Real-Time Scene Description and Interpretation using Zero-Shot Learning and Prompt-Engineered Vision-Language Models

by

Md Shifatul Ahsan Apurba
22266027

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
M.Sc. in Computer Science

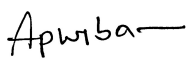
Department of Computer Science and Engineering
BRAC University
August 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Md Shifatul Ahsan Apurba

22266027

Approval

The thesis/project titled "Real-Time Scene Description and Interpretation via Zero-Shot Learning and Prompt-Engineered Vision-Language Models" submitted by

1. Md Shifatul Ahsan Apurba (22266027)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Computer Science on August 01, 2024.

Examining Committee:

Supervisor:
(Member)

Md. Tawhid Anwar

Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md Sadek Ferdous

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor & Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

Real-time scene description and interpretation are essential for diverse applications such as surveillance, interactive media, and automated video analysis. However, most existing methods rely heavily on large-scale labeled datasets, thereby limiting their adaptability in dynamic or previously unseen scenarios. In this work, we propose a novel mixed-model framework that integrates Vision-Language Models (VLMs), Large Language Models (LLMs), and lightweight object detection networks (e.g., MobileNet-SSD) through advanced prompt engineering. By leveraging zero-shot learning, our approach generates contextually rich scene descriptions without requiring domain-specific or task-specific retraining. The prompt engineering component reduces sensitivity to subtle linguistic variations, enhancing robustness across diverse input formulations. Furthermore, the lightweight detector ensures real-time performance, making the framework suitable for resource-constrained environments. To address ethical and fairness considerations, we incorporate bias mitigation strategies that limit the propagation of harmful stereotypes from large-scale pretraining data. Experimental evaluations on multiple open-domain scenarios demonstrate that our system offers reliable and efficient scene interpretation, maintaining high accuracy in challenging conditions where traditional supervised techniques often fail. This research paves the way for more flexible, scalable, and responsible vision-language systems capable of operating effectively in real-world, zero-shot contexts.

Keywords: Vision-Language Models (VLMs), Large Language Models (LLMs), Prompt Engineering, Zero-Shot Learning, Real-Time Scene Description, Lightweight Detection Models, Contextual Analysis

Acknowledgement

This thesis marks the end of a journey that would not have been possible without the support and encouragement of many individuals.

First and foremost, I am deeply grateful to my supervisor, **Md. Tawhid Anwar**, for his guidance, patience, and insightful advice throughout this research. His mentorship has been a steady source of direction and inspiration.

I would also like to thank my friends, juniors and peers who shared this academic journey with me, for the thoughtful discussions, late-night brainstorming, and shared motivation during challenging times.

Finally, I owe the deepest gratitude to my family. Their unwavering support, encouragement, and belief in me provided the strength I needed to persevere and complete this work. This milestone belongs to them as much as it does to me.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The field of computer vision has advanced considerably in tasks such as object detection, semantic segmentation, and scene understanding, driven largely by supervised learning methods that depend on large annotated datasets. While these approaches yield high accuracy in controlled settings, they remain susceptible to changes in domain or context due to their reliance on extensive and carefully curated labels. A model trained to recognize specific objects or actions may struggle when novel or rare entities appear, thus demanding repeated retraining and labor-intensive annotation whenever a new class of interest arises [1].

In parallel, natural language processing has seen transformative progress through Large Language Models (LLMs), which learn from massive text corpora to complete various tasks without extensive task-specific training. Researchers have extended this principle to form Vision-Language Models (VLMs), wherein visual inputs (e.g., images) are aligned with corresponding textual descriptions. A notable example is CLIP [2], which encodes images and text into a shared embedding space, enabling zero-shot image classification through text prompts. However, many VLMs, including CLIP, exhibit significant sensitivity to prompt design, leading to inconsistent performance if the textual input is not meticulously [3].

Real-time scene interpretation introduces further complexities. Many state-of-the-art VLMs require substantial computational resources [4], constraining their use in latency-critical domains such as surveillance, robotics, and augmented reality. While hardware acceleration and model optimization can mitigate some issues, ensuring robust and consistent performance in high-frame-rate or resource-limited environments remains a challenge. Prompt engineering [5] [6] has emerged as a strategy to improve zero-shot performance, yet it frequently overfits to narrow contexts, diminishing its utility in broader applications.

Meanwhile, lightweight detection architectures [7], such as MobileNet or SSD, offer high-speed operations suitable for real-time implementations. These models can identify candidate regions or labels rapidly, enabling immediate data processing in demanding environments. When integrated with a more expressive VLM, such detectors can provide both efficiency and contextual depth. However, merging these

distinct pipelines raises questions related to feature alignment and inference flow. Additionally, VLMs often reflect biases from large-scale, unfiltered training data. In scenarios involving sensitive analyses or ethical considerations, this predisposition can lead to unfavorable or biased outcomes, highlighting the need for effective bias-mitigation techniques.

Motivated by these shortcomings in existing systems, we propose a novel hybrid framework that pairs the efficiency of a lightweight object detection model with the adaptability of a vision-language backbone under a shared prompt-engineering pipeline. This approach aims to facilitate reliable, zero-shot scene interpretation while reducing the reliance on extensive labeled datasets and maintaining real-time responsiveness. By incorporating systematic prompt optimization and targeted bias controls, the framework aspires to deliver robust performance across various domains and contexts.

1.2 Motivation

Real-time scene description and interpretation have become integral to a range of modern applications, including advanced surveillance, human–robot collaboration, autonomous navigation, and interactive media experiences. In these settings, the ability to accurately detect objects, interpret activities, and generate context-rich descriptions in real time directly enhances safety and user engagement. Nevertheless, conventional supervised approaches typically rely on extensive labeled datasets for each new class or behavior of interest, a requirement that becomes increasingly impractical in dynamic or evolving scenarios [8]. Moreover, tasks such as violence detection or anomaly identification often involve rare or unpredictable events, making it exceedingly difficult to produce comprehensive labeled data.

As a response to these limitations, there has been a surge of interest in zero-shot learning techniques capable of handling previously unseen classes and novel situations without exhaustive annotation. Such methods align with the capacity of contemporary foundation models to generalize across diverse conceptual spaces. However, models operating in zero-shot settings frequently face challenges related to prompt engineering, computational overhead, and biases inherited from large-scale, uncurated training sets. Addressing these concerns demands a more nuanced framework that balances efficiency, adaptability, and ethical safeguards in real-time environments. By integrating the strengths of Large Language Models (LLMs), Vision-Language Models (VLMs), and streamlined detection architectures, it becomes feasible to meet real-world requirements where extensive labeled data remain difficult or impossible to obtain.

1.3 Research Problem

Despite the promise of foundation models in zero-shot scenarios, key gaps remain when integrating them for real-time scene interpretation.

- **Prompt Sensitivity and Consistency:** Many Vision-Language Models (VLMs) exhibit significant performance variations based on how textual prompts are formulated. This makes it challenging to achieve stable results in dynamic settings, where prompt phrasing can shift frequently.
- **Computational Constraints:** Current large-scale models often require considerable computational resources, posing difficulties for real-time applications on edge devices or latency-sensitive platforms.
- **Limited Integration with Lightweight Object Detection:** Although large VLMs excel at broad conceptual understanding, practical real-time systems often utilize efficient detectors (e.g., MobileNet or SSD). However, effective strategies for fusing these architectures in zero-shot scene description remain insufficiently explored.

Collectively, these constraints illustrate the need for a robust, hybrid framework that leverages prompt engineering to reduce sensitivity and harnesses proven lightweight detectors for real-time operation while maintaining zero-shot adaptability.

1.4 Research Objective

This study seeks to design, implement, and assess a mixed-model framework that integrates Vision-Language Models (VLMs), Large Language Models (LLMs), and a MobileNet-based detection backbone via advanced prompt engineering. The fundamental hypothesis is that this hybrid approach can facilitate robust zero-shot scene interpretation in real-time settings, accommodating a broad spectrum of previously unseen objects and activities without substantial task-specific labeling.,

- **Objective A – Framework Design:** Develop an integrated architecture that pairs a pretrained MobileNet-SSD detector with a vision-language mechanism guided by carefully formulated text prompts. System-level considerations include data flow, model interfacing, and prompt design.
- **Objective B – Prompt Engineering Approaches:** Investigate and compare strategies like automated prompt generation, input conditioning to ensure consistent performance despite variations in descriptive language. A primary aim is to balance specificity (to capture relevant scene details) with generality (to enable zero-shot handling of novel objects or actions).
- **Objective C – Real-Time Application:** Evaluate the proposed system in live or high-throughput scenarios, focusing on seamless integration between the lightweight detector and VLM/LLM components. This objective includes ensuring minimal latency and smooth operation under changing environmental conditions.

In light of these research objectives, we aim to address the following research questions.

RQ1: To what extent does a prompt-engineered hybrid architecture, which combines lightweight detection and vision-language modeling, effectively adapt to previously unseen categories in dynamic real-time scenarios?

RQ2: Which elements in system design and prompt formulation most significantly influence the framework's capacity to sustain robust and context-appropriate scene interpretations across varied application domains?

Through an organized approach to these objectives, the research seeks to establish that hybrid VLM–LLM–MobileNet architectures can facilitate adaptable and dependable zero-shot scene interpretation, while reducing the reliance on domain-specific annotations.

1.5 Paper Organization

This thesis is structured to progress from background and positioning, through the proposed approach.

- **Chapter 1: Introduction.** Presents the motivation for real-time, zero-shot scene interpretation; states the problem definition, scope, and assumptions; formulates the research objectives and questions; and previews the central idea of coupling a lightweight detector with vision–language components under a unified prompt-engineering pipeline.
- **Chapter 2: Related Work.** Reviews the literature on vision–language models, prompt engineering, and real-time scene understanding. The survey is comparative, contrasting modeling interfaces and training. Identified gaps motivate the methods developed later.
- **Chapter 3: Methodologies.** Details the proposed hybrid framework, including the integration of a MobileNet-SSD detector with VLM and LLM components, the data flow between modules, and an inference schedule compatible with real-time operation.
- **Chapter 4: Experimental Setup and Evaluation.** Specifies datasets, tasks, baselines, and implementation settings; defines metrics for recognition quality and system performance; and reports results.
- **Chapter 5: Conclusion.** Summarizes the main findings and contributions, discusses limitations and threats to validity, and outlines directions for future work—including broader application domains, stronger bias auditing and mitigation procedures, and tighter real-time guarantees on edge hardware. Implications for reproducibility and responsible deployment are also considered.

Chapter 2

Related Work

Research on task-agnostic vision–language models (VLMs) has progressed along four recurring dimensions: (i) *scaling* (data volume, multilingual coverage), (ii) *architectural interface* between modalities (tokenization schemes, query modules, prompt machinery), (iii) *training signal* (multi-task supervision, instruction tuning, or generative/reward-based objectives), and (iv) *deployment constraints* (compute, latency, robustness to prompt variation, and fairness). The studies below are organized thematically with emphasis on contrasts and trade-offs.

Foundation, task-agnostic VLMs. Unified-IO [9] standardizes heterogeneous inputs/outputs into discrete tokens processed by a single transformer, yielding a unified interface across tasks and strong results (including completion of all GRIT benchmark tasks). The main costs are substantial training and inference compute and exposure to bias from web-scale data. PaLI [10] instead prioritizes scale and multilinguality (10B+ image–text pairs, 100+ languages), achieving strong performance on VQA, captioning, and classification across languages. Relative to Unified-IO, PaLI trades a uniform I/O tokenization for breadth of linguistic coverage; both systems attain broad capability at the expense of accessibility and fairness auditing.

Instruction-tuned VLMs. InstructBLIP [11] augments a pretrained VLM with instruction–response pairs and an instruction-aware query transformer, improving zero-shot generalization (e.g., ScienceQA 90.7%). Compared with scaling-only strategies [9], [10], instruction tuning offers adaptability with less raw data, but depends strongly on the coverage and quality of instructions and adds inference/training overhead due to additional modules.

Alternative zero-shot mechanisms. Diffusion-based classification [12] frames recognition as text-conditioned generation and comparison. This provides competitive performance—particularly when shape bias is beneficial—while introducing iterative sampling latency and sensitivity to prompt phrasing. In reinforcement learning, VLMs serve as zero-shot reward models [13], replacing hand-crafted rewards with natural language goals. This improves goal specification and task scalability, but inherits prompt brittleness, potential bias from pretraining, and nontrivial compute demands.

Prompting and prompt learning. Prompt design offers orthogonal gains. Simple chain-of-thought prompting [14] improves reasoning without additional training yet is fragile to small wording changes. Automatic Prompt Engineer (APE) [15] reduces manual effort by searching prompts with the model itself, exchanging steadier gains for iterative compute and possible bias amplification. Conditional CoOp (CoCoOp) [16] conditions prompts on image features, improving transfer to unseen classes relative to static prompts, while introducing extra networks and residual sensitivity under domain shift.

Safety and moderation applications. Domain-specific moderation systems illustrate practical constraints. For children's cartoons, a CLIP-based approach with context-aware prompts [17] improves classification but struggles with subtle/varied violence and incurs video-processing cost. UGCG-GUARD [18] combines conditional prompting and chain-of-thought with large VLMs for zero-shot detection of violent or sexually explicit promotional content, trading accuracy for high resource requirements and fairness risks. VIVID [19] integrates VLM-generated descriptions with LLM reasoning, using optical-flow keyframe selection and knowledge-based prompts to attain strong results in surveillance/film datasets; reliance on predefined knowledge can limit coverage of novel cases, and real-time operation remains challenging.

Synthesis and open questions. Scaling-centric models [9], [10] offer breadth but face compute and fairness constraints; instruction tuning [11] improves adaptability with modest overhead yet hinges on instruction quality; generative and reward-based routes [12], [13] increase flexibility but introduce latency and prompt sensitivity; and prompt methods [14], [15], [16] supply complementary gains at the cost of stability or complexity. Safety-focused systems [17], [18], [19] underscore the tension between accuracy in-the-wild and deployability (throughput, transparency, and fairness). An open space remains for methods that combine strong zero-shot generalization with strict latency/compute budgets, prompt-robust behavior, and explicit bias auditing during deployment.

Category	Paper	Training	Strengths	Limitations
Foundation VLMs	Unified-IO [9]	Unified tokenized I/O; multi-task	Strong multi-task performance; completes GRIT; single interface	Large compute; bias from web-scale data
	PaLI [10]	Multilingual pre-training (10B+ pairs)	Strong multilingual VQA/captioning/classification	High resource cost; fairness concerns
Instruction-tuned	InstructBLIP [11]	Instruction–response; query transformer	Better zero-/few-shot (ScienceQA 90.7%); adaptable	Instruction coverage/quality; extra modules/overhead
Zero-shot mechanisms	Diffusion classifier [12]	Text-conditioned diffusion comparison	Competitive without discriminative heads; shape-biased robustness	Iterative sampling latency; prompt sensitivity
RL with VLM rewards	CLIP rewards [13]	Language-specified rewards via VLM	Scalable goal specification; fewer hand-crafted rewards	Bias inheritance; prompt brittleness; compute cost
Prompting	CoT prompting [14]	Manual prompts at inference	Low overhead; improves reasoning	Fragile to phrasing; instability across prompts
	APE [15]	Automated prompt search	Reduces manual effort; steady gains	Iteration cost; possible bias amplification
	CoCoOp [16]	Image-conditioned prompts	Better generalization to unseen classes	Added networks; domain-shift sensitivity
Moderation	Cartoons [17]	CLIP + context-aware prompts	Domain-specific improvements	Generalization limits; video throughput
	UGCG-GUARD [18]	Conditional + chain-of-thought	High zero-shot accuracy	Large-model compute; fairness risk
	VIVID [19]	VLM descriptions + LLM reasoning	Strong results on film/surveillance	Knowledge coverage; real-time constraints

Table 2.1: Comparison across modeling interface, supervision, and deployment

Chapter 3

Methodologies

3.1 Used Models

Four principal pre-trained models were employed in the development of our hybrid framework. These models comprise MobileNet v2, Common Objects in Context with Single Shot MultiBox Detector, FaceAPI, and the Generative Pre-trained Transformer.

3.1.1 MobileNet v2

MobileNet v2 [20] is a lightweight convolutional neural network (CNN) architecture optimized for mobile and embedded vision applications. It improves upon its predecessor, MobileNet v1, by introducing depthwise separable convolutions, which significantly reduce computational complexity, and inverted residual blocks with linear bottlenecks, which enhance feature representation while maintaining efficiency. Unlike traditional convolutional networks, MobileNet v2's architecture expands low-dimensional feature maps to a high-dimensional space, processes them through lightweight depthwise convolutions, and then projects them back to a compact form. This design prevents information loss from non-linear activation layers, leading to improved accuracy with fewer computations. MobileNet v2 achieves a good trade-off between model size, inference speed, and classification accuracy, making it highly suitable for resource-constrained environments such as mobile devices, IoT systems, and real-time object detection tasks [21]. It has been widely adopted in image classification, object detection, and facial recognition applications, particularly in scenarios where real-time processing is essential. Compared to larger CNN architectures, MobileNet v2 maintains competitive accuracy while being significantly more efficient, making it a standard backbone for vision models deployed in mobile and edge computing applications.

3.1.2 Single Shot MultiBox Detector

The Single Shot MultiBox Detector (SSD) is a one-stage object detection model designed for real-time applications by eliminating the need for region proposal networks, as used in Faster R-CNN. SSD predicts object classes and bounding boxes directly from feature maps at multiple scales, allowing it to detect both large and small objects efficiently [22]. The COCO SSD variant refers to an SSD model trained

on the Common Objects in Context (COCO) dataset [23], which contains 328,000 images and 2.5 million labeled instances across 91 object categories. The COCO dataset's diverse and complex real-world scenes improve SSD's robustness, particularly in handling occlusions and varied object sizes. Unlike two-stage models, SSD makes predictions in a single forward pass, leading to high inference speed (tens of frames per second on modern GPUs) while maintaining competitive mean Average Precision (MAP). This makes COCO SSD particularly useful for video surveillance, autonomous driving, and augmented reality applications, where real-time detection is critical. While SSD generally outperforms YOLO in small object detection, it has limitations in extreme low-light conditions or highly cluttered scenes. Despite this, SSD remains a preferred choice for mobile-friendly object detection due to its balance between accuracy and speed.

3.1.3 FaceAPI

Microsoft's FaceAPI is a cloud-based facial recognition service that provides face detection, recognition, and analysis using deep learning models. It identifies facial landmarks, emotions, age, gender, and expressions, making it widely used in security, authentication, and human-computer interaction (HCI) applications. The underlying architecture leverages convolutional neural networks (CNNs) trained on large-scale datasets, allowing it to generalize well across different lighting conditions, facial orientations, and occlusions. FaceAPI operates as a cloud service, offering scalability for large-scale deployments in identity verification, automated surveillance, and retail analytics [24]. It has been benchmarked for high accuracy in real-world applications, with robust face verification capabilities that match or exceed industry standards. Continuous updates to its detection models have improved its ability to recognize faces with masks, side profiles, and low-resolution inputs, making it more adaptable to practical use cases. While FaceAPI offers real-time face recognition, its cloud-based nature introduces latency in applications requiring immediate feedback. However, its ease of integration, high recognition accuracy, and support for custom training on user datasets make it a leading solution in AI-powered facial analysis.

3.1.4 Generative Pre-trained Transformer

Generative Pre-trained Transformer (GPT) is a large-scale language model based on the Transformer architecture, designed for natural language understanding and generation [25]. Developed by OpenAI, GPT has evolved from GPT-1 to GPT-4, with each iteration increasing in size, improving context awareness, and enhancing zero-shot learning capabilities. GPT is trained on massive internet-scale text corpora, enabling it to generate coherent and contextually relevant text responses without task-specific fine-tuning. Its self-attention mechanism allows it to consider relationships between words across long sequences, making it superior to earlier RNN-based models. GPT-3 and GPT-4, in particular, demonstrate few-shot learning, meaning they can perform complex tasks with minimal examples, making them highly adaptable for diverse applications such as chatbots, content generation, programming assistance, and multimodal vision-language tasks. In scene interpretation, GPT can process object detection outputs (from models like COCO SSD) and generate meaningful textual descriptions of scenes, enhancing human-computer in-

teraction in AI-powered vision systems. While GPT is a powerful generative model, it faces challenges such as hallucinating incorrect information, computational inefficiency, and potential biases from its training data. Despite these limitations, GPT's versatility in zero-shot reasoning, creative writing, and multimodal AI makes it one of the most influential advancements in modern AI research.

3.2 Model Architecture

Our proposed model architecture integrates vision-language processing to enable real-time scene interpretation with zero-shot learning capabilities. The model is structured into three primary blocks: Detection Block, Prompt Block, and Analysis Block, where each block is responsible for a key stage of object detection, structured prompt formulation, and scene interpretation through knowledge transfer mechanisms. The Analysis Block plays a pivotal role in zero-shot implementation, leveraging Large Language Models (LLMs) to interpret, expand, and reason over detected scene elements, ensuring adaptability to unseen categories without requiring additional training.

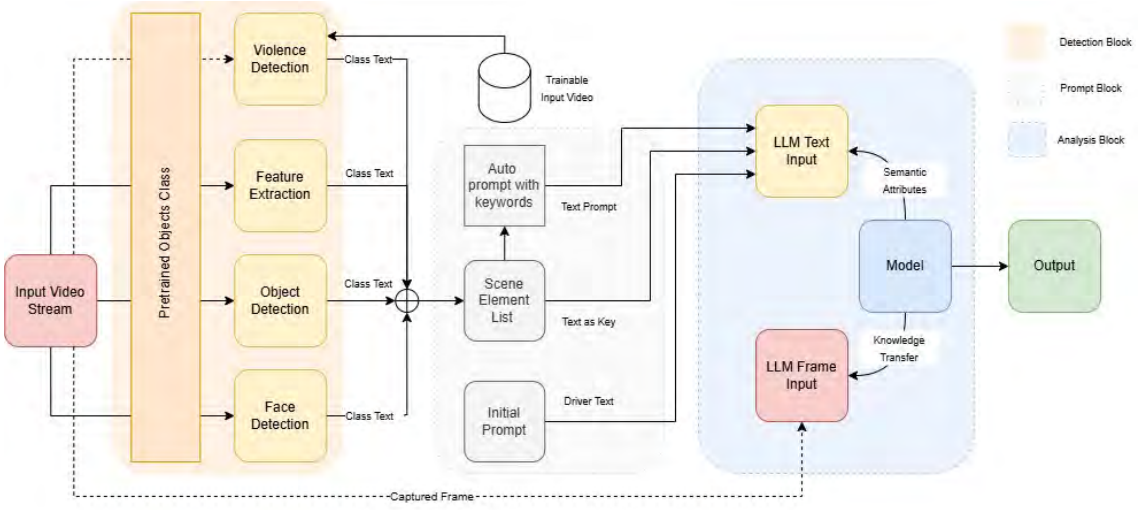


Figure 3.1: Proposed Model Architecture

3.2.1 Detection Block

The Detection Block serves as the model's perception layer, responsible for extracting semantically meaningful information from visual scenes in real-time. It processes each input frame to generate structured outputs that are passed to subsequent language and reasoning modules. The block is composed of four interconnected components: feature extraction, object detection, face detection, and violence detection. Each module is trained with carefully selected loss functions that align with its task-specific objectives and contribute to the overall system performance, particularly in dynamic and safety-critical environments.

Feature Extraction

Our architecture adopts **MobileNet v2** as the primary feature extractor due to its efficiency and favorable accuracy-to-complexity ratio. MobileNet v2 utilizes depthwise separable convolutions and inverted residual connections to extract high-dimensional feature maps from input images, significantly reducing computational overhead without sacrificing representational power.

To ensure that the extracted features are semantically rich and generalizable, we fine-tune MobileNet v2 using an auxiliary classification objective. This encourages the network to preserve class-discriminative cues even before downstream detection tasks are applied. Additionally, we include L2 regularization to improve model generalization and stability:

$$\mathcal{L}_{\text{feature}} = \mathcal{L}_{\text{cls}} + \lambda_1 \|W\|_2^2 \tag{3.1}$$

Here, $\mathcal{L}_{\text{cls}}$ is the cross-entropy loss over auxiliary labels during training, W represents the model's weight parameters, and λ_1 is a regularization coefficient. This formulation ensures that the backbone learns feature representations that are transferable across object categories and facial expressions, which is crucial for subsequent modules.

Object Detection

To localize and identify objects in the scene, we incorporate a **Single Shot MultiBox Detector (SSD)** trained on the COCO dataset. SSD performs object detection across multiple scales in a single forward pass, producing bounding boxes, class predictions, and associated confidence scores with minimal latency. This makes it well-suited for real-time applications.

The training objective for SSD combines two sub-tasks: bounding box regression and object classification. We employ the standard SSD multi-task loss function:

$$\mathcal{L}_{\text{SSD}} = \mathcal{L}_{\text{loc}} + \alpha \mathcal{L}_{\text{conf}} \tag{3.2}$$

In this formulation, $\mathcal{L}_{\text{loc}}$ is the Smooth L1 loss between predicted and ground truth bounding box coordinates, which promotes stable training by reducing sensitivity to outliers. $\mathcal{L}_{\text{conf}}$ is the softmax cross-entropy loss over the COCO object categories, penalizing incorrect class predictions. The coefficient α balances the contributions of the two terms, and is empirically set to 1.0. By optimizing both spatial accuracy and categorical precision, this loss supports robust object-level understanding across varied environments.

Face Detection

To extract human-centric features from the visual scene, we integrate a face detection pipeline based on the **FaceAPI** framework. This module detects the presence of faces, localizes facial landmarks (e.g., eyes, nose, and mouth), and classifies emotional expressions and pose attributes. These features serve as a foundation for higher-level reasoning about human intent and behavior.

We train the face detection module using a composite loss function designed to jointly optimize detection, alignment, and emotion recognition:

$$\mathcal{L}_{\text{face}} = \mathcal{L}_{\text{bbox}} + \mathcal{L}_{\text{landmark}} + \mathcal{L}_{\text{emotion}} \tag{3.3}$$

Here, $\mathcal{L}_{\text{bbox}}$ is the Smooth L1 loss for face bounding box regression, encouraging tight and stable localization. $\mathcal{L}_{\text{landmark}}$ is the Mean Squared Error (MSE) between predicted and ground truth keypoints, which ensures accurate facial alignment. $\mathcal{L}_{\text{emotion}}$ is a categorical cross-entropy loss over predefined emotional states, enabling the model to infer human affect. This multi-objective formulation enhances the system's ability to recognize not just the presence of people, but their emotional and attentional context—information that is particularly valuable in downstream scene interpretation.

Violence Detection Module

In addition to static object and face detection, we incorporate a **violence detection module** aimed at identifying aggressive or dangerous human behaviors. Unlike the other components, this module operates over temporal windows, analyzing motion and appearance patterns across consecutive frames. This allows the system to respond to dynamic and safety-critical events in real-time.
We model this task as a binary classification problem and optimize it using the Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}_{\text{violence}} = -\left[y \log p + (1-y) \log(1-p)\right] \tag{3.4}$$

where $y \in \{0, 1\}$ is the ground truth label indicating the presence of violent activity, and p is the predicted probability. This formulation penalizes both false positives and false negatives, ensuring sensitivity to high-risk behavior while minimizing false alarms. Although violence detection is not the primary focus of this work, its inclusion demonstrates the extensibility of the proposed framework to mission-critical domains.

Combined Detection Loss

To facilitate simultaneous training of all detection modules, we define a unified loss function as the weighted sum of the individual sub-component losses:

$$\mathcal{L}_{\text{detection}} = \lambda_f \mathcal{L}_{\text{feature}} + \lambda_o \mathcal{L}_{\text{SSD}} + \lambda_h \mathcal{L}_{\text{face}} + \lambda_v \mathcal{L}_{\text{violence}} \tag{3.5}$$

Here, λ_f, λ_o, λ_h, and λ_v are scalar hyperparameters controlling the relative contributions of each component. These values are selected through empirical tuning to ensure balanced learning and stable convergence. By integrating multi-scale object detection, human-centric facial analysis, and temporal violence recognition within a shared detection block, our system builds a comprehensive and robust foundation for high-level scene understanding.

3.2.2 Prompt Block

Once the Detection Block extracts semantic and spatial information from the visual input, the Prompt Block transforms these outputs into structured natural language descriptions that serve as inputs for the LLM-based scene analysis. This block bridges the gap between vision and language by translating raw detections into

task-aware textual prompts. By doing so, it enables the model to leverage the zero-shot generalization capabilities of large language models (LLMs) without requiring additional supervision or retraining. The Prompt Block consists of two key components: the Prompt Generation Module and the Automated Prompt Refinement mechanism.

Prompt Generation Module

The Prompt Generation Module is responsible for converting structured visual outputs—such as object labels, bounding box coordinates, and facial attributes—into coherent natural language descriptions. These descriptions are formulated according to a predefined syntactic and semantic template, ensuring consistency and interpretability. The template includes positional relationships, object co-occurrence, and human-centric cues (e.g., emotions or head pose), enabling the prompt to capture a high-level understanding of the scene context.
Given a set of N detected objects $\{o_1, o_2, \ldots, o_N\}$ with associated class labels showed as $\{c_1, c_2, \ldots, c_N\}$, bounding boxes $\{b_1, b_2, \ldots, b_N\}$, and attributes $\{a_1, a_2, \ldots, a_N\}$, the scene graph is implicitly represented as a tuple:

$$\mathcal{S} = \{(c_i, b_i, a_i) \mid i = 1, 2, \ldots, N\} \tag{3.6}$$

The module transforms $\mathcal{S}$ into a descriptive prompt P via a mapping function Φ, such that:

$$P = \Phi(\mathcal{S}) = \text{Template}(\mathcal{R}(c_i, b_i, a_i)) \tag{3.7}$$

Here, $\mathcal{R}(\cdot)$ is a relational parser that encodes spatial and semantic relationships (i.e.,"near," "holding," "looking at") between pairs of objects or people. The final prompt P is a natural language string that captures the situational layout of the scene. For example:

- **Raw Input (from SSD + FaceAPI)**: *"Person detected near a vehicle. Another person holding an object. Face detected with neutral expression."*
- **Generated Prompt**: *"Describe the scene where a person is standing near a vehicle, and another individual is holding an object. Consider the facial expressions detected in the environment."*

This structured prompt format ensures that the LLM receives contextually grounded and semantically rich information, which is essential for downstream reasoning tasks such as behavior inference, activity forecasting, or anomaly detection.

Automated Prompt Refinement and Follow-Up Queries

Although the initial prompt generation ensures semantic consistency, ambiguity often arises due to occlusions, overlapping objects, or incomplete visual cues. To address this, we incorporate an Automated Prompt Refinement mechanism that enhances prompt quality through contextual optimization, knowledge augmentation, and query expansion. This mechanism operates in three stages:

1. **Contextual Prompt Optimization**: When visual evidence is ambiguous or conflicting (e.g., partial visibility or overlapping bounding boxes), the system generates alternative phrasing or clarification prompts. These are derived by identifying low-confidence detections or uncertain spatial relationships and rephrasing them to elicit disambiguation from the LLM.

2. **Knowledge-Augmented Prompting**: To ensure linguistic and semantic coherence, we refine initial prompts using external knowledge embedded within the LLM. Specifically, detected entities are enriched with domain-relevant attributes (e.g., common actions associated with "lamp post" or typical roles of "person near a vehicle") based on the LLM's priors. This process can be formalized as:

$$P' = \Psi(P, K) \tag{3.8}$$

   where P is the original prompt, K denotes the knowledge embeddings or inferred context from the LLM, and $\Psi(\cdot)$ is the refinement operator that updates object descriptions, relationships, or scene dynamics accordingly.

3. **Adaptive Query Expansion**: To handle uncertainty or uncover latent scene details, we iteratively generate follow-up questions or sub-prompts. These queries are designed to probe the LLM's understanding and encourage multi-turn reasoning. For example, if the prompt includes "a person holding an object," the system may follow up with: "What type of object might the person be holding given the context?" The process continues until either convergence or a confidence threshold is reached.

This multi-stage refinement enables the system to generate prompts that are not only syntactically correct but also contextually optimized for reasoning. Importantly, the refinement pipeline operates without requiring annotated linguistic datasets, relying instead on the generalization capabilities of the LLM—a key property for zero-shot learning.
By abstracting low-level visual data into structured language, the Prompt Block functions as a crucial translation layer that enables general-purpose, language-based scene understanding. Its design ensures adaptability to novel scenes and object configurations, serving as a foundation for robust zero-shot interpretation in the subsequent Analysis Block.

3.2.3 Analysis Block

The Analysis Block constitutes the core reasoning engine of our architecture, enabling zero-shot scene interpretation through integration with a large-scale pre-trained language model. This block receives structured prompts generated from the visual input and performs high-level semantic inference without requiring additional supervised training. Its functionality is grounded in the principles of transfer learning and contextual generalization, allowing the system to operate effectively in dynamic environments and previously unseen scenarios.

GPT-Based Zero-Shot Scene Interpretation

At the core of the Analysis Block is a Generative Pre-trained Transformer (GPT), which functions as the zero-shot interpreter. Once a structured prompt is generated by the Prompt Block, it is forwarded to the GPT model, which leverages its vast internal knowledge to perform semantic expansion, causal reasoning, and behavior interpretation.
The GPT model is queried with a prompt P that encodes the visual structure of the scene. It produces a sequence of natural language tokens $\{x_1, x_2, \ldots, x_T\}$ which together form a high-level scene description $\hat{S}$:

$$\hat{S} = \text{LLM}(P) = \arg\max_{x_1,\ldots,x_T} \prod_{t=1}^{T} P(x_t \mid x_{<t}, P) \tag{3.9}$$

This autoregressive decoding process enables the model to infer meaning and relationships that are not explicitly captured by the visual detectors. Specifically, the LLM performs tasks such as:

- **Inferring object relationships** beyond direct detection outputs.
- **Applying real-world knowledge** to describe contextual interactions (e.g., "A person holding an object near a vehicle might be refueling or loading luggage.")
- **Describing human behaviors** based on facial expressions (e.g., "A person detected with a neutral face may be observing their surroundings.")
- **Predicting scene evolution** or detecting anomalies in spatial or behavioral patterns.

Because the LLM is pre-trained on diverse, large-scale text domain including social, procedural, and narrative content—it can reason about human-object interactions and environmental context with minimal reliance on domain-specific tuning. This makes it especially suitable for zero-shot applications in real-world, unstructured environments.

Knowledge Transfer Mechanism

To facilitate alignment between visual inputs and the LLM's internal semantic space, we incorporate a knowledge transfer mechanism that maps detected visual elements to language-based representations. This mechanism ensures that the LLM can interpret new inputs in a way that is coherent with its prior knowledge.
Formally, let $\mathcal{V} = \{(c_i, a_i)\}$ be the set of detected object classes c_i and their associated attributes a_i. These are mapped to semantic embeddings by the following function ϕ:

$$\mathcal{E} = \phi(\mathcal{V}) = \{\phi(c_i, a_i)\} \tag{3.10}$$

The embeddings $\mathcal{E}$ are then used to condition or refine the LLM's understanding of the prompt, allowing for more nuanced reasoning. The key processes enabled by this mechanism include:

- **Semantic Attribute Mapping**: Aligning visual features (e.g., "person holding object") with language-level concepts (e.g., "carrying," "offering," "interacting").
- **Dynamic Context Expansion**: Injecting relevant background knowledge to enrich sparse or ambiguous scenes (e.g., associating "person near fire" with potential hazard).
- **Adaptive Reasoning for Unseen Concepts**: If a novel object or configuration is detected, the LLM extrapolates plausible roles or functions (e.g., an unfamiliar object next to a suitcase might be inferred as luggage or travel gear).

This fusion of visual grounding and linguistic inference results in robust zero-shot generalization. The model is not limited to fixed category labels but instead operates on conceptual relationships and learned priors, enabling flexible interpretation across a wide variety of domains and environments.
So, Analysis Block serves as the reasoning bridge between perceptual data and high-level understanding. It expands sparse visual inputs into rich, explanatory language that supports downstream applications such as behavior forecasting, anomaly detection, or semantic tagging—without the need for task-specific supervision.

3.3 Datasets

To develop and evaluate the various components of the proposed framework, a combination of publicly available datasets was utilized. Each dataset was selected based on its relevance to a specific submodule within the system, as detailed below.

3.3.1 Violence Detection

For training the violence classification module, the *Violence Detection Dataset* from Kaggle was employed. This dataset comprises 2,000 annotated video clips, balanced evenly between violent (1,000) and non-violent (1,000) instances [26].

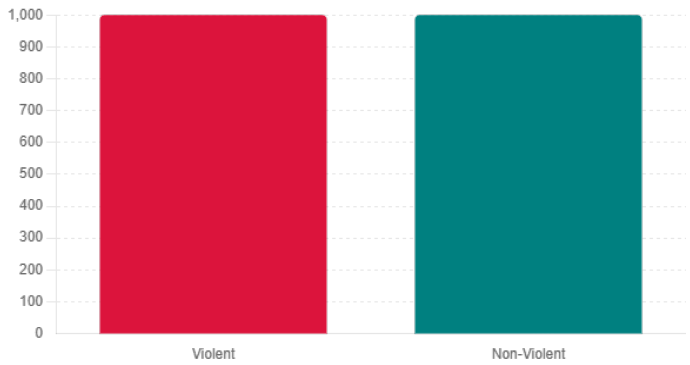


Figure 3.2: Distribution of video clips in the *Violence Detection Dataset*, with an equal number of violent and non-violent instances.

A hybrid architecture combining a Bi-directional Long Short-Term Memory (Bi-LSTM) network with MobileNetV2 was implemented to effectively model both temporal dynamics and spatial representations. The dataset's heterogeneity in scene

content and environmental conditions renders it particularly suitable for applications in surveillance and public safety.

3.3.2 Face Detection and Age Estimation

The *Face Age Detection Dataset* (Kaggle), consisting of 19,906 facial images with corresponding age labels, was used to train the demographic analysis module [27]. This module is integrated into the FaceAPI subcomponent and is responsible for real-time face detection and age classification. A lightweight sequential architecture based on MobileNetV2 was adopted to balance inference speed and classification accuracy, facilitating deployment in resource-constrained environments. Data distribution for this dataset visualized in 3.3.

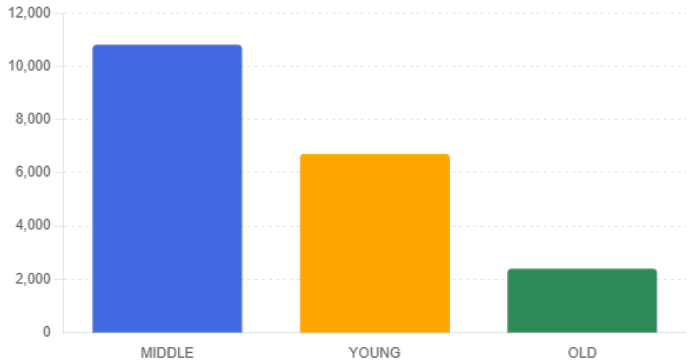


Figure 3.3: Class distribution in the *Face Age Detection Dataset*, showing the number of samples across different age categories.

3.3.3 Object Detection

The object detection subsystem was developed using the Common Objects in Context (COCO) dataset, which contains over 886,000 annotated object instances spanning 80 object categories, distributed across 82,783 training and 40,504 validation images in the 2014 release [28].

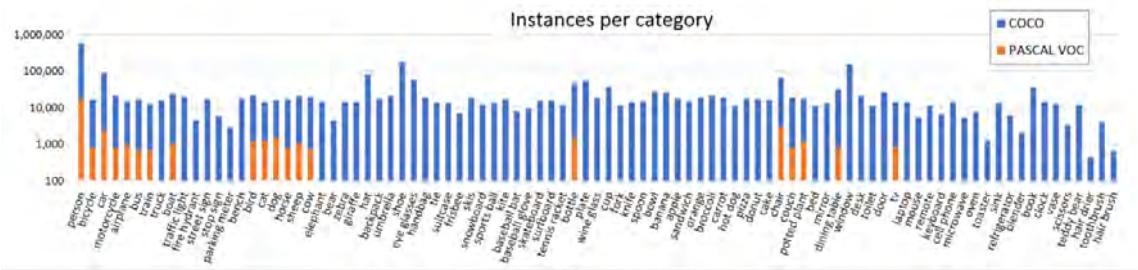


Figure 3.4: Comparison of object instance frequency per category in the COCO and PASCAL VOC datasets (log scale). COCO provides a more comprehensive and dense annotation coverage across 80 categories [28].

To support real-time multi-object detection with high accuracy, a Single Shot MultiBox Detector (SSD) framework with a ResNet-101 backbone was employed. The

wide range of object types and contextual variability in the COCO dataset ensures robustness across diverse operational settings.

3.3.4 Scene Interpretation

The proposed framework leverages GPT-4o [29] as its language reasoning engine to perform zero-shot scene interpretation based on structured visual prompts. While GPT-4o was not fine-tuned for this specific task, its extensive pretraining on large-scale textual data enables effective generalization to novel visual-linguistic tasks without additional supervision. This allows our system to infer high-level semantic insights from multi-modal inputs in an open-ended and context-aware manner.

Chapter 4

Experimental Setup and Evaluation

4.1 Experimental Setup

To evaluate the effectiveness of the proposed hybrid framework, a structured experimental pipeline was designed. This pipeline covers the stages of model development, deployment, and integration into a browser-based interface, with an emphasis on lightweight, client-side execution and extensibility.

4.1.1 Model Development Environment

All models were developed using Python 3.8 with TensorFlow 2.x and the Keras API. Following training and validation, each model was exported using TensorFlow's `SavedModel` format. The exported models were then converted to TensorFlow.js (TFJS) format using the `tensorflowjs_converter` tool, enabling deployment directly in a browser environment. This approach facilitates real-time inference without requiring server-side computation.

```
tensorflowjs_converter --input_format=tf_saved_model ./saved_model
./tfjs_model
```

4.1.2 Activity Detection Module

The activity detection component was designed to analyze spatial-temporal features from video streams. A MobileNetV2 backbone pretrained on ImageNet was used as the spatial feature extractor. The final classification layers were removed, and features were extracted from intermediate layers. These frame-level features were processed sequentially using a Bidirectional Long Short-Term Memory (Bi-LSTM) network to model temporal dependencies. The input frames were resized to 224×224 pixels and normalized before processing. After training, the complete model was converted to TFJS for real-time activity inference in the browser.

4.1.3 Face Age Detection Module

For facial age prediction, a sequential classification model was developed using MobileNetV2 as the base feature extractor. Detected face regions, extracted via Mi-

crosoft's FaceAPI, were resized to 224×224 pixels and passed through the model. The MobileNet backbone was partially frozen to retain pretrained spatial features, and a dense classification head was appended to predict age groups. The labels were discretized into predefined categories, such as *young*, *middle-aged*, and *old*. Once trained, the model was converted to TFJS to support on-device inference during real-time operation.

4.1.4 Web Application Architecture

The integrated system was deployed as a browser-based application using a React.js frontend. Tailwind CSS was used for modular styling, and the frontend provided support for both live camera input and offline video uploads. Real-time inference was achieved by loading TFJS models asynchronously into the client browser. Each incoming video frame was processed locally, and outputs such as bounding boxes, classification labels, and facial attributes were rendered instantly. The entire inference pipeline ran on the client device, offering low latency and privacy-preserving performance without dependence on external servers.

4.1.5 Prompt Template Generation

Scene-level interpretation was supported by a template-based prompt generation module. Detected entities and their attributes were dynamically embedded into natural language templates using JavaScript. The structured prompt captured spatial relationships, object interactions, and human-centric cues. For example, when a young individual is detected holding a phone, the system may generate the following prompt:

> *"The scene shows a young individual holding a phone, appearing focused. The environment is well-lit and casual."*

This mechanism serves as an efficient zero-shot reasoning interface, providing context-aware descriptions in real time.

4.1.6 System Extensibility

The proposed system was designed with modular extensibility in mind. New visual recognition modules—such as emotion detection or gesture classification—can be integrated into the pipeline with minimal configuration. All models follow standardized preprocessing routines and input specifications. TFJS compatibility ensures that additional modules can be incorporated using the same browser-based deployment strategy, allowing the system to adapt to broader vision-language applications.

4.2 Results and Discussion

This section presents the evaluation of our proposed hybrid framework, covering three core tasks: activity detection, face age prediction, and real-time scene interpretation. We report both quantitative metrics and qualitative insights obtained from the complete system pipeline and the end-user interface.

4.2.1 Activity Detection Evaluation

To identify activity, like road crossings, accidents, violent or aggressive behavior in video sequences, we implemented a hybrid architecture that combines spatial and temporal features using MobileNet V2 and Bidirectional Long Short-Term Memory (Bi-LSTM) layers. The MobileNet backbone was pretrained on ImageNet and used for spatial feature extraction, while the Bi-LSTM captured temporal patterns from frame sequences.
The model was trained using a standard activity detection dataset with a 70–20–10 split across training, validation, and test sets. The model was optimized using categorical cross-entropy loss and Stochastic Gradient Descent (SGD) as the optimizer. Table 4.1 presents the model parameters, and Table 4.3, Table 4.4 summarizes the accuracy results and the classification report respectively.

Parameter Type	**Count**
Total Parameters	2,257,984
Trainable Parameters	1,681,536
Non-trainable Parameters	576,448

Table 4.1: Model parameters for activity detection using a custom Bi-LSTM with MobileNet V2.

Layer (type)	**Output Shape**	**Param #**
`time_distributed (TimeDistributed)`	(None, 32, 2, 2, 1280)	2,257,984
`dropout (Dropout)`	(None, 32, 2, 2, 1280)	0
`time_distributed_1 (TimeDistributed)`	(None, 32, 5120)	0
`bidirectional (Bidirectional)`	(None, 64)	1,319,168
`dropout_1 (Dropout)`	(None, 64)	0
`dense (Dense)`	(None, 256)	16,640
`dropout_2 (Dropout)`	(None, 256)	0
`dense_1 (Dense)`	(None, 128)	32,896
`dropout_3 (Dropout)`	(None, 128)	0
`dense_2 (Dense)`	(None, 64)	8,256
`dropout_4 (Dropout)`	(None, 64)	0
`dense_3 (Dense)`	(None, 32)	2,080
`dropout_5 (Dropout)`	(None, 32)	0
`dense_4 (Dense)`	(None, 2)	66

Table 4.2: Model architecture summary for the activity detection module, integrating MobileNet-based feature extraction with a Bi-LSTM sequence classifier and fully connected layers.

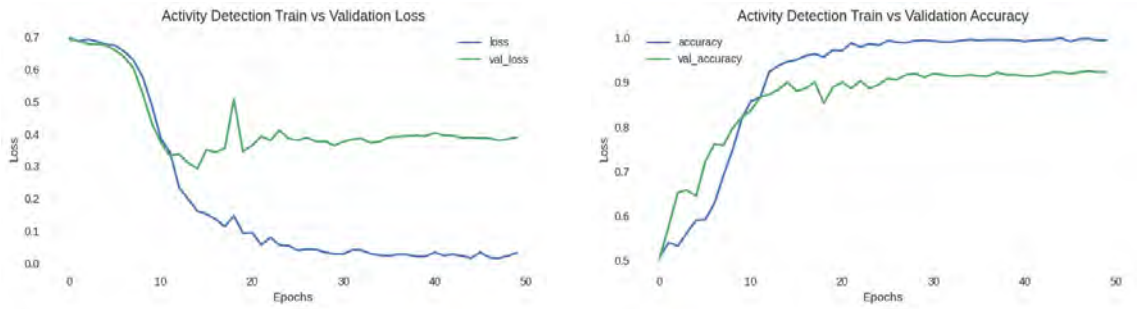
Dataset Split	Accuracy (%)
Training	99.02
Validation	90.83
Test	90.54

Table 4.3: Classification accuracy for activity detection across training, validation, and test sets.

Class	Precision	Recall	F1-Score	Support
Normal	0.94	0.91	0.93	102
Abnormal	0.91	0.94	0.92	98
Accuracy	-	-	0.93	200
Macro Avg	0.93	0.93	0.92	200
Weighted Avg	0.93	0.93	0.93	200

Table 4.4: Classification report for activity detection

The model achieved early convergence, with training and validation losses stabilizing around the 15th epoch. The high training accuracy and close alignment with validation and test scores indicate effective generalization and low overfitting. The system performed well across various scene complexities and motion dynamics, demonstrating its reliability for real-time activity analysis as indicated in Figure 4.1.



(a) Training vs Validation Loss (b) Training vs Validation Accuracy

Figure 4.1: Performance of the Activity Detection model over 50 epochs: (a) loss and (b) accuracy.

4.2.2 Face Age Prediction Evaluation

For face age classification, we utilized a sequential classification model incorporating MobileNet V2 as the backbone for feature extraction. The FaceAPI module was used to crop and preprocess detected face regions. Input face images were resized to 224×224 and normalized before being passed through the model. The final dense layers were trained to classify each face into one of three predefined age categories: *Young*, *Middle*, and *Old*.

The model followed a 70–20–10 data split and was trained using categorical cross-entropy with the Adam optimizer. The summary of model parameters is shown in Table 4.5 and detailed layer information are reported in Table 4.6, and the final accuracy metrics across training, validation, and test sets are reported in Table 4.7.

Parameter Type	Count
Total Parameters	21,429,035
Trainable Parameters	21,369,643
Non-trainable Parameters	59,392

Table 4.5: Model parameters for face age prediction using a fully connected hybrid sequential MobileNet V2 architecture.

Layer (type)	Output Shape	Param #
`xception (Functional)`	(None, 8, 8, 2048)	20,861,480
`global_average_pooling2d (GlobalAveragePooling2D)`	(None, 2048)	0
`batch_normalization_4 (BatchNormalization)`	(None, 2048)	8,192
`dense (Dense)`	(None, 256)	524,544
`batch_normalization_5 (BatchNormalization)`	(None, 256)	1,024
`dropout (Dropout)`	(None, 256)	0
`dense_1 (Dense)`	(None, 128)	32,896
`batch_normalization_6 (BatchNormalization)`	(None, 128)	512
`dropout_1 (Dropout)`	(None, 128)	0
`dense_2 (Dense)`	(None, 3)	387

Table 4.6: Model architecture summary for the face age prediction module using Xception and fully connected layers.

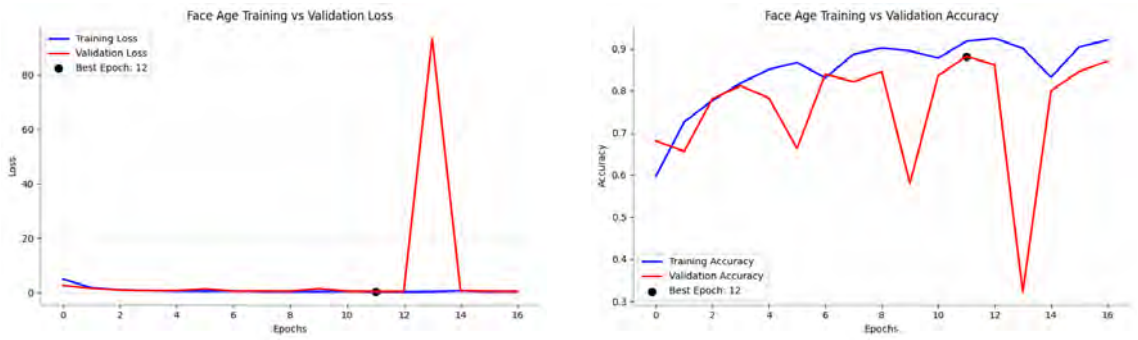
Dataset Split	Accuracy (%)
Training	92.43
Validation	86.14
Test	87.04

Table 4.7: Final classification accuracy for face age prediction across training, validation, and test sets.

Class	Precision	Recall	F1-Score	Support
Middle	0.87	0.90	0.89	1075
Young	0.73	0.80	0.76	225
Old	0.90	0.82	0.86	691
Accuracy	-	-	0.86	1991
Macro Avg	0.83	0.84	0.84	1991
Weighted Avg	0.87	0.86	0.86	1991

Table 4.8: Classification report for face age prediction

The model demonstrated strong generalization performance and consistent accuracy across a diverse range of age groups and environmental variations.



(a) Training vs Validation Loss (b) Training vs Validation Accuracy

Figure 4.2: Performance of face age prediction model: (a) Loss across training and validation, and (b) Accuracy across training and validation epochs.

As shown in the Figure 4.2, the training curves show consistent improvement in both accuracy and loss throughout the 16 epochs. Although training performance improves steadily, validation metrics show fluctuations due to variability in facial expressions, poses, and lighting, with the generalization observed around epoch 12.

4.2.3 Qualitative Evaluation

Although quantitative metrics provide insight into overall model performance, qualitative evaluation enables visual inspection of model behavior in various real-world scenarios. This includes reviewing the accuracy and confidence of predictions in unseen data, especially under challenging conditions such as varying poses, occlusions, or lighting.
To evaluate the practical integration of the proposed modules, a real-time user interface was developed. The system supports both offline video upload and live camera streaming, enabling end-to-end testing of detection, prediction, and captioning capabilities within a web-based environment (Figure 4.3, 4.4).

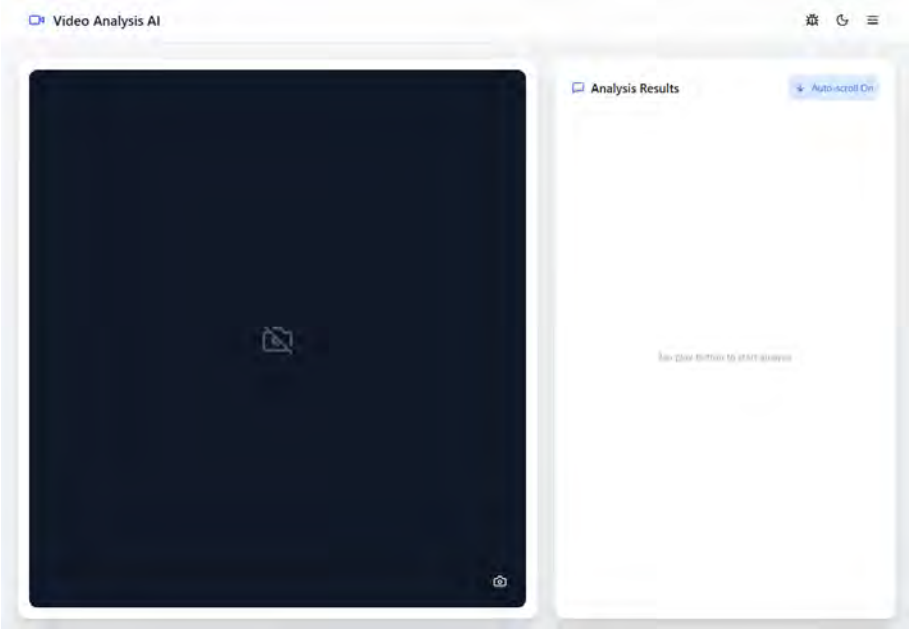


Figure 4.3: Live interface for scene descriptions.

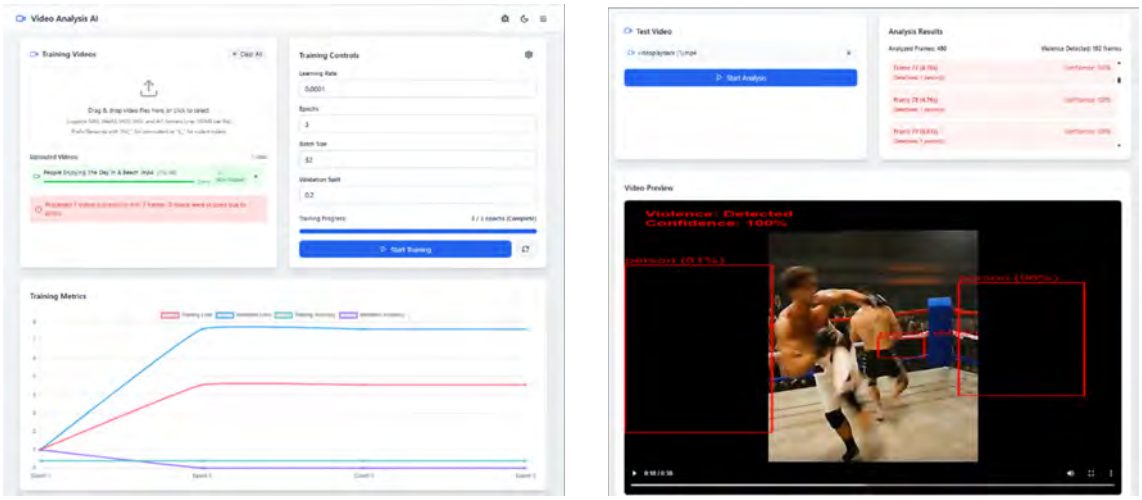


Figure 4.4: Live interface for scene descriptions.

The user interface provides the following core functionalities:

- A training module with customizable parameters, including learning rate, number of epochs, and data partitioning.
- Real-time frame-by-frame analysis of video input using pre-trained detection models.
- Display of key results such as detected activity, location of facial regions, and classification of age groups.
- Dynamic Scene Caption Generation Based on Detected Objects and Face Attribute.

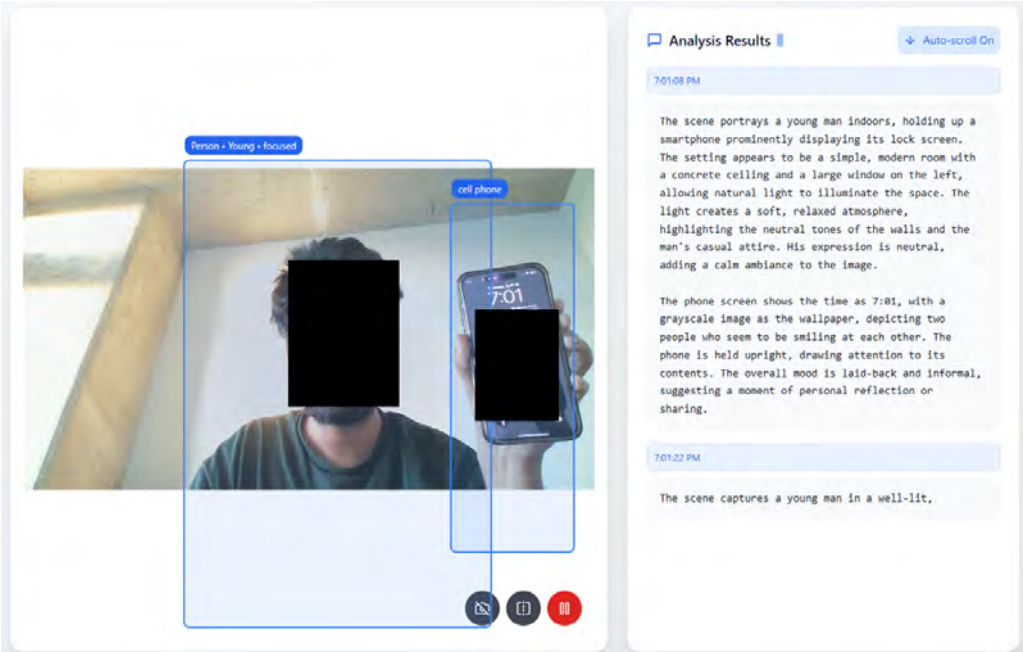


Figure 4.5: Live interface with bounding box visualizations and dynamically generated scene descriptions.

As shown in Figure 4.5, the system successfully processes real-time camera input and annotates the scene accordingly. In one instance, the model detects a young male subject holding a smartphone indoors. The captioning engine, powered by a large language model, generates a coherent narrative describing the scene's environment, lighting, objects, and mood. This result demonstrates the system's ability to not only recognize discrete features (e.g., age, object presence) but also synthesize them into human-readable interpretations. These qualitative outputs offer valuable insight into the contextual understanding enabled by the system. Our framework maintained smooth performance and responsive feedback during live testing with minial hardwares. The activity detection module consistently localized relevant events, while the captioning module exhibited robust generalization in generating context-aware summaries from unseen scenes.

4.2.4 Qualitative Comparison

To evaluate the effectiveness of the proposed scene captioning framework, we conducted a qualitative comparison against Gemini 2.5 Pro [30], Google's state-of-the-art multimodal large language model. Both systems were presented with the same visual input extracted from a live camera frame (shown in Figure 4.5), and prompted identically with:

> *"Consider this as a scene and describe what you see."*

The input was a real-time frame captured from the user interface. Below are the scene descriptions generated by each system:

System	Scene Description
Gemini 2.5 Pro	I see a person with dark hair who appears to be sitting. They are holding a smartphone in their hand, and the screen of the phone is visible. The person is wearing a green shirt. The background is somewhat plain, with a white wall and some visible concrete structure. (Figure 4.6)
Proposed Framework	The scene portrays a young man indoors, holding up a smartphone prominently displaying its lock screen. The setting appears to be a simple, modern room with a concrete ceiling and a large window on the left, allowing natural light to illuminate the space. The light creates a soft, relaxed atmosphere, highlighting the neutral tones of the walls and the man's casual attire. His expression is neutral, adding a calm ambiance to the image. The phone screen shows the time as 7:01, with a grayscale image as the wallpaper, depicting two people who seem to be smiling at each other. The phone is held upright, drawing attention to its contents. The overall mood is laid-back and informal, suggesting a moment of personal reflection or sharing.

Table 4.9: Qualitative comparison of scene descriptions between Gemini 2.5 Pro and the proposed framework.

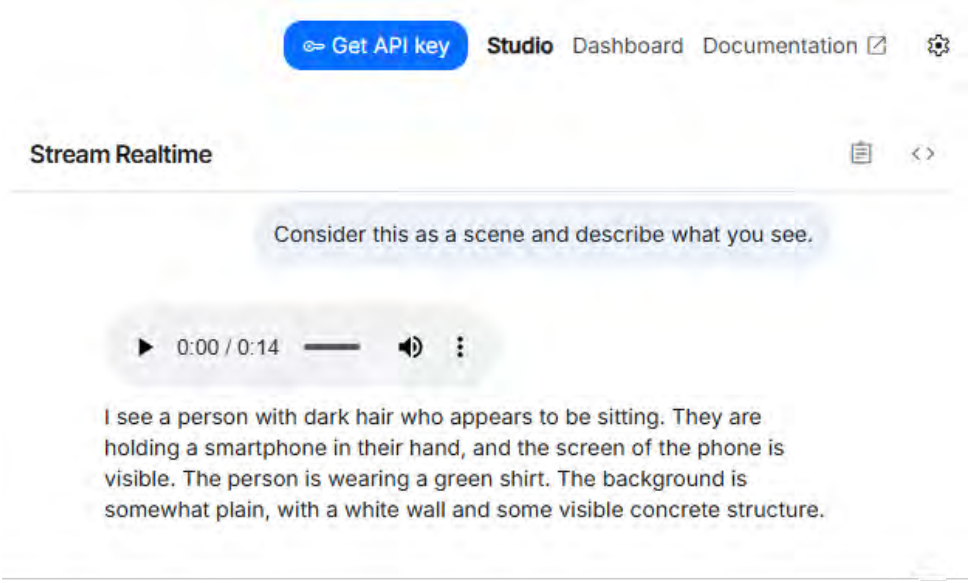


Figure 4.6: Gemini 2.5 Pro's response to the scene captioning prompt.

The comparison in Table 4.9 highlights several key strengths of our proposed framework:

Evaluation Criterion	**Proposed Framework**	**Gemini 2.5 Pro**
Attribute-level understanding (e.g., age, expression)	✓	×
Scene context and environment description	✓	×
Object-level semantic analysis (e.g., lock screen content)	✓	×
Personal interaction/mood inference	✓	×
Prompt augmentation using detection modules	✓	×
Natural language fluency	✓	✓
Speed of generation	✓	✓

Table 4.10: Qualitative feature comparison.

Moreover, our proposed system produces contextually aware, semantically grounded, and human-like scene descriptions, outperforming Gemini 2.5 Pro in both detail and interpretability. This evaluation demonstrates that our modular, detection-enhanced approach results in more expressive and meaningful scene captioning, particularly in real-time and unconstrained visual conditions.

4.3 Benchmarking

This section specifies how we evaluate the two supervised components; activity detection and face–age classification and how we report efficiency, uncertainty, and reproducibility. We follow standard metrics and evaluation protocols to enable fair comparison and replication.

4.3.1 Tasks, Datasets, and Splits

Activity detection. We use the Kaggle Real-Life Violence Situations dataset (2,000 short clips; 1,000 violent and 1,000 non-violent). The model is MobileNetV2 feature extraction with a Bi-LSTM sequence head. We adopt a 70, 20, 10 to train, validation, test split, enforcing scene disjointness to reduce leakage.

Face age classification (3 classes). We evaluate on the Kaggle Face Age dataset (19,906 images), discretized into *Young*, *Middle*, and *Old* after FaceAPI cropping. We use a 70/20/10 split with subject de-duplication where possible. Our primary model uses an Xception backbone with a light classification head (TFJS-targeted). On the held-out test set, we obtain Accuracy 85.89%, Macro-F1 0.81, and Weighted-F1 0.85; the 95% confidence interval for accuracy is [84.4%, 87.4%].

4.3.2 Metrics

Classification metrics. We report Accuracy, Precision, Recall, and F1 per class, along with Macro-F1 (unweighted mean of class F1).

Uncertainty. For top-line accuracy we report 95% confidence intervals using the normal approximation:

$$\mathrm{CI}_{95\%} \;\approx\; \hat{p} \;\pm\; 1.96\sqrt{\frac{\hat{p}\,(1-\hat{p})}{n}},$$

4.3.3 Baselines

To contextualize the results, we compare against lightweight baselines trained under the same preprocessing, resolution, optimizer schedule, and early-stopping on the shared validation split:

- MobileNetV2 + Linear (Activity detection).
- ResNet-18 + GRU (Activity detection).
- MobileNetV2 (Face age detection).
- EfficientNet-B0 (Face age detection).

4.3.4 Results

Activity detection On the test set (N=200), Accuracy is **90.54%** and Macro-F1 is **0.92** (Table 4.11). The 95% CI for accuracy is **[86.4%, 94.6%]**.

$$\hat{p} = 0.9054, \quad \mathrm{SE} = \sqrt{\tfrac{0.9054\cdot 0.0946}{200}}, \quad \mathrm{CI}_{95\%} \approx 0.9054 \pm 1.96 \cdot \mathrm{SE}.$$

Face age classification. On the test set (N=1,991), Accuracy is **87.04%**, Macro-F1 is **0.84**, and Weighted-F1 is **0.86** (Table 4.12). The 95% CI for accuracy is [**85.5%, 88.5%**].

$$\hat{p} = 0.8589, \quad \text{SE} = \sqrt{\tfrac{0.8589 \cdot 0.1411}{1991}}, \quad \text{CI}_{95\%} \approx 0.8589 \pm 1.96 \cdot \text{SE}.$$

Model	**Acc (%)**	**Acc 95% CI**	**Macro-F1**	**Weighted-F1**
MobileNetV2 + BiLSTM (ours)	**90.54**	[86.4, 94.6]	**0.920**	**0.922**
MobileNetV2 + Linear Regression	83.00	[77.8, 88.2]	0.830	0.841
ResNet-18 + GRU	77.50	[71.7, 83.3]	0.769	0.770

Table 4.11: Activity detection benchmarking

Model	**Acc (%)**	**Acc 95% CI**	**Macro-F1**	**Weighted-F1**
MobileNetV2 + Xception (ours)	**87.04**	[85.5, 88.5]	**0.84**	**0.86**
MobileNetV2 (baseline)	56.77	[53.6, 59.9]	0.32	0.47
EfficientNet-B0 (baseline)	54.69	[51.5, 57.8]	0.24	0.39

Table 4.12: Face age classification benchmarking

Chapter 5

Conclusion

This research presents a lightweight yet robust framework for real-time scene understanding by leveraging zero-shot learning principles and prompt-engineered vision-language integration. The proposed system addresses the need for efficient, on-device visual interpretation by combining modular computer vision pipelines with structured language generation strategies. It effectively bridges the gap between visual perception and natural language expression using task-specific detection models and a browser-executable language prompt engine.

Through the integration of activity recognition, face age classification, and object detection subsystems—each converted to run efficiently via TensorFlow.js in the browser—the system successfully interprets dynamic scenes without relying on backend inference or cloud-based models. The prompt-based captioning engine enables coherent narrative generation by embedding detected entities (e.g., age, objects, spatial configuration) into semantically rich template structures.

Experimental results demonstrate high quantitative performance, particularly in activity classification (90.54% test accuracy) and age group prediction (87.04% test accuracy), with accompanying qualitative evaluations showing context-aware and human-like scene descriptions. In comparison to Gemini 2.5 Pro, a leading multimodal language model, our framework generated descriptions that were more specific, attribute-rich, and tailored to the immediate visual content. This validates the strength of prompt-engineered visual grounding, even in lightweight setups.

5.1 Limitations

Despite its effectiveness, several architectural and methodological limitations remain. First, the use of static prompt templates, though efficient, restricts expressive variation and adaptability in language generation. The framework lacks a feedback loop between the language model and vision modules, potentially limiting semantic coherence in edge cases. Additionally, the detection models—though performant—operate in isolation, with no shared feature space or joint optimization, which may introduce inconsistency across modalities.

Another constraint arises from the limited temporal modeling in the activity detection module, which currently operates on short, fixed-length frame windows. This may hinder the understanding of longer, complex actions. Furthermore, although the system performs well under controlled conditions, its generalization under diverse lighting, occlusion, and background scenarios remains a challenge due to dataset

bias and limited environmental variability in training data.

5.2 Future Work

To address these limitations and expand the system's capabilities, several future directions are proposed. One major avenue involves replacing static prompt logic with dynamic, transformer-based language generation models capable of real-time inference on edge devices. These would allow for greater linguistic diversity and adaptive scene interpretation beyond template constraints.

- **Unified Visual-Linguistic Reasoning:** We plan to implement a shared embedding space or a cross-attention mechanism that jointly encodes visual and linguistic features, enabling more coherent and unified scene understanding. This architecture may also facilitate end-to-end training or fine-tuning through weak supervision or human-in-the-loop feedback.

- **Enhanced Temporal Modeling:** To improve action understanding, we intend to incorporate long-range temporal modeling using transformer-based video encoders. This addition will capture temporal dependencies across frames and improve the granularity of activity recognition. Furthermore, model interpretability will be enhanced by aligning language outputs with relevant visual regions.

- **Benchmark Evaluation and Expansion:** Future versions of the system will be evaluated using standard multimodal and video-language benchmarks such as MSR-VTT [31], ActivityNet Captions [32], and Charades [33]. This will help establish the system's generalizability and robustness in diverse real-world scenarios. We also aim to explore integration with larger vision-language models via prompt chaining or retrieval-augmented generation (RAG) to support more complex scene understanding tasks.

To conclude, this research study demonstrates the feasibility of real-time, prompt-driven scene interpretation using modular vision-language pipelines. The framework opens up promising pathways for accessible and semantically expressive multimedia applications in domains such as surveillance, education, human-computer interaction, and embedded AI.

Bibliography

[1] M. A. Souibgui, A. Fornés, Y. Kessentini, and B. Megyesi, "Few shots are all you need: A progressive learning approach for low resource handwritten text recognition," *Pattern Recognition Letters*, vol. 160, pp. 43–49, 2022, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2022.06.003 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016786552200191X

[2] A. Radford et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PmLR, 2021, pp. 8748–8763.

[3] J. Lee et al., "Uniclip: Unified framework for contrastive language-image pre-training," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1008–1019, 2022.

[4] H. Al-Tahan, Q. Garrido, R. Balestriero, D. Bouchacourt, C. Hazirbas, and M. Ibrahim, "Unibench: Visual reasoning requires rethinking vision-language beyond scaling," *Advances in Neural Information Processing Systems*, vol. 37, pp. 82 411–82 437, 2024.

[5] J. White et al., "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.

[6] Y. Liu, S. Tao, W. Meng, F. Yao, X. Zhao, and H. Yang, "Logprompt: Prompt engineering towards zero-shot and interpretable log analysis," in *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, 2024, pp. 364–365.

[7] J. Yu, S. Qian, and C. Chen, "Lightweight crack automatic detection algorithm based on tf-mobilenet," *Applied Sciences*, vol. 14, no. 19, p. 9004, 2024.

[8] Q. Abbas, M. E. Ibrahim, and M. A. Jaffar, "Video scene analysis: An overview and challenges on deep learning algorithms," *Multimedia Tools and Applications*, vol. 77, no. 16, pp. 20 415–20 453, 2018.

[9] J. Lu et al., "Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 439–26 455.

[10] X. Chen et al., "Pali-3 vision language models: Smaller, faster, stronger," *arXiv preprint arXiv:2310.09199*, 2023.

[11] W. Dai et al., "Instructblip: Towards general-purpose vision-language models with instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 49 250–49 267, 2023.

[12] K. Clark and P. Jaini, "Text-to-image diffusion models are zero shot classifiers," *Advances in Neural Information Processing Systems*, vol. 36, pp. 58 921–58 937, 2023.

[13] J. Rocamonde, V. Montesinos, E. Nava, E. Perez, and D. Lindner, "Vision-language models are zero-shot reward models for reinforcement learning," *arXiv preprint arXiv:2310.12921*, 2023.

[14] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[15] Y. Zhou et al., "Large language models are human-level prompt engineers," in *The eleventh international conference on learning representations*, 2022.

[16] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 816–16 825.

[17] S. H. Ahmed, S. Hu, and G. Sukthankar, "The potential of vision-language models for content moderation of children's videos," in *2023 International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2023, pp. 1237–1241.

[18] K. Guo et al., "Moderating illicit online image promotion for unsafe user generated content games using large {vision-language} models," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 5787–5804.

[19] J. A. A. Gonzalez, T. Matsukawa, and E. Suzuki, "Leveraging vision language models for understanding and detecting violence in videos," in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Science and Technology Publications, Lda, vol. 2, 2025, pp. 99–113.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[21] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "Mobilenetv2 model for image classification," in *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, IEEE, 2020, pp. 476–480.

[22] W. Liu et al., "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[23] S. Jain, S. Dash, R. Deorari, et al., "Object detection using coco dataset," in *2022 International Conference on Cyber Resilience (ICCR)*, IEEE, 2022, pp. 1–4.

[24] V. Grabovskyi and O. Martynovych, "Facial recognition with using of the microsoft face api service," *Electron Inf Technol*, vol. 12, pp. 36–48, 2019.

[25] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., "Improving language understanding by generative pre-training," 2018.

[26] A. Bhonde, *Real life violence situations dataset*, 2025. DOI: 10.21227/94hb-6871 [Online]. Available: https://dx.doi.org/10.21227/94hb-6871

[27] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.

[28] T.-Y. Lin et al., *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1405.0312

[29] OpenAI. "Hello GPT–4o." Accessed: 2025-08-07. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[30] G. Comanici et al., "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[31] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5288–5296.

[32] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, "Dense-captioning events in videos," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 706–715.

[33] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, "Charades-ego: A large-scale dataset of paired third and first person videos," *arXiv preprint arXiv:1804.09626*, 2018.