CMPE 491
Senior Project 1

Coffee Machine Assistance for the Visually Impaired:
A Computer Vision-Based Mobile Application

Project Supervisor
Dr. Hakki Gökhan Ilk

Project Team Members
Murad Huseynov
Bedir Esen
Hazem Mustafa
Alperen Dalgıç

Course Coordinator
Dr. Gökçe Nur Yilmaz

Table of Contents

1. Introduction

The daily routine of brewing coffee, a simple pleasure for many, presents a significant obstacle for individuals with visual impairments. Modern coffee machines, with their visually dependent interfaces and controls, create barriers to independent operation. This thesis addresses the need for assistive technologies that can bridge this accessibility gap, focusing on empowering visually impaired individuals to confidently use everyday appliances like coffee machines and enhance their daily independence and quality of life.

Existing assistive technologies for appliance operation often rely on generic solutions or costly modifications. This thesis explores a more innovative and accessible approach: utilizing computer vision (CV) and smartphone technology. We focus on developing a mobile application that uses the smartphone camera to "see" and interpret coffee machine interfaces. This project aims to create a specific, real-time guidance system for a chosen coffee machine model, providing precise voice instructions for its operation. The scope of this research is the design, development, and evaluation of a CV-based mobile application to assist visually impaired individuals in independently operating a defined coffee machine model through voice guidance.

Assistive technology research has explored various methods for appliance accessibility, from tactile aids to auditory feedback and voice control. However, many solutions are limited in adaptability, integration ease, or real-world usability. While voice and gesture interfaces offer potential, challenges in robustness and accuracy remain. This thesis builds upon this foundation by investigating CV, a rapidly advancing field – to address coffee machine operation specifically. By using visual understanding and real-time guidance, this project aims to contribute a novel, potentially more universally applicable, and context-aware approach to appliance accessibility.

This dissertation investigates the feasibility and effectiveness of a CV-based mobile application for coffee machine assistance, guided by the following questions:

1. Can a mobile CV application accurately recognize a coffee machine model in real-time?
2. Can voice instructions effectively guide visually impaired users to make coffee on the recognized machine?
3. Is the application usable, accessible, and helpful for independent coffee machine operation by visually impaired individuals?
4. What are the key design challenges and effective mitigation strategies for such an assistive application?

To answer these questions, this project aims to achieve the following objectives:

1. Design and implement an EfficientDet-based CV system for coffee machine recognition.
2. Develop a voice guidance system using Android TTS and iOS VoiceOver for clear coffee machine operation instructions.
3. Integrate these systems into an accessible mobile application.
4. Thoroughly test and evaluate the application's performance and usability with visually impaired users.

2. Proposed System

2.1 Overview

This project proposes a mobile application designed to assist visually impaired individuals in operating specific models of coffee machines independently. Leveraging computer vision (CV) and smartphone cameras, the application recognizes the coffee machine and its components (buttons, indicators, etc.) in real-time. It then provides sequential, context-aware voice instructions via Text-to-Speech (TTS) to guide the user through the coffee-making process. The system aims to bridge the accessibility gap presented by modern appliances with complex visual interfaces, thereby enhancing user independence. The core technologies include the EfficientDet object detection model for CV, React Native for cross-platform mobile development, and native platform TTS engines (Android TTS, iOS VoiceOver) for voice guidance. The system is designed for offline functionality and prioritizes accessibility and usability through user-centered design principles.

2.2 Functional Requirements

1. **Coffee Machine Recognition:** The application must accurately detect and identify the target coffee machine model and its key components (e.g., power button, water reservoir, coffee dispenser, control panel, specific option buttons, status indicators) from the phone's camera.
2. **Image Capture:** The application must capture images from the device camera at appropriate intervals for CV processing. This includes handling camera permissions and providing feedback for optimal positioning.
3. **Voice Guidance Generation:** Based on the recognized components and the current state of the coffee-making process, the application must generate clear, concise, and sequential voice instructions.
4. **Text-to-Speech Output:** The application must convert the generated instructions into audible speech using the device's native TTS engine (Android TTS or iOS VoiceOver).
5. **State Management:** The application must track the progress of the coffee-making task (e.g., machine off, needs water, brewing, ready) based on CV detections and elapsed time to provide contextually relevant guidance.
6. **User Interaction:** The application must allow minimal, accessible user interaction, primarily through voice output and potentially simple screen reader-compatible gestures or voice commands (if implemented) to start, pause, or potentially adjust settings.
7. **Cross-Platform Compatibility:** The core functionality must operate on both Android (7.0+) and iOS (13+) platforms.
8. **Offline Operation:** The core CV processing and voice guidance must function without requiring an active internet connection after initial installation and model download.

2.3 Nonfunctional Requirements

1. **Performance:**
   Real-time Recognition: The CV model inference time should be low enough (target < 200ms on mid-range devices) to provide responsive guidance.
   TTS Latency: The delay between state detection and corresponding voice output should be minimized.
   Resource Efficiency: The application should manage CPU, GPU, memory, and battery usage efficiently, especially during camera operation and CV processing, utilizing interval-based capture rather than continuous video processing.
2. **Accuracy:**
   Detection Accuracy: The CV model must achieve high precision and recall (target mAP > TBD threshold) for detecting coffee machine components under various lighting conditions, angles, and distances typical of user interaction. Minimize false positives/negatives.
   Guidance Accuracy: Voice instructions must accurately reflect the detected state and the required user action.
3. **Accessibility:**
   Screen Reader Compatibility: The UI must be fully compatible with native screen readers (TalkBack on Android, VoiceOver on iOS), including proper labeling, focus order, and announcements for dynamic changes.
   WCAG Compliance: Adhere to WCAG 2.1 AA guidelines where applicable to mobile applications.
   Auditory Clarity: TTS output must be clear and intelligible. Speech rate should be configurable (target default 175-200 wpm).
   Interaction Design: UI elements (if any require direct interaction) must have large touch targets (min 48x48dp) and sufficient contrast (min 4.5:1). Navigation should be simple and linear.
4. **Usability:**
   User-Centered Design: The application flow and instructions must be intuitive and easy to follow for visually impaired users, validated through user testing.
   Learnability: Users should be able to understand how to use the application quickly.
   Robustness: The application should handle errors gracefully (e.g., machine not detected, incorrect user action) and provide helpful recovery instructions.
5. **Maintainability:**
   The code should follow a modular design (Input, CV, Voice Guidance, UI) to facilitate updates, bug fixes, and potential future expansion (e.g., adding new coffee machine models).
6. **Compatibility:**
   The application must function correctly on the specified target OS versions (Android 7.0+, iOS 13+) and representative device hardware.
7. **Privacy:**
   All image processing and user data handling should occur locally on the device to protect user privacy. No image data should be transmitted externally for core functionality.

2.4 Pseudo Requirements

1. **Intuitive Guidance Language:** Voice instructions should use simple, unambiguous language, potentially incorporating spatial cues (e.g., "button at 2 o'clock") and confirmation feedback ("You should hear a beep").
2. **User Customization:** Provide options for users to adjust settings like TTS speech rate, volume, and potentially the level of instruction detail. Low-vision users might benefit from adjustable font sizes and contrast themes.
3. **Haptic Feedback:** Strategically use haptic feedback to complement audio cues (e.g., confirm camera positioning, acknowledge successful detection).
4. **Scalability (Future):** While the initial focus is on one machine model, the architecture should ideally allow for easier integration of additional models in the future (e.g., by swapping or extending the CV model and guidance logic).
5. **Aesthetics (Minimal):** While primarily auditory, any visual elements should follow accessibility guidelines (high contrast, simple layout) and not distract from the core function.

2.5 System Models

2.5.1 Scenarios

Scenario 1: First-time use - Making Coffee

Preconditions: The user has installed the application. The target coffee machine is present. The user has visual impairment.

Steps:
User launches the application.
Application requests camera permission (if not already granted) with an explanatory audio prompt. User grants permission.
The application provides initial audio instructions: "Point your phone camera towards the coffee machine." Includes audio/haptic cues for optimal distance/angle.
User points the phone at the coffee machine.
The application captures images, and the CV module processes them.
Application detects the known coffee machine model. Audio confirmation: "Coffee machine detected. Ready to start. First, locate the power button."
Application guides the user sequentially using voice instructions based on CV detection: "The power button is a round button at the top right, near 3 o'clock. Press the power button."
User presses the button. The app detects the change (e.g., indicator light turns on).
App confirms: "Power on detected. Now, check the water reservoir..." (continues guidance for filling water, adding coffee, selecting brew option).
User follows prompts until coffee is brewed. The app provides final confirmation: "Brewing complete. Your coffee is ready."
User closes the app or lets it time out.

Postconditions:
Coffee is successfully brewed
Application returns to standby mode
Usage data saved locally

Scenario 2: Handling Uncertainty/Error

Preconditions: The user is part way through making coffee using the app. The phone is moving, losing sight of the machine.

Steps:
The application provides guidance (e.g., "Press the 'Strong Brew' button").
Users accidentally move the phone significantly.
CV module fails to detect expected components in subsequent image captures.
Application pauses guidance and provides feedback: "Lost sight of the machine. Please reposition your phone, aiming towards the control panel." Audio/haptic cues assist repositioning.
User repositions the phone.
Application re-acquires detection of the machine components.
Application provides reorientation and resumes guidance: "Machine back in view. You were about to press the 'Strong Brew' button."

Postconditions:
Error corrected without aborting process
User successfully completes brewing
Error logged for system improvement

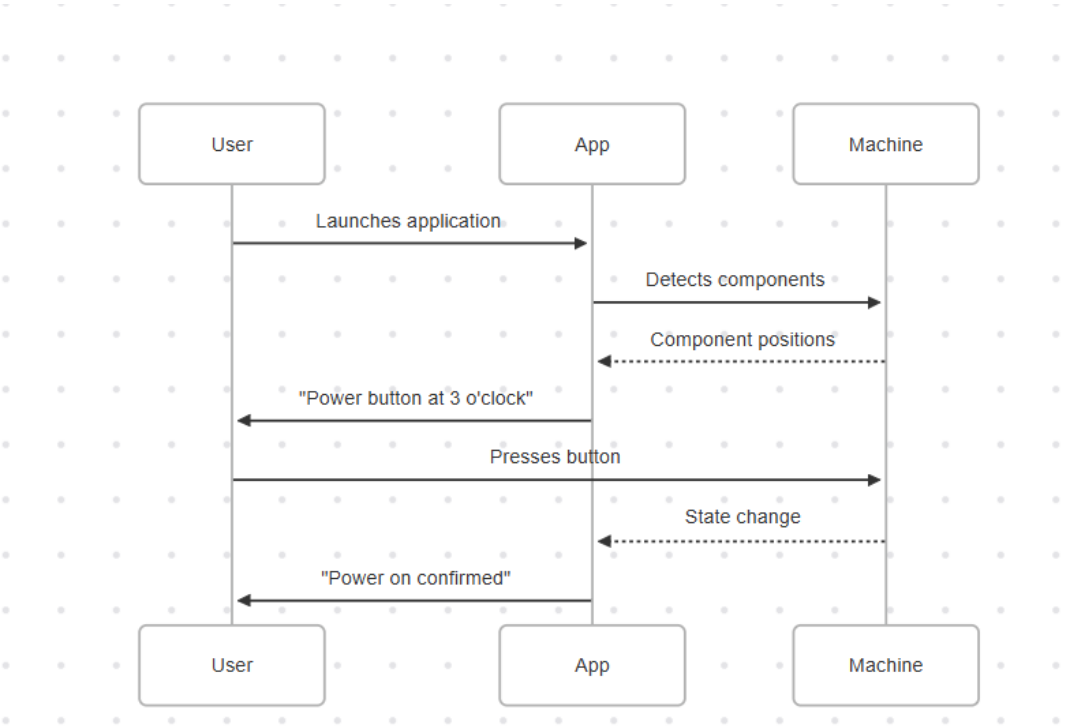
Scenario 3: Adjusting TTS Settings

Preconditions: The user finds the default speech rate too fast.

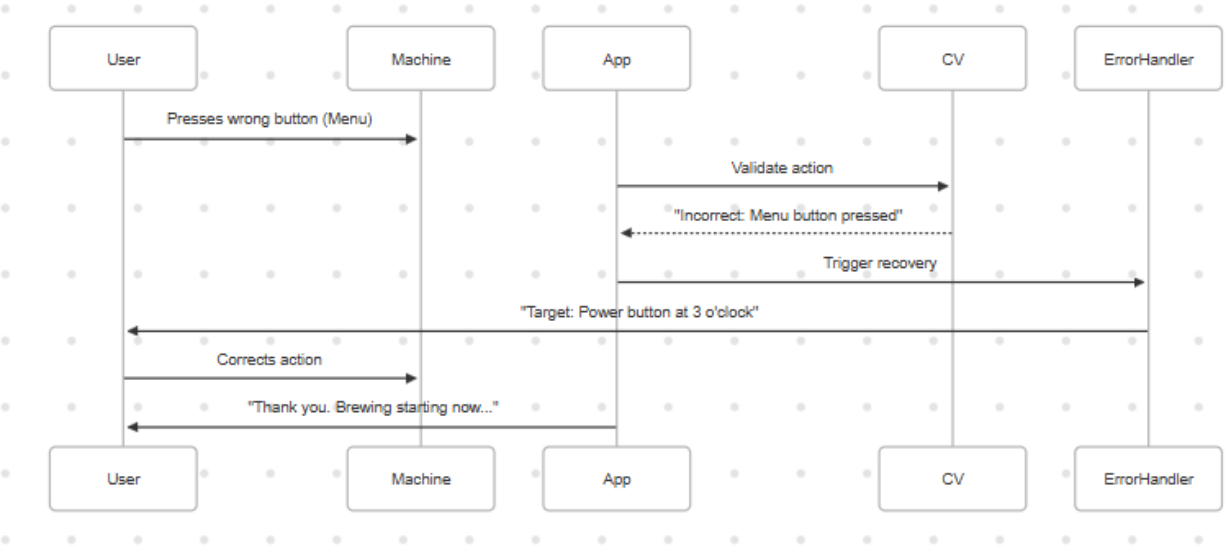
Steps:
User navigates to a settings menu within the app (assuming an accessible method is provided, e.g., specific gesture or voice command recognized by screen reader).
User accesses the "Voice Guidance Settings".
User interacts with a control labeled "Speech Rate".
Users decrease the speech rate setting using screen reader navigation.
User saves or exits the settings menu.
Subsequent voice guidance from the application uses the newly adjusted speech rate.

Postconditions:
The application's Text-to-Speech configuration (specifically the speech rate) has been updated and stored.
All subsequent voice guidance delivered by the application utilizes the new speech rate setting.
The user has successfully navigated back from the settings menu to the main application interface or closed the application.

2.5.2 Use Case Model



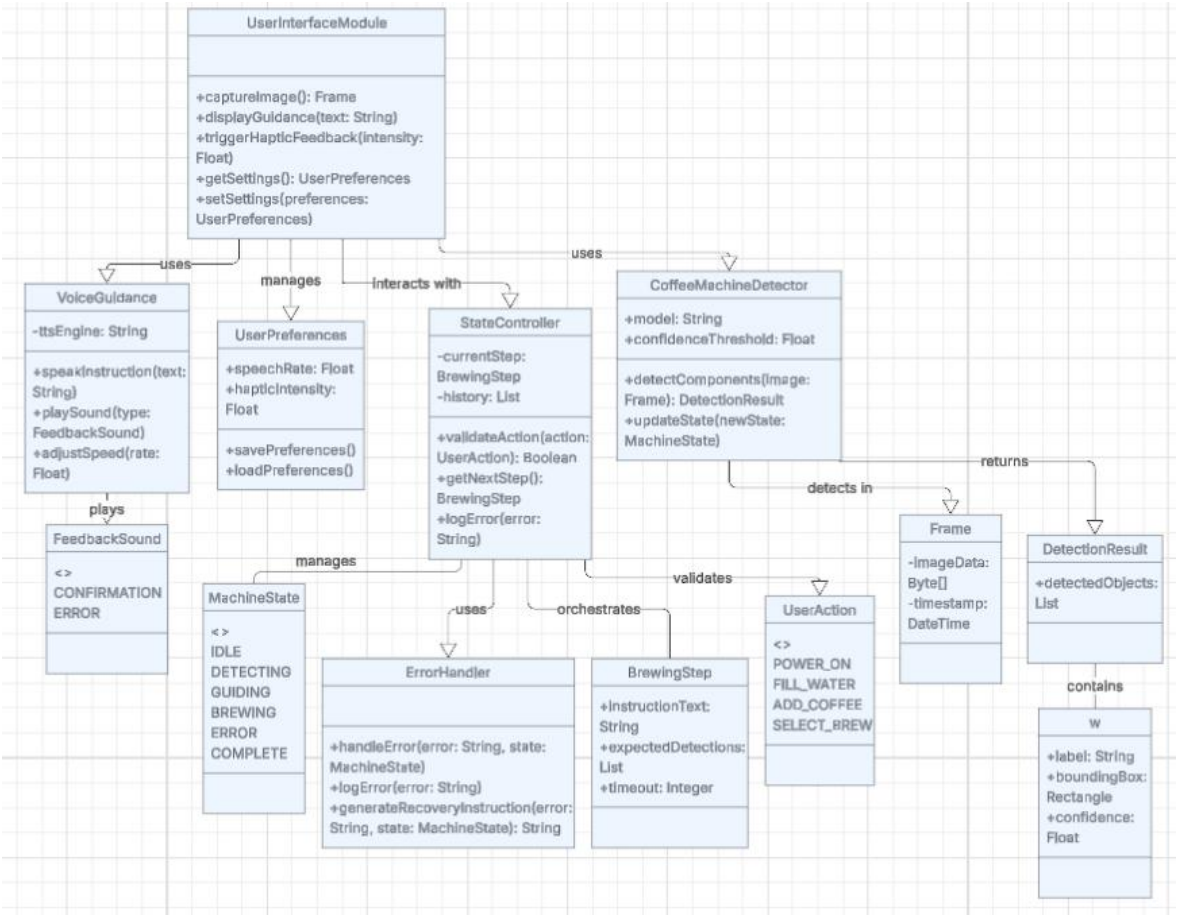
Demonstrates ideal system flow with high-confidence detections and clear user compliance. Tests core CV accuracy and voice instruction effectiveness



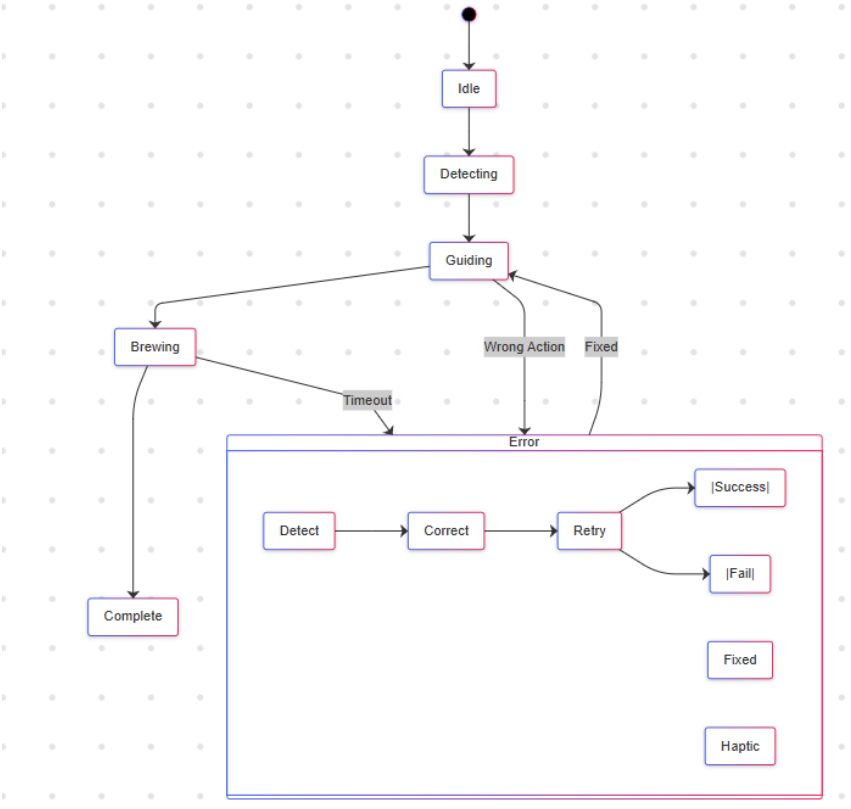
Tests the system's ability to detect and correct user errors through real-time feedback, validating fault tolerance when users press incorrect buttons. Measures both recovery speed (time-to-correction) and instruction clarity during mistakes.

Key Difference: While the first scenario checks ideal-path performance, this one evaluates how gracefully the system handles real-world human errors

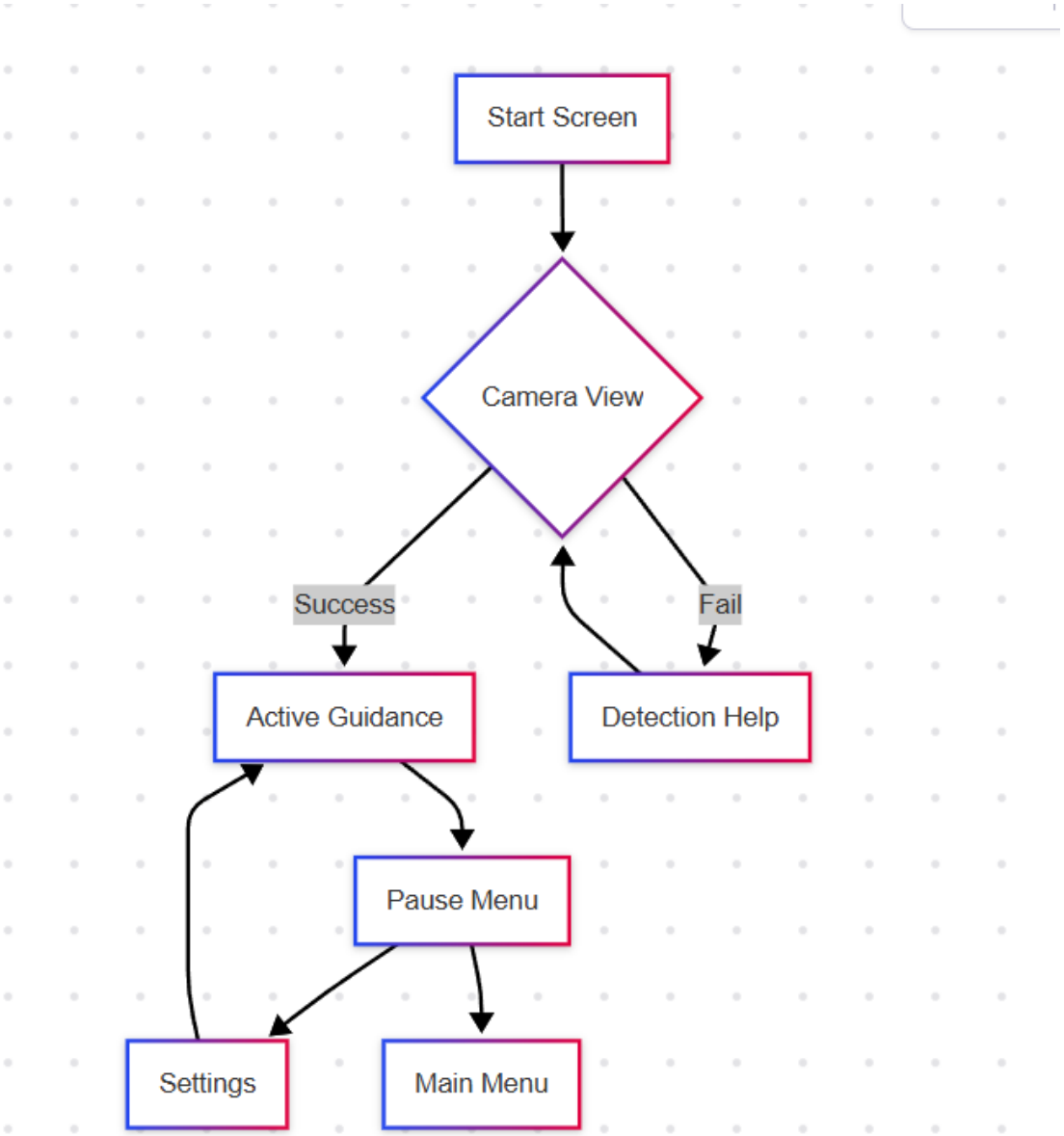
2.5.3 Object and Class Model



2.5.4 Dynamic Models



2.5.5 Navigational Path



3. Glossary

- **Accessibility:** Designing systems usable by people with disabilities, including visual impairments. Focuses on compatibility with assistive technologies like screen readers.
- **Android TTS:** Android's built-in Text-to-Speech engine API.
- **Assistive Technology (AT):** Hardware or software designed to help people with disabilities perform tasks.
- **AVSpeechSynthesizer:** Apple's framework for Text-to-Speech on iOS.
- **BiFPN (Bi-directional Feature Pyramid Network):** A component within EfficientDet that improves multi-scale feature fusion for better object detection, especially for small objects.
- **Computer Vision (CV):** A field of artificial intelligence that enables computers to "see" and interpret visual information from images or videos.
- **Cross-Platform Development:** Building applications that can run on multiple operating systems (like Android and iOS) from a single codebase (e.g., using React Native).
- **Data Augmentation:** Techniques used to artificially increase the size and diversity of a training dataset by applying transformations (rotations, brightness changes, etc.) to existing images.
- **EfficientDet:** A state-of-the-art, efficient object detection model known for balancing accuracy and computational cost.
- **Inference:** The process of using a trained machine learning model to make predictions on new data.
- **Interval-Based Capture:** Capturing images from the camera at discrete time intervals rather than processing a continuous video stream, often to save resources.
- **iOS VoiceOver:** Apple's built-in screen reader for iOS devices.
- **JavaScript:** The primary programming language used with React Native for application logic.
- **mAP (mean Average Precision):** A standard metric for evaluating the accuracy of object detection models.
- **MobileNet:** An efficient convolutional neural network architecture often used as a backbone for mobile vision tasks, including as a base for some EfficientDet variants.
- **Modular Architecture:** Designing a system as a set of independent, interchangeable components (modules) with specific responsibilities.
- **Object Detection:** A CV task that involves identifying the presence, location (using bounding boxes), and class of objects within an image.
- **Offline Functionality:** The ability of an application to perform its core tasks without an active internet connection.
- **OpenCV:** An open-source library widely used for computer vision tasks, including image processing and augmentation.
- **React Native:** A popular JavaScript framework for building native mobile applications for both Android and iOS.
- **Screen Reader:** Assistive software that reads out the content displayed on a screen, used primarily by visually impaired individuals (e.g., TalkBack, VoiceOver).
- **SQLite:** A lightweight, file-based relational database engine often used for local storage on mobile devices.
- **State Machine:** A computational model used to design systems that transition between different modes (states) of operation based on inputs or events.

- **TensorFlow/Keras:** Popular open-source libraries/APIs for developing and training machine learning models, particularly deep learning models.
- **TensorFlow Lite (TFLite):** A framework for optimizing and deploying TensorFlow models on mobile, embedded, and IoT devices.
- **Text-to-Speech (TTS):** Technology that converts written text into audible speech.
- **Transfer Learning:** A machine learning technique where a model pre-trained on a large dataset (like COCO) is adapted (fine-tuned) for a different, specific task (like coffee machine detection).
- **User-Centered Design (UCD):** A design philosophy that prioritizes the needs, wants, and limitations of the end-user throughout the design process, often involving user feedback and iterative testing.
- **WCAG (Web Content Accessibility Guidelines):** Widely adopted international standards for making web and digital content accessible to people with disabilities.

4. References

[1] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. arXiv:1911.09070.

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.

[3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., "TensorFlow: A System for Large-Scale Machine Learning," in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016, pp. 265-283.

[4] OpenCV team, "OpenCV (Open Source Computer Vision Library)". [Online]. Available: https://opencv.org/. [Accessed: March 28, 2025].

[5] Android Developers, "TextToSpeech | Android Developers". [Online]. Available: https://developer.android.com/reference/android/speech/tts/TextToSpeech. [Accessed: March 29, 2025].

[6] Apple Inc., "VoiceOver | Apple Developer Documentation," [Online]. Available: https://developer.apple.com/documentation/accessibility/voiceover. [Accessed: March 29, 2025].

[7] TensorFlow, "TensorFlow Lite Guide". [Online]. Available: https://www.tensorflow.org/lite/guide. [Accessed: March 30, 2025].

[8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems, 2015. arXiv:1506.01497.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in European Conference on Computer Vision, 2016. arXiv:1512.02325.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. arXiv:1506.02640.

[13] Google Cloud, 'Text-to-Speech Al', [Online].
Available: https://cloud.google.com/text-to-speech. [Accessed: April 1, 2025].

[14] J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, and Å. Cajander, "Key principles for user-centred systems design," Behaviour & Information Technology, vol. 22, no. 6, pp. 397-409, 2003.

[15] World Wide Web Consortium, "Web Content Accessibility Guidelines (WCAG) 2.1," W3C Recommendation, 2018. [Online]. Available: https://www.w3.org/TR/WCAG21/. [Accessed: April 1, 2025].

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in European Conference on Computer Vision (ECCV), 2014, pp. 740-755. arXiv:1405.0312.