrix: misclassifications concentrate among affectively adjacent categories (e.g., neutral - sad, fear - surprise, angry - disgust). The matrix image is saved alongside the JSON report for inclusion in the appendix/screenshots.

Relative to the random baseline, the model more than doubles top-1 accuracy and improves top-3 by ~18 percentage points, indicating that it learns signal beyond chance despite the challenging data.

4.5 Error Analysis

Three factors dominate the observed errors: -

- Image quality (blur/low contrast). Many faces exhibit motion blur, compression artefacts, or poor lighting, which weakens edge and texture cues relied upon by the CNN backbone. Even after filtering, a sizeable subset remains marginal.
- Occlusion/domain shift (masks). Lower-face occlusion removes strong emotion cues (mouth curvature, nasolabial folds), shifting reliance to peri-ocular regions. Without large, well-labelled masked data, separation between subtle classes is weak.
- Class imbalance & label noise. The distribution across the seven labels is uneven and some samples are ambiguously labelled. The model therefore over-predicts relatively frequent/“easier” classes (happy) and under-predicts rare/ambiguous ones (disgust, fear).

5 Conclusion

The project delivers an end-to-end, reproducible pipeline for masked-face emotion recognition, spanning dataset curation with quality checks, augmentation (including a synthetic mask overlay), MobileNetV2 based feature extraction with a reset classification head, staged optimization (head warm-up followed by selective unfreezing), and a lightweight Streamlit interface with explanation maps. On the held-out validation set of 4,953 images over seven classes, the final model achieved 28.7% top-1 accuracy and 61.0% top-3 accuracy above random chance (14.3% top-1; 42.9% top-3) and indicative of non-trivial learning from peri-ocular cues despite occlusion and blur. Per-class F1 scores, however, reveal uneven performance: the classifier favors more salient expressions and struggles with subtle or under-represented classes. Explainability maps consistently highlight the eye and brow regions, supporting that the network is attending to plausible anatomical cues, while the user interface's low-confidence warnings make predictions transparent during demonstration.

Several lessons emerged. First, data quality dominates under heavy occlusion: blurred or low-contrast inputs sharply constrain attainable accuracy, regardless of backbone choice. Second, class balance matters: directory counts and the confusion matrix show that minority classes require more clean, diverse examples; class weighting alone offers limited gains when signals are intrinsically weak. Third, the two-stage training schedule is reliable but sensitive to learning-rate selection around the unfreeze point. Mix-up improved stability and top-3 performance but can dilute rare-class signals if not paired with targeted rebalancing. Finally, evaluating beyond top-1 and preserving training artifacts proved essential for trustworthy analysis and replicability.

The work remains constrained by image quality and domain mismatch, mask-induced information loss (lower-face cues removed), class imbalance and probable label noise, and the capacity–budget trade-off of a lightweight backbone. Augmentation diversity and probability calibration were not exhaustively explored, and generalization to unconstrained, real-world settings is unverified. Future iterations should prioritize higher-clarity, better-balanced data; targeted peri-ocular supervision; calibrated probabilities; and modest architecture and

augmentation upgrades improvements that are likely to yield larger gains than further tuning alone.

References

Agarwal, A., & Seba Susan. (2025, August 13). *Emotion Recognition from Masked Faces using Inception-v3*. Retrieved from IEEE Xplore: https://ieeexplore.ieee.org/document/10126777

Karadbhajne, M., Gomkale, M., A, C. N., Agrawal, R., & Dhule, C. (2025, August 13). *EmoNet: Exploring Facial Emotion Recognition Through Machine Learning*. Retrieved from IEEE Xplore: https://ieeexplore.ieee.org/document/10465804

Pratardanudu, N. S., Jayanth, Y., Madhur, S., Sumanth, D. S., Vishruth, G. S., & Reddy, D. C. (2025, August 12). *Emotion Detection of People Wearing Face Mask*. Retrieved from IEEE Xplore: https://ieeexplore.ieee.org/document/10104665

Yang, Z., Nayan, K., Fan, Z., & Cao, H. (2025, August 12). *Multimodal Emotion Recognition with Surgical and Fabric Masks*. Retrieved from IEEE Xplore: https://ieeexplore.ieee.org/document/9746414