Technical Documentation

1. System Architecture

1.1 Overview

The system implements a multi-stage pipeline for processing customer complaints using Azure OpenAI services. The pipeline consists of five main components:

- Audio transcription (Whisper)
- Image generation (DALL-E)
- Image analysis and annotation (GPT-4V)
- Complaint classification (GPT-4)
- Workflow orchestration

1.2 Component Interaction

```
[Audio Input] → [Whisper] → [Text] → [DALL-E] → [Image] → [GPT-4V] → [Annotated Image]
                                                                    ↓
                                                          [Classification (GPT-4)]
```

2. Implementation Details

2.1 Audio Transcription (`whisper.py`)

- **Technology**: Azure OpenAI Whisper model
- **Key Features**:
  - Supports MP3 and WAV formats
  - Creates sample-specific output directories
  - Implements error handling for failed transcriptions
- **Output**: Transcribed text saved as `transcription.txt`

2.2 Image Generation (`dalle.py`)

- **Technology**: Azure OpenAI DALL-E model
- **Key Features**:
  - Focused prompt generation
  - Retry mechanism for failed generations
  - Sample-specific image storage
- **Output**: Generated image saved as `generated_image.png`

2.3 Image Analysis (`vision.py`)

- **Technology**: Azure OpenAI GPT-4V
- **Key Features**:
  - Complaint-focused image analysis
  - Coordinate-based annotation system
  - Semi-transparent text overlays
- **Outputs**:
  - Annotated image (`annotated_image.png`)
  - Image description (`image_description.txt`)

2.4 Classification (`gpt.py`)

- **Technology**: Azure OpenAI GPT-4
- **Key Features**:
  - Category/subcategory classification
  - Confidence scoring
  - Reasoning explanation
- **Output**: Classification results in `classification.txt`

2.5 Workflow Orchestration (`main.py`)

- **Key Features**:
  - Logging system
  - Error handling and retries
  - Batch processing
  - Accuracy calculation

3. Challenges and Solutions

3.1 Image Generation Reliability

Challenge: DALL-E occasionally fails to generate images

Solution:

- Implemented retry mechanism with configurable attempts
- Added delay between retries
- Improved prompt engineering for better results

3.2 Annotation Readability

Challenge: Text annotations were difficult to read on various backgrounds

Solution:

- Added semi-transparent background to text
- Optimized font size and color
- Implemented marker system for better visibility

3.3 Classification Accuracy

Challenge: Initial classifications were inconsistent

Solution:

- Enhanced prompt engineering
- Added structured category format
- Implemented confidence scoring
- Added reasoning requirement

3.4 Error Handling

Challenge: System failures in one component affected entire pipeline

Solution:

- Implemented comprehensive error handling
- Added logging system
- Created sample-specific directories
- Added retry mechanisms

4. Performance Optimization

4.1 Processing Speed

- Batch processing for multiple samples
- Efficient file handling
- Optimized API calls

4.2 Resource Usage

- Sample-specific directory structure
- Cleanup of temporary files
- Efficient image processing

4.3 Accuracy Improvements

- Enhanced prompt engineering
- Structured output formats
- Validation against predefined categories

5. Future Improvements

5.1 Proposed Enhancements

1. Parallel processing for multiple samples
2. Enhanced retry strategies
3. Additional classification metrics
4. User interface for system monitoring
5. Real-time processing capabilities

5.2 Scalability Considerations

1. Load balancing for API calls
2. Database integration for results
3. Containerization for deployment
4. API endpoint creation

6. Maintenance and Monitoring

6.1 Logging System

- Comprehensive logging of all operations
- Error tracking and reporting
- Performance metrics collection

6.2 Quality Assurance

- Input validation
- Output verification
- Category validation
- Accuracy monitoring

7. Configuration Guide

7.1 Environment Variables

```
AZURE_OPENAI_API_KEY=your_key
AZURE_OPENAI_ENDPOINT=your_endpoint
WHISPER_DEPLOYMENT=deployment_name
WHISPER_VERSION=api_version
GPT_DEPLOYMENT=deployment_name
GPT_VERSION=api_version
DALLE_DEPLOYMENT=deployment_name
DALLE_VERSION=api_version
```

7.2 Output File Structure

```
output/
  └── sample_name/
      ├── transcription.txt
      ├── generated_image.png
      ├── annotated_image.png
      ├── image_description.txt
      └── classification.txt
```

8. Testing and Validation

8.1 Test Cases

- Audio format compatibility
- Image generation quality
- Classification accuracy
- Error handling effectiveness

8.2 Validation Methods

- Accuracy metrics
- Manual review process
- Error rate monitoring
- Performance benchmarking