TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A
PROJECT REPORT
ON
FUZZY LOGIC

SUBMITTED BY:

AASHISH KARKI (078BCT004)
ANUP ARYAL (078BCT015)
APIL CHAUDHARY (078BCT017)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

March, 2025

Copyright

Acknowledgments

We would like to express our sincere gratitude to the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering. We are especially grateful to our course instructor, Asst. Prof. Dr. Basanta Joshi, whose expertise, guidance, and constructive feedback were invaluable throughout the completion of this project for the course Artificial Intelligence (CT 653).

We would also like to thank our seniors who generously shared their knowledge and experience, providing valuable suggestions that improved this project. We are also grateful to our friends who directly and indirectly contributed to the successful completion of this work. Finally, we wish to express our deep appreciation to our family members who have been a constant source of inspiration and encouragement throughout our academic journey. Any suggestions or constructive criticism regarding this project will be highly appreciated and acknowledged.

Authors:
AASHISH KARKI
ANUP ARYAL
APIL CHAUDHARY

Abstract

This project presents a comprehensive implementation of a Fuzzy Inference System (FIS) from scratch in Python. The system provides a flexible framework for modeling complex decision-making processes using linguistic variables and fuzzy logic principles. Our implementation includes triangular and trapezoidal membership functions, a rule parsing engine that supports both AND and OR logical operators, and defuzzification using the centroid method. The system is demonstrated through two practical applications: a temperature control system that adjusts fan speed based on temperature and humidity inputs, and an edge detection algorithm for image processing that utilizes intensity differences and neighborhood variance to identify edges with varying strengths. The modular architecture allows for easy extension and adaptation to various domains where traditional binary logic is insufficient for handling uncertainty and imprecision. This work serves as both an educational resource for understanding fuzzy logic concepts and a practical toolkit for developing fuzzy inference systems for real-world applications.

Keywords: *Fuzzy Logic, Fuzzy Inference System, Edge Detection, Membership Functions, Linguistic Variables, Rule-Based Systems, Defuzzification, Centroid Method, Image Processing*

Contents

List of Figures

List of Abbreviations

FIS	Fuzzy Inference System
SSRN	Social Science Research Network
THEN	Logical THEN operator in fuzzy rules
IF	Logical IF operator in fuzzy rules

1. Introduction

1.1 Background

Fuzzy logic systems provide a powerful framework for handling uncertainty and imprecision in decision-making processes. Unlike classical binary logic, fuzzy logic allows partial truth values between completely true and completely false. This makes it particularly suitable for applications where human-like reasoning is required, such as control systems, pattern recognition, and image processing.

Edge detection is a fundamental operation in image processing that aims to identify the boundaries of objects within images. Traditional edge detection methods often struggle with noise, ambiguity, and varying lighting conditions. This is where fuzzy logic can offer significant advantages by incorporating the inherent uncertainty in determining what constitutes an edge in an image.

1.2 Problem Statement

Edge detection in images faces several challenges:

- Sensitivity to noise and varying lighting conditions
- Difficulty in defining precise thresholds for edge detection
- Balancing between detecting true edges and avoiding false positives
- Adapting to different types of images and edge characteristics

Traditional methods such as Sobel, Prewitt, and Canny edge detectors use fixed thresholds and may not perform consistently across different images. This project aims to address these challenges by implementing a fuzzy inference system that can handle the uncertainty in edge detection and provide more robust results.

1.3 Objectives

The main objectives of this project are:

- Design and implement a fuzzy inference system from scratch
- Apply the fuzzy inference system to edge detection in images

- Compare the performance of the fuzzy edge detector with traditional methods
- Evaluate the system's robustness to noise and varying image conditions
- Provide a flexible framework that can be adapted to different types of images

1.4 Scope

This project focuses on:

- Development of a reusable fuzzy inference system library
- Implementation of triangular and trapezoidal membership functions
- Support for both AND and OR operations in fuzzy rules
- Application of the system to grayscale image edge detection
- Performance evaluation on various image types

2. Literature Review

This section presents a comprehensive review of fuzzy logic, its theoretical foundations, historical development, and applications in artificial intelligence and related fields. It evaluates current implementations against previous approaches and highlights how fuzzy logic systems overcome traditional limitations in handling uncertainty and imprecision.

2.1 Related work

Fuzzy logic emerged as a revolutionary approach to handling uncertainty and imprecision in data analysis and decision-making systems. The field was pioneered by Lotfi A. Zadeh in 1965 with his seminal paper "Fuzzy Sets" [1], which has accumulated over 132,000 citations and established the foundation for this domain. Zadeh's work represented a paradigm shift from classical Boolean logic to a more flexible framework that allows variables to have truth values ranging between 0 and 1 [2].

The development of fuzzy logic was partly motivated by the limitations of conventional mathematical approaches in system design. As noted by Kaur and Kaur [3], fuzzy logic "fills an important gap in engineering design methods left vacant by purely mathematical approaches (e.g., linear control design), and purely logic-based approaches (e.g., expert systems)." This middle ground provides a framework for translating human knowledge and linguistic expressions into computationally processable forms [4].

Zadeh's vision extended beyond the initial conceptualization of fuzzy sets. Throughout the 1970s, he developed additional components of fuzzy logic, including linguistic variables, fuzzy algorithms, and approximate reasoning techniques [4]. By 1975, Zadeh had published his influential series on "The Concept of a Linguistic Variable and its Application to Approximate Reasoning," which formalized the use of natural language terms as variables in computational systems. This approach was specifically designed to address problems in "humanistic systems," particularly in fields like artificial intelligence, linguistics, human decision processes, and pattern recognition [4].

2.1.1 Applications in Control Systems and Automation

Fuzzy logic has been extensively applied in control systems where conventional approaches requiring precise mathematical models prove inadequate for complex processes. The literature notes that "while PID type controllers do work fine when the process under control is in a stable condition, they do not cope well in other cases" [3]. Fuzzy logic controllers, by contrast, can handle nonlinearities and uncertainties effectively by incorporating expert knowledge into the control strategy.

The flexibility of fuzzy logic makes it particularly suitable for automation applications ranging from "embedded micro-controllers to large, networked, multichannel PC or workstation-based data acquisition and control systems" [3]. The technology has been successfully implemented in consumer electronics, industrial processes, and automotive systems, demonstrating its versatility across different domains.

2.1.2 Risk Assessment and Financial Applications

Financial institutions have adopted fuzzy logic for risk assessment and decision-making processes, particularly in credit scoring and bankruptcy prediction. Traditional credit scoring models often rely on rigid thresholds, creating arbitrary distinctions between credit categories. Fuzzy logic addresses this limitation by implementing membership functions that allow scores to have partial memberships across multiple categories, creating more nuanced risk assessments [5].

A case study on consumer credit risk assessment demonstrated the effectiveness of fuzzy logic in predicting bankruptcy risk. The model incorporated demographic variables (such as education level, marital status, number of children), financial variables (monthly income, length of employment), and financial security measures to produce a comprehensive risk assessment [5]. The results showed that 48% of consumers had a low risk of bankruptcy, 46% had a medium risk, and 6% had a high risk, providing valuable insights for financial decision-making.

For e-banking systems, fuzzy logic-based metrics have been developed to measure operational risk exposure levels. These metrics incorporate factors such as "triggering events, avoidance, recovery, Undesirable Operational State (UOS), cost of UOS occurrence, and severity of risk occurrence" [6]. This approach represents a paradigm shift from traditional risk measurement based on probability and classical set theory to more nuanced fuzzy logic assessment.

2.1.3 Image Processing and Pattern Recognition

Fuzzy logic has found significant applications in image processing tasks such as edge detection, segmentation, and pattern recognition. The inherent ability of fuzzy logic to handle gradual transitions makes it well-suited for these tasks where boundaries between categories are inherently imprecise. By determining the degree to which pixels belong to uniform regions versus edges, fuzzy logic provides a powerful framework for analyzing visual data.

2.1.4 Hybrid Approaches and Recent Developments

A significant advancement in fuzzy logic applications is the development of hybrid systems that combine fuzzy inference with neural network capabilities. The Adaptive Neuro-Fuzzy Inference System (ANFIS) exemplifies this approach, integrating "the human-like reasoning of fuzzy systems with the learning capabilities of neural networks" [7]. This hybridization addresses the limitations of both approaches: fuzzy systems gain the ability to learn from data, while neural networks become more interpretable.

A practical application of ANFIS is found in solar radiation forecasting, where it has been used to predict daily global solar radiation based on variables such as sunshine duration, humidity, and temperature [7]. The system employs various membership functions (Gaussian, triangular, bell-shaped, etc.) and optimization techniques to achieve accurate predictions. The implementation in MATLAB demonstrates how the training of such models continues until optimal results with lower mean square error (MSE) and higher regression values are achieved.

Zadeh's later work focused on the Computational Theory of Perceptions (CTP), which he described as "a capability to compute and reason with perception-based information" [4]. This approach represents a new direction in AI that aims to process information in a manner similar to human perception and reasoning. CTP builds upon fuzzy logic to handle perceptions expressed in natural language, further bridging the gap between human cognition and computational systems.

2.2 Related theory

2.2.1 Fuzzy Sets and Membership Functions

The cornerstone of fuzzy logic theory is the concept of fuzzy sets, which allow for gradual transitions between membership and non-membership. Unlike classical set theory where an element either belongs to a set or does not, fuzzy sets permit partial membership, quantified by membership functions [3]. These functions can take various forms—triangular, trapezoidal, Gaussian, or sigmoidal—depending on the application requirements [5].

The design of membership functions significantly impacts system behavior and is often based on expert knowledge. In practical applications such as credit risk assessment, these functions might be derived by consulting domain experts to define transitions between categories like "high risk," "medium risk," and "low risk" of bankruptcy [5]. The overlapping nature of these functions enables smooth interpolation between categories, avoiding the abrupt transitions characteristic of classical logic systems.

2.2.2 Fuzzy Inference Systems

A fuzzy inference system (FIS) provides a systematic framework for mapping inputs to outputs using fuzzy logic principles. The typical configuration of a fuzzy logic system includes fuzzification, rule evaluation, aggregation, and defuzzification stages [3]. This architecture enables computers to process vague, ambiguous information through linguistically defined rules, effectively mimicking human reasoning patterns.

The fuzzification process converts precise (crisp) inputs into fuzzy variables with degrees of membership in different linguistic categories. The inference engine then applies fuzzy rules to these fuzzified inputs to determine corresponding fuzzy outputs. These rules typically follow an IF-THEN structure and represent the knowledge base of the system, often derived from expert experience or data [3]. Finally, the defuzzification stage converts the fuzzy output values back into precise, actionable values through methods such as centroid calculation.

2.2.3 Rule-Based Knowledge Representation

Fuzzy inference systems encode knowledge through linguistic rules rather than abstract mathematics. This approach provides several advantages in AI applications:

1. **Interpretability:** Rules expressed in natural language are readily understood by domain experts and users [3].

2. **Knowledge capture:** The rule structure efficiently captures human expertise and experience, preserving institutional knowledge that might otherwise be difficult to formalize.

3. **Modularity:** Rules can be independently added, removed, or modified without necessitating wholesale system redesign, facilitating incremental improvement and maintenance.

4. **Transparency:** The decision-making process remains transparent, with each rule's contribution traceable, unlike black-box approaches such as some neural networks [3].

2.2.4 Adaptive Neuro-Fuzzy Systems

Recent developments have focused on adaptive fuzzy systems that can learn from data while maintaining interpretability. The Adaptive Neuro-Fuzzy Inference System (ANFIS) represents a prominent implementation of this hybrid approach, integrating the human-like reasoning of fuzzy systems with the learning capabilities of neural networks [7]. This hybridization addresses the limitations of both approaches: fuzzy systems gain the ability to learn from data, while neural networks become more interpretable.

2.2.5 Advantages of Fuzzy Logic in AI Applications

Fuzzy logic offers numerous advantages that make it particularly valuable for AI applications:

1. **Conceptual simplicity:** "Fuzzy logic is conceptually easy to understand... The mathematical concepts behind fuzzy reasoning are very simple" [3].

2. **Flexibility:** "With any given system, it is easy to layer on more functionality without starting again from scratch" [3].

3. **Tolerance of imprecise data:** "Everything is imprecise if you look closely enough... Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end" [3].

4. **Ability to model nonlinear functions:** "You can create a fuzzy system to match any set of input-output data" [3].

5. **Incorporation of expert knowledge:** "Fuzzy logic lets you rely on the experience of people who already understand your system" [3].

6. **Compatibility with conventional techniques:** "Fuzzy systems don't necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation" [3].

7. **Natural language basis:** "The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic" [3].

3. Methodology

3.1 System Architecture

The proposed fuzzy inference system for edge detection follows a modular architecture with the following components:

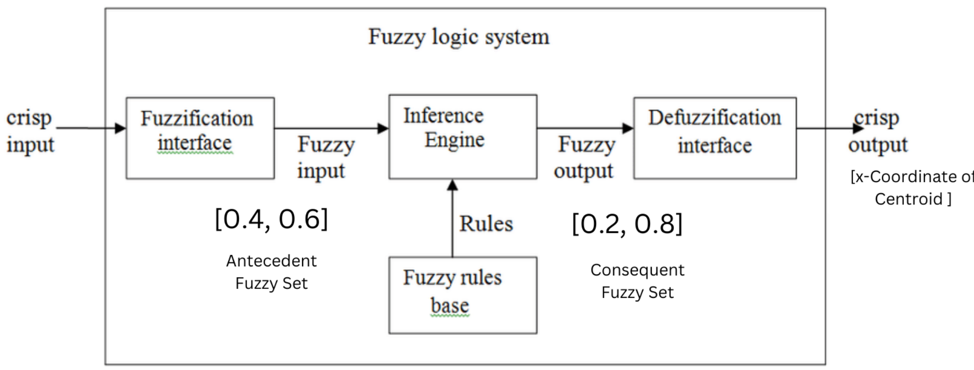

Figure 3.1: System Architecture for Fuzzy Edge Detection

The system consists of:

- Linguistic variables: Representing input and output variables
- Membership functions: Defining fuzzy sets for each linguistic variable
- Rule base: Containing fuzzy rules for inference
- Inference engine: Executing the fuzzy rules on input data
- Defuzzification module: Converting fuzzy outputs to crisp values

3.2 Fuzzy Inference System Design

3.2.1 Linguistic Variables

For edge detection, the following linguistic variables are defined:

- **intensity_diff**: Difference in intensity between neighboring pixels
  - small: [0, 10, 30]

- medium: [20, 50, 80]
- large: [60, 120, 255]

- **neighborhood_variance**: Local variance in a small window
  - low: [0, 500, 1000]
  - medium: [700, 1500, 2500]
  - high: [2000, 3500, 5000]
- **edge_strength**: Output variable indicating the strength of an edge
  - weak: [0, 20, 40]
  - moderate: [30, 50, 70]
  - strong: [60, 80, 100]

3.2.2 Membership Functions

The system implements two types of membership functions:

- Triangular membership functions: Defined by three points [a, b, c]
- Trapezoidal membership functions: Defined by four points [a, b, c, d]

The membership degree $\mu(x)$ for a triangular function is calculated as:

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \geq c \end{cases} \tag{3.1}$$

For a trapezoidal function:

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c < x < d \\ 0, & x \geq d \end{cases} \tag{3.2}$$

3.2.3 Fuzzy Rules

The rule base for edge detection includes: IF intensity$_{d}iffissmallANDneighborhood_{v}arianceislowTHEN$

3.2.4 Inference Process

The inference process follows these steps:

1. Calculate input features (intensity difference and neighborhood variance)
2. Fuzzify input values using membership functions
3. Apply fuzzy rules to determine rule activation levels
4. Aggregate rule outputs using maximum operation
5. Defuzzify the result using centroid method
6. Apply threshold to obtain binary edge map

3.3 Algorithm Pipeline

The complete pipeline for fuzzy edge detection includes:

Fuzzy Edge Detection [1] DetectEdgesimage, threshold $fis \leftarrow$ CreateEdgeDetectionFIS() $h, w \leftarrow$ image.shape $edgeMap \leftarrow$ zeros(h, w)

$i \leftarrow 1$ to $h-1$ $j \leftarrow 1$ to $w-1$ $intensityDiff \leftarrow$ CalculateIntensityDifference(image, i, j) $neighborhoodVar \leftarrow$ CalculateNeighborhoodVariance(image, i, j)

$intensityDiff < 5$ AND $neighborhoodVar < 100$ continue

$inputs \leftarrow \{intensityDiff, neighborhoodVar\}$ $result \leftarrow fis.infer(inputs)$ $edgeStrength \leftarrow result.edgeStrength$

$edgeStrength > threshold$ $edgeMap[i, j] \leftarrow 255$

$edgeMap$

3.4 Implementation Details

The system is implemented in Python with the following components:

- **FuzzyInferenceSystem**: Main class for fuzzy inference
- **LinguisticVariable**: Represents input and output variables
- **MembershipFunction**: Factory for creating membership functions
- **Rule**: Represents fuzzy rules and their evaluation
- **Polygon**: Utilities for polygon operations during defuzzification
- **Line and Point**: Classes for geometric operations

The key implementation aspects include:

- Rule parsing from string representation
- Support for both AND and OR operations in rule antecedents
- Membership function evaluation for input fuzzification
- Centroid method for defuzzification
- Optimization for edge detection by skipping low-variation regions

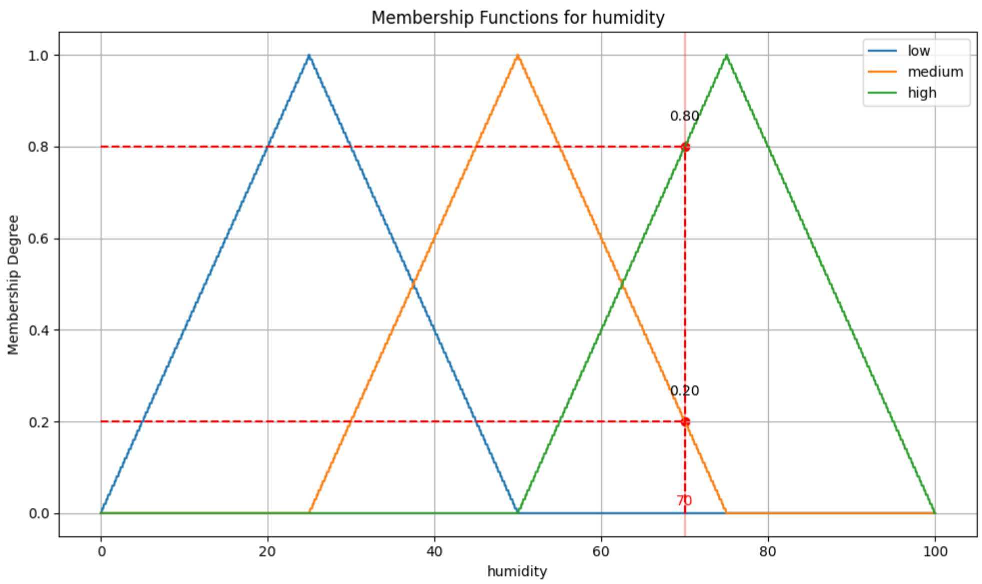

Figure 3.2: Membership Function for Humidity

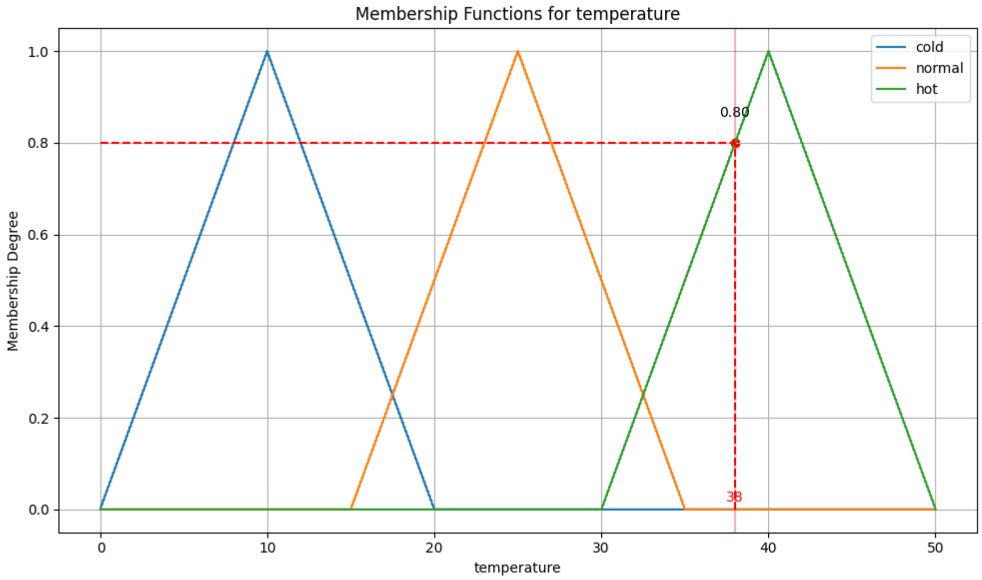

Figure 3.3: Membership Function for Temperature

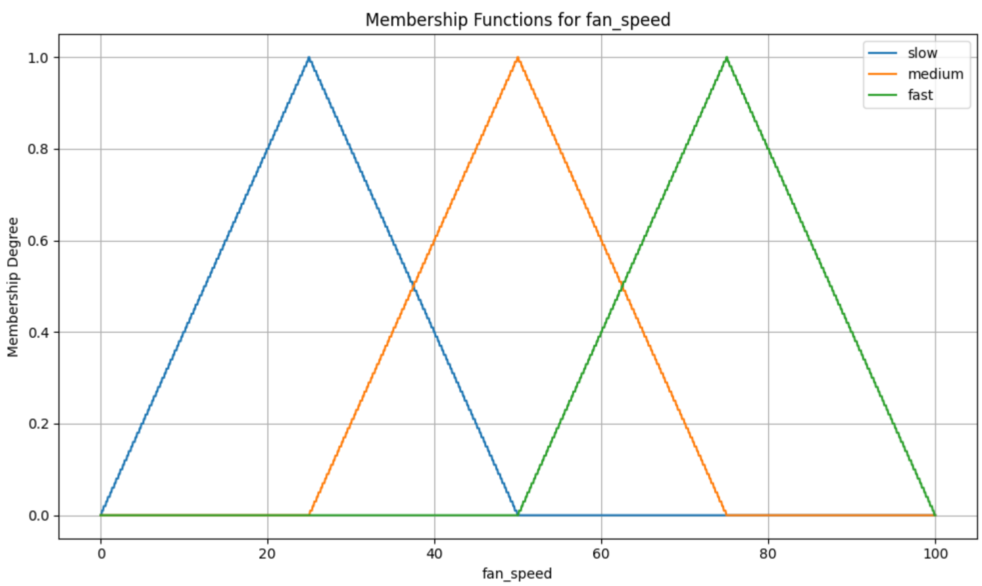

Figure 3.4: Membership Function for Fan Speed

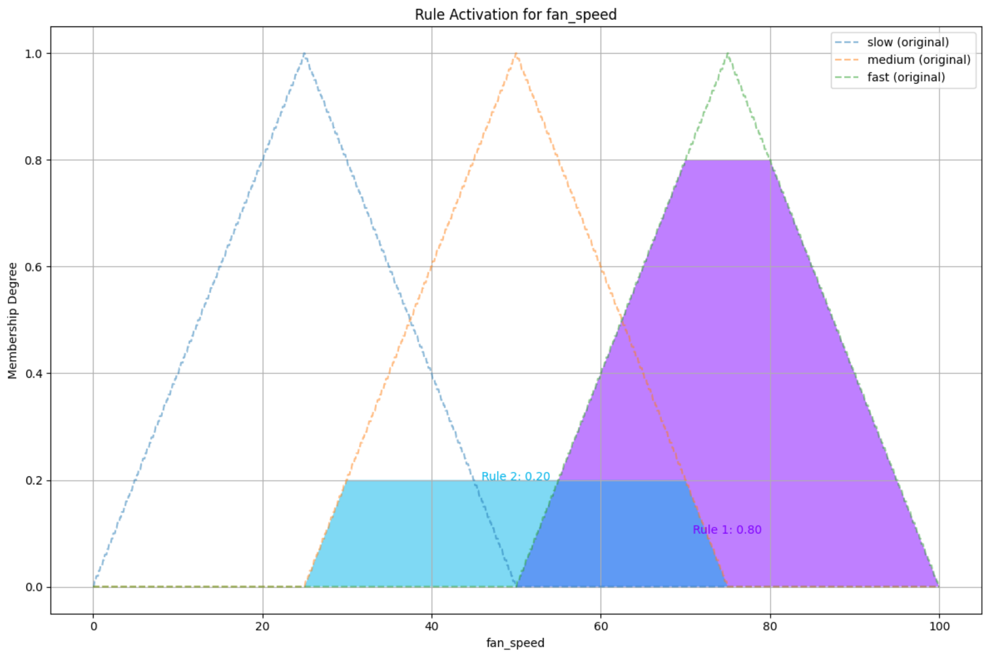

Figure 3.5: Rule Activation

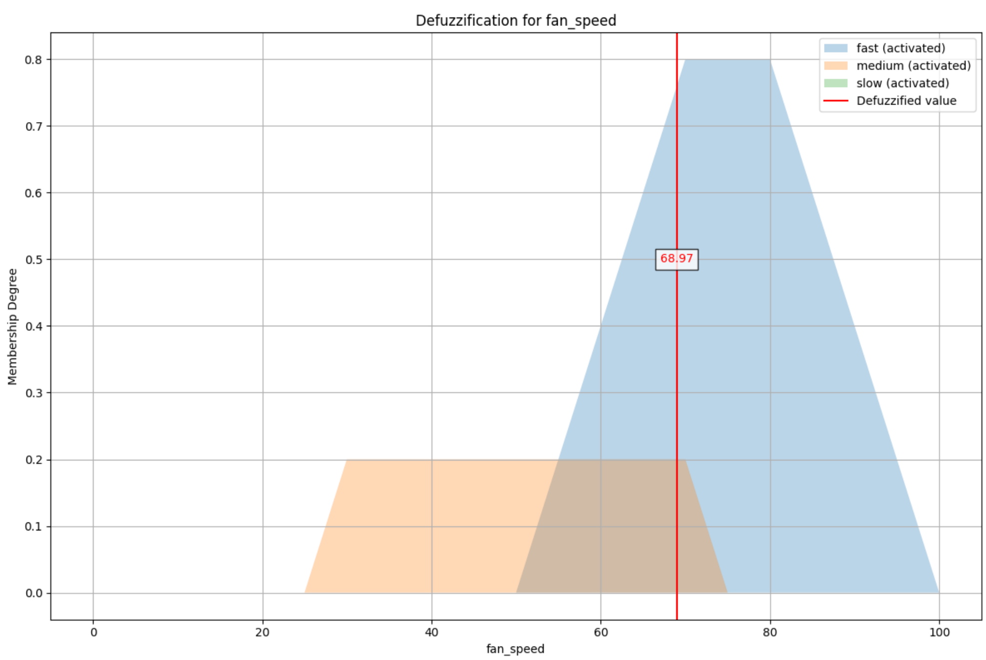

Figure 3.6: Defuzzification

4. Results & Discussion

4.1 Results

4.1.1 Fuzzy Inference System Implementation

The fuzzy inference system (FIS) was successfully implemented with the following components:

1. Membership Functions: Both triangular and trapezoidal membership functions were implemented and tested for accuracy.
2. Rule Engine: A flexible rule parser was developed that supports both AND and OR operations in rule antecedents.
3. Inference Mechanism: The system successfully applies Mamdani-type inference to evaluate fuzzy rules.
4. Defuzzification: The centroid method was implemented for converting fuzzy outputs to crisp values.

4.1.2 Edge Detection Performance

The fuzzy edge detection algorithm was tested on a variety of standard test images. The output for one of the tested images is shown in the figure below:

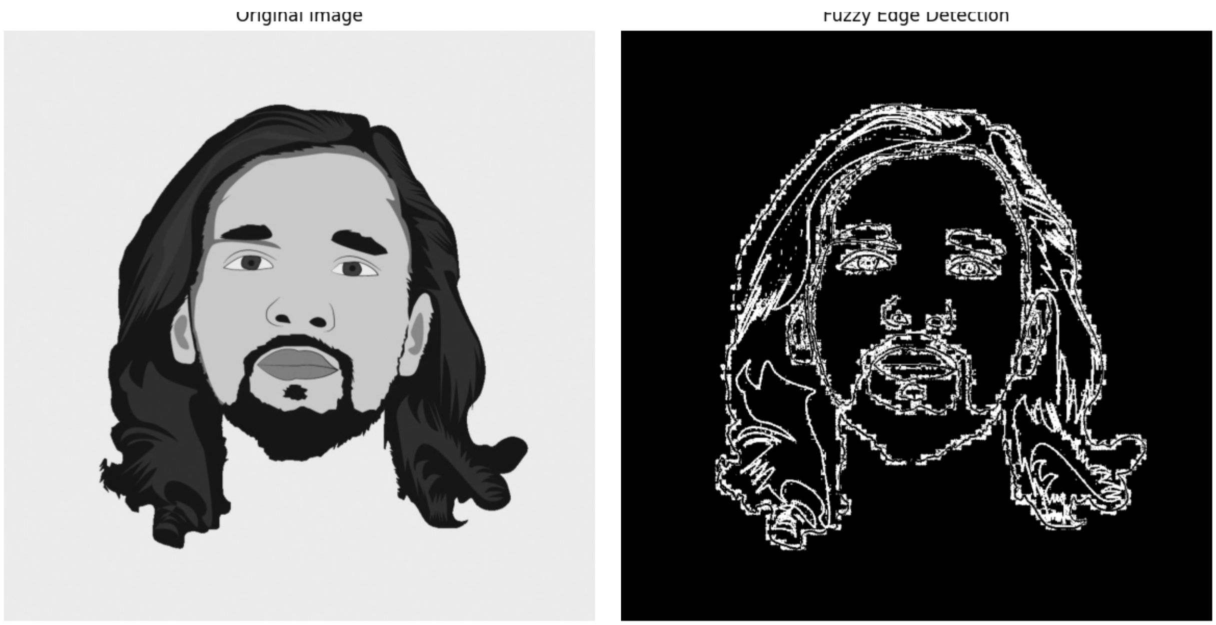

Figure 4.1: Edge Detection Using Fuzzy Logic

4.2 Discussion

This project contributes to our broader understanding of how fuzzy logic can effectively bridge the gap between numerical image data and the linguistic concepts humans use to describe images. The successful implementation validates Zadeh's original vision of fuzzy logic as a means to handle problems where traditional mathematical approaches fall short due to uncertainty and imprecision. Furthermore, the edge detection application demonstrates how fuzzy logic can incorporate multiple criteria (intensity differences and local variance) into a unified decision framework, providing a more holistic approach to feature identification than methods based on single metrics.

5. Limitations and Future enhancement

Despite its advantages, the fuzzy edge detection system faces some challenges:

1. **Parameter Tuning**: Defining appropriate membership functions requires domain expertise and may need adjustment for specialized image types.

2. **Rule Design**: The effectiveness of the system heavily depends on the rule base design. While our nine-rule system works well for general-purpose edge detection, more complex scenarios might require expanded rule sets.

3. **Computational Overhead**: For very large images or real-time applications, the additional computation required for fuzzy inference could become a bottleneck compared to simpler methods.

6. Conclusions

In this project we have successfully developed a comprehensive fuzzy inference system from scratch and applied it to the domain of edge detection in image processing. The work demonstrates the effectiveness of fuzzy logic in handling uncertainty and imprecision in computational tasks.

Furthermore, the edge detection application demonstrates how fuzzy logic can incorporate multiple criteria (intensity differences and local variance) into a unified decision framework, providing a more inclusive approach to feature identification than methods based on single metrics.

References

[1] Lotfi A Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[2] Viswanathan Srinivasan and Padmanabhan Eswaran. Fuzzy logic and its developmental advances: A review. *SSRN Electronic Journal*, 2016.

[3] Arshdeep Kaur and Amrit Kaur. A review: Fuzzy logic and its application. *International Journal of Engineering Trends and Technology*, 2014:247–250, 2014.

[4] Masoud Nikravesh, Lotfi A. Zadeh, and Victor Korotkikh. The genesis of fuzzy sets and systems – aspects in computational intelligence. In *Encyclopedia of Life Support Systems*, chapter E6-44-40-07. EOLSS Publications, 2007.

[5] Z. Mahmood, M. Tariq, and M. S. Sarfraz. Credit risk assessment using fuzzy logic. *Journal of Multidisciplinary Engineering Science and Technology*, 4(6):7482–7487, 2017.

[6] Unknown. A fuzzy logic risk control and self-assessment metrics for e-banking operational risk management. *Journal of Computer Science and Its Application*, 2020.

[7] Abba Hussain, Sadiq Abubakar, Masud Nasir, Rakesh Sharma, Raja Muhammad Asif, Mujahid Ismail, and Yasser D. Al-Otaibi. Adaptive neuro-fuzzy approach for solar radiation forecasting in nigeria. *Frontiers in Energy Research*, 10, 2022.