STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–17.
Published online in International Academic Press (www.IAPress.org)

NEUTRO-FUP: Neutrosophic Fuzzy Pooling

Chaymae Rajae Fillah [1], Karim El Moutaouakil [1], Mouhcine Kahlaoui [1], Abderazzak Mouiha [2,*]

[1] *Laboratory of Mathematics and Data Science, Faculty of Polydisciplinary Studies of Taza, Taza, Morocco*
[2] *Euromed Research Center, Euromed University of Fez (UEMF), Fez, Morocco*

Abstract Convolutional Neural Networks (CNNs) are widely used for tasks such as image segmentation, object recognition, and classification, with pooling operations playing a key role in reducing computational complexity and model parameters. While traditional and fuzzy-based pooling methods have been extensively studied, recent approaches have introduced intuitionistic fuzzy pooling to address local imprecision in feature maps. However, this method relies on the assumption that the indeterminacy is simply the complement of membership and non-membership degrees—an assumption that can be inaccurate in complex environments.
To overcome this limitation, we propose a novel pooling operation based on Neutrosophic Fuzzy Sets (NFSs), which explicitly incorporates a degree of indeterminacy. Our method, called Neutrosophic Pooling, operates in four stages: bi-fuzzification using membership, non-membership, and indeterminacy maps; a first aggregation converting Neutrosophic Fuzzy Set NFS into a classical fuzzy set; a second aggregation using a sum operator; and finally, defuzzification through a max operation. This new pooling layer can be seamlessly integrated into CNN architectures, replacing standard, fuzzy, or intuitionistic pooling layers.
Experimental evaluations across several benchmark datasets demonstrate that the proposed NFS-based pooling significantly enhances classification performance, especially in uncertain or noisy environments, outperforming state-of-the-art pooling methods.

Keywords Convolutional Neural Networks (CNNs), Neutrosophic Fuzzy Sets (NFS), Neutrosophic Pooling, Image Classification, Innovation

DOI: 10.19139/soic-2310-5070-3466

1. Introduction

Neutrosophic fuzzy sets are an extension of fuzzy sets and intuitionistic fuzzy sets (IFSs), introduced by Smarandache in 1998[1] then the concept of a single-valued neutrosophic set (SVNS) presented by wang[2] until be oriented for real scientific and engineering applications . Fuzzy sets were first introduced by L. Zadeh in 1965 to describe the degree of belonging of each element to a group or class by a value called membership to any non-deterministic event, where this value is less than or equal to one. This concept addresses the vagueness in classical (crisp) sets, where an element either fully belongs to a class (1) or does not belong at all (0).

In 1983, K. Atanassov[3] proposed the intuitionistic fuzzy set (IFS) as a generalization of fuzzy sets. IFS consider not only the membership of an element to a class but also the non-membership, addressing the fact that an element could not belong to a class, where the sum of both membership and non-membership less or equal 1. Intuitionistic fuzzy sets can only handle incomplete information no the indeterminate information and inconsistent information[4].

Neutrosophic sets extend these ideas further by handling information that encompasses not only uncertainty and vagueness but also indeterminacy and inconsistent information. Neutrosophic sets comes with three functions T,

*Correspondence to: Abderazzak Mouiha (Email: a.mouiha@ueuromed.org). Euromed Research Center, Euromed University of Fez (UEMF), Fez, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

F and I which indicate truthiness, Falsity and indeterminacy respectively where all the function are independent [5] and belong to $]0^-, 1^+[$, and their summation between 0 and 3. In fuzzy sets, the membership function handles vagueness, and in IFS, there is an additional function gives attention to non-membership, addressing uncertainty. Neutrosophic sets study cases where an element might partially belong to a set, addressing the problem of indeterminacy along with the issues handled by fuzzy sets and IFSs, Furthermore NFSs handles uncertainty better than fuzzy logic where they are case when fuzzy logic couldn't perform will with uncertainty problem [6] .

In neutrosophic sets, truth, indeterminacy, and falsity are independent, which is particularly important in situations such as information fusion, where data from different sensors are combined, whereas in intuitionistic sets, they are dependent. A Neutrosophic Fuzzy Set (NFS) becomes an IFS when the sum of truth, indeterminacy, and falsity is less than or equal to 1.

SVNS provides a more practical and efficient approach for applying neutrosophic logic to real-world problems, making it preferable in many scientific and engineering contexts.

Now, in our work we seek to applied the concept mathematics of neutrosophic fuzzy set into field of deep learning more specially in CNNs.

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing structured grid data, such as images. They leverage convolutional layers to automatically extract hierarchical features from the input data, allowing the model to learn spatial hierarchies effectively. CNNs consist of several key components: convolutional layers that apply filters to capture local patterns, pooling layers that decrease the size of activation maps to reduce dimensionality and while retaining important information, and fully connected layers that interpret the features for classification or regression tasks. This architecture makes CNNs particularly powerful for tasks such as image recognition, object detection, and segmentation, achieving state-of-the-art performance across various benchmarks. Their ability to learn directly from raw pixel values, combined with techniques like data augmentation and transfer learning, has revolutionized computer vision and enabled significant advancements in related fields.

The main contributions of this paper can be summarized as follows: We propose a novel pooling method based on neutrosophic fuzzy sets, termed Neutrosophic Fuzzy Pooling, for CNN architectures. We introduce a mathematical formulation of pooling using neutrosophic logic to better handle uncertainty and indeterminacy in feature maps. Furthermore, we conduct a comparative study between the proposed method and standard pooling techniques, including average pooling and fuzzy pooling. In addition, we validate the effectiveness of the proposed approach through experiments on benchmark datasets such as MNIST, CIFAR10, and STL10. Finally, we demonstrate that the proposed method achieves improved performance in terms of accuracy and loss while maintaining robustness. In this work we will focus on pooling. Where, in pooling layer we reduce the spatial dimension using neutrosophic fuzzy sets

Our work is outlined as follows: In the first section, we begin with an introduction that provides background information on our study. Next, we explore previous pooling methods in related work and discuss the important concepts in mathematics related to neutrosophic fuzzy sets , including the mathematical equations used in pooling in neutrosophic fuzzy sets section. Following that, we describe the datasets used in this study, namely MNIST, CIFAR10, and STL10, in a dedicated Datasets section. Then, we present experimental results and analysis on different datasets when we present experimental results and analysis on different datasets, concluding with a future work and summary of our findings.

2. Related work

The first pooling introduced and used in CNN architecture was Average pooling, as implemented in LeNet by LeCun. later, Max pooling became popular following its successful use in AlexNet. Since then, numerous pooling approaches have been introduced including methods that combine both average and max pooling, such as Mix-pooling [7] and gated mix-average pooling. or approaches incorporating fuzzy logic.

Images typically exhibit strong correlations between neighboring pixels. Traditional pooling methods like mixed max-average pooling and gated mix-average pooling often overlook these relationships by handling each pooling

region separately. To address this limitation, Dynamic Correlation Pooling was proposed, enabling the use of inter-pixel correlation during pooling [8, 9].

A wide range of soft pooling techniques have been developed to provide smoother and more adaptable alternatives to traditional pooling operations. These methods aim to retain more nuanced feature information by using continuous, differentiable functions rather than hard selection mechanisms like max or average pooling. Notable examples include Log-Sum-Exp (LSE) pooling[10], which approximates the maximum through a smooth logarithmic function; Polynomial pooling [11], which uses polynomial transformations to capture nonlinear relationships; and Learned-Norm pooling [12], which learns an optimal norm parameter during training. Other variants such as l_p pooling [13] generalize between max and average pooling depending on the value of p, while α-Integration (α-I) pooling [14] introduces a parameterized integration framework to unify different pooling behaviors.

Additional soft pooling strategies include Rank-based pooling [15], which considers the order of feature values; Dynamic pooling [16], which adapts pooling based on feature importance; and Smooth-Maximum pooling [17], which uses smooth approximations to the max operator. Furthermore, Soft pooling [18] aims to preserve detailed spatial information, while Maxfun pooling [19] selects features based on a learned maximization function. Lastly, Ordinal pooling [20] introduces ranking-based weights to better capture feature significance across pooling regions. These approaches collectively contribute to more expressive and flexible neural network architectures, particularly beneficial in domains like medical imaging and fine-grained visual recognition where preserving subtle feature differences is critical.

To address overfitting in neural networks, several advanced pooling strategies have been proposed. One such approach is Mixed Pooling [7] and Hybrid Pooling [21], which aim to reduce overfitting by introducing randomness into the pooling operation, this method combine both max pooling and average pooling by randomly selecting either operation during training. The selection made during training is also retained during testing[9].

Another technique, Stochastic Pooling [22], addresses two key limitations: the down-weight effect of average pooling and the overfitting tendency of max pooling. Instead of deterministically selecting a maximum or average value, stochastic pooling randomly chooses an activation within each pooling region based on a probability distribution derived from the activations themselves. This probabilistic selection helps avoid dominant activations from skewing the learning process, which is especially important when training data is limited. To further tackle the issue of dominant activations and overfitting in limited data scenarios, Rank-based Stochastic Pooling [15] was introduced. This method selects activations based on their rank, thereby offering a more stable and generalizable pooling behavior during training. While stochastic pooling selects a single activation from each region, Max-Pooling Dropout [23] introduces an extension by randomly sampling a subset of activations from each pooling region. Pooling (typically max) is then applied to this sampled subset rather than the entire region [9]. Both stochastic pooling and max-pooling dropout rely on randomness and sample activations based on multinomial distributions. All these methods introduce some form of stochasticity during the pooling stage to enhance generalization. However, other strategies introduce randomness not in the pooling values, but in the spatial sampling process. For example, S3 Pooling [24] and Fractional Max Pooling [25] incorporate stochasticity into the spatial location selection process within the pooling window, further contributing to improved generalization and robustness.

For images containing objects with varying poses, *part-based pooling* [26] is effective due to its ability to detect and pool features from diverse object parts, which are then concatenated to form a comprehensive image representation [9]. In scenarios involving rotated objects, *Concentric Circle Pooling (CCP)* [27] and *Polycentric Circle Pooling* [28] offer robust solutions by addressing rotation variance in CNNs.

While many pooling methods focus on capturing global spatial structure, *Geometric l_p-Norm pooling* [26] targets local structural information, providing a finer-grained representation [9].

Pooling can also model interactions between different feature maps and regions. Techniques like *Improved Bilinear Pooling* [29] and *Second-Order Pooling* [30] preserve pairwise feature correlations for richer representations.

Additionally, *self-attentive pooling* [31] surpasses traditional local pooling by capturing complex, non-local dependencies among features across activation maps.

The latest pooling were in using the concept of fuzzy first time in 2022 in Fuzzy pooling, later mixed max and fuzzy pooling that show more accurate result later Intuitionistic pooling [32] is presented and show outperform result by taking into consideration in real world the sum of membership and non membership is not necessarily equal to one, then the hesitation function as added to deal with this case.

Overall, the evolution of pooling methods reflects a trade-off between computational efficiency, feature extraction, and information retention. As deep learning architectures become increasingly complex and data large.

3. Neutrosophic Fuzzy Sets

In this section, we outline some fundamental definitions that are necessary to understand the research context and the technical terms used throughout the paper.

Definition[1] Let the universal set X of discourse be fixed. A neutrosophic fuzzy sets (NFSs) A in X is defined as an object of the form A = $\{< x, T_A(x), I_A(x), F_A(x) > | x \in X\}$ where the functions $T_A : X \rightarrow]0^-, 1^+[$, $I_A : X \rightarrow]0^-, 1^+[$ and $F_A : X \rightarrow]0^-, 1^+[$ define truth, indeterminacy, and falsity membership functions of the element x$\in X$ respectively, and for every x$\in X$ in A, $0^- \leq T_A(x) + I_A(x) + F_A(x) \leq 3^+$.

Now, we will define SVNS equation, as SVNS provides more practical and efficient approach for applying neutrosophic logic to real-world problems.

Definition[1] Let the universal set X of discourse(object) be fixed. A single-valued neutrosophic sets (SNFs) A in X is defined as an object of the form A = $\{< x, T_A(x), I_A(x), F_A(x) > | x \in X\}$ where the functions $T_A : X \rightarrow]0, 1[$, $I_A : X \rightarrow]0, 1[$ and $F_A : X \rightarrow]0, 1[$ define truth, indeterminacy, and falsity membership functions of the element x$\in X$ respectively, and for every x$\in X$ in A, $0 \leq T_A(x) + I_A(x) + F_A(x) \leq 3$.

Definition A triangular single-valued neutrosophic number A=$< (a, b, c); \mu_A, \nu_A, \rho_A >$ is a special type of neutrosophic set on X is denoted as whose truth, indeterminacy, and falsity membership functions are illustrated 1, which are defined as:

$$< T_A(x), I_A(x), F_A(x) >= \begin{cases} < (\frac{x-a}{b-a})\mu_A, \frac{b-x+(x-a)\nu_A}{b-a}, \frac{b-x+(x-a)\rho_A}{b-a} > & \text{if} \quad a \leq x \leq b \\ < (\frac{c-x}{c-a})\mu_A, \frac{x-b+(c-x)\nu_A}{c-a}, \frac{x-b+(c-x)\rho_A}{c-a} > & \text{if} \quad b \leq x \leq c \\ < 0, 1, 1 > & otherwise \end{cases} \tag{1}$$

$0 \leq T_A(x) + I_A(x) + F_A(x) \leq 3$.

Definition [33] A triangular single-valued neutrosophic number of type 1 is a specific kind of neutrosophic set on the real line, defined as A = $(a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3)$. Where the truth, indeterminacy, and falsity membership functions are illustrated in Figure2 and defined as follows:

$$T(x) = \begin{cases} \frac{p^n_{i,j} - a_1}{b_1 - a_1} & \text{if} \quad a_1 \leq p^n_{i,j} \leq b_1 \\ \frac{c_1 - p^n_{i,j}}{c_1 - b_1} & \text{if} \quad b_1 \leq p^n_{i,j} \leq c_1 \\ 0 & Otherwise \end{cases} \tag{2}$$

$$I(x) = \begin{cases} \frac{b_2 - p^n_{i,j}}{b_2 - a_2} & \text{if} \quad a_2 \leq p^n_{i,j} \leq b_2 \\ \frac{p^n_{i,j} - b_2}{c_2 - b_2} & \text{if} \quad b_2 \leq p^n_{i,j} \leq c_2 \\ 1 & Otherwise \end{cases} \tag{3}$$

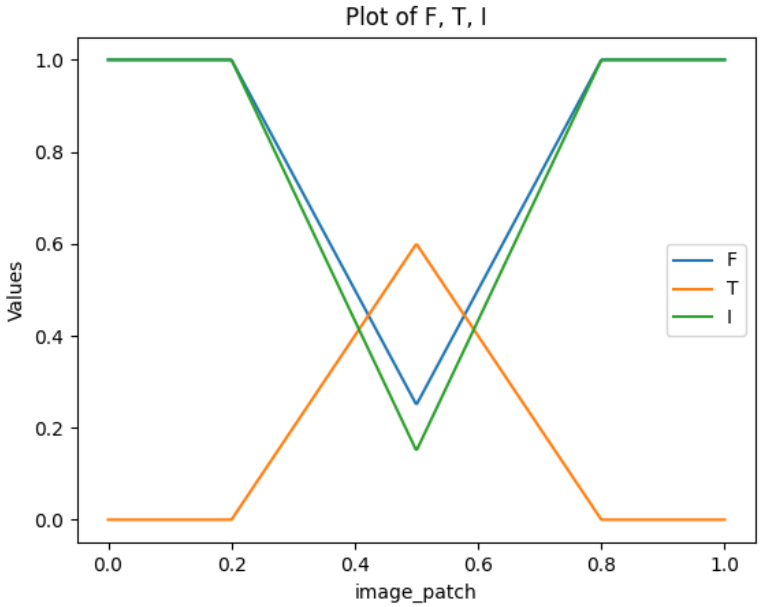

Figure 1. Graph of single-valued triangular neutrosophic number.

$$F(x) = \begin{cases} \frac{b_3 - p^n_{i,j}}{b_3 - a_3} & \text{if} \quad a_3 \leq p^n_{i,j} \leq b_3 \\ \frac{p^n_{i,j} - b_3}{c_3 - b_3} & \text{if} \quad b_3 \leq p^n_{i,j} \leq c_3 \\ 1 & Otherwise \end{cases} \tag{4}$$

4. Neutrosophic pooling method

In this section, we present and define our pooling method, which can be regarded as an extension of intuitionistic fuzzy pooling. A single-valued triangular neutrosophic number is a special type of neutrosophic set whose truth, indeterminacy, and falsity membership functions, respectively defined by:

$$T_1(x) = \begin{cases} 1 & \text{if} \quad p^n_{i,j} < a_1 \\ \frac{b_1 - p^n_{i,j}}{b_1 - a_1} & \text{if} \quad a_1 \leq p^n_{i,j} \leq b_1 \\ 0 & \text{if} \quad p^n_{i,j} > b_1 \end{cases} \tag{5}$$

$$I_1(x) = \begin{cases} 0 & \text{if} \quad p^n_{i,j} < a'_1 \\ \frac{p^n_{i,j} - a'_1}{b'_1 - a'_1} & \text{if} \quad a'_1 \leq p^n_{i,j} \leq b'_1 \\ 1 & \text{if} \quad p^n_{i,j} > b'_1 \end{cases} \tag{6}$$

$$F_1(x) = \begin{cases} 0 & \text{if} \quad p^n_{i,j} < a''_1 \\ \frac{p^n_{i,j} - a''_1}{b''_1 - a''_1} & \text{if} \quad a''_1 \leq p^n_{i,j} \leq b''_1 \\ 1 & \text{if} \quad p^n_{i,j} > b''_1 \end{cases} \tag{7}$$

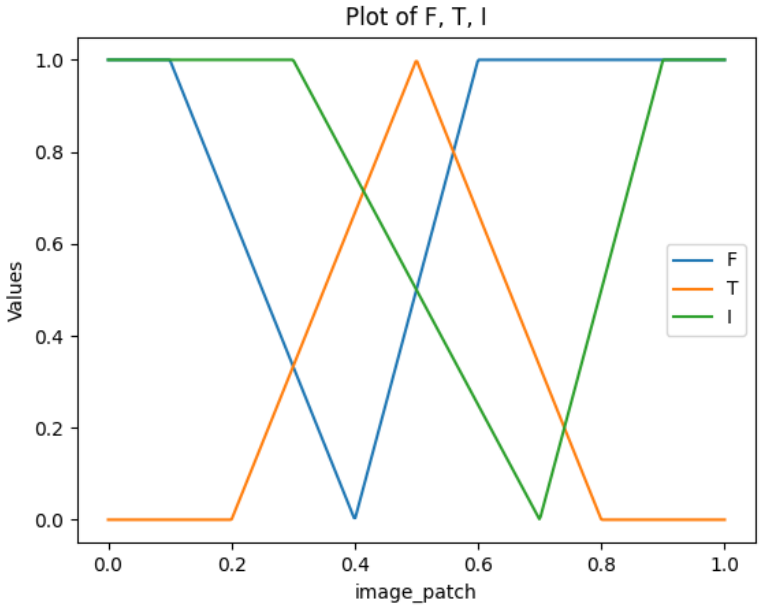

Figure 2. Graph of single-valued triangular neutrosophic number of type 1.

$$T_2(x) = \begin{cases} \frac{p^n_{i,j}-a_2}{b_2-a_2} & \text{if} \quad a_2 \leq p^n_{i,j} \leq b_2 \\ \frac{c_2-p^n_{i,j}}{c_2-b_2} & \text{if} \quad b_2 \leq p^n_{i,j} \leq c_2 \\ 0 & Otherwise \end{cases} \tag{8}$$

$$I_2(x) = \begin{cases} \frac{b'_2-p^n_{i,j}}{b'_2-a'_2} & \text{if} \quad a'_2 \leq p^n_{i,j} \leq b'_2 \\ \frac{p^n_{i,j}-b'_2}{c'_2-b'_2} & \text{if} \quad b'_2 \leq p^n_{i,j} \leq c'_2 \\ 1 & Otherwise \end{cases} \tag{9}$$

$$F_2(x) = \begin{cases} \frac{b''_2-p^n_{i,j}}{b''_2-a''_2} & \text{if} \quad a''_2 \leq p^n_{i,j} \leq b''_2 \\ \frac{p^n_{i,j}-b''_2}{c''_2-b''_2} & \text{if} \quad b''_2 \leq p^n_{i,j} \leq c''_2 \\ 1 & Otherwise \end{cases} \tag{10}$$

$$T_3(x) = \begin{cases} 0 & \text{if} \quad p^n_{i,j} < a_3 \\ \frac{p^n_{i,j}-a}{b-a} & \text{if} \quad a_3 \leq p^n_{i,j} \leq b_3 \\ 1 & \text{if} \quad p^n_{i,j} > b_3 \end{cases} \tag{11}$$

$$I_3(x) = \begin{cases} 1 & \text{if} \quad p^n_{i,j} < a'_3 \\ \frac{b'-p^n_{i,j}}{b'-a'} & \text{if} \quad a'_3 \leq p^n_{i,j} \leq b'_3 \\ 0 & \text{if} \quad p^n_{i,j} > b'_3 \end{cases} \tag{12}$$

$$F_3(x) = \begin{cases} 1 & \text{if} \quad p^n_{i,j} < a''_3 \\ \frac{b''-p^n_{i,j}}{b''-a''} & \text{if} \quad a''_3 \leq p^n_{i,j} \leq b''_3 \\ 0 & \text{if} \quad p^n_{i,j} > b''_3 \end{cases} \tag{13}$$

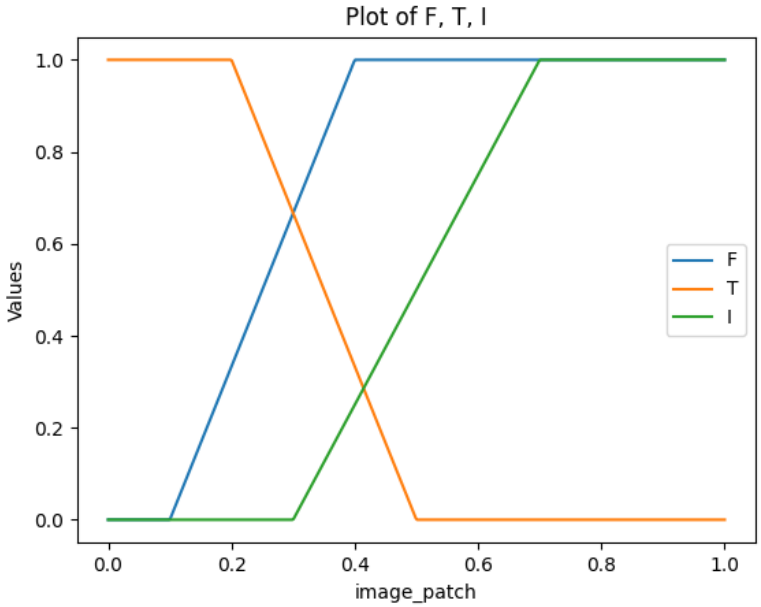

Figure 3. Graphical Representation of a Single-Valued Triangular Neutrosophic Number (T_1, I_1, F_1).

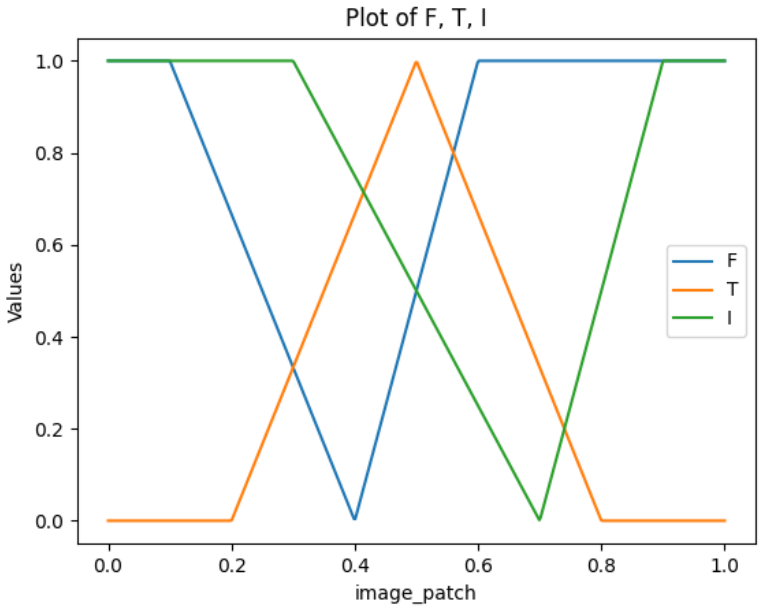

Figure 4. Graphical Representation of a Single-Valued Triangular Neutrosophic Number (T_2, I_2, F_2)

Where $0 \leq a_1 \leq a_2 \leq b_1 \leq b_2 \leq a_3 \leq c_2 \leq b_3$ defines the ordering of the coefficients, which can apply to both the prime and second prime coefficients.
Figures 3, 4, and 5 illustrate the graphical representations of three single-valued triangular neutrosophic numbers. Specifically, they correspond to (T_1, I_1, F_1), (T_2, I_2, F_2), and (T_3, I_3, F_3), respectively. Each figure depicts the

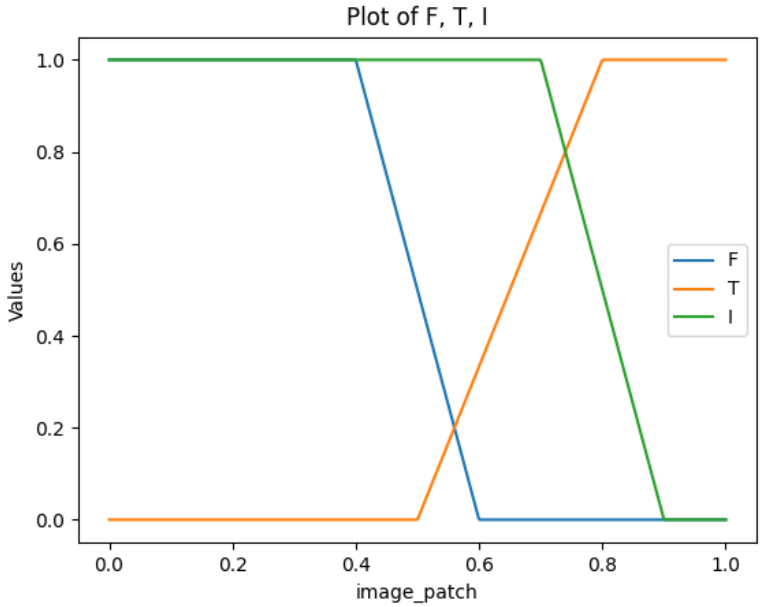

Figure 5. Graphical Representation of a Single-Valued Triangular Neutrosophic Number (T_3, I_3, F_3) .

variation of the truth, indeterminacy, and falsity membership functions, emphasizing the independent behavior of these three components. The triangular shapes highlight the distribution of each membership degree over the domain, where the peaks represent the maximum contribution of each component. Furthermore, the differences between the three figures demonstrate how varying the parameters of (T, I, F) affects the representation of uncertainty, allowing for flexible modeling of imprecision, inconsistency, and noise. This illustrates the advantage of the neutrosophic framework in capturing complex and uncertain information compared to traditional fuzzy and intuitionistic fuzzy representations.

Let z be the total number of batches, and p^n the patch at position n, of size $k \times k$. For each real patch p^n, n= 1,...,z. The three neutrosophic membership $\pi^n_{\nu,T}$, $\pi^n_{\nu,I}$ and $\pi^n_{\nu,F}$ for the z patches are defined as follow:

$$\langle \pi^n_{\nu,T}, \pi^n_{\nu,I}, \pi^n_{\nu,F} \rangle = \langle T_\nu(p^n), I_\nu(p^n), F_\nu(p^n) \rangle = \begin{pmatrix} \langle T_\nu(p^n_{1,1}), I_\nu(p^n_{1,1}), F_\nu(p^n_{1,1}) \rangle & \cdots & \langle T_\nu(p^n_{1,k}), I_\nu(p^n_{1,k}), F_\nu(p^n_{1,k}) \rangle \\ \vdots & \ddots & \vdots \\ \langle T_\nu(p^n_{k,1}), I_\nu(p^n_{k,1}), F_\nu(p^n_{k,1}) \rangle & \cdots & \langle T_\nu(p^n_{k,k}), I_\nu(p^n_{k,k}), F_\nu(p^n_{k,k}) \rangle \end{pmatrix} \tag{14}$$

where ν=1,2,3.

Lets $S^n_\nu(p^n)$ be the sum of the three membership function such that

$$S^n_\nu = T_\nu(p^n) + I_\nu(p^n) + F_\nu(p^n) = \begin{pmatrix} (T_\nu(p^n_{1,1}) + I_\nu(p^n_{1,1}) + F_\nu(p^n_{1,1})) & \cdots & (T_\nu(p^n_{1,k}) + I_\nu(p^n_{1,k}) + F_\nu(p^n_{1,k})) \\ \vdots & \ddots & \vdots \\ (T_\nu(p^n_{k,1}) + I_\nu(p^n_{k,1}) + F_\nu(p^n_{k,1})) & \cdots & (T_\nu(p^n_{k,k}) + I_\nu(p^n_{k,k}) + F_\nu(p^n_{k,k})) \end{pmatrix}, \tag{15}$$

and P_ν^n be the product of the three membership function exhibit as follows

$$P_\nu^n(p^n) = T_\nu(p^n) * I_\nu(p^n) * F_\nu(p^n) = \begin{pmatrix} (T_\nu(p_{1,1}^n) * I_\nu(p_{1,1}^n) * F_\nu(p_{1,1}^n)) & \cdots & (T_\nu(p_{1,k}^n) * I_\nu(p_{1,k}^n) * F_\nu(p_{1,k}^n)) \\ \vdots & \ddots & \vdots \\ (T_\nu(p_{k,1}^n) * I_\nu(p_{k,1}^n) * F_\nu(p_{k,1}^n)) & \cdots & (T_\nu(p_{k,k}^n) * I_\nu(p_{k,k}^n) * F_\nu(p_{k,k}^n)) \end{pmatrix}, \tag{16}$$

For each patch $p^n = (p_{ij}^n)_{i,j=1,\ldots,k}$, $n = 1, \ldots, z$, the output $pool(p^n)$, representing the "crisp" output associated with the given membership function. This value is derived after applying neutrosophic processing to the input batch.

$$pool(p^n) = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} p_{ij}^n out(p_{ij}^n)}{\sum_{i=1}^{k}\sum_{j=1}^{k} out(p_{ij}^n)} \tag{17}$$

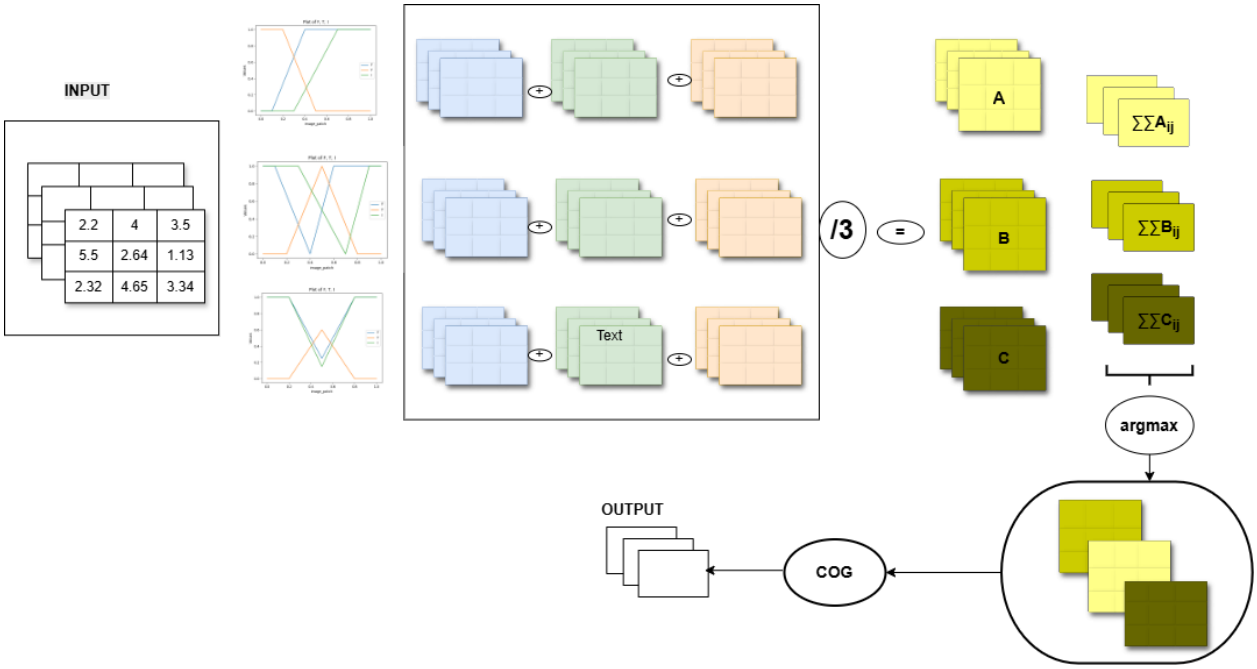

Figure 6. Graphical representation of the Neutrosophic pooling workflow.

Figure 6 depicts the step-by-step process of the Neutrosophic Pooling mechanism. It begins with a 3 × 3 patch taken from a set of feature maps consisting of three filters. The figure outlines how each stage transforms the patch using intuitionistic fuzzy logic principles, progressively refining the information to generate the final pooled result. This schematic helps visualize the internal computation flow of the pooling method.

Before conducting our experiments, we first highlight the key differences between fuzzy pooling, intuitionistic fuzzy pooling, and the proposed pooling method, as summarized in Table 1. Fuzzy pooling relies on a single membership function to assign importance to each element. Intuitionistic fuzzy pooling (IFS) considers both membership (truth) and non-membership (falsity) values and incorporates a hesitation function to handle cases where the membership value does not precisely reflect the element; in IFS, the sum of truth and falsity is less than or equal to 1. In contrast, neutrosophic pooling (NFS), can be viewed as an augmented extension of intuitionistic fuzzy pooling, uses three independent components—truth (T), falsity (F), and indeterminacy (I)—which allow it

to effectively handle noise and uncertainty; in NFS, the sum of the three components can be less than or equal to 3(each component less or equal to 1).As a conclusion, neutrosophic pooling addresses a key limitation of intuitionistic fuzzy pooling, where the sum of components is constrained to 1, the components are dependent and the indeterminacy derived from membership and non-membership degrees. In neutrosophic pooling, all three components—truth, falsity, and indeterminacy—are independent, enabling it to better handle real-world scenarios and provide more flexible and robust treatment of uncertainty and noise.

Table 1. Comparison of Fuzzy, Intuitionistic Fuzzy, and Neutrosophic Fuzzy Pooling

Aspect	**Fuzzy Pooling**	**Intuitionistic Fuzzy Pooling (IFS)**	**Neutrosophic Fuzzy Pooling (NFS)**
Basic idea	Weighted aggregation using membership	Aggregation using membership (truth) and non-membership (falsity)	Aggregation using truth (T), indeterminacy (I), and falsity (F)
Components	Single membership function (μ)	Two components: truth-membership (μ) and falsity-membership (ν)	Three independent components: truth (T), falsity (F), and indeterminacy (I)
Handling indeterminacy / hesitation	Not handled	Implicitly defined as $\pi = 1 - \mu - \nu$	Explicitly handled via independent indeterminacy (I)
Summation constraints	None	$\mu(x) + \nu(x) \leq 1$	$0 \leq T(x) + F(x) + I(x) \leq 3$
Dependency	N/A	Components are dependent due to the sum constraint	Components are independent, allowing flexible handling of noise and uncertainty
Robustness to noise	Low	Moderate	High

5. Datasets

This section describes the different datasets used to evaluate the proposed network and to compare its performance with existing methods.

5.1. MNIST

The MNIST dataset 7, short for the Modified National Institute of Standards and Technology database, is a classic benchmark in the field of machine learning, particularly for handwritten digit recognition. It consists of 70,000 grayscale images of handwritten digits ranging from 0 to 9, with each image sized at 28x28 pixels. The dataset is divided into 60,000 training samples and 10,000 test samples, providing a rich resource for developing and evaluating classification algorithms. MNIST is notable for its simplicity and accessibility, making it a popular

choice for beginners and researchers alike. Its well-defined structure and the clarity of its digit representations allow for straightforward comparisons of different machine learning models, serving as a foundational dataset in the study of neural networks and image processing.

Figure 7. Examples images of 10 classes of MNIST .

5.2. cifar-10

The CIFAR-10 dataset8 is a widely used benchmark in the field of machine learning and computer vision, consisting of 60,000 32x32 color images categorized into 10 distinct classes. These classes include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck, with each class containing 6,000 images. The dataset is divided into 50,000 training images and 10,000 test images, making it ideal for evaluating the performance of various machine learning models, particularly convolutional neural networks (CNNs). Due to its relatively small size and diverse set of images, CIFAR-10 serves as a foundational dataset for researchers and practitioners to test algorithms and develop new techniques in image classification and recognition tasks.

5.3. STL10

The STL10 dataset [34] is a challenging image classification benchmark consisting of 96×96 pixel color images from 10 different classes, such as airplane, bird, cat, deer, and others . It contains a total of 13,000 labeled images, with 5,000 images used for training and 8,000 for testing. Inspired by the CIFAR-10 dataset, STL10 includes several improvements Figure 9. Unlike simpler datasets such as MNIST or CIFAR-10, STL10 provides fewer labeled training samples per class (only 500), while the images are more complex—featuring higher resolution and greater intra-class variability—which increases the difficulty of the classification task.

Figure 8. Examples images of 10 classes of CIFAR-10 .

6. Experimentation and results

This section focuses on the experiment and comparison of Neutrosophic Fuzzy Pooling with different pooling methods across various datasets to highlight the differences and strengths of Neutrosophic Fuzzy Pooling compared to other pooling methods in in feature extraction and information preservation.

All the data are normalized within the interval $[0, R_{\max}]$, where $R_{\max} = 6$, using the ReLU6 function defined as

$$\mathrm{ReLU6}(x) = \min(\max(0, x), 6).$$

$$R_{\max} = 6$$

$$a'' = \frac{R_{\max}}{10}, \quad b'' = \frac{R_{\max}}{9}$$

$$a = \frac{R_{\max}}{10} + \frac{a''}{10}, \quad b = \frac{R_{\max}}{9} + \frac{b''}{10}$$

$$a' = \frac{R_{\max}}{10} + \frac{a}{10}, \quad b' = \frac{R_{\max}}{9} + \frac{b}{10}$$

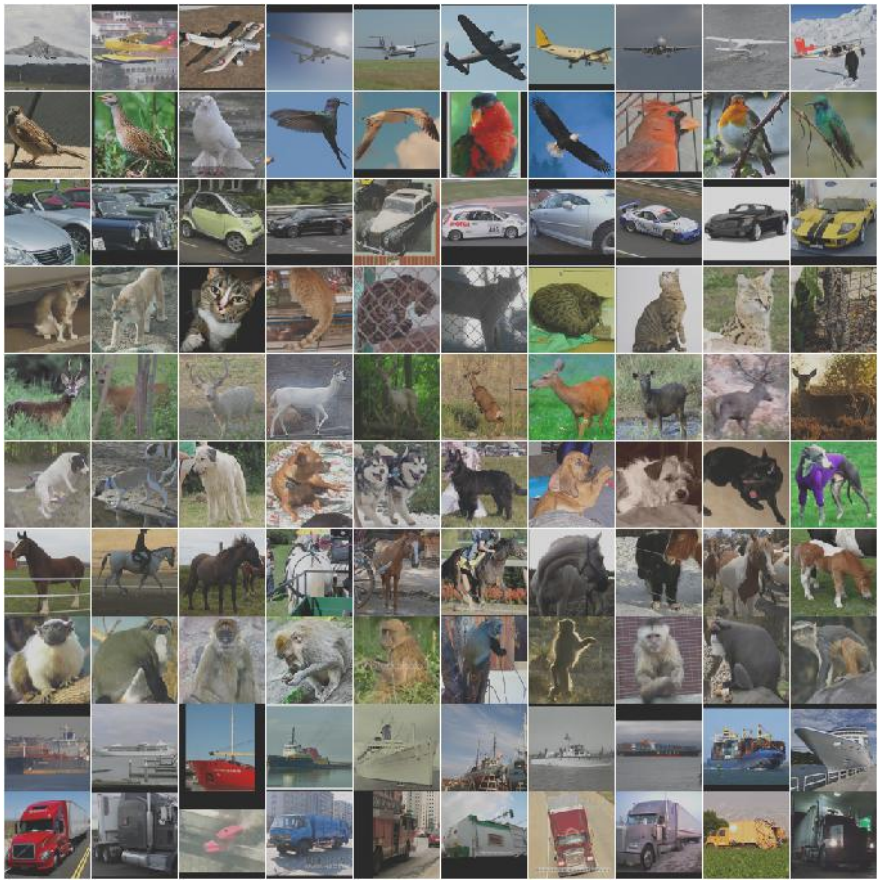

Figure 9. Examples of STL10 images [34].

$$a_2' = \frac{R_{\max}}{6}, \quad a_2 = \frac{R_{\max}}{6} + \frac{a_2'}{10}, \quad a_2'' = \frac{R_{\max}}{6} + \frac{a_2}{10}$$

$$b_2' = \frac{R_{\max}}{5}, \quad b_2 = \frac{R_{\max}}{5} + \frac{b_2'}{10}$$

$$c_2' = \frac{R_{\max}}{4}, \quad b_2'' = \frac{R_{\max}}{4} + \frac{b_2'}{10}$$

$$c_2 = \frac{R_{\max}}{4}, \quad c_2'' = \frac{R_{\max}}{4} + c_2$$

$$a_3'' = \frac{R_{\max}}{3}, \quad a_3 = \frac{R_{\max}}{3} + \frac{a_3''}{10}, \quad a_3' = \frac{R_{\max}}{3} + \frac{a_3}{10}$$

$$b_3'' = \frac{R_{\max}}{2}, \quad b_3 = \frac{R_{\max}}{2} + \frac{b_3''}{10}, \quad b_3' = \frac{R_{\max}}{2} + \frac{b_3}{10}$$

All parameters are initialized randomly in accordance with the constraints specified in Section 4 and subsequently updated iteratively during the training process.

By comparing different pooling strategies under controlled conditions, this experiment aims to provide insights into

Methodology	Accuracy	Precision	Recall	F1 score
Max Pooling (%)	98.30	98.30	98.29	98.29
Average pooling (%)	98.38	98.37	98.36	98.37
Soft Pooling (%)	98.30	98.30	98.30	98.30
Fuzzy pooling (%)	98.43	98.42	98.42	98.42
Neutrosophic Fuzzy pooling (%)	98.48	98.47	98.47	98.47

Table 2. Comparison results of 5 types of pooling on MNIST dataset.

how pooling choices influence feature representation, learning dynamics, and overall classification effectiveness in CNNs.

In our experiment, we first employed the MNIST dataset with simple architecture model to systematically compare the performance of several pooling techniques. These techniques include Max Pooling, Average Pooling, Fuzzy Pooling, and Neutrosophic Pooling. Each pooling method was implemented to evaluate its effectiveness in feature extraction and overall model performance.

The base CNN architecture consists of a single convolutional layer followed by a pooling operation, feature flattening, and a fully connected output layer for classification. This structure is deliberately kept simple to isolate the effects of the pooling method on model performance. All other architectural and training components—including the number of filters, activation functions, optimizer, and training schedule—are held constant across experiments to ensure fair comparisons.

The model is trained and evaluated on the MNIST dataset, a widely used benchmark for handwritten digit classification. Training is conducted over a fixed number of epochs (50 epochs) using the Adam optimizer and a negative log-likelihood loss function. Performance is assessed using standard classification metrics such as accuracy, precision, recall, and F1 score. These metrics provide a comprehensive evaluation of the model's ability to generalize across all classes.

Table 2 presents a comparative analysis of five different pooling strategies—Max Pooling, Average Pooling, Soft Pooling, Fuzzy Pooling, and Neutrosophic Fuzzy Pooling—applied to a consistent CNN architecture and evaluated on the MNIST dataset. The performance metrics considered include Accuracy, Precision, Recall, and F1 score, all expressed as percentages.

The results show that **Neutrosophic Fuzzy Pooling** achieves the best overall performance, with an accuracy of **98.48%** and the highest scores in precision, recall, and F1 score (all at **98.47%**). This indicates the effectiveness of integrating neutrosophic logic into the pooling process, enabling more nuanced feature aggregation and improved generalization.

Fuzzy Pooling also performs well, with an accuracy of **98.43%** and consistent values for the other metrics (**98.42%**), demonstrating that fuzzy logic alone can enhance the pooling operation beyond traditional methods.

Average Pooling achieves slightly better performance than **Max Pooling**, with an accuracy of **98.38%** compared to **98.30%**, and shows similarly small gains across precision, recall, and F1 score. This suggests that average pooling may provide a more balanced summary of features than max pooling in this context.

Overall, the results indicate that advanced pooling strategies—particularly those incorporating fuzzy and neutrosophic logic—can provide measurable improvements in classification performance. These improvements, though numerically modest, can be significant in high-precision applications or tasks requiring enhanced robustness and interpretability.

In this second experiment, a custom deep convolutional neural network was developed using PyTorch. The model comprises three convolutional blocks using Batch Normalization, ReLU activations, and Average Pooling layers, followed by two fully connected layers with a dropout mechanism to reduce overfitting. The STL10 images were resized to 96×96 pixels and normalized. For data loading, the training set was passed to a DataLoader with a batch size of 64, shuffling enabled, and configured with 2 worker threads and pinned memory for efficient GPU transfer.

The test set used a batch size of 128 and the same threading and memory settings, but without shuffling. First Block: A 3×3 convolution with 32 filters and padding of 1, followed by batch normalization, ReLU activation, and 2×2 pooling, reducing the resolution from 96×96 to 48×48.

Second Block: A 3×3 convolution with 64 filters, batch normalization, ReLU, and 2×2 pooling, reducing the resolution to 24×24.

Third Block: A 3×3 convolution with 128 filters, batch normalization, ReLU, and 2×2 pooling, reducing the resolution to 12×12. Training was carried out for 20 epochs using the Adam optimizer with a learning rate of 0.001. After each epoch, model performance was evaluated using accuracy, precision, recall, and F1 score on the test set. These metrics highlighted the model's robustness in handling high-resolution and limited-label image classification tasks.

Methodology	Accuracy	Precision	Recall	F1 score
Max Pooling (%)	59.80	59.44	59.80	57.62
Average pooling (%)	61.61	62.30	61.61	61.30
Soft Pooling (%)	62.85	63.04	62.85	62.51
Fuzzy pooling (%)	61.50	62.23	61.50	61.29
Neutrosophic Fuzzy pooling (%)	64.15	65.02	64.15	64.39

Table 3. Comparison results of 5 types of pooling on STL10 dataset (20 epoch).

Among the five pooling methods evaluated on the STL10 dataset over 20 epochs table 3, Neutrosophic Fuzzy Pooling achieved the best overall performance. It outperformed all other methods in terms of accuracy (64.15%), precision (65.02%), recall (64.15%), and F1-score (64.39%). This indicates that incorporating neutrosophic logic into the fuzzy pooling framework significantly enhances the model's ability to generalize and capture more discriminative features.

In contrast, Max Pooling, the most basic technique, yielded the lowest performance across all metrics, suggesting that it may discard too much information. Soft Pooling and Average Pooling offered moderate improvements, while Fuzzy Pooling showed slightly better performance than Average Pooling but was still outperformed by Soft Pooling and Neutrosophic Fuzzy Pooling. Despite these promising results, the proposed Neutrosophic Fuzzy Pooling introduces a slightly higher computational cost compared to standard pooling methods and involves a larger number of parameters, which may increase model complexity.

The third experiment, focuses on the ResNet18 architecture applied to the CIFAR-10 dataset. The network is based on a deep residual learning framework composed of an initial 7×7 convolutional layer followed by batch normalization, ReLU activation, and pooling layer. It then consists of four residual stages built using BasicBlock structures, which progressively extract hierarchical feature representations. The final classification is performed using global average pooling followed by a fully connected layer. For this experiment, the pooling strategies were evaluated under identical training conditions using the Adam optimizer and cross-entropy loss.

Methodology	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Loss
Best Results					
Average Pooling	76.45	76.72	76.45	76.49	0.0121
Fuzzy Pooling	75.19	75.42	75.19	75.27	0.0133
Neutrosophic Fuzzy Pooling	**76.75**	**76.62**	**76.75**	**76.65**	**0.0113**
Mean Results					
Average Pooling	74.62	74.87	74.62	74.58	0.1069
Fuzzy Pooling	73.23	73.51	73.23	73.19	0.1170
Neutrosophic Fuzzy Pooling	**74.89**	**75.12**	**74.89**	**74.85**	**0.1066**

Table 4. Comparison of pooling methods: best and mean performance results.

The results in Table 4 clearly highlight the effectiveness of Neutrosophic Fuzzy Pooling, which consistently achieves the best performance across all evaluation metrics in both the best and mean scenarios. This method slightly but systematically surpasses Average Pooling in terms of accuracy, precision, recall, F1-score, and loss, demonstrating its ability to better capture and represent informative features. The improvement can be attributed to its capacity to handle uncertainty and indeterminacy more effectively than conventional approaches, leading to more robust learning. In contrast, while Fuzzy Pooling provides reasonable results, its performance remains lower. This suggests that its efficiency is highly dependent on precise parameter selection; without careful tuning, it may not fully exploit its potential. Overall, Neutrosophic Fuzzy Pooling shows greater stability and reliability, making it a more suitable choice for achieving consistent and improved performance.

Methodology	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Loss
Best Results					
Avg Pooling	65.64	67.23	65.64	65.84	0.0743
Soft Pooling	66.16	68.18	66.16	66.45	0.0760
Max Pooling	65.20	67.40	65.20	65.55	0.0759
Fuzzy Pooling	64.77	67.65	64.97	65.41	0.0749
Neutrosophic Fuzzy Pooling	66.24	67.11	66.24	66.28	0.0755
Mean Results					
Avg Pooling	63.07	64.76	63.07	63.29	0.2936
Soft Pooling	63.15	65.29	64.06	63.30	0.2886
Max Pooling	62.69	65.00	62.69	63.01	0.2960
Fuzzy Pooling	62.24	63.96	62.24	62.24	0.985
Neutrosophic Fuzzy Pooling	63.16	64.78	63.16	63.32	0.2911

Table 5. Best and mean performance metrics of different pooling methods on CIFAR-10 with Gaussian noise ($\sigma = 0.3$).

On CIFAR-10 with Gaussian noise corruption ($\sigma = 0.3$), the experimental results demonstrate that the proposed Neutrosophic Fuzzy Pooling remains competitive under highly noisy conditions. As shown in Table 5, the proposed method achieved the highest best accuracy of 66.24% and the highest F1-score of 66.28%, indicating a better balance between precision and recall compared with conventional pooling strategies. Although Soft Pooling obtained the highest precision value (68.18%), its overall classification balance remained slightly lower than the proposed approach in terms of F1-score.

The mean results across all epochs further confirm the robustness of the proposed pooling mechanism. Neutrosophic Fuzzy Pooling obtained the best mean accuracy (63.16%) and mean F1-score (63.32%), outperforming Average Pooling and Max Pooling under noisy conditions. These improvements suggest that incorporating neutrosophic representations helps preserve discriminative information while reducing the negative impact of noise propagation through the network.

Moreover, Max Pooling exhibited the lowest mean performance, which may be explained by its sensitivity to noisy activations, as the maximum operation can amplify corrupted responses. Average Pooling produced more stable behavior but tended to smooth important discriminative features. Soft Pooling improved robustness by weighting activations continuously; however, the proposed Neutrosophic Fuzzy Pooling achieved superior overall consistency by integrating truth, indeterminacy, and falsity information during feature aggregation.

Overall, these results indicate that the proposed pooling strategy is more resilient to severe Gaussian perturbations and can provide more stable feature extraction in noisy image classification scenarios.

T, I, F are called neutrosophic components. Remarkably, in the same NFS one can have elements that have paraconsistent information (sum of components > 1), others incomplete information (sum of components < 1), others consistent information (in the case when the sum of components = 1), and others components with interval values (without restriction on their superior or inferior sums).

Future work: In future research, genetic algorithms will be explored as a parameter optimization strategy for the proposed neutrosophic pooling method. By enabling automatic selection of optimal parameter configurations, this approach has the potential to improve accuracy, robustness, and generalization while reducing reliance on manual tuning.

7. Conclusion

In this paper, we have introduced a novel pooling operation for Convolutional Neural Networks based on Neutrosophic Fuzzy Sets (NFSs), aiming to explicitly account for the degree of indeterminacy—an aspect often overlooked in classical, fuzzy, and intuitionist fuzzy pooling schemes. Unlike intuitionist fuzzy sets, which restrict the indeterminacy to the complement of membership and non-membership degrees, the proposed approach allows for an independent treatment of uncertainty, thus offering a more expressive and robust representation of feature map imprecision. Moreover, unlike Soft Pooling, which relies on weighted averaging to preserve dominant activations and does not explicitly capture uncertainty, the proposed Neutrosophic Fuzzy Pooling method models truth, indeterminacy, and falsity components, enabling richer feature representations and improved robustness in noisy or ambiguous scenarios.

Our proposed NFS-based pooling layer integrates seamlessly into existing CNN architectures and operates through a four-step process: bi-fuzzification, two-stage aggregation, and defuzzification. Extensive experiments across diverse image classification tasks confirm the effectiveness of this approach. Notably, the neutrosophic pooling layer consistently outperforms traditional pooling methods, including recent state-of-the-art models, particularly in environments characterized by uncertainty and noise.

These promising results demonstrate that incorporating explicit modeling of indeterminacy into CNN pooling operations can lead to significant gains in robustness and accuracy. Future work may explore the integration of NFS pooling into other neural architectures and its application to broader tasks such as object detection, semantic segmentation, and adversarial learning.

References

1. Florentin Smarandache. Neutrosophy: neutrosophic probability, set, and logic: analytic synthesis & synthetic analysis. 1998.
2. Haibin Wang, Florentin Smarandache, Yanqing Zhang, and Rajshekhar Sunderraman. Single valued neutrosophic sets. *Infinite study*, 12:20110, 2010.
3. K. Atanassov. Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, 20(1):87–96, 1986.
4. Lotfi Asker Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.
5. Amirhossein Nafei, YUAN Wenjun, and Hadi Nasseri. A new method for solving interval neutrosophic linear programming problems. *Gazi University Journal of Science*, 33(4):796–808, 2020.
6. Selin Yalçın and Ihsan Kaya. Analyzing of process capability indices based on neutrosophic sets. *Computational and Applied Mathematics*, 41(6):287, 2022.
7. Dingjun Yu, Hanli Wang, Peiqiu Chen, and Zhihua Wei. Mixed pooling for convolutional neural networks. In *Rough Sets and Knowledge Technology: 9th International Conference, RSKT 2014, Shanghai, China, October 24-26, 2014, Proceedings 9*, pages 364–375. Springer, 2014.
8. Junfeng Chen, Zhoudong Hua, Jingyu Wang, and Shi Cheng. A convolutional neural network with dynamic correlation pooling. In *2017 13th International Conference on Computational Intelligence and Security (CIS)*, pages 496–499. IEEE, 2017.
9. Rajendran Nirthika, Siyamalan Manivannan, Amirthalingam Ramanan, and Ruixuan Wang. Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study. *Neural Computing and Applications*, 34(7):5321–5347, 2022.
10. Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1713–1721, 2015.
11. Zhen Wei, Jingyi Zhang, Li Liu, Fan Zhu, Fumin Shen, Yi Zhou, Si Liu, Yao Sun, and Ling Shao. Building detail-sensitive semantic segmentation networks with polynomial pooling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7115–7123, 2019.
12. Caglar Gulcehre, Kyunghyun Cho, Razvan Pascanu, and Yoshua Bengio. Learned-norm pooling for deep feedforward and recurrent neural networks. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, pages 530–546. Springer, 2014.
13. Joan Bruna Estrach, Arthur Szlam, and Yann LeCun. Signal recovery from pooling representations. In *International conference on machine learning*, pages 307–315. PMLR, 2014.
14. Hayoung Eom and Heeyoul Choi. Alpha-integration pooling for convolutional neural networks. *arXiv preprint arXiv:1811.03436*, 2018.

15. Zenglin Shi, Yangdong Ye, and Yunpeng Wu. Rank-based pooling for deep convolutional neural networks. *Neural Networks*, 83:21–31, 2016.
16. Bhaskar Navaneeth and M Suchetha. A dynamic pooling based convolutional neural network approach to detect chronic kidney disease. *Biomedical Signal Processing and Control*, 62:102068, 2020.
17. Florentin Bieder, Robin Sandkühler, and Philippe C Cattin. Comparison of methods generalizing max-and average-pooling. *arXiv preprint arXiv:2103.01746*, 2021.
18. Alexandros Stergiou, Ronald Poppe, and Grigorios Kalliatakis. Refining activation downsampling with softpool. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10357–10366, 2021.
19. Wojciech Czaja, Weilin Li, Yiran Li, and Mike Pekala. Maximal function pooling with applications. *Excursions in Harmonic Analysis, Volume 6: In Honor of John Benedetto's 80th Birthday*, pages 413–429, 2021.
20. Ashwani Kumar. Ordinal pooling networks: for preserving information over shrinking feature maps. *arXiv preprint arXiv:1804.02702*, 2018.
21. Zhiqiang Tong, Kazuyuki Aihara, and Gouhei Tanaka. A hybrid pooling method for convolutional neural networks. In *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part II 23*, pages 454–461. Springer, 2016.
22. Matthew D Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.
23. Haibing Wu and Xiaodong Gu. Max-pooling dropout for regularization of convolutional neural networks. In *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part I 22*, pages 46–54. Springer, 2015.
24. Shuangfei Zhai, Hui Wu, Abhishek Kumar, Yu Cheng, Yongxi Lu, Zhongfei Zhang, and Rogerio Feris. S3pool: Pooling with stochastic spatial sampling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4970–4978, 2017.
25. Benjamin Graham. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*, 2014.
26. Jiashi Feng, Bingbing Ni, Qi Tian, and Shuicheng Yan. Geometric p-norm feature pooling for image classification. In *CVPR 2011*, pages 2609–2704. IEEE, 2011.
27. Kunlun Qi, Qingfeng Guan, Chao Yang, Feifei Peng, Shengyu Shen, and Huayi Wu. Concentric circle pooling in deep convolutional networks for remote sensing scene classification. *Remote Sensing*, 10(6):934, 2018.
28. Kunlun Qi, Chao Yang, Chuli Hu, Qingfeng Guan, Wenwen Tian, Shengyu Shen, and Feifei Peng. Polycentric circle pooling in deep convolutional networks for high-resolution remote sensing image recognition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:632–641, 2020.
29. Tsung-Yu Lin and Subhransu Maji. Improved bilinear pooling with cnns. *arXiv preprint arXiv:1707.06772*, 2017.
30. Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 430–443. Springer, 2012.
31. Fang Chen, Gourav Datta, Souvik Kundu, and Peter A Beerel. Self-attentive pooling for efficient deep learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3974–3983, 2023.
32. Chaymae Rajafillah, Karim El Moutaouakil, Alina-Mihaela Patriciu, Ali Yahyaouy, and Jamal Riffi. Int-fup: Intuitionistic fuzzy pooling. *Mathematics*, 12(11):1740, 2024.
33. Alaa Fouad Momena, Rakibul Haque, Mostafijur Rahaman, Soheil Salahshour, and Sankar Prasad Mondal. The existence and uniqueness conditions for solving neutrosophic differential equations and its consequence on optimal order quantity strategy. *Logistics*, 8(1):18, 2024.
34. Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.