A Study on the Comparison between Fuzzy Logic and Neural Networks Systems as 2D-Geometric Shape Identifier

Reginald Geoffrey L. Bayeta IV
Electronics and Communication Department
De La Salle University - Manila
Manila, Philippines
reginald_geoffrey_bayetaiv@dlsu.edu.ph

Teofilo M. Contreras Jr.
Electronics and Communication Department
De La Salle University - Manila
Manila, Philippines
teofilo_contreras@dlsu.edu.ph

***Abstract*— Computer vision is a growing field of computer science that aims to extract useful information from images. In computer vision applications, the ability to discern shapes in pictures is frequently required. The recognition and classification of geometric shape is useful in variety of field such as robotics. In this paper, Convolutional Neural Network (CNN) and fuzzy logic system is developed to successfully classify 2-D geometric shapes. The dataset used in this study for the training phase of the program consists of 16,000 images of shapes of triangle, circle, star and square. A 4 layered sequential model was developed for the CNN using the TensorFlow library. Meanwhile, Fuzzy Vision Computer Toolbox was utilized for the development of the fuzzy shape classifier. The study aims to compare and analyze the accuracy and precision of Fuzzy Logic and Convolutional Neural Network in categorizing four image shapes in a given test. The two different methods are compared in terms of accuracy and Root Mean Square Error (RMSE). For the result of the CNN, the percent accuracy is 97.6 % and the RMSE is 0.076. On the other hand, the percent accuracy and RMSE of fuzzy logic system is 97.6 % and 0.484, respectively. The result of the experimentation indicates that CNN is more accurate in predicting image shapes compared to the fuzzy logic system.**

Keywords—fuzzy logic, neural networks, computer vision, pattern recognition, shape classification

I. INTRODUCTION

A. Background of the Study

Neural networks were first created as very simplified models of the human nervous system that could learn, generalize, and abstract information. It is simply a network of numerous simple highly interconnected processing units (also known as nodes) that executes in parallel at the same time. The efficiency, fault-tolerance, and self-adaptability of neural networks enable them to adapt to changing environments [1]. Since it is hard to understand how neural networks obtain their results, they are regarded as a kind of black-box model [2]. However, they are trained to utilize the actual data from real experiments in their learning process. Nowadays, neural networks and machine learning are often utilized in the field of algorithm creation and development that allows computers to learn from data collected from sensors or databases [3].

In theory, a neural can be utilized for any application that involves function computation. Nevertheless, neural networks excel in the application of classification, regression, and simulation that are tolerant of imprecision [4]. Neural networks are capable of simulating exceedingly complicated functions, particularly nonlinear problems because of their extensive modeling techniques.

Fuzzy logic is a computation and reasoning system in which the objects of computation and reasoning are classes with unsharp borders [5]. As a result, the truth value in fuzzy logic is a matter of degree, including degrees, or is permitted to be. The goal of fuzzy logic is to make computers think like people. Fuzzy logic understands that its nature is different from randomness and is used to cope with the ambiguity inherent in human thought and natural language.

Fuzzy logic is applied to machines to make them understand and respond to abstract human notions such as hot, cold, large, small, and so on. Unlike neural networks, Fuzzy logic offers a verbal rather than a mathematical advantage in modeling complicated and nonlinear situations when employing natural language processing [6].

Before using either fuzzy or neural networks, engineers must first grasp its limitations in order to get the most out of this crucial design process and employ it when it is advantageous or seek an alternative when it is not [7]. Furthermore, using a fuzzy system and neural network also requires comprehensive expert knowledge in order to effectively simulate the sets of rules.

The study of how computers and machines can examine the environment, learn to detect features of interest, and make good and logical conclusions about forms is known as pattern recognition. Pattern recognition is used in a wide range of applications, including recognition of printed and handwritten text, speech recognition, and human face recognition, etc. [8] [9].

Before the convolutional neural networks and fuzzy logic were discovered, a comprehensive statistical methodology has been used in this area of science [10].

B. Problem Statement

Humans can easily classify simple shapes in images or drawings by counting the number of their edges. The computer can also learn to classify simple shapes through the use of machine learning and a sufficient dataset of the shapes. In this study, a neural network and fuzzy logic system is developed to successfully classify 2-D geometric shapes. The study aims to determine the most accurate and precise system.

C. *Objectives of the Study*

- To develop a neural network system that can classify four simple shapes (circle, square, star, and triangle) in a dataset of geometric shapes.
- To develop a fuzzy logic system that can classify four simple shapes in a dataset of geometric shapes.
- To compare and analyze the accuracy of Fuzzy Logic and Neural Network in classifying four shapes in a given data set.

II. Procedure

A. *Dataset*

The dataset used in this study consists of 16,000 images of shapes of triangle, circle, star, and square. This dataset will then be feed to the training phase of the program. After the training phase, the system is expected to classify the test input images with high accuracy.

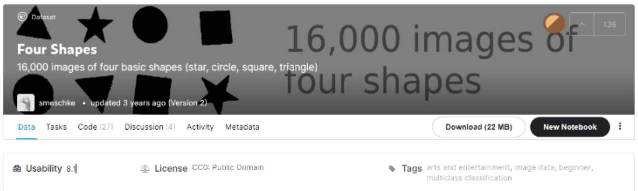

Fig 1. Four Shapes Dataset [11]

B. *Design of Neural Network Model*

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 198, 198, 32)      320
_________________________________________________________________
activation (Activation)      (None, 198, 198, 32)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 99, 99, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 97, 97, 32)        9248
_________________________________________________________________
activation_1 (Activation)    (None, 97, 97, 32)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 48, 48, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 73728)             0
_________________________________________________________________
dense (Dense)                (None, 32)                2359328
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 132
_________________________________________________________________
activation_2 (Activation)    (None, 4)                 0
=================================================================
Total params: 2,369,028
Trainable params: 2,369,028
Non-trainable params: 0
```

Fig 2. Summary of the Neural Network Model

A sequential model with 4 layers, excluding the input layer, was used by using the TensorFlow library. The summary of the trained network model is shown in Figure 2. The *None* variable in the output shape refers to the batch size in the model training process.

C. *Fuzzy System Classifier*

For the classification of the images using fuzzy logic, Fuzzy Computer Vision Toolbox was used which contains functions for image acquisition, image pre-processing, feature extraction, and image classification for developing the fuzzy logic system with computer vision [12]. In the toolbox, there are two available methods for image classification which are CRISP and Fuzzy Qualitative Rank Classifier (FQRC). Between the two methods, FQRC was chosen. FQRC is presented to overcome the constraints of Crisp classification and conventional Fuzzy Inference System (FIS) in a computer vision problem.

Furthermore, FQRC can provide an answer in terms of the ranking result with different confidence values annotated. This interpretation is more similar to how humans think in uncertain situations. FQRC consists of four main stages:

1. Pre-processing
2. Learning model
3. Inference
4. Ranking interpretation

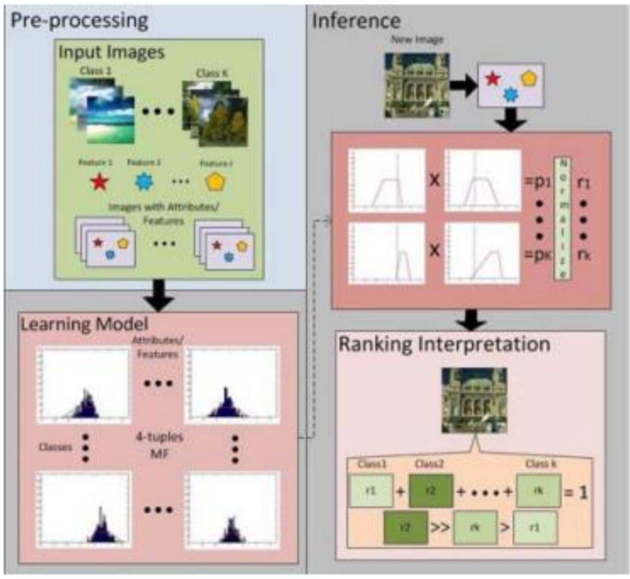

Fig 3. Overall steps of FQRC [13]

It is critical that the characteristics extracted from the training image be recognizable even when the picture size , orientation, and lighting are changed in order to conduct reliable to perform reliable recognition. In this paper, the feature descriptor used in the FQRC is Scale-space extrema detection (SIFT).

III. Results And Discussion

The code used to obtain the results of this paper is done by using Python v3.8.10, OpenCV v4.5.2, and TensorFlow v2.30, and sci-kit learning libraries.

A. *Trained Neural Network Model*

The sequential model was compiled using categorical cross-entropy loss function, *adam* optimizer, and *accuracy* as its main metrics. For the training process, a batch size of 256 was used for the duration of 5 epochs. Results of the training process are reflected in Figure 4. For the training input data, 13969 images were used, while 1000 images were set for the testing data.

```
Epoch 1/5
 1/31 [..............................] - ETA: 0s - loss: 1.3699 - accuracy: 0.2812WARNING:tensorflow:From C:\ProgramData\Anaconda3\en
op (from tensorflow.python.eager.profiler) is deprecated and will be removed after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
31/31 [==============================] - 316s 10s/step - loss: 1.4060 - accuracy: 0.8012 - val_loss: 0.0286 - val_accuracy: 0.9973
Epoch 2/5
31/31 [==============================] - 316s 10s/step - loss: 0.0152 - accuracy: 0.9972 - val_loss: 0.0101 - val_accuracy: 0.9989
Epoch 3/5
31/31 [==============================] - 324s 10s/step - loss: 0.0071 - accuracy: 0.9987 - val_loss: 0.0056 - val_accuracy: 0.9989
Epoch 4/5
31/31 [==============================] - 310s 10s/step - loss: 0.0030 - accuracy: 0.9994 - val_loss: 0.0026 - val_accuracy: 0.9992
Epoch 5/5
31/31 [==============================] - 305s 10s/step - loss: 0.0017 - accuracy: 0.9997 - val_loss: 0.0021 - val_accuracy: 1.0000
```

Fig 4. Training the model

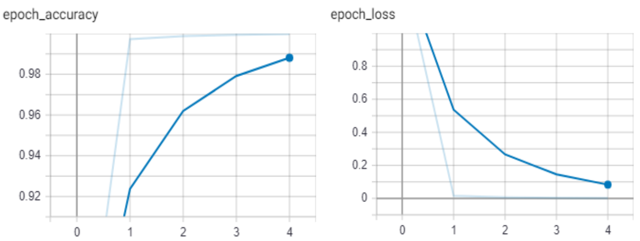

Fig 5. Training accuracy and loss graph

Figure 5 shows the training accuracy and the loss graph of the trained model. It follows the ideal case where the accuracy is approaching the max value of 1, and loss approaching 0, for each epoch. After evaluating the model, it outputs an accuracy level of 99.43%, which is a very good result. A portion of the results is highlighted in Figure 6.

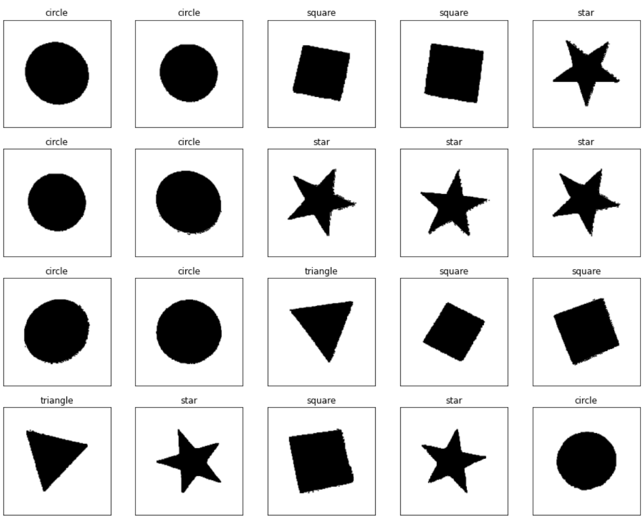

Fig 6. Sample predictions from the trained model

B. *Trained Fuzzy System Classifier*

The FCVT library's *Image_Classification* function provides a method to automatically generate membership function which can be visualized in either plain graph or histogram form. The histogram membership generation of the trained fuzzy system is shown in Figure 7. For the fuzzy prediction of the shape, it takes the maximum of the degree of memberships. In Figure 8, a sample instance of a fuzzy prediction was shown, and by taking the max, it would conclude that the sample is predicted to be a *circle*.

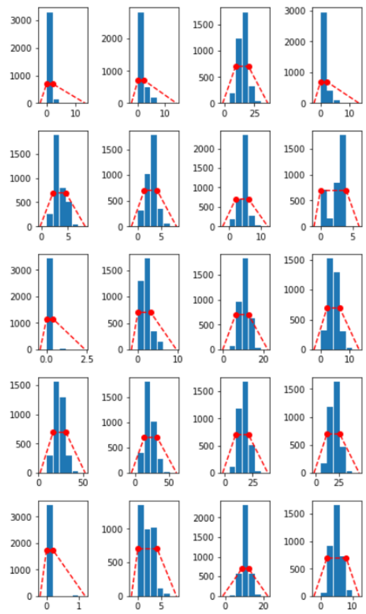

Fig 7. Fuzzy Membership Generation from Histogram

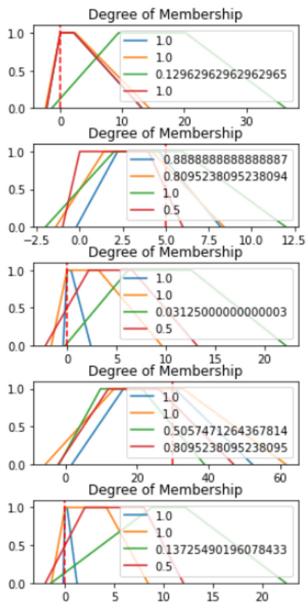

Fig 8. Sample instance degree of membership prediction

C. *Classification Results of CNN and Fuzzy System*

After the training of both models, the verification test data set used is an array of size 1000, that consists of 250 images per category. To get the percent accuracy of each classifier, their prediction array was compared to the verification test data set. For the CNN, it outputs a 97.6% accuracy and 0.076 RMSE, which indicates a good result as high value of accuracy and low RMSE indicates a better fit of the model. Similarly, the fuzzy system classified 84.7% of data to be valid results and 0.484 which is slightly worse than CNN. From this, it can be concluded that in identifying a pattern of a shape, CNN performs better than fuzzy logic systems.

TABLE I. CLASSIFICATION RESULTS

Classifier	**Shapes**				**% accuracy**	**RMSE**
	Circle	***Square***	***Square***	***Triangle***		
CNN	227	273	249	251	97.6%	0.076
Fuzzy System	200	285	250	265	84.7%	0.484

D. *Storing Results to Data Frame*

The final results are stored into a data frame which is then exported to a *.csv* file in order not to repeat the process of training the model again in the future. As seen in Figure 9 and complementary of the results to Figure 8, it shows that the fuzzy and CNN predictions were similar but there's a huge difference when it comes to their confidence level.

	fuzzy_predict	fuzzy_confidence	cnn_predict	cnn_confidence	actual_description	fuzzy_accuracy	cnn_accuracy
0	circle	0.330067	circle	0.999994	circle	1	1
1	circle	0.330067	circle	0.999995	circle	1	1
2	circle	0.323188	circle	0.999994	circle	1	1
3	circle	0.331852	circle	0.999993	circle	1	1
4	circle	0.320238	circle	0.999994	circle	1	1
5	circle	0.313899	circle	0.999991	circle	1	1
6	circle	0.331584	circle	0.999996	circle	1	1
7	circle	0.325449	circle	0.999997	circle	1	1
8	circle	0.309622	circle	0.999996	circle	1	1
9	circle	0.318097	circle	0.999996	circle	1	1
10	circle	0.315984	circle	0.999994	circle	1	1
11	circle	0.315482	circle	0.999996	circle	1	1
12	square	0.332500	circle	0.999988	circle	0	1
13	circle	0.315984	circle	0.999991	circle	1	1
14	circle	0.317366	circle	0.999987	circle	1	1
15	square	0.309682	circle	0.999997	circle	0	1
16	circle	0.314411	circle	0.999990	circle	1	1
17	circle	0.308950	circle	0.999997	circle	1	1
18	circle	0.306784	circle	0.999996	circle	1	1
19	circle	0.316234	circle	0.999998	circle	1	1

Fig 9. Sample entries to the results data frame

IV. CONCLUSION

In this paper, a comparison of performance between fuzzy logic and neural network systems as 2D-geometric shape identifier is discussed. Implementation of trained models was presented using Python v3.8.10, OpenCV v4.5.2, and TensorFlow v2.30, and sci-kit learning libraries. Based on the results, it showed that CNN performed better as a shape categorizer as it outputted 97.6% accuracy compared to the 84.7% performance accuracy of the fuzzy system. For future work, an addition of colored shapes is recommended to be fed to the model, and real-time testing should be conducted to test which system performs better in a live testing environment, which is the most common application of shape identifier especially in various complex computer vision systems.

ACKNOWLEDGMENT

We would like to thank Dr. Ira C. Valenzuela and Dr. Carlo Noel E. Ochotorena for their teachings and guidance in our CpE Elective 1 Laboratory (LBYCPF3) and Lecture (CPECOG1) courses.

REFERENCES

[1] M. Firdaus, S. E. Pratiwi, D. Kowanda, and A. Kowanda, "Literature review on artificial neural networks techniques application for stock market prediction and as decision support tools," in *Proceedings of the 3rd International Conference on Informatics and Computing, ICIC 2018*, Oct. 2018, pp. 1–4, doi: 10.1109/IAC.2018.8780437.

[2] B. Robbins, "MACHINE LEARNING: How Black is This Beautiful Black Box ," *Towards Data Science*, Dec. 01, 2016. https://towardsdatascience.com/machine-learning-how-black-is-this-black-box-f11e4031fdf (accessed May 02, 2021).

[3] N. L. W. Keijsers, "Neural Networks," in *Encyclopedia of Movement Disorders*, Elsevier Inc., 2010, pp. 257–259.

[4] M. Y. Kiang, "Neural Networks," in *Encyclopedia of Information Systems*, Elsevier, 2003, pp. 303–315.

[5] L. A. Zadeh, "Fuzzy logic - A personal perspective," *Fuzzy Sets Syst.*, vol. 281, pp. 4–20, Dec. 2015, doi: 10.1016/j.fss.2015.05.009.

[6] A. M. Ibrahim, "Embedded systems," in *Fuzzy Logic for Embedded Systems Applications*, Elsevier, 2004, pp. 1–21.

[7] A. M. Ibrahim, "Critique of fuzzy logic," in *Fuzzy Logic for Embedded Systems Applications*, Elsevier, 2004, pp. 99–109.

[8] R. Plamondon, "A Fuzzy-Syntactic Approach to Allograph Modeling for Cursive Script Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 7, pp. 702–712, 1995, doi: 10.1109/34.391412.

[9] M. Y. Chen, A. Kundu, and J. Zhou, "OffLine Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 481–496, 1994, doi: 10.1109/34.291449.

[10] T. H. Kim, "Pattern recognition using artificial neural network: A review," in *Communications in Computer and Information Science*, 2010, vol. 76 CCIS, pp. 138–148, doi: 10.1007/978-3-642-13365-7_14.

[11] "Four Shapes | Kaggle." https://www.kaggle.com/smeschke/four-shapes (accessed May 02, 2021).

[12] "FuzzyComputerVision: A toolbox for Fuzzy Computer Vision." https://github.com/cs-chan/FuzzyComputerVision (accessed May 29, 2021).

[13] C. H. Lim, A. Risnumawan, and C. S. Chan, "A Scene Image is Nonmutually Exclusive-A Fuzzy Qualitative Scene Understanding," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 6, pp. 1541–1556, Dec. 2014, doi: 10.1109/TFUZZ.2014.2298233.