

Article

INT-FUP: Intuitionistic Fuzzy Pooling

Chaymae Rajafillah ^{1,†} , Karim El Moutaouakil ^{1,*,†} , Alina-Mihaela Patriciu ^{2,*,†} , Ali Yahyaouy ^{3,†}  and Jamal Riffi ^{3,†}¹ Laboratory of Engineering Sciences, Multidisciplinary Faculty of Taza, Sidi Mohamed Ben Abdellah University, Taza 35000, Morocco; chaymae.rajafillah@usmba.ac.ma² Department of Mathematics and Computer Sciences, Faculty of Sciences and Environment, Dunărea de Jos University of Galați, 800201 Galați, Romania³ Computer Science, Signals, Automatics and Cognitivism Laboratory, Sciences Faculty of Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fès-Atlas 30000, Morocco; ali.yahyaouy@usmba.ac.ma (A.Y.); jamal.riffi@usmba.ac.ma (J.R.)

* Correspondence: karim.elmoutaouakil@usmba.ac.ma (K.E.M.); alina.patriciu@ugal.ro (A.-M.P.)

† These authors contributed equally to this work.

Abstract: Convolutional Neural Networks (CNNs) are a kind of artificial neural network designed to extract features and find out patterns for tasks such as segmentation, recognizing objects, and drawing up classification. Within a CNNs architecture, pooling operations are used until the number of parameters and the computational complexity are reduced. Numerous papers have focused on investigating the impact of pooling on the performance of Convolutional Neural Networks (CNNs), leading to the development of various pooling models. Recently, a fuzzy pooling operation based on type-1 fuzzy sets was introduced to cope with the local imprecision of the feature maps. However, in fuzzy set theory, it is not always accurate to assume that the degree of non-membership of an element in a fuzzy set is simply the complement of the degree of membership. This is due to the potential existence of a hesitation degree, which implies a certain level of uncertainty. To overcome this limitation, intuitionistic fuzzy sets (IFS) were introduced to incorporate the concept of a degree of hesitation. In this paper, we introduce a novel pooling operation based on intuitionistic fuzzy sets to incorporate the degree of hesitation heretofore neglected by a fuzzy pooling operation based on classical fuzzy sets, and we investigate its performance in the context of image classification. Intuitionistic pooling is performed in four steps: bifuzzification (by the transformation of data through the use of membership and non-membership maps), first aggregation (through the transformation of the IFS into a standard fuzzy set, second aggregation (through the transformation and use of a sum operator), and the defuzzification of feature map neighborhoods by using a max operator. IFS pooling is used for the construction of an intuitionistic pooling layer that can be applied as a drop-in replacement for the current, fuzzy (type-1) and crisp, pooling layers of CNN architectures. Various experiments involving multiple datasets demonstrate that an IFS-based pooling can enhance the classification performance of a CNN. A benchmarking study reveals that this significantly outperforms even the most recent pooling models, especially in stochastic environments.

Keywords: intuitionistic fuzzy sets; pooling; compression; CNNs; classification**MSC:** 68T07; 03F55

Citation: Rajafillah, C.; El Moutaouakil, K.; Patriciu, A.-M.; Yahyaouy, A.; Riffi, J. INT-FUP: Intuitionistic Fuzzy Pooling. *Mathematics* **2024**, *12*, 1740. <https://doi.org/10.3390/math12111740>

Academic Editors: Cheung Chieh Ku and Chang-Hua Lien

Received: 28 April 2024

Revised: 23 May 2024

Accepted: 27 May 2024

Published: 3 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Deep learning is a subset of machine learning that involves training artificial neural networks with multiple layers to perform complex tasks such as image recognition [1], advisory frameworks [2], image classification and sequencing, health image processing, natural speech processing [3], brain-computer interfaces [4], and economics time series [5].

Generally, CNNs consist of several layers such as convolutional layers, pooling layers, and fully connected layers, where pooling layers down-sample the feature maps to reduce

their size and improve efficiency. Pooling is a critical component of CNNs that plays an important role by reducing the spatial dimensions of the input data, while at the same time retaining the important information. Max pooling and average pooling are two of the most popular pooling operations used, due to the fact that they are simple and fast.

In many different fields, such as engineering, finance, medicine, and natural language processing, the information provided might be imprecise due to various factors, which requires giving a greater degree of precision to every member; thus, fuzzy sets appear as a viable solution to handle these types of problems by assigning each element with a degree of membership. However, the available information corresponding to a fuzzy concept may be incomplete, where the sum of the membership degrees and the non-membership degrees may be less than one.

As a solution, Atanassov [6,7] introduced a flexible extension of traditional fuzzy sets that generalizes fuzzy sets into intuitionistic fuzzy sets (IFS) by adding a “hesitation degree” as a new function, defining the lack of knowledge and thereby providing a tool to deal with the hesitancy of the decision-maker in assigning an element to a set or its complement. Thus, IFS has turned out to be an important tool in modeling real situations [8,9].

The goal of this article is to explore and to apply the concept of intuitionistic fuzzy sets (IFS) to the pooling operation so as to lead to more accurate and robust feature representations and thereby demonstrate the advantages of using IFS-based pooling instead of other more traditional pooling methods.

The rest of the paper is structured as follows: Section 2 provides an overview of various pooling operations through a discussion on related work. Section 3 introduces the concept of intuitionistic fuzzy logic and the proposed intuitionistic pooling model. In Section 4, experimental results are presented, including performance measures and CNN classification tests conducted on different pooling methods. Finally, Section 5 concludes the paper and offers perspectives for future research.

2. Related Work

Convolutional Neural Networks (CNNs) are a deep learning algorithm, widely used in computer vision, based on two operations: convolution (which extracts features through filtering) and pooling (which reduces the dimensionality). There are multiple types of pooling operators that are used for different purposes as will be shown below in this section.

The two more common types of pooling are max and average pooling, due to their simplicity in the absence of the parameters to tune, where average pooling summarizes all the features in the pooling region, then reduces noisy features, and the background region dominates. In contrast, max pooling selects the strongest activation in the pooling region, thus avoiding the effect of unwanted background and capturing noisy features [10]. In this direction, some new pooling has emerged through the use of both max and average pooling in order to take greater advantage of them, such as mixed max-average pooling [11] which combines them linearly, thereby giving it a weight that determines the proportions of using each type of pooling, or like gated mix-average pooling that depends on the characteristics of each image instead of the characteristics of the dataset [10,11] because of the high correlation between adjacent pixels of the image and due to both the mixed max-average pooling and gated mix-average pooling and thus considering each pooling region independently. Dynamic Correlation pooling [10,12] is introduced to use the correlation information between adjacent pixels of the image. In medical imaging, soft pooling approaches are widely used compared to linear combination by using a smooth differentiable function to approximate the max and the average pooling for different parameter settings [10]. Log-Sum-Exp pooling (LSE), Polynomial pooling [13], Learned-Norm pooling [14], l_p pooling [15], α Integration (αI) pooling [16], rank-based pooling [17], Dynamic pooling [18], Smooth-Maximum pooling [19], soft pooling [20], Maxfun pooling [21] and Ordinal pooling [22] are all types of soft pooling approaches.

Moreover, there are other soft pooling approaches based on the characteristics of the pooling region, such as Polynomial pooling, which enhances the detail sensitivity of

a segmentation network and is compatible with any pre-trained classification [13], and l_p pooling, which provides a flexible way to transition smoothly from max to average pooling [17] and is characterized by the order of the unit that is learned according to geometrical perspective rather than pre-defined [14]. In the case of rank-based pooling, the top k elements in each pooling region are averaged together as the pooled representation. Ordinal pooling and Multiactivation pooling [23] are similar to rank-based pooling since they also use the rank of the elements when applying pooling.

Some variants of pooling aim to handle overfitting such as Mixed-pooling [24] and Hybrid pooling [25], which are randomly selected since either max or average operations will be performed during training and the mode used in training will also be used in testing [10], in addition to stochastic pooling [26] which solves the down-weight caused by average pooling as well as the overfitting problems caused by max pooling. In case the training data are limited, over-fitting occurs due to strong activations that dominate the updating process, and therefore, rank-based stochastic pooling [17] could be used. In contrast to stochastic pooling that uses only one value from each pooling region, Max-pooling dropout [27] uses a set of values that could be randomly sampled, then pooling can be applied on these randomly sampled activations [10]. Comparing with stochastic pooling, both randomly sample activation based on multinomial distributions at the pooling stage, with better performance by Max-pooling dropout. To date, this approach introduces randomness in the pooling stage. S3 pooling [28] and fractional max pooling [29] introduce randomness in the spatial sampling stage.

There are other pooling approaches used for specific purposes such as encoding spatial structure information. Spatial Pyramid pooling (SPP) [30] is a popular one and is useful for rigid structures. Cell Pyramid Matching (CPM) [31] is proposed for cell image classification through the incorporation of two spatial structure Dual Region (DR) descriptors and Spatial Pyramid pooling (also known as Spatial Pyramid Matching (SPM) [30]).

In case of images that include objects with various poses [10], part-based pooling [10,32] is useful as a solution according to its ability to detect diverse parts of each image and pool these features, finally concatenating them together as the final image representation. In the case of rotated objects, Concentric Circle pooling (CCP) [33] and Polycentric Circle pooling [34] are efficient in dealing with the rotation variance problem in CNNs.

Unlike previous methods, which aimed to capture large-scale spatial structure information, a Geometric l_p -Norm pooling [35] aims to capture local structure information [10].

Pooling can also be used for capturing the interaction between different feature maps, and different regions of feature maps, such as Improved Bilinear pooling [36], and Second-Order pooling [37], which preserves information about their pairwise correlations.

Grouping Bilinear Pooling (GBP) [38] is an improvement of Bilinear pooling aimed at fine-grained image classification, as well as achieving good accuracy with the fewest parameters, in addition to Context-aware Attentional Pooling (CAP) for Fine-grained Visual Classification, where the correlation among the regions and their spatial layouts to encode complementary partial information are taken into account [39]. Moreover, self-attentive pooling extracts more complex relationships between the different features from non-local features of the activation maps in comparison to existing local pooling layers [40].

Certain types of structured data utilize special types of pooling better suited to their structure, for instance, graph-structured data that use graph pooling like Self-attention graph pooling SAGPool [41], a method based on self-attention that considers both node features and graph topology, or Adaptive Structure Aware Pooling (ASAP) [42], a sparse pooling operator able to capture local sub-graph information using a new self-attention, Master2Token (M2T), and a modified GNN formulation to capture the importance of each node within a given graph. In addition, Graph multihead attention pooling with self-supervised learning (GMAPS) [43] enables the construction of graph pooling through a differentiable node assignment depending on a multihead attention mechanism and a hierarchical objective based on maximizing mutual information, along with Coarsened

Graph Infomax Pooling (CGIPool), which maximizes the mutual information between the input and the coarsened graph of each pooling layer. Also, dual-sampling attention pooling was proposed [44] for 3D mesh, and a tripool pooling method [45] was proposed for 3D action recognition from skeleton data.

During implement pooling, some discriminative details can be lost; as a solution, Preserving pooling (DPP) [46] and Local Importance-Based pooling (LIP) [47] were proposed for pooling so as to preserve important features. Without ignoring the popular problem of computational complexity, RNNPool [48] is an efficient pooling for reducing computational complexity and peak memory usage for inference without a substantial loss in accuracy. Also, fuzzy pooling based on fuzzification, aggregation, and the defuzzification of feature map neighborhoods to cope with the uncertainty of feature values can preserve the important features of the pooling areas by transforming the crisp input volume space into a fuzzy feature space [49].

In this paper, we propose an intuitionistic fuzzy pooling methodology that can be integrated into any existing Convolutional Neural Network (CNN) architecture as a replacement for traditional pooling layers.

Our proposed approach extends the concept of fuzzy pooling by addressing an important limitation. In fuzzy pooling, the membership of an element to a fuzzy set is represented by a single value between zero and one. However, this value may not accurately capture the underlying uncertainty. To overcome this limitation, we introduce a new function called the hesitation function which handles cases where the membership value does not precisely correspond to the element.

By incorporating the hesitation function, our method ensures that the neglected value in fuzzy pooling is appropriately considered. This enhancement leads to a more comprehensive and accurate representation of uncertainty during the pooling process.

Overall, our intuitionistic fuzzy pooling methodology presents a valuable extension to the existing fuzzy pooling technique, enabling a more refined treatment of uncertainty in CNN architectures.

3. Intuitionistic Fuzzy Pooling

3.1. Intuitionistic 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, as well as the diagram in Figure 1, which further explains our method. In what follows, the set U represents a universe of discourse (the set of values that a fuzzy variable can take).

Definition 1 ([8]). An intuitionistic fuzzy set (IFS) I is obtained by associating two non-negative values to each element x of U . In other words, to construct an intuitionistic set, we need two functions $\mu_I : U \rightarrow [0, 1]$ and $\eta_I : U \rightarrow [0, 1]$, which represent the degrees of membership and non-membership of each element $x \in U$ to I , respectively. In this sense, I is explicitly given by $I = \{ \langle x, \mu_I(x), \eta_I(x) \rangle / x \in U \}$. Moreover, $\forall x \in U$, Equation (1) is valid:

$$0 \leq \mu_I(x) + \eta_I(x) \leq 1. \quad (1)$$

Note that μ_I and η_I model the experts' knowledge of our agent's environment.

The fact that $\mu_I(x) + \eta_I(x)$ does not expect 1 means that there is a lack of information or certainty about whether x belongs to I . The following definition quantifies the degree of hesitation.

Definition 2 ([8]). The intuitionistic fuzzy indicator or the hesitation indicator from x in I is given by the formula $\phi_I(x) = 1 - \mu_I(x) - \eta_I(x)$.

$\phi_I(x)$ is the degree of indeterminacy of $x \in U$ at the IFS I . $\phi_I(x)$ reflects the lack of knowledge of whether or not each $x \in U$ belongs to the IFS. Evidently, whatever $x \in U$, we have $0 \leq \phi_I(x) \leq 1$.

Definition 3 ([50]). The intuitionistic triangular fuzzy distribution of I can be expressed by the next equation:

$$\langle \mu_I(x), \eta_I(x) \rangle = \begin{cases} \langle 0, 1 - \varepsilon \rangle & \text{if } x \leq s_2 \\ \langle \frac{x-s_2}{s_1-s_2} - \varepsilon, 1 - \frac{x-s_2}{s_1-s_2} \rangle & \text{if } s_2 < x \leq s_1 \\ \langle \frac{s_0-x}{s_0-s_1} - \varepsilon, 1 - \frac{s_0-x}{s_0-s_1} \rangle & \text{if } s_1 \leq x < s_0 \\ \langle 0, 1 - \varepsilon \rangle & \text{if } x \geq s_0 \end{cases} \quad (2)$$

ε is an arbitrary non-negative number such that $\mu_A(x) + \eta_I(x) + \varepsilon = 1$ and $0 \leq \varepsilon \leq 1$.

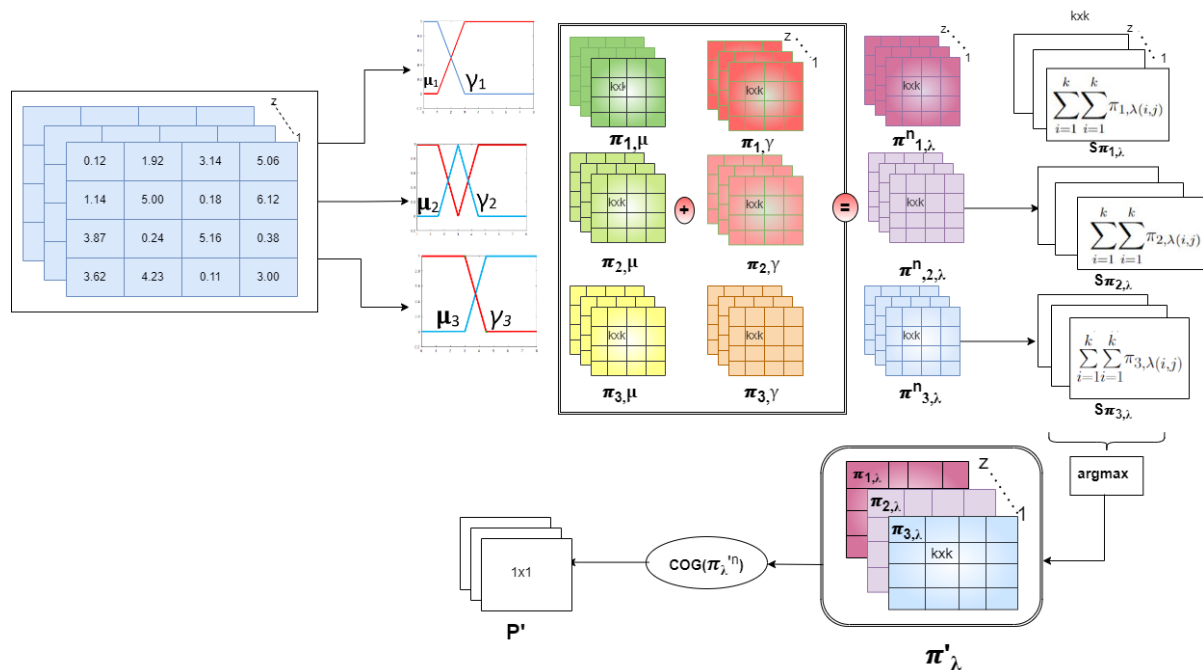


Figure 1. Schematic representation of different steps of intuitionistic pooling.

3.2. Defuzzification of IFSs

In this section, we take up the challenge of defuzzifying an IFS I [51]. A typical way of associating a real number with an IFS I can be illustrated by the steps below:

- Converts the IFS I into a fuzzy (normal) entity;
- Assesses the standard fuzzy pattern by means of a defuzzification strategy.

Regarding step (i), in [52], the contributors gave the name “de-i-fuzzification” for the scheme for generating a convenient fuzzy set out of an IFS. Moreover, they suggested utilizing the operator presented in [53]:

$$D_\alpha(I) = \{ \langle x, \mu_I(x) + \alpha\phi_I(x), \gamma_I(x) + (1 - \alpha)\phi_I(x) \rangle; x \in X \}$$

with $\alpha \in [0, 1]$. Note that $D_\alpha(I)$ is a standard fuzzy subset with membership function

$$\mu_{\alpha, I}(x) = \mu_I(x) + \alpha\phi_I(x)$$

In particular, they proposed $\alpha = 0.5$ as a solution for the minimum problem:

$$\min_{\alpha \in [0, 1]} d(D_\alpha(I), I)$$

where d is the Euclidean distance. In this case, the fuzzy set $D_\alpha(I)$ is characterized by the following membership function:

$$\mu_{0.5,I}(x) = \frac{1}{2}(1 + \mu_I(x) - \eta_I(x))$$

For step (ii), in agreement with the approach suggested in ([54], Section 10), we may evaluate the IFS I by computing the center of gravity (COG) of the obtained fuzzy set, that is,

$$Val_\alpha(I) = \frac{\int_{-\infty}^{+\infty} x \mu_{\alpha,I}(x) dx}{\int_{-\infty}^{+\infty} \mu_{\alpha,I}(x) dx}$$

with $\alpha = 0.5$.

3.3. Intuitionistic Pooling Model

To introduce the fuzzy intuitionistic pooling operator, we consider the following triangular fuzzy intuitionistic membership/non-membership functions [50]:

$$\langle \mu_1(p), \gamma_1(p) \rangle = \begin{cases} \langle 0, 1 - \varepsilon \rangle & \text{if } p > d \\ \langle \frac{d-p}{d-c} - \varepsilon, 1 - \frac{d-p}{d-c} \rangle & \text{if } c \leq p \leq d \\ \langle 1 - \varepsilon, 0 \rangle & \text{if } p < c \end{cases} \quad (3)$$

$$\langle \mu_2(p), \gamma_2(p) \rangle = \begin{cases} \langle 0, 1 - \varepsilon \rangle & \text{if } p \leq a \\ \langle \frac{p-a}{m-a} - \varepsilon, 1 - \frac{p-a}{m-a} \rangle & \text{if } a \leq p \leq m \\ \langle \frac{b-p}{b-m} - \varepsilon, 1 - \frac{b-p}{b-m} \rangle & \text{if } m < p < b \\ \langle 0, 1 - \varepsilon \rangle & \text{if } p \geq b \end{cases} \quad (4)$$

$$\langle \mu_3(p), \gamma_3(p) \rangle = \begin{cases} \langle 0, 1 - \varepsilon \rangle & \text{if } p < r \\ \langle \frac{p-r}{q-r} - \varepsilon, 1 - \frac{p-r}{q-r} \rangle & \text{if } r \leq p \leq q \\ \langle 1 - \varepsilon, 0 \rangle & \text{if } p > q \end{cases} \quad (5)$$

where $r_{\max} = 6$, $d = r_{\max}/2$, $c = d/3$, $a = r_{\max}/4$, $m = r_{\max}/2$, $b = m + a$, $r = r_{\max}/2$, $q = r + r_{\max}/4$; these choices are inspired by the paper that introduced the fuzzy pooling operator [49]. The non-negative real number ε expresses the amount of missing information and p is a non-negative real number. It is worth noting that fuzzification and defuzzification operations are easier to perform for the system using intuitionistic triangular fuzzy numbers (in comparison with Gaussian numbers) [51].

Let $\lambda \in [0 \ 1]$. For $v = 1, 2, 3$, the fuzzy membership function $\mu_{v,\lambda}$ is obtained by aggregating μ_v and γ_v using weight λ :

$$\mu_{v,\lambda}(x) = \mu_v(x) + \lambda \phi_v(x) \quad (6)$$

For each patch $p^n = (p_{ij}^n)_{i,j=1,\dots,k}$, $n = 1, \dots, z$, and based on $\mu_{v,\lambda}$, we define the summary batch by

$$\pi_{v,\lambda}^n = \mu_{v,\lambda}(p^n) = \begin{pmatrix} \mu_{v,\lambda}(p_{1,1}^n) & \cdots & \mu_{v,\lambda}(p_{1,k}^n) \\ \vdots & \ddots & \vdots \\ \mu_{v,\lambda}(p_{k,1}^n) & \cdots & \mu_{v,\lambda}(p_{k,k}^n) \end{pmatrix} = \begin{pmatrix} \mu_v(p_{1,1}^n) & \cdots & \mu_v(p_{1,k}^n) \\ \vdots & \ddots & \vdots \\ \mu_v(p_{k,1}^n) & \cdots & \mu_v(p_{k,k}^n) \end{pmatrix} + \lambda \begin{pmatrix} \phi_v(p_{1,1}^n) & \cdots & \phi_v(p_{1,k}^n) \\ \vdots & \ddots & \vdots \\ \phi_v(p_{k,1}^n) & \cdots & \phi_v(p_{k,k}^n) \end{pmatrix} \quad (7)$$

Pooling starts with the aggregation of the intuitionistic fuzzy patch as follows:

$$S_{\pi_{v,\lambda}}^n = \sum_{i=1}^k \sum_{j=1}^k \pi_{v,\lambda,i,j}^n \quad n = 1, 2, \dots, z \quad (8)$$

Based on these scores, for each p , another patch π'_λ is built by selecting the spatial intuitionistic fuzzy patches $\pi_{v,\lambda}^n, v = 1, \dots, V$, that have the large scores $S_{\pi_{v,\lambda}}^n$:

$$\pi' = \{\pi'_\lambda = \pi_{v,\lambda} \mid v = \operatorname{argmax}(S_{\pi_{v,\lambda}}^n), n = 1, \dots, z\} \quad (9)$$

For each patch $p^n = (p_{ij}^n)_{i,j=1,\dots,k}, n = 1, \dots, z$, if $v^n = \operatorname{argmax}(S_{\pi_{v,\lambda}}^n)$, then the intuitionistic fuzzy crisp value $\operatorname{Pool}_\lambda(p^n)$, associated with the batch n , is given by Equation (10):

$$\operatorname{Pool}_\lambda(p^n) = \frac{\sum_{i=1}^k \sum_{j=1}^k \pi_{v^n, \lambda, i, j}^n p_{ij}^n}{\sum_{i=1}^k \sum_{j=1}^k \pi_{v^n, \lambda, i, j}^n} \quad (10)$$

In Appendix A, Figure A1 illustrates an example of the different steps of the INT-FUPooling operation, starting from a patch extracted from a set of maps with a shape of 3×3 and a number of filters equal to 3.

3.4. Fuzzy Average to Fuzzy System

The premature aggregation of the blocks (sum), after the immediate application of intuitionistic functions, makes the direct clarification of the fuzzy process almost impossible. But this does not prevent the mathematical rules governing the fuzzy averaging of the various tasks from being spelled out in equation form. Indeed, $\forall n = 1, \dots, z$, the transformation of the bloc $p^n = (p_{ij}^n)_{i,j=1,\dots,k}$ into a single crisp value can be seen as the calculation of a fuzzy average of the set P^n that implements the membership $\mu_{v,\lambda}$ determined by Equation (9). In this sense, each patch n is governed by the following mathematical rule:

$$\begin{aligned} \text{IF } (\mathbf{1} * \chi_{1,\lambda}(p^n)) \vee (\mathbf{1} * \chi_{2,\lambda}(p^n)) \vee (\mathbf{1} * \chi_{3,\lambda}(p^n)) &= \mathbf{1} * \chi_{v,\lambda}(p^n) \\ \text{THEN } \operatorname{Pool}_\lambda(p^n) &= \frac{\chi_{v,\lambda}(p^n) * p^n}{\mathbf{1} * \chi_{v,\lambda}(p^n)}, \end{aligned} \quad (11)$$

where $*$ is the convolution operator, $\chi_{v,\lambda}(p^n) = (\mu_{v,\lambda}(p_{ij}^n))_{i,j=1,\dots,k}$, $\mathbf{1} = (1)_{i,j=1,\dots,k}$, and $\vee$ is the max logical operator. As the proposed method performs several aggregations (fuzzy summations before reaching the conclusion part), it is difficult to extract the fuzzy rules governing the proposed intuitionistic pooling operation.

In order to transform the fuzzy averaging procedure into an explicit fuzzy system (with inputs, outputs, rules, fuzzification, and defuzzification operators), we consider a sample of N images, which we break down into $k \times k$ blocks. Then, we perform intuitionistic pooling for each block, forming an input–output dataset (vectors formed by the $k \times k$ components, intuitionist pooling values). Next, we use enhanced self-generated dynamic fuzzy Q-learning (EDSGFQL) to systematically construct fuzzy inference systems (FISs) [55]. In the EDSGFQL process, the structural recognition and parameter approximation of SIFs are carried out using fuzzy cmeans [56] for grouping the input data space when generating SIFs. Meanwhile, the frame and preconditioning components of an SIF are created by reinforcement learning, i.e., the fuzzy rules are tuned and deleted based on the reinforcement signals. In this sense, $\forall n = 1, \dots, z$, and for $k = 2$, we obtain the sub-fuzzy-system presented in Figure 2. In this figure, $p_{i,j}^n$, $p_{i,j+1}^n$, $p_{i+1,j}^n$, and $p_{i+1,j+1}^n$ represent the components of the patch n .

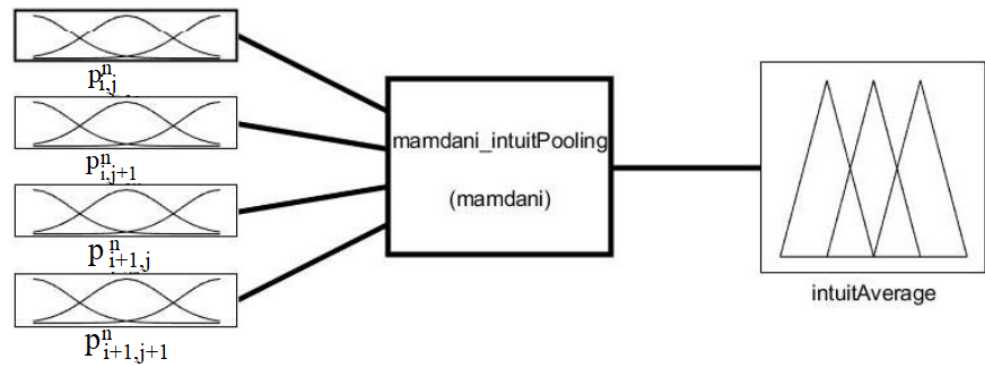


Figure 2. Intuitionistic fuzzy pooling system for $k = 2$ associated with patch n .

In the following, we give different components of the intuitionistic fuzzy system given in Figure 2:

[System]: Fis type = 'mamdani'; NumInputs = 4; NumOutputs = 1; NumRules = 3; AndMethod = 'min'; OrMethod = 'max'; ImpMethod = 'min'; AggMethod = 'max'; DefuzzMethod = 'centroid'.

[Input1 $p_{i,j}^n$]: Range = [0.057 0.897]; MF1 = ['Low', 'gaussmf', sd = 0.0375, mean = 0.447]; MF2 = ['Medium', 'gaussmf', sd = 0.034, mean = 0.484]; MF3 = ['High', 'gaussmf', sd = 0.042, mean = 0.531].

[Input2 $p_{i,j+1}^n$]: Range = [0.034 0.992], MF1 = ['Low', 'gaussmf', sd = 0.037, mean = 0.448]; MF2 = ['Medium', 'gaussmf', sd = 0.034, mean = 0.482]; MF3 = ['High', 'gaussmf', sd = 0.042, mean = 0.530].

[Input3 $p_{i+1,j}^n$]: Range = [0.050 1], MF1 = ['Low', 'gaussmf', sd = 0.037, mean = 0.448]; MF2 = ['Medium', 'gaussmf', sd = 0.034, mean = 0.482]; MF3 = ['High', 'gaussmf', sd = 0.042, mean = 0.530].

[Input4 $p_{i+1,j+1}^n$]: Range = [0 0.93], MF1 = ['Low', 'gaussmf', sd = 0.037, mean = 0.447]; MF2 = ['Medium', 'gaussmf', sd = 0.034, mean = 0.482]; MF3 = ['High', 'gaussmf', sd = 0.043, mean = 0.532].

[output intuitAverage]: Range = [0.227 1.559]; MF1 = ['Low', 'gaussmf', sd = 0.025, mean = 0.447], MF2 = ['medium', 'gaussmf', sd = 0.023, mean = 0.482]; MF3 = ['High', 'gaussmf', sd = 0.032, mean = 0.530].

where sd represents the standard deviation of different Gaussian membership functions and MF is the abbreviation of the membership function.

[Rules]:

1. If ($p_{i,j}^n$ is Low) and ($p_{i,j+1}^n$ is Low) and ($p_{i+1,j}^n$ is Low) and ($p_{i+1,j+1}^n$ is Low), then (intuitAverage is Low) (1).
2. If ($p_{i,j}^n$ is Medium) and ($p_{i,j+1}^n$ is Medium) and ($p_{i+1,j}^n$ is Medium) and ($p_{i+1,j+1}^n$ is Medium), then (intuitAverage is Medium) (1).
3. If ($p_{i,j}^n$ is High) and ($p_{i,j+1}^n$ is High) and ($p_{i+1,j}^n$ is High) and ($p_{i+1,j+1}^n$ is High), then (intuitAverage is High) (1).

It should be noted that all the rules have the same weight, equal to 1. In addition, the components of this system will be modified if we enrich the learning dataset with other images.

3.5. Optimal Control Problem to Train Deep Neural Network

Given a set of labeled images $(I_1, l_1), \dots, (I_M, l_M)$, in order to automate the prediction of the label of each image, we build a CNN with P convolution layers $C_1, \dots, C_P$, such as $C_p = \text{Conv}_p \oplus \text{ReLU} \oplus \text{Pool}$, whose last layer is connected to a multilayer perceptron $\text{MLP}(\theta)$, where θ is the matrix of the connection between its different layers. The primary objective of learning is to shorten the distance between the predicted label of a given input

image and the target label. Considering the collection of training and labeled image from $\mathcal{I} \times \mathcal{L}$, Golobal's loss function, e.g., root mean square error (RMSE), is given in [57] by:

$$L(\theta, \mathcal{I}) = \frac{1}{M} \sum_{i=1}^M (MLP(\theta) \otimes \mathcal{V}(I_i) - l_i)^2 \quad (12)$$

where $\mathcal{V}(I_i) = \bigotimes_{p=1}^P C_p(I_i)$ and θ is the matrix of connection between neurons of the MLP component of CNN. To minimize the loss L , given by Equation (12), the backpropagation (BP) algorithm based on stochastic gradient descent is implemented to update the weights at the $k + 1$ th step as follows:

$$\theta_{ij}^{k+1} = \theta_{ij}^k - \alpha \frac{\partial L(\theta^k, I_m)}{\partial \theta_{ij}} \quad (13)$$

where I_m is a random image from $\mathcal{I}$, and α is the time step. If α is sufficiently small, one can perform the following approximation:

$$\dot{\theta}_{ij} = - \frac{\partial L(\theta^k, I_m)}{\partial \theta_{ij}} \quad (14)$$

Let $\{I_{m_1}, \dots, I_{m_T}\}$ be a time series of images uniformly generated from $\mathcal{I}$, and $\{\theta_{m_1}, \dots, \theta_{m_T}\}$ a time of the MLP weights associated with these images. The aim behind training the CNN via BP is to tune all the parameters and therefore optimize the loss function. In this sense, the problem of training CNN can be reformulated as an optimal control problem [58,59]:

$$(P) : \begin{cases} J(\theta, \mathcal{I}) = \frac{1}{M} \int_{t_0}^{t_f} \sum_{i=1}^M (MLP(\theta) \otimes \mathcal{V}(I_i) - l_i)^2 dt \\ \text{Subject to :} \\ \dot{\theta}_{ij} = -u \frac{\partial L}{\partial \theta_{ij}} \end{cases} \quad (15)$$

The control u is nothing but the time step of the BP algorithm. We can use Pontryagin's Minimum Principle [59] or a local search method [60] to solve the problem (P).

4. Experimental Setup and Results

4.1. Metrics

The measures used in this paper are used to calculate the quality of compressed images and how many are similar to the original image. In general, the results of the measures on datasets are obtained by calculating the average of measures taken over all the images.

Mean Squared Error (MSE) measures the average squared difference between the pixels of two images; it strongly depends on the image intensity scaling:

$$MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (I_{ij}^1 - I_{ij}^2)^2, \text{ where } (M \times N) \text{ is the size of both input images } I^1 \text{ and } I^2.$$

A lower MSE value indicates high similarity.

Peak Signal-to-Noise Ratio (PSNR) is the ratio between the maximum possible power of an image and the power of corrupting noise that affects the quality of its representation [61]. $PSNR = 10 \log_{10}(\frac{R^2}{MSE})$, $R = 2^n - 1$ ($n = 8, 8$ bits); let R denote the maximum pixel value of the original image, which is set at $R = 255$.

A higher PSNR value indicates higher image quality, while a lower value suggests a significant difference between the images.

This metric is typically used to evaluate the quality of reconstructed or compressed images.

Structural Similarity Index Measure (SSIM) [61] is a well-known metric used for measuring the similarity, based on three factors: loss of correlation, luminance distortion and contrast distortion. SSIM varies between -1 and 1 , where 1 indicates perfect similarity.

Root Mean Squared Error (RMSE) is the square root of the Mean Squared Error (MSE) and calculates the differences between values predicted values and the true values. RMSE, similarly to MSE, is frequently used as a metric for evaluating the quality of predictions from a model, the smaller value referring to the better performance of models or similarity for images.

Signal-to-Reconstruction Error (SRE) is a metric widely used to measure the quality of speech signals. It can also be used to measure quality of images. SRE measures the errors related to the mean image intensity and is relevant for making errors comparable between images that have different brightness levels [62]. Higher SRE values indicate better quality.

Feature-Based Similarity Index (FSIM) [63] is a measure to show the similarity rates between two images based on two basic features: the primary feature Phase Congruency (PC) and the second feature Gradient Magnitude (GM) [64]. The value of FSIM ranges between 0 and 1 , a high value referring to similar images.

Universal image quality index (UIQ) is designed by modeling any image distortion as a combination of three factors: loss of correlation, luminance distortion, and contrast distortion [65]. Its values range from -1 to 1 , values approaching 1 in the case of more similar images.

4.2. Data Sets

4.2.1. MNIST Dataset

MNIST is a collection of 70,000 divided into two parts: the first contains 60,000 images for training, while the second part contains 10,000 testing images. This collection of images consists of handwritten digits from 0 to 9 with a size of 28×28 as shown in Figure 3.



Figure 3. Examples of images for 10 classes of MNIST dataset [66].

4.2.2. Fashion MNIST Dataset

The fashion MNIST dataset contains 70,000 images of clothes, each image being a 28×28 grayscale of 10 categories of fashion products ("T-shirt", "Trouser", "Pullover", "Dress", "Coat", "Sandal", "Shirt", "Sneaker", "Bag", "Ankle boot"). The fashion MNIST

dataset Figure 4 is divided into 60,000 for training datasets and 10,000 for testing, and every class of images contains 7000 images.



Figure 4. Examples of images for 10 classes of Fashion MNIST [67].

4.3. Image Reconstruction: INT-FUP vs. Classical Pooling Models

The pooling operation reduces the dimensionality of images by summarizing a block of information into a single value. The reverse operation consists of reconstructing the images by using only the values obtained through aggregation. In this section, the INT-FUP model is tested against average pooling, max pooling, and fuzzy pooling using some of the image similarity measures mentioned earlier. Table 1 shows the high performance of our proposed method over all measures metric, and the evaluation performed for 1500 images of handwritten digits. The measure in Table 1 shows the similarities and differences between the real images and the images after pooling, in the second table, there are results of measures between pooling and depooling images. We will also show pooling experimental using various types of noises.

In order to evaluate the efficacy of our model, we conducted experiments on two widely accessible National Institute of Standards and Technology (MNIST) datasets: handwritten digits-MNIST [66] and Fashion-MNIST [67] datasets.

Table 1. Comparison of four pooling methods using a metric measure (Rmax = 6) on MNIST dataset of handwritten digits.

Method	PSNR	FSIM (e^{-2})	UIQ	MSE	SSIM (e^{-5})	SRE	RMSE (e^{-5})
Max pooling	86.32	88.16	0.28	5.98	99,998.55	512.78	4.88
Average Pooling	91.746	95.18	0.715	4.30	99,999.861	5.495	2.619
Fuzzy pooling	91.742	95.26	0.719	4.18	99,999.865	15.493	2.621
Intuitionistic fuzzy	91.764	95.25	0.72	4.23	99,999.866	15.503	2.614

The experiments presented in Table 1 were performed using the same set of parameters as mentioned in Section 3, involving a comparison of four pooling approaches. The results from the experiments provide clear evidence that intuitionistic fuzzy pooling outperforms the other pooling methods. This superiority is evident across multiple measure scales, such as PSNR, FSIM, UIQ, SSIM, and SRE, where intuitionistic pooling consistently achieves higher values, indicating better similarity when compared to the other pooling methods. Although there is a slight difference favoring fuzzy pooling in the case of the FSIM measure,

intuitionistic pooling outperforms the other methods in all other measures. Furthermore, intuitionistic pooling demonstrates a lower RMSE value, indicating a small difference compared to alternative pooling methods.

Note: Compared with fuzzy polishing, Int-FUP slightly improves PNSR with 0.022, UIQ with 0.001, MSE with 0.05, SSIM with e^{-7} , and SRE with 0.01. To highlight the ability of the hesitation membership function to process the main information in the stochastic environment, the reduction and reconstruction operator will be performed on noisy images in the next section.

4.4. Noisy Image Reconstruction: Int-FUP vs. Classical Pooling Models

To demonstrate the ability of the hesitation membership function to retain key information in blurred environments, the reduction and reconstruction operations are applied to noisy images in this section using max, min, average, random, fuzzy, and intuitionistic pooling. The noise types considered here are Gaussian, Poisson, Salt, Pepper, and Speckle. In addition to the MNIST and Fashion datasets, we use another public dataset available at UCI [68], and we compare the pooling operators on RMSE and PSNR performance measures.

To challenge the pooling approaches, we add different degrees of noise to the images, resulting in four data images: Images + Gauss, Images + Poisson, Images + Salt_Pepper, and Images + Speckle.

Table 2 gives the average RMSE for the different pooling methods used to compress the images, to which we have added four types of noise. We note that intuitionistic pooling has the lowest RMSE, far ahead of all traditional methods (max, min, average and rand). Furthermore, Int-FUP significantly outperforms fuzzy pooling and achieves a significant improvement in terms of RMSE: 1.519 for Gaussian noise, 1.573 for Poisson noise, 0.999 for Salt-Pepper, and 1.072 for Speckle noise. In this sense, the proposed method improves fuzzy clustering by around 20.20%.

Table 2. Average RMSE of the different pooling methods used to compress the different images to which we added four types of noise.

Images + Noise	RMSE_Max	RMSE_Avg	RMSE_Fuzz	RMSE_Intuit	RMSE_Min	RMSE_Rand
Images + Gauss	44.285	29.171	9.136	7.617	37.356	40.208
Images + Poisson	43.388	29.572	9.359	7.786	37.051	40.068
Images + Salt-Pepper	38.427	28.408	8.309	7.310	36.177	39.746
Images + Speckle	31.943	28.040	7.788	6.716	35.015	33.774

Table 3 gives the average PSNR for the different pooling methods used to compress the various images considered, to which we add four types of noise. We can see that intuitionistic pooling has by far the best PSNR of all the traditional methods (max, min, average, and rand). Moreover, Int-FUP far outperforms fuzzy pooling and achieves a significant improvement in term of PSNR: 0.123 for Gaussian noise, 0.083 for Poisson noise, 0.088 for Salt-Pepper, and 0.093 for Speckle noise. In this sense, in terms of PSNR, the proposed method improves fuzzy pooling by almost 37%.

Table 3. Average PSNR of the different pooling methods used to compress the different images to which we added four types of noise.

Images + Noise	PSNR_Max	PSNR_Avg	PSNR_Fuzz	PSNR_Intuit	PSNR_Min	PSNR_Rand
Images + Gauss	0.199	0.158	0.208	0.331	0.072	0.104
Images + Poisson	0.255	0.215	0.268	0.351	0.121	0.150
Images + Salt-Pepper	0.255	0.213	0.265	0.353	0.11632	0.159
Images + Speckle	0.290	0.276	0.283	0.376	0.198	0.214

The cause of this success is the ability of intuitionistic logic to quantify the degree of hesitation thanks to the non-membership function which implements the epsilon parameter.

We notice that in order to deal with uncertainty, this parameter must be increased largely in comparison with the case where there is no noise.

To study the sensitivity of the fuzzy and intuitionistic pooling methods, while considering different noises, we represent, in terms of boxes with whiskers, the two series RMSE and PSNR, associated with the compressions of all the images and all the noises; see Figures 5–7.

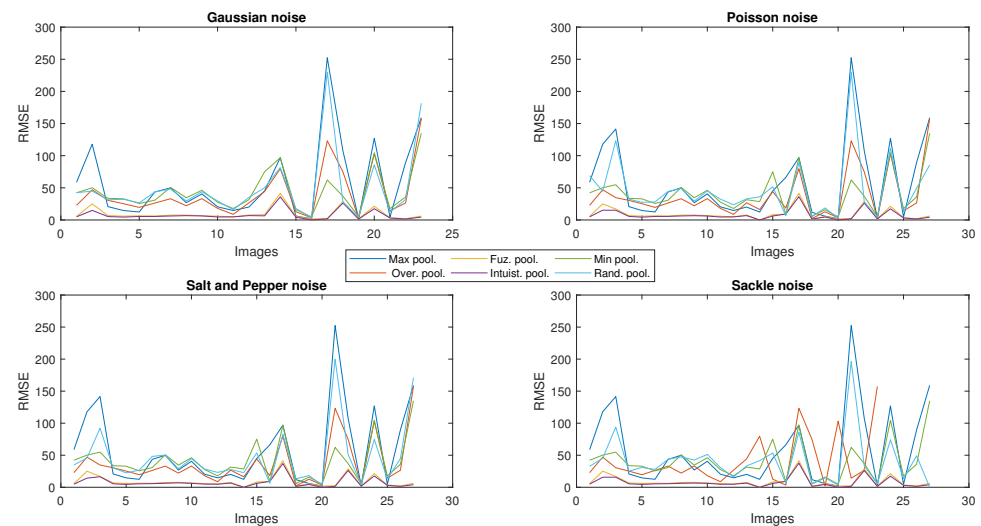


Figure 5. RMSE of pooling operators for different types of noises.

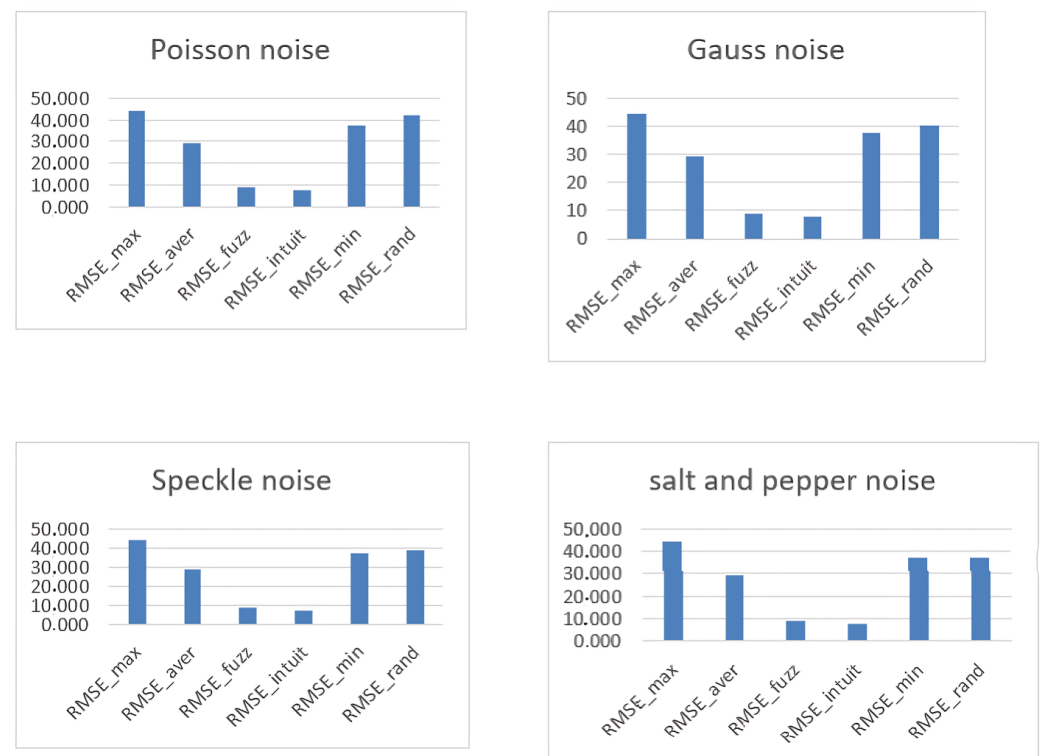


Figure 6. Average RMSE of the different pooling methods used to compress the different images to which we added the Poisson noise, Gaussian noise, Speckle noise, and Salt–Pepper noise.

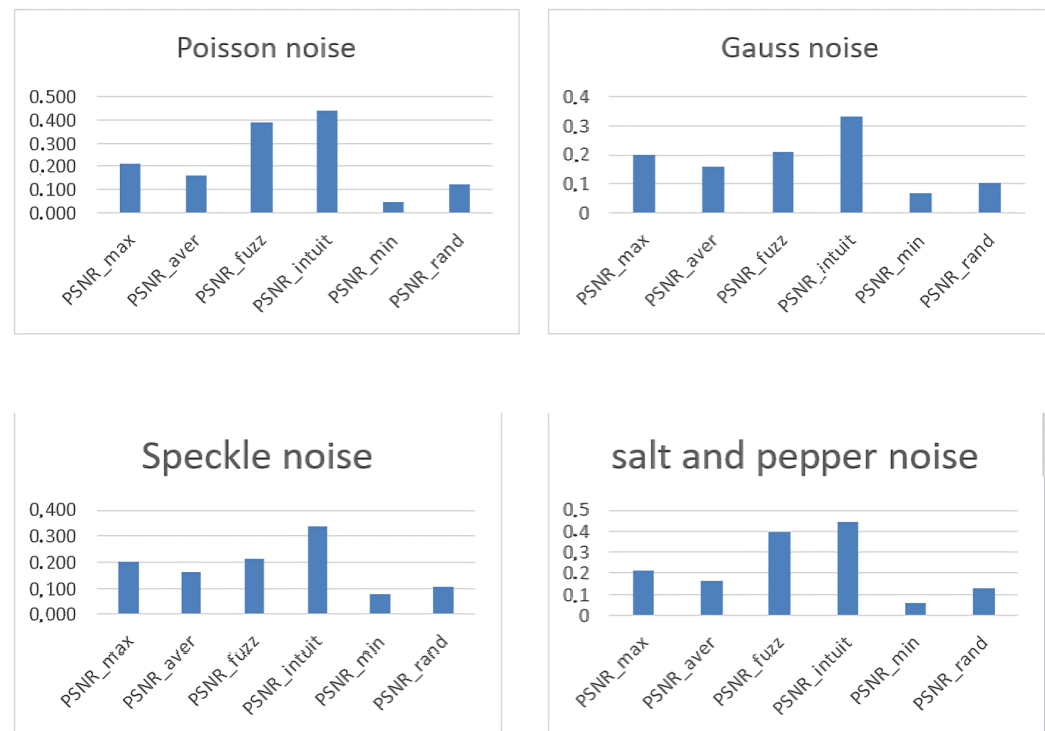


Figure 7. Average PSNR of the different pooling methods used to compress the different images to which we added the Poisson noise, Gaussian noise, Speckle noise, and Salt–Pepper noise.

Figure 8 shows the fuzzy and intuitionistic pooling whiskers of the RMSEs series corresponding to the four noises. We notice that the boxes associated with intuitionistic pooling are below the boxes associated with fuzzy pooling. Moreover, the boxes associated with Int-FUP are small compared to those of fuzzy pooling, which means that the quartile ranges and deciles of the RMSEs-Int-FUP series are very small and therefore have a low sensitivity in comparison to fuzzy pooling (in the RMSE sense).

Figure 9 shows the fuzzy and intuitionistic pooling whiskers of the PSNRs series corresponding to the four noises. We notice that the boxes associated with intuitionistic pooling are above the boxes associated with fuzzy pooling. Moreover, the boxes associated with intuitionistic pooling are considerable compared to those of fuzzy pooling, which means that the quartile ranges and deciles of the PSNRs–intuitionistic pooling series are better; therefore, intuitionistic pooling has a low sensitivity compared to fuzzy pooling (in the sense of PSNR).

Towards the end, the fact of considering non-membership functions endows intuitionistic logic with a higher capacity to reason correctly (in comparison with fuzzy logic) in stochastic environments.

An experiment on Int-FUP is performed to show the impact of noises used in the experiments above. As shown in Figure 10, Poisson noise has high RMSE values; as a result, Int-FUP performs poorly in a noisy Poisson environment, while the results in other noisy environments remain acceptable.

Note: It should be noted that the inclusion of a non-adhesion function leads to a slight increase in the processor operating time. Indeed, to perform the pooling operation, fuzzy pooling requires 0.2381375 s, max pooling requires 0.002265 s, average pooling requires 0.02757 s, intuitionistic pooling requires 0.2529685 s, and random pooling requires 0.0089365 s. We can reduce the CPU time corresponding to the intuitionistic pooling operation by performing a parallel calculation, as the membership and non-membership functions work with the same input data.

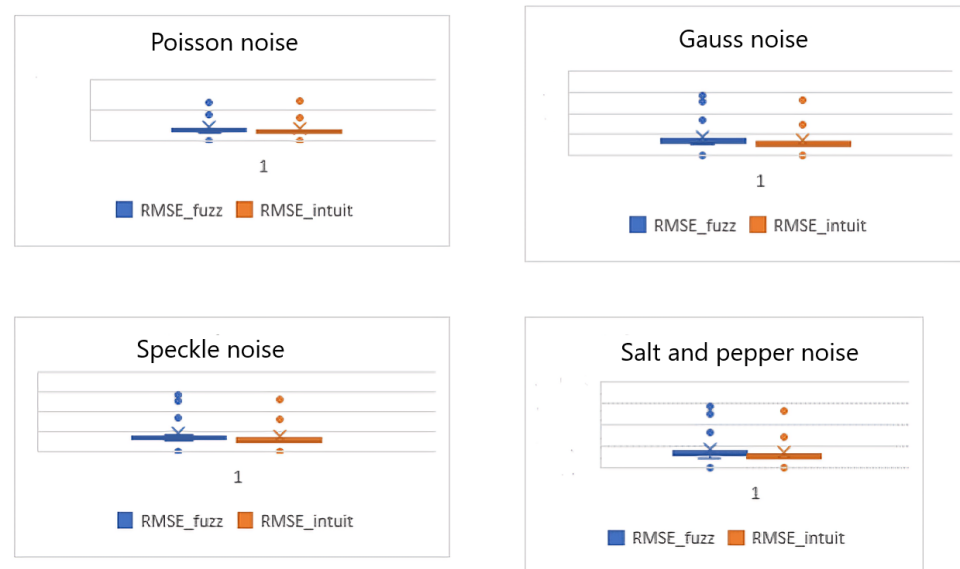


Figure 8. Box plots of the average RMSE of the two pooling methods, fuzzy and intuitionistic, used to compress the different images to which we added the Poisson noise, Gaussian noise, Speckle noise, and Salt–Pepper noise.

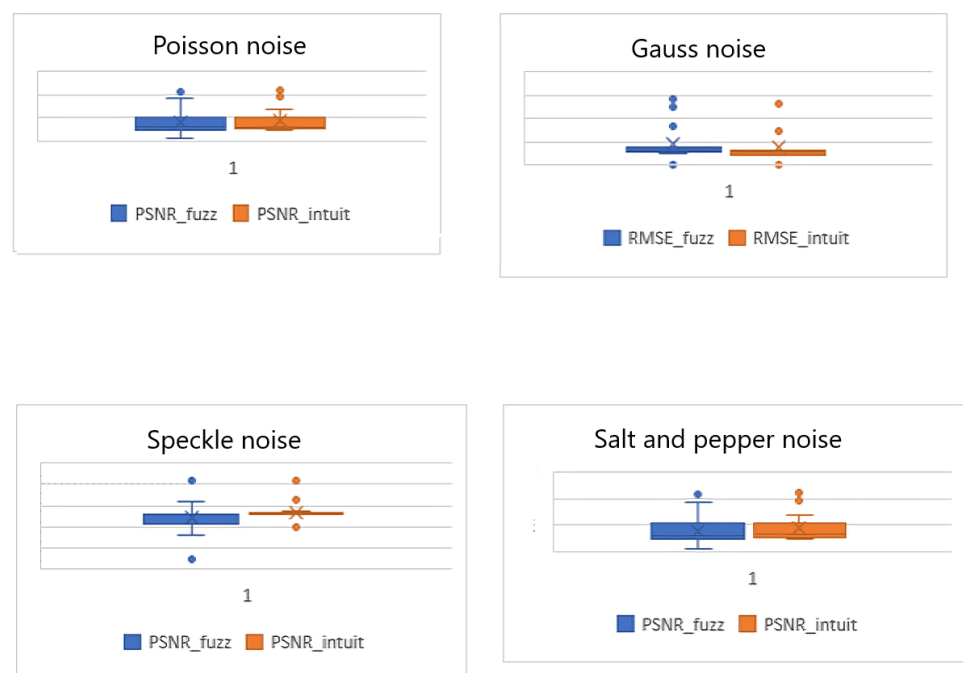


Figure 9. Box plots of the average PSNR of the two pooling methods, fuzzy and intuitionistic, used to compress the different images to which we added the Poisson noise, Gaussian noise, Speckle noise, and Salt–Pepper noise.

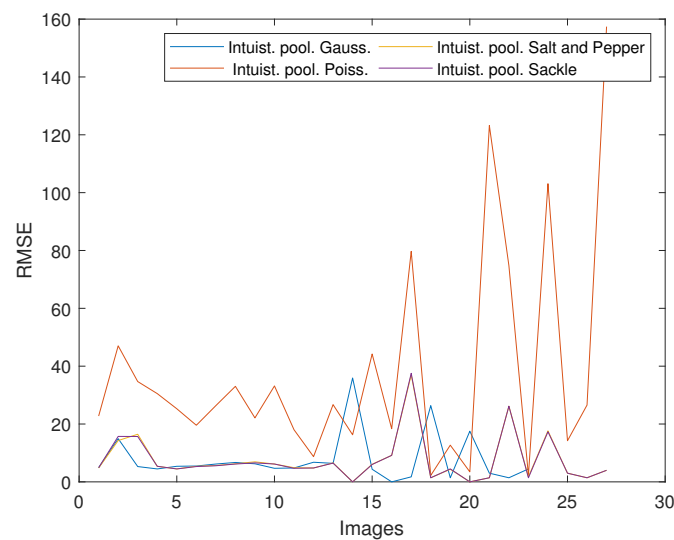


Figure 10. RMSE of intuitionistic pooling operator for different types of noises.

4.5. Deep CNN Classification: INT-FUP-CNN vs. Classical Pooling-CNN

To perform the classification task via CNN with intuitionistic pooling, several layers [convolution operator + relu + intuitionistic pooling] are introduced: the convolution layers extract relevant information from images using an appropriate number of masks; the Rectified Linear Unit (ReLU) function is used to correct the values of each patch (transforms negative grey levels into 0); and intuitionistic fuzzy clustering reduces the image size to optimize the CNN architecture. Next, the final matrix is flattened to transform the matrix into vectors. The final convolution layers are connected to a fully connected artificial neural network [69–72], which consists of three types of layers: the input layer, the hidden layers, and the decision layer (or classification layer). The input layer performs no processing but represents the resulting vectors to the neural network. The hidden layers perform almost all the processing so that the neural network can store the images in the image dataset. The decision layer, equipped with an appropriate activation function, makes the classification decision in terms of probability.

The pooling methods are mainly used in CNN to reduce computational complexity and memory requirements, and this section aims to show the performance of our proposed method compared to other common pooling approaches. Table 4 shows the result obtained for different evaluation metrics for one epoch on the MNIST dataset, where the results of our model clearly exceed the other pooling methods, along with Figure 11 that shows the AUC-ROC (Area Under The Curve–Receiver Operating Characteristics) curve performance when compared to the other approaches.

Table 4. Comparison results of 4 types of pooling on MNIST dataset of handwritten digits.

Method	Accuracy	Precision	Recall	AUC-ROC	F1 Score	Cohens Kapp	g-Mean
Max Pooling (%)	82.03	85.79	81.91	89.96	82.30	80.03	89.66
Average pooling (%)	90.76	91.06	90.69	94.83	90.64	89.73	94.78
Fuzzy pooling (%)	91.99	92.07	91.97	95.54	91.91	91.10	95.48
Intuitionistic Fuzzy pooling (%)	94.51	94.52	94.45	96.92	94.46	93.90	96.91

In this experiment, various performance measures are employed to compare the effectiveness of our model against commonly used pooling methods such as Max and Average pooling, as well as the fuzzy pooling method, which our model extends. The outcomes of our model are presented in Tables 4 and 5, thus demonstrating the superior performance of our proposed model.

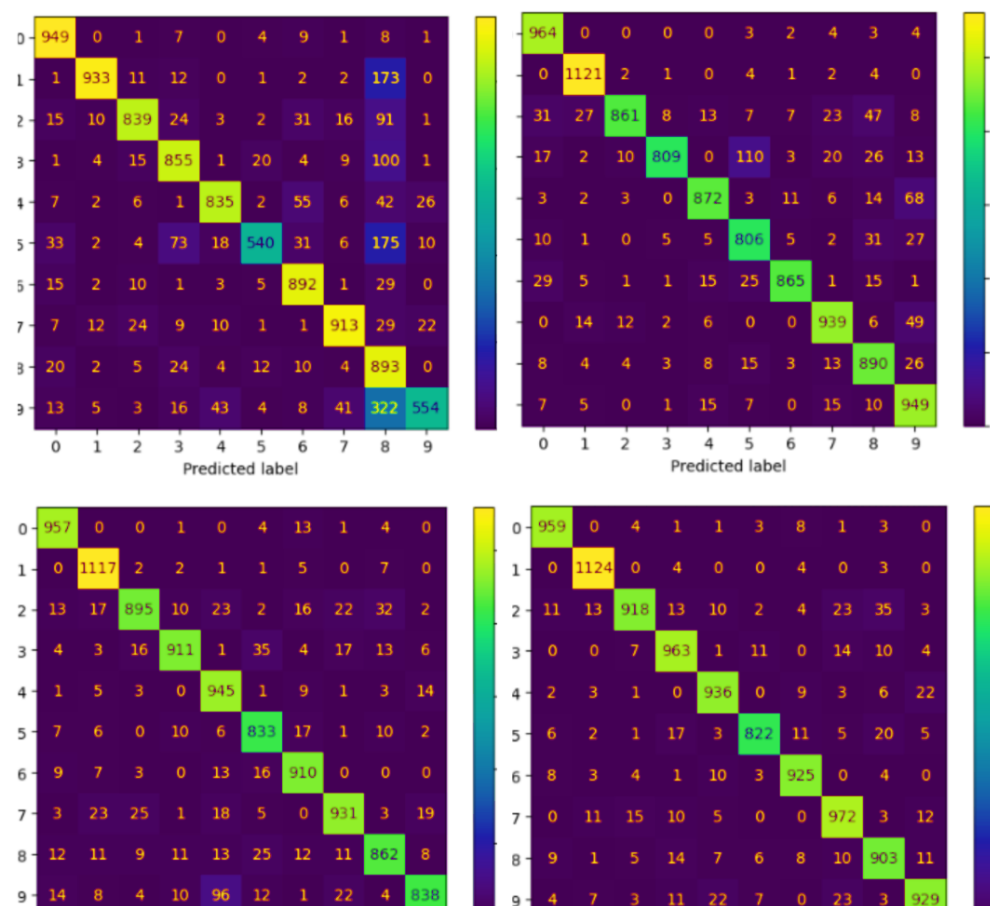
Table 5. Comparison results of 4 types of pooling on the MNIST dataset of Fashion mnist.

Method	Accuracy	Precision	Recall	AUC-ROC	F1 Score	Cohens Kapp
Max Pooling (%)	84.5	84.78	84.17	84.53	82.77	91.12
Average pooling (%)	87.06	88.48	87.05	87.05	85.62	92.63
Fuzzy pooling (%)	89.25	89.23	89.14	89.15	88.05	93.85
Intuitionistic Fuzzy pooling (%)	91.29	91.55	91.18	91.19	90.32	95.08

In Figure 11, we can observe the confusion matrices representing the performance of the proposed model on the MNIST handwritten digits dataset. The top row showcases images representing the “Max” and “Average” approaches, where the image on the right side corresponds to the “Average” approach. In the bottom row, we have matrices illustrating the “Fuzzy” and “Intuitionistic” approaches, with the matrix on the right side representing the “Intuitionistic” approach.

By comparing these matrices, it is evident that the proposed model demonstrates a higher level of confidence in classifying the images. This suggests that the proposed model shows improved accuracy and effectiveness in accurately identifying and categorizing the handwritten digits, which is clearly demonstrated in both Tables 4 and 5.

Moreover, Figure 12 shows the performance of our proposed method through the AUC-ROC curve, thereby pointing out the effectiveness of the fuzzy and intuitionistic pooling approach. In order to further highlight the capabilities of our model, in the next section, we introduce additional noise to the images, demonstrating its robustness and enhanced performance on fuzzy sets.

**Figure 11.** Confusion matrices of max, average, fuzzy and intuitionistic approaches on mnist handwritten digits.

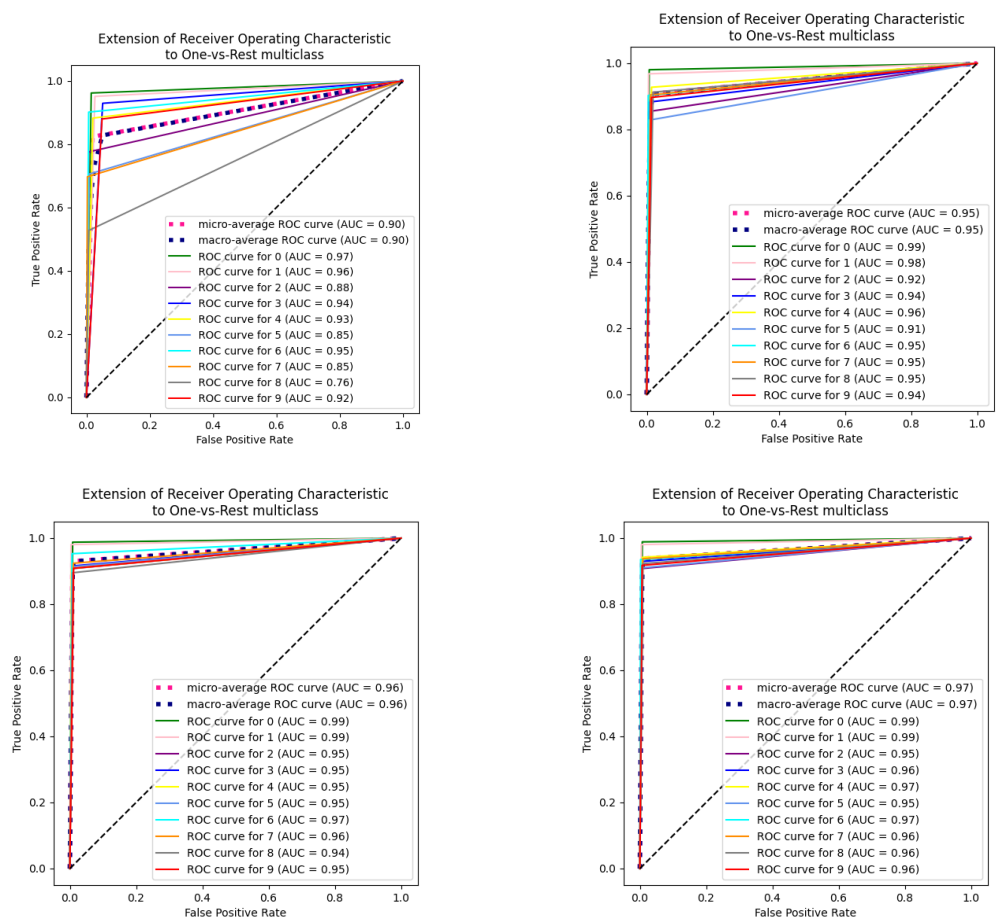


Figure 12. AUC-ROC curve results of Max-pooling, Avg-pooling, fuzzy pooling and intuitionistic pooling.

5. Conclusions

In this work, the authors introduced an original intuitionistic pooling operation for CNN architectures to handle uncertainty in the characteristic values. From the experiments, it can be seen that the proposed operator significantly improves the efficiency of CNN classification when compared to other existing clustering operators. As a result, we show that the use of intuitionistic aggregation, instead of the best-known clustering methods, in CNN improves the generalization ability of the resulting system. In addition, experience indicates that the suggested model better retains the relevant characteristics of the pool regions. This has been validated on the basis of well-known performance measures.

Selecting the most active blocks on the basis of a global comparison, using Equation (9), neglects the local information that may characterize certain regions, resulting in a certain loss of information in fuzzy pooling.

It is possible to use genetic algorithms to select the optimal parameters of the membership and non-membership functions and the parameters of the first aggregation based on a representative mini-batch. Other promising avenues of research concern the extension of the proposed model to neutrosophic sets.

Author Contributions: Conceptualization, C.R. and K.E.M.; methodology, K.E.M.; software, C.R.; validation, A.-M.P., A.Y. and J.R.; writing—original draft preparation, C.R.; writing—review and editing, A.-M.P.; supervision, K.E.M., A.Y. and J.R.; project administration, K.E.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset used in the experiments is publicly available in [66] and can be found at <http://yann.lecun.com/exdb/mnist/> (accessed on 1 September 2023) for the fashion dataset [67] can be found at <https://www.cs.595toronto.edu/kriz/cifar.html> (accessed on 1 September 2023).

Acknowledgments: The authors thank all those who contributed to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IFS Intuitionistic fuzzy sets
CNNs Convolutional Neural Networks

Appendix A

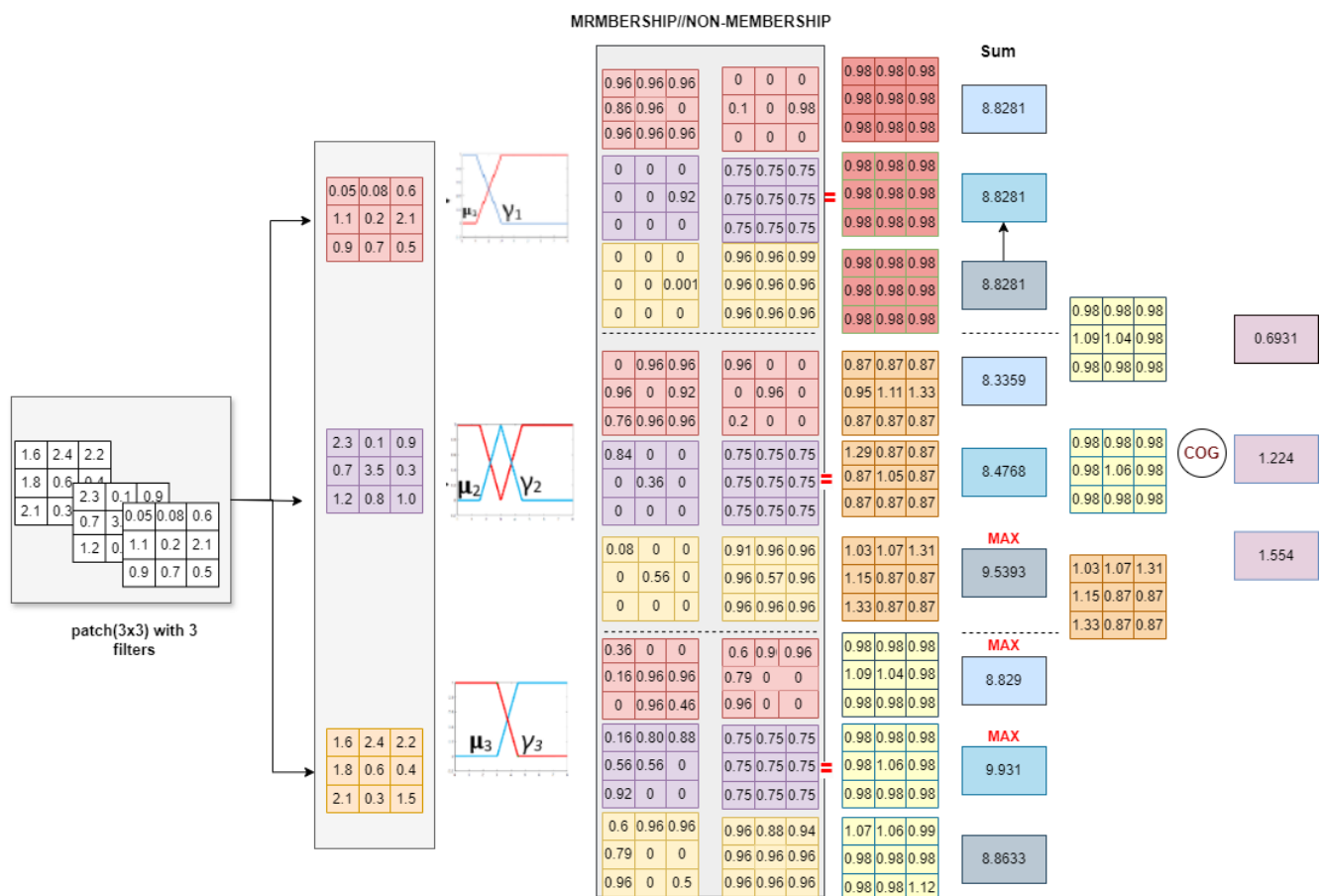


Figure A1. Numeric example of the proposed INT-FU Pooling operation applied to a 3 × 3 patch with 3 filters.

In Figure A1, we illustrate an example of the different steps of the INT-FUPooling operation, starting from a patch extracted from a set of maps with a shape of 3 × 3 and a number of filters equal to 3. The patch is then processed through intuitionistic membership functions to obtain membership and non-membership values. This operation is known as fuzzification. From each filter in the image patch, we obtain three blocks for membership and three blocks for non-membership. As the final step of fuzzification, we apply Equation (6). In the next step, defuzzification is performed by calculating the sum of the values for each patch obtained from fuzzification. Then, we select the patch with the

highest sum. Finally, we apply the center of gravity method as indicated in Equation (10) to obtain a single output value for each single input filter.

References

- Li, Y. Research and application of deep learning in image recognition. In Proceedings of the IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 21–23 January 2022; pp. 994–999.
- Oord, A.V.; Dieleman, S.; Schrauw, B. Deep content-based music recommendation. In Proceedings of the Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013.
- Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
- Avilov, O.; Rimbart, S.; Popov, A.; Bougrain, L. Deep learning techniques to improve intraoperative awareness detection from electroencephalographic signals. In Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 142–145.
- Tsantekidis, A.; Passalis, N.; Tefas, A.; Kannianen, J.; Gabbouj, M.; Iosifidis, A. Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks. In Proceedings of the 2017 IEEE 19th Conference on Business Informatics (CBI), Thessaloniki, Greece, 24–27 July 2017; pp. 7–12.
- Atanassov, K.T. Intuitionistic fuzzy sets. In *Fuzzy Sets and Systems*; Physica Verlag: New York, NY, USA, 1986; pp. 87–96.
- Atanassov, K.T. Intuitionistic fuzzy sets: Past, present and future. In Proceedings of the EUSFLAT Conference 2003, Zittau, Germany, 10–12 September 2003; pp. 12–19.
- Atanassov, K.T. *Intuitionistic Fuzzy Sets: Theory and Applications*, 1st ed.; Physica Verlag: New York, NY, USA, 1999; pp. 1–137.
- Lei, Y.; Hua, J.; Yin, H.; Lei, Y. Normal Technique for Ascertaining Non-membership functions of Intuitionistic Fuzzy Sets. In Proceedings of the IEEE Chinese Conference on Control and Decision, Yantai, China, 2–4 July 2008; pp. 2604–2607.
- Nirthika, R.; Manivannan, S.; Ramanan, A.; Wang, R. Pooling in convolutional neural networks for medical image analysis: A survey and an empirical study. *Neural Comput. Appl.* **2022**, *34*, 5321–5347. [[CrossRef](#)] [[PubMed](#)]
- Lee, C.Y.; Gallagher, P.W.; Tu, Z. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In Proceedings of the 18th International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, 9–11 May 2016; pp. 464–472.
- Chen, J.; Hua, Z.; Wang, J.; Cheng, S. A Convolutional Neural Network with Dynamic Correlation Pooling. In Proceedings of the 2017 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, China, 15–18 December 2017; pp. 496–499.
- Wei, Z.; Zhang, J.; Liu, L.; Zhu, F.; Shen, F.; Zhou, Y.; Liu, S.; Sun, Y.; Shao, L. Building Detail-Sensitive Semantic Segmentation Networks with Polynomial Pooling. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7108–7116.
- Gülçehre, Ç.; Cho, K.; Pascanu, R.; Bengio, Y. Learned-Norm Pooling for Deep Feedforward and Recurrent Neural Networks. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, 15–19 September 2014; pp. 530–546.
- Bruna, J.; Szlam, A.; LeCun, Y. Signal recovery from pooling representations. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 307–315.
- Eom, H.; Choi, H. Alpha-Integration Pooling for Convolutional Neural Networks. *arXiv* **2022**, arXiv:1811.03436.
- Shi, Z.; Ye, Y.; Wu, Y. Rank-based pooling for deep convolutional neural networks. *Neural Netw.* **2016**, *83*, 21–31. [[CrossRef](#)] [[PubMed](#)]
- Navaneeth, B.; Suchetha, M. A dynamic pooling based convolutional neural network approach to detect chronic kidney disease. *Biomed. Signal Process. Control* **2020**, *62*, 102068. [[CrossRef](#)]
- Bieder, F.; Sandkühler, R.; Cattin, P.C. Comparison of Methods Generalizing Max- and Average-Pooling. *arXiv* **2021**, arXiv:2103.01746.
- Stergiou, A.; Poppe, R.; Kalliatakis, G. Refining activation downsampling with softpool. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10337–10346.
- Czaja, W.; Li, W.; Li, Y.; Pekala, M. Maximal function pooling with applications. In *Excursions in Harmonic Analysis, Volume 6: In Honor of John Benedetto's 80th Birthday*; Birkhäuser: Cham, Switzerland, 2021; pp. 413–429.
- Kumar, A. Ordinal pooling networks: For preserving information over shrinking feature maps. *arXiv* **2018**, arXiv:1804.02702.
- Zhao, Q.; Lyu, S.; Zhang, B.; Feng, W. Multiactivation pooling method in convolutional neural networks for image recognition. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 8196906. [[CrossRef](#)]
- Yu, D.; Wang, H.; Chen, P.; Wei, Z. Mixed pooling for convolutional neural networks. In Proceedings of the Rough Sets and Knowledge Technology: 9th International Conference, RSKT 2014, Shanghai, China, 24–26 October 2014; pp. 364–375.
- Tong, Z.; Aihara, K.; Tanaka, G. A hybrid pooling method for convolutional neural networks. In Proceedings of the Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, 16–21 October 2016; pp. 454–461.
- Zeiler, M.D.; Fergus, R. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv* **2013**, arXiv:1301.3557.
- Wu, H.; Gu, X. Max-pooling dropout for regularization of convolutional neural networks. In Proceedings of the Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, 9–12 November 2015; pp. 46–54.

28. Zhai, S.; Wu, H.; Kumar, A.; Cheng, Y.; Lu, Y.; Zhang, Z.; Feris, R. S3pool: Pooling with stochastic spatial sampling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4970–4978.
29. Graham, B. Fractional max-pooling. *arXiv* **2014**, arXiv:1412.6071.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 1904–1916. [[CrossRef](#)]
31. Wiliem, A.; Sanderson, C.; Wong, Y.; Hobson, P.; Minchin, R.F.; Lovell, B.C. Automatic classification of human epithelial type 2 cell indirect immunofluorescence images using cell pyramid matching. *Pattern Recognit.* **2015**, *47*, 2315–2324. [[CrossRef](#)]
32. Zhang, N.; Farrell, R.; Darrell, T. Pose pooling kernels for sub-category recognition. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3665–3672.
33. Qi, K.; Guan, Q.; Yang, C.; Peng, F.; Shen, S.; Wu, H. Concentric Circle Pooling in Deep Convolutional Networks for Remote Sensing Scene Classification. *Remote Sens.* **2018**, *10*, 934. [[CrossRef](#)]
34. Qi, K.; Yang, C.; Hu, C.; Guan, Q.; Tian, W.; Shen, S.; Peng, F. Polycentric circle pooling in deep convolutional networks for high-resolution remote sensing image recognition. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 632–641. [[CrossRef](#)]
35. Feng, J.; Ni, B.; Tian, Q.; Yan, S. Geometric ℓ_p -norm feature pooling for image classification. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 2609–2704.
36. Lin, T.Y.; Maji, S. Improved bilinear pooling with cnns. *arXiv* **2017**, arXiv:1707.06772.
37. Carreira, J.; Caseiro, R.; Batista, J.; Sminchisescu, C. Semantic segmentation with second-order pooling. In Proceedings of the Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 430–443.
38. Zeng, R.; He, J. Grouping Bilinear Pooling for Fine-Grained Image Classification. *Appl. Sci.* **2022**, *12*, 5063. [[CrossRef](#)]
39. Behera, A.; Wharton, Z.; Hewage, P.R.; Bera, A. Context-aware attentional pooling (cap) for fine-grained visual classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; pp. 929–937.
40. Chen, F.; Datta, G.; Kundu, S.; Beerel, P.A. Self-Attentive Pooling for Efficient Deep Learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 3974–3983.
41. Lee, J.; Lee, I.; Kang, J. Self-attention graph pooling. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 3734–3743.
42. Ranjan, E.; Sanyal, S.; Talukdar, P. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 5470–5477.
43. Wang, Y.; Hu, L.; Wu, Y.; Gao, W. Graph Multihead Attention Pooling with Self-Supervised Learning. *Entropy* **2022**, *24*, 1745. [[CrossRef](#)] [[PubMed](#)]
44. Wen, T.; Zhuang, J.; Du, Y.; Yang, L.; Xu, J. Dual-sampling attention pooling for graph neural networks on 3D mesh. *Comput. Methods Programs Biomed.* **2021**, *208*, 106250. [[CrossRef](#)]
45. Peng, W.; Hong, X.; Zhao, G. Tripool: Graph triplet pooling for 3D skeleton-based action recognition. *Pattern Recognit.* **2021**, *115*, 107921. [[CrossRef](#)]
46. Saeedan, F.; Weber, N.; Goesele, M.; Roth, S. Detail-preserving pooling in deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9108–9116.
47. Gao, Z.; Wang, L.; Wu, G. Lip: Local importance-based pooling. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3354–3363.
48. Saha, O.; Kusupati, A.; Simhadri, H.V.; Varma, M.; Jain, P. RNNPool: Efficient non-linear pooling for RAM constrained inference. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 20473–20484.
49. Diamantis, D.E.; Iakovidis, D.K. Fuzzy pooling. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 3481–3488. [[CrossRef](#)]
50. Radhika, C.; Parvathi, R. Intuitionistic fuzzification functions. *Glob. J. Pure Appl. Math.* **2016**, *12*, 1211–1227.
51. Anzilli, L.; Facchinetti, G. A new proposal of defuzzification of intuitionistic fuzzy quantities. In *Novel Developments in Uncertainty Representation and Processing: Advances in Intuitionistic Fuzzy Sets and Generalized Nets, Proceedings of the 14th International Conference on Intuitionistic Fuzzy Sets and Generalized Nets IWIFSGN@FQAS, Warsaw, Poland, 12–14 October 2016*; Springer International Publishing: Cham, Switzerland, 2016; pp. 185–195.
52. Ban, A.; Kacprzyk, J.; Atanassov, K. On de-I-fuzzification of intuitionistic fuzzy sets. *C. R. L'Academie Bulg. Sci.* **2008**, *61*, 1535–1540.
53. Atanassov, K.T. Intuitionistic fuzzy sets: Theory and applications. In *Studies in Fuzziness and Soft Computing*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 87–96.
54. Yager, R.R. Some aspects of intuitionistic fuzzy sets. *Fuzzy Optim. Decis. Mak.* **2009**, *8*, 67–90. [[CrossRef](#)]
55. Er, M.J.; Zhou, Y. Automatic generation of fuzzy inference systems via unsupervised learning. *Neural Netw.* **2008**, *21*, 1556–1566. [[CrossRef](#)] [[PubMed](#)]
56. El Moutaouakil, K.; Palade, V.; Safouan, S.; Charroud, A. FP-Conv-CM: Fuzzy Probabilistic Convolution C-Means. *Mathematics* **2023**, *11*, 1931. [[CrossRef](#)]
57. Alkawaz, A.N.; Abdellatif, A.; Kanesan, J.; Khairuddin, A.S.M.; Gheni, H.M. Day-Ahead Electricity Price Forecasting Based on Hybrid Regression Model. *IEEE Access* **2022**, *10*, 108021–108033. [[CrossRef](#)]
58. El Boukhari, N. Constrained optimal control for a class of semilinear infinite dimensional systems. *J. Dyn. Control Syst.* **2018**, *24*, 65–81.

59. Alkawaz, A.N.; Kanesan, J.; Khairuddin, A.S.M.; Badruddin, I.A.; Kamangar, S.; Hussien, M.; Baig, M.A.A.; Ahammad, N.A. Training Multilayer Neural Network Based on Optimal Control Theory for Limited Computational Resources. *Mathematics* **2023**, *11*, 778. [CrossRef]
60. El Moutaouakil, K.; El Ouissari, A.; Palade, V.; Charroud, A.; Olaru, A.; Baizri, H.; Chellak, S.; Cheggour, M. Multi-objective optimization for controlling the dynamics of the diabetic population. *Mathematics* **2023**, *11*, 2957. [CrossRef]
61. Hore, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
62. Lanaras, C.; Bioucas-Dias, J.; Galliani, S.; Baltasvias, E.; Schindler, K. Super-resolution of Sentinel-2 images: Learning a globally applicable deep neural network. *ISPRS J. Photogramm. Remote Sens.* **2018**, *146*, 305–319. [CrossRef]
63. Zhang, L.; Zhang, L.; Mou, X.; Zhang, D. FSIM: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **2011**, *20*, 2378–2386. [CrossRef] [PubMed]
64. Aljanabi, M.A.; Hussain, Z.M.; Shnain, N.A.A.; Lu, S.F. Design of a hybrid measure for image similarity: A statistical, algebraic, and information-theoretic approach. *Eur. J. Remote Sens.* **2019**, *52*, 2–15. [CrossRef]
65. Wang, Z.; Bovik, A.C. A universal image quality index. *IEEE Signal Process. Lett.* **2002**, *9*, 81–84. [CrossRef]
66. LeCun, Y. The MNIST Database of Handwritten Digits. 1998. Available online: <http://yann.lecun.com/exdb/mnist/> (accessed on 1 September 2023).
67. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
68. Machine Learning Repository UCI. Available online: <http://archive.ics.uci.edu/ml/datasets.html> (accessed on 1 September 2023).
69. Charroud, A.; El Moutaouakil, K.; Palade, V.; Yahyaouy, A. XDLL: Explained Deep Learning LiDAR-Based Localization and Mapping Method for Self-Driving Vehicles. *Electronics* **2023**, *12*, 567. [CrossRef]
70. Bahri, A.; Bourass, Y.; Badi, I.; Zouaki, K.; El Moutaouakil, K.; Satori, K. Dynamic CNN combination for Morocco aromatic and medicinal plant classification. *Int. J. Comput. Digit. Syst.* **2022**, *11*, 239–249. [CrossRef]
71. Aharrane, N.; Dahmouni, A.; Ensah, K.E.M.; Satori, K. End-to-end system for printed Amazigh script recognition in document images. In Proceedings of the 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Fez, Morocco, 22–24 May 2017; pp. 1–6.
72. Badr, E.; Ourabah, L.; Sekkat, H.; Mouad Ouhasni, M.; Rachid, A.; Khani, C.; El Moutaouakil, K. Multi-Dataset Convolutional Neural Network Model for Glaucoma Prediction in OCT Fundus Scans. *Stat. Optim. Inf. Comput.* **2024**, *12*, 630–645.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.