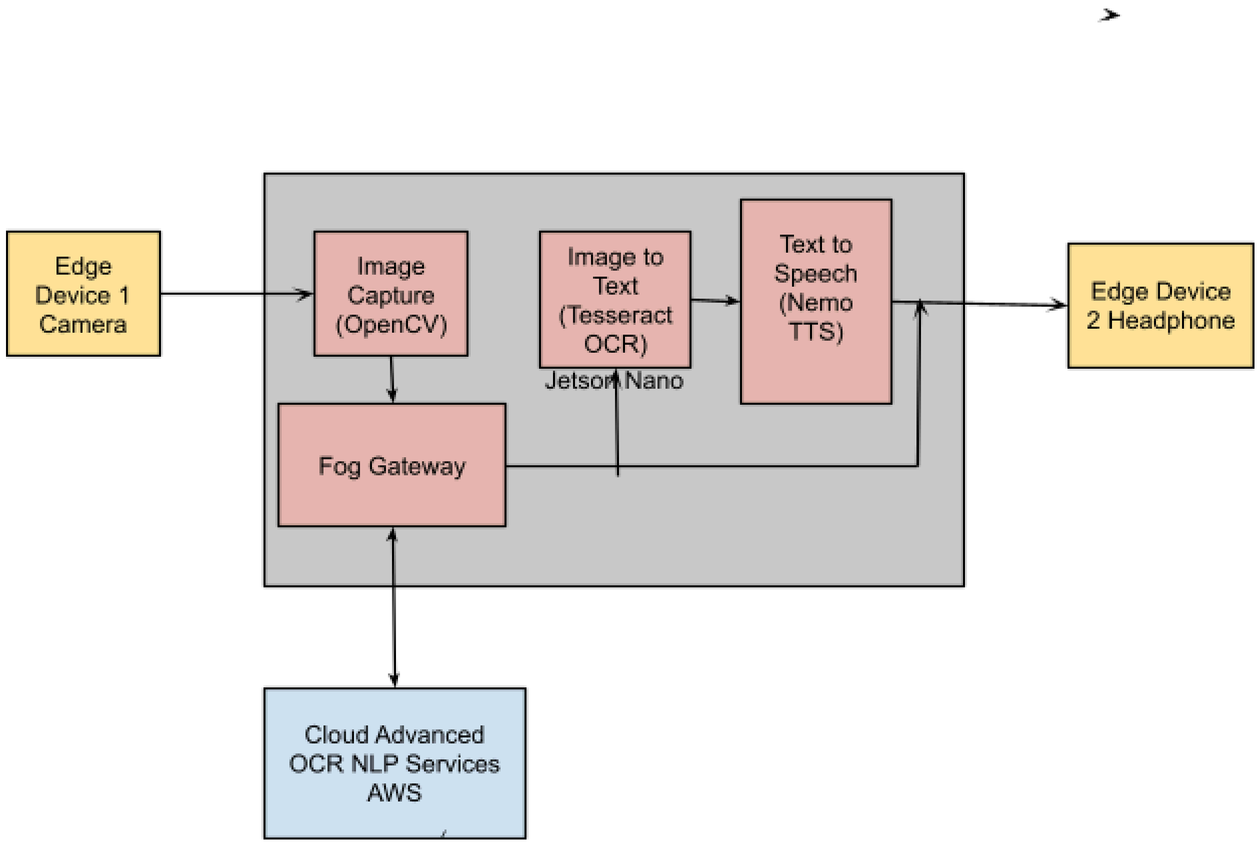
Final Phase Report: Real-Time OCR & TTS System

Team Information

Team Number: 20
Team Members:

- **Nhan Nguyen** (nnguy403@ucr.edu)
- **Pranay Thakur** (pthak008@ucr.edu)

1. Network Architecture Diagram & System Overview


Edge Device 1 Camera
Image Capture (OpenCV)
Image to Text (Tesseract OCR)
Text to Speech (Nemo TTS)
Edge Device 2 Headphone
Jetson Nano
Fog Gateway
Cloud Advanced OCR NLP Services AWS

System Overview

The system is designed to provide visually impaired individuals with real-time text-to-speech conversion using a distributed architecture consisting of edge devices, a fog gateway, and optional cloud services. The primary objective is to balance low latency, privacy, and accuracy by leveraging local computing resources while selectively offloading tasks to the cloud when necessary.

Key Components & Responsibilities

Edge Device #1 - Camera Node

- **Hardware:** Jetson Nano with a camera module
- **Tasks:**
    - Captures images based on user input
    - Preprocesses images (resizing, denoising, contrast enhancement, and text detection)
    - Performs local OCR when feasible

Fog Gateway

- **Hardware:** Jetson Nano
- **Tasks:**
    - Receives image data from Edge Device #1
    - Determines whether OCR processing is done locally or forwarded to cloud services
    - Manages device coordination
    - Ensures encrypted communication between edge devices

Edge Device #2 - TTS Node

- **Hardware:** Jetson Nano connected to headphones/speaker
- **Tasks:**
    - Converts recognized text into speech using Piper TTS
    - Performs OCR if computational resources allow

Cloud Services (Optional - Used Only When Necessary)

- **Services Used:** AWS Textract
- **Tasks:**
    - Handles complex OCR tasks when local processing is insufficient
    - Supports multilingual recognition for enhanced accessibility

2. Device Catalog with Hardware Features

Device	Model	Features
Edge Device #1	Jetson Nano	Camera module, ARM CPU, 4GB RAM
Fog Gateway	Jetson Nano	API handling, Orchestration, Load Balancing
Edge Device #2	Jetson Nano	Headphone output, Speech synthesis
Cloud	AWS Textract	High-accuracy OCR processing

3. Network Operation Report

Challenges Encountered & Solutions

1. Limited Processing Power on Jetson Nano

- **Challenge:** Running OCR locally resulted in slow processing and high resource usage.
- **Solution:** Implemented selective offloading to AWS Textract for complex OCR tasks while keeping simple text processing local.

2. Latency in Image Processing

- **Challenge:** Image capture and transmission introduced processing delays.
- **Solution:** Applied **image preprocessing techniques** (denoising, binarization) to enhance OCR efficiency, reducing network congestion and improving response time.

3. Privacy Concerns for Cloud Processing

- **Challenge:** Some users may not want their text data transmitted to the cloud.
- **Solution:** Designed the system to prioritize local processing and send data to the cloud only when necessary. Also, implemented end-to-end encryption for secure data transmission.

4. OCR Accuracy Issues

- **Challenge:** OCR sometimes detects incorrect or out-of-order text.
- **Solution:** improved image preprocessing with cv2 for better text alignment.

Services & Tools Used

- **PyTesseract:** Primary OCR engine for local text recognition.
- **OpenCV**: Preprocessing image.
- **Amazon Textract:** Cloud processing when accuracy falls below threshold on local device.
- **Flask REST API (on Fog Gateway):** Handles communication between edge devices and cloud services.
- **Piper TTS:** Converts recognized text into speech locally.

4. Quality Attributes Implemented

- **Security**: Encrypted communication between devices.
- **Privacy**: User data remains on local devices unless cloud offloading is necessary.
- **Reliability**: Automatic fallback to local processing if cloud services are unavailable.
- **Low Latency**: Optimized edge computing and reduced network congestion

5. Messaging Patterns & Communication Protocols

- MQTT / HTTP REST API: Used for communication between edge devices and Fog Gateway.

6. Next Steps & Improvements

1. Improve Local OCR Performance

- **Optimization:** Further fine-tuning of preprocessing techniques and using more efficient OCR models to reduce dependency on cloud services.
- **GPU Acceleration:** Utilize Jetson Nano's GPU for better OCR processing speed.

2. Develop an Intuitive User Interface

- **Voice Commands:** Implement a speech-based interface for hands-free operation.
- **Better document detection:** Help detect the document to be used for OCR better with real time feedback.
- **Mobile App Integration:** Provide an alternative interface for controlling the system.

3. Enhance Multilingual Support

- **Expanded Language Models:** Integrate additional language support to improve accessibility for non-English speakers.
- **Real-Time Translation:** Implement on-the-fly text translation before TTS conversion.

4. Real-World Deployment & Testing

- **User Testing:** Conduct extensive testing with visually impaired individuals to gather feedback.
- **Performance Evaluation:** Measure latency, accuracy, and usability in different lighting conditions and environments.

Conclusion

The final deployment of our Real-Time OCR & TTS System successfully addresses the challenge of providing a cost-effective, privacy-conscious solution for visually impaired individuals. By leveraging Jetson Nano, optimized OCR techniques, and cloud integration, the system ensures efficient, reliable, and secure text-to-speech conversion. With future enhancements in document detection, live feedback, multilingual support, and real-world testing, this project can become a good assistive technology solution.

Some test results:

OCR Accuracy

	Local	AWS
	84.52%	99.4%
	84.63%	99.5%
	84.76%	99.6%
	84.88%	99.35%
	84.87%	99.55%
Average	84.732%	99.48%