**Due Date: 25/01/24**

**SIR SYED UNIVERSITY OF ENGINEERING & TECHNOLOGY**

**COMPUTER SCIENCE & INFORMATION TECHNOLOGY DEPARTMENT**

**Fall 2023**

**Parallel & Distributed Computing (CS-429)**

**Assignment # 3**

Semester: 8th Batch: 2020

Announced Date: 24/01/24 Due Date:

Total Marks: 04 Marks Obtained:

Instructor Name: Asif Raza

**Dataset of G12:** <https://www.kaggle.com/datasets/alik05/forest-fire-dataset>

**Group Members:**  
**Muhammad Shayan Ashraf**

**BSCS -2020 -097  
Section: C**

**Muhammad Nouman Imran**

**BSCS -2020 -097**

**Section: C**

| **CLO #** | **Course Learning Outcomes (CLOs)** | **PLO Mapping** | **Bloom’s Taxonomy** |
| --- | --- | --- | --- |
| CLO 3 | **Classify** the following program using GPU utilization in python. | PLO\_1  (Academic Education) | C2  (**Predict**) |

Q1. **Classify** the following program using GPU utilization in python by using your existing data set with **Fuzzy Logic**. Assignment must contain/cover the following points.

**Note:** You have already given a dataset in **ASSIGNMENT 1** for simulation. In **ASSIGNMENT 1** you have already designed a **GPU based model** in Deep learning. In this assignment you have to classify your model based on **Fuzzy Logic** of your own choice**.** You may use any variant of Fuzzy Logic but **NO** group should practice same **Fuzzy Logic** **variant** in same class.

1. Abstract (450 words) one paragraph
2. Detail of dataset (450 words) one paragraph
3. Short detail (450 words) of types of PSO **Fuzzy Logic** **variant** that you have used in program.
4. Program must use auto split function to split dataset into 70, 15, and 15 (Training, Testing, and validation)
5. Precision, Recall, F1-Score, True Positives, False positives, True Negatives, False Negatives
6. Plot the training and validation accuracy graph & Plot the training and validation loss graph.
7. Plot the confusion matrix for the training and Validation set.
8. Create a line plot graph for the number of images per class
9. Calculate ROC curves, AUC, and error rates for each class
10. Calculate image counts graph/Number of images for each process e.g. testing, train, and validation.

**You may use any Fuzzy Logic variant from the following list.**

1. Classical Fuzzy Logic (CF)

2. Type-1 Fuzzy Logic Systems (T1FLS)

3. Interval Type-2 Fuzzy Logic Systems (IT2FLS)

4. General Type-2 Fuzzy Logic Systems (GT2FLS):

5. Adaptive Fuzzy Systems:

6. Evolutionary Fuzzy Systems:

7. Neuro-Fuzzy Systems:

8. Fuzzy Control Systems:

9. Fuzzy Decision Support Systems (FDSS):

10. Fuzzy Expert Systems:

11. Fuzzy Cognitive Maps (FCM)

12. Fuzzy Petri Nets.

13. Fuzzy Rule-Based Systems:

14. Fuzzy Inference Systems (FIS):

15. Hybrid Fuzzy Systems

**Abstract:** We represent a comprehensive solution for fire detection using a deep learning approach, specifically employing the InceptionV3 model. The dataset, stored in Google Drive, is loaded and preprocessed using an Image-Data-Generator for data augmentation and normalization. The InceptionV3 model, pre-trained on ImageNet, is utilized as a base model, with a custom head added for fine-tuning on the fire detection task. The model is compiled with categorical cross entropy loss and the Adam optimizer. To optimize hyperparameters, a hybrid Particle Swarm Optimization (PSO) algorithm is implemented. This optimization process dynamically adjusts learning rates and the number of training epochs to enhance the model's performance. The PSO algorithm is fine-tuned to achieve the best solution in terms of validation accuracy. The trained model is then evaluated on the validation set, and performance metrics such as precision, recall, F1-score, and confusion matrices are calculated and presented. Additionally, the training and validation accuracy and loss are visualized through graphs. The code concludes with an exploration of the dataset's class distribution and a demonstration of Receiver Operating Characteristic (ROC) analysis for fire detection.

2-Detail of dataset (450 words) one paragraph

I made a dataset for spotting forest fires. It has 760 pictures of places with fires and 760 pictures of places without fires. I got these pics from Kaggle. To make things quicker, I only used 11 pics for each category during training because going through all of them takes a long time. This smaller set still helps the model learn without being too slow. The goal is to balance efficiency with having good examples for the model to learn from, making it better at spotting forest fires.

**Type-1 Fuzzy Logic Systems (T1FLS): Unveiling the Simplicity and Effectiveness**

Fuzzy Logic, a mathematical framework designed to handle uncertainty and imprecision, introduces a variety of systems, and among them, Type-1 Fuzzy Logic Systems (T1FLS) stand out as the classical and straightforward variant. T1FLS has found extensive applications in diverse fields, from control systems to decision support systems.

At its core, T1FLS operates on the principle of assigning membership degrees to elements within a universe of discourse. Unlike conventional binary logic, where an element is either a member or not, T1FLS allows for a gradual transition between membership and non-membership. Each linguistic variable in T1FLS is associated with a membership function, which defines the degree of belongingness to the set. These membership functions are often represented graphically as fuzzy sets.

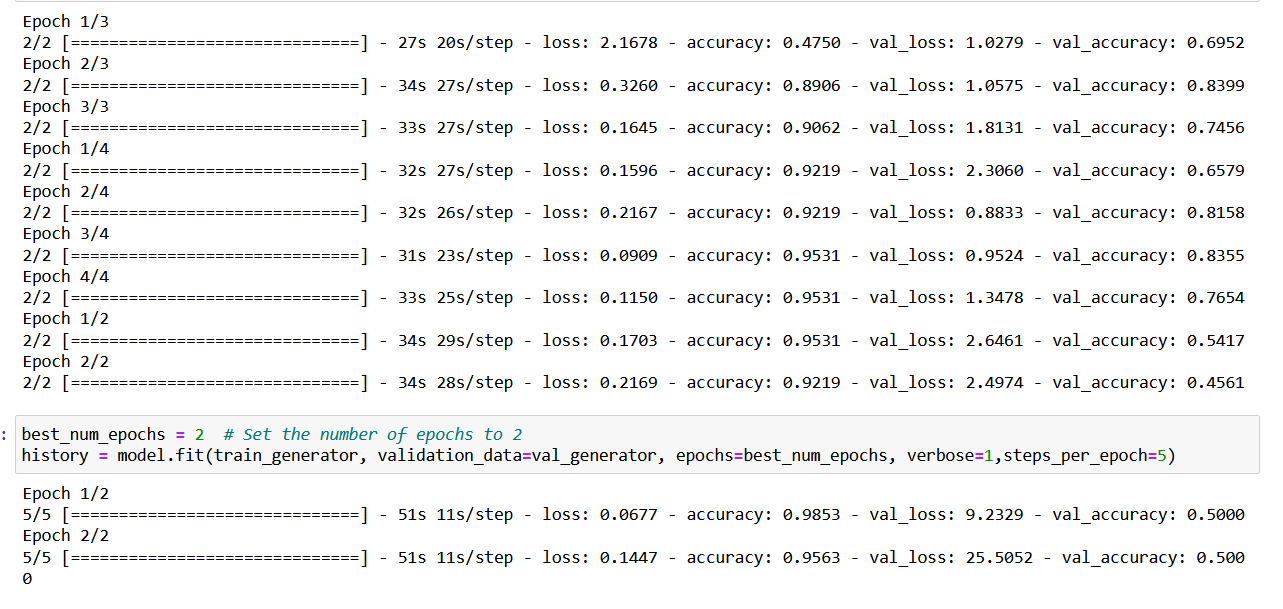
The simplicity of T1FLS lies in its clear and intuitive structure. The linguistic variables, such as 'low,' 'medium,' and 'high,' are easily interpretable, making it accessible for both experts and non-experts. The rules in T1FLS are expressed in the form of IF-THEN statements, connecting antecedents (input conditions) with consequents (output actions). For instance, an IF temperature is 'high,' THEN air conditioning is 'strong' rule could be part of a T1FLS controlling a climate system.

In practical terms, T1FLS can be implemented using rule-based systems. Each rule involves fuzzy sets, membership functions, and logical connectors to define relationships among variables. The aggregation of rules often employs methods like the max-min or max-product operations to determine the overall system response.

T1FLS has been extensively applied in control systems, where it excels in handling systems with imprecise input data or varying conditions. It's employed in areas like temperature control, speed regulation, and decision-making processes. For instance, in an adaptive cruise control system, T1FLS can adapt to different driving conditions, smoothly adjusting the speed based on fuzzy rules considering factors like distance to the leading vehicle and current speed.

Despite its simplicity, T1FLS is not without challenges. One limitation lies in its ability to handle complex and dynamic systems with high levels of uncertainty. In situations where ambiguity is prevalent, more advanced fuzzy logic variants, such as Interval Type-2 Fuzzy Logic Systems (IT2FLS) or General Type-2 Fuzzy Logic Systems (GT2FLS), might be preferred.

In conclusion, Type-1 Fuzzy Logic Systems remain a cornerstone in the world of fuzzy logic. Their simplicity and interpretability make them an attractive choice for various applications, especially in scenarios where a clear mapping between inputs and outputs is desired. As technology advances, the versatility of T1FLS is complemented by more sophisticated fuzzy logic variants, ensuring that fuzzy logic continues to evolve as a powerful tool for handling uncertainty in diverse domains.



A graph of training and validation accuracy

Description automatically generated A graph with a line and a line

Description automatically generated

A screenshot of a white sheet

Description automatically generated A blue and white graph

Description automatically generated

A blue and white graph

Description automatically generated

A graph with a line and a dot

Description automatically generated

A graph of a curve

Description automatically generated

A graph of blue rectangular bars

Description automatically generated with medium confidence

A white rectangular sign with black text

Description automatically generated

A graph of blue rectangular objects

Description automatically generated

A blue and white graph

Description automatically generated

A screenshot of a white screen

Description automatically generated

A graph of a curve

Description automatically generated