DESIGN OF A HIGHLY EFFECTIVE MULTI-LAYERED FUZZY LOGIC BASED IMAGE ENHANCEMENT FILTER

M. Mohamed Anver
Faculty of Engineering, University of Sri Jayewardenepura, Sri Lanka
dr.mohamed.anver@gmail.com

ABSTRACT: This paper describes the development of an effective fuzzy image filter which consists of a multi-layered fuzzy structure for the removal of noise from images heavily corrupted by impulse noise, while preserving the intricate details of the image. The introduction of multi-layered fuzzy systems substantially decreases the number of rules to be learnt. We then show how Evolutionary Algorithms (EAs) can be used to effectively learn the fuzzy rules in each knowledge base. Results are presented for impulse noise corruption of the well-known 'Lena' image.

Keywords: Fuzzy filter, image enhancement, multilayered fuzzy system, evolutionary learning, evolutionary algorithm.

INTRODUCTION

Conventional image enhancement techniques such as mean and median filtering have been employed in various applications in the past and are still being used. However techniques using Fuzzy Logic (FL) which mimics human reasoning and tolerates ambiguities well are increasingly being looked into as alternatives to these conventional techniques. In this paper, both FL and EAs are employed to show how they could be used in a practical digital image processing system to remove heavy impulse noise from corrupted images.

Section 2 of the paper gives an introduction to MLFL systems. In Section 3 we present a simple analysisof the Weighted Fuzzy Blend Filter presented in [1].It will be used to construct a two layered fuzzy imageenhancement algorithm in Section 4, whose fuzzy rulesin each layer will be learnt from corrupted data by anevolutionary algorithm. Application will be made towell-known 'Lena' image and comparisons are madewhere applicable.

MULTI-LAYERED FUZZY LOGIC SYSTEMS

A key consideration when designing FL systems is the size of the rule base, which increases exponentially with the number of inputs - the so called 'curse of dimensionality'. For example, consider the 8 input – single output single-layer FL system shown in Figure 1 and the MLFL structure shown in Figure 2. In the MLFL structure the output of the first layer is combined with the output of the second layer to obtain The Fuzzy Knowledge Base (FKB) for the single-layer fuzzy system consists of 2^8 = 256 rules assuming each input is represented using a membership function having two fuzzy sets. For the MLFL system shown in Figure 2, the first layer will have 2^4 = 16 rules and the second layer will have another 2^4 = 16 rules, making the total number of rules to 32. The total number of rules in the FKB has been reduced by a factor of 8 in this example, by employing a multi-layered approach. There are two possible ways to solve this problem according to recent papers published in the area of MLFL systems. One method is to break up the inputs so that a physically meaningful output is produced from the first layer and then use it together with the second layer to obtain the final output. This kind of approach has been proven to be successful in [3] where a MLFL controller has been developed for the control of a robot. Different applications that show how multi-layered fuzzy systems and EAs can be used to find fuzzy rule bases are described in [2] and [4].

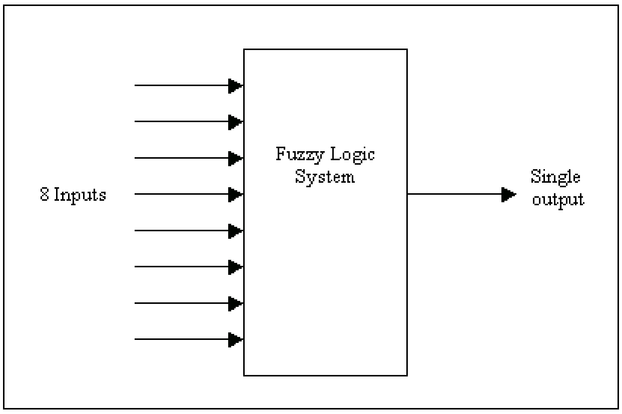

Figure 1: Single-Layer Fuzzy Logic System

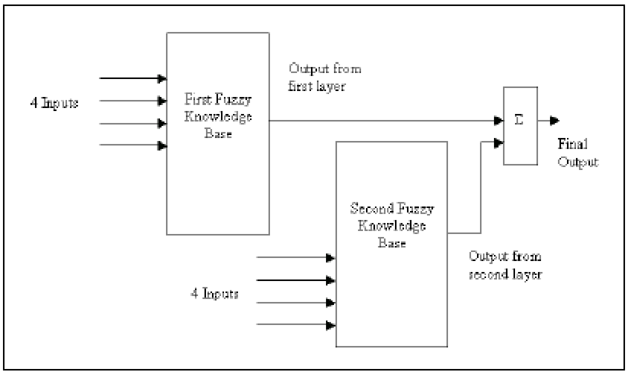

Figure 2: A Multi-Layered Fuzzy Logic System

THE WEIGHTED FUZZY BLEND FILTER

We begin with a discussion of the Weighted Fuzzy Blend Filter (WFBF) presented in [1]. From this filterour multi-layered fuzzy filter will be developed.The WFBF consists of two components to recoveroriginal pixel values from a corrupted image. Thefirst component is aimed at detecting noisy pixels bycomparing the intensity differences between a selectedpixel and the neighbouring pixels in a sliding windowof 3 × 3. The second component concentrates on thepixel data re-construction when necessary. Both componentsuse fuzzy reasoning to detect if the pixel isnoisy and to reconstruct the pixel by using mean ormedian filters. A 3×3 sliding window is shown below.

$$\begin{bmatrix} I_1 & I_2 & I_3 \\ I_4 & I_0 & I_5 \\ I_6 & I_7 & I_8 \end{bmatrix}$$

Let us suppose the pixel to be processed is I_0 and pixels I_1 to I_4 have been processed. The input variables to the first FKB are the n = 4 intensity differences given by,

$$x_j = abs(Ij^{'} - I_0) \quad j = 1, \cdots, 4 \qquad (1)$$

Since the difference values for pixels I_5 to I_8 are uncertain as they may be highly corrupted by noise, they are not used as fuzzy inputs. Two input fuzzy sets, named DH = Difference High and DL = Difference Low, are defined with membership functions in terms of constants c_1 = 10 and c_2 = 40, given as follows:

$$U_{DH}(x) = \begin{cases} 0 & x \leq c_1, \\ (x - c_1)/(c_2 - c_1) & c_1 < x < c_2, \\ 1 & x \geq c_2. \end{cases}$$

$$U_{DL}(x) = \begin{cases} 1 & x \leq c_1, \\ (c_2 - x)/(c_2 - c_1) & c_1 < x < c_2, \\ 0 & x \geq c_2. \end{cases}$$

The input variables lie in the interval [0, 255] for the processing of grey scale images. The output variable of the fuzzy inference engine out ∈ [0, 1], is a member of two fuzzy sets, B1 = V L and B2 = VH respectively. The centers of these sets are taken to be $y^1 = 0$ and $y^2 = 1$. There are M = 16 rules in the fuzzy rule base. The rules are built using intuition of how the intensity differences determine the existence of a noisy pixel. For example rule ℓ may be of the form:

$$\text{If } (x_1 \text{ is } A_1^\ell) \text{ and } (x_2 \text{ is } A_2^\ell) \text{ and } (x_3 \text{ is } A_3^\ell) \\ \text{and } (x_4 \text{ is } A_4^\ell) \text{ then } (out \text{ is } B^\ell)$$

where on the antecedent side of the rule A_1^l = DH, A_2^l= DH, A_3^l = DH and A_4^l = DH, and in the consequent B^l= VL. Each rule has been designed to deal with a particular pattern of intensity difference among the neighbouring pixels. Given a fuzzy rule base with M rules, the output out as given in Equation 2 uses a singleton fuzzifier, Mamdani product inference engine and centre average defuzzifier, see [10].

$$out = \frac{\sum_{\ell=1}^{M} \overline{y}^{\ell} (\prod_{i=1}^{n} \mu_{A_i^\ell}(x_i))}{\sum_{\ell=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^\ell}(x_i))} \qquad (2)$$

where $\underline{y}_l$ are centres of the output sets *B.* The enhanced pixel output is given by,

$$I_{enhanced} = out \times I_0 + (1 - out) \times I_{recons} \qquad (3)$$

where$I_{enhanced}$is the output of the pixel to be estimated, *out* is the fuzzy output, *I*0 is the input valueof the pixel and $I_{recons}$is the reconstructed value for that pixel. Two recovery methods are used to obtain I_{recons}, one is the median value when intensity gradient is high, and the other is the mean value when the intensity gradient is low. The intensity gradients are defined in 4 directions:

$$\begin{aligned} G(1) &= abs(I_4' - I_5), \\ G(2) &= abs(I_2' - I_7), \\ G(3) &= abs(I_3' - I_6), \\ G(4) &= abs(I_1' - I_8) \end{aligned}$$

and the minimum gradient is defined as,

$$G_{min} = \min\{G(1), G(2), G(3), G(4)\} \qquad (4)$$

The median value $I_{median}$is calculated using a 3 × 3 sliding window, while the mean value $I_{mean}$is calculated using the 2 outer pixels corresponding to the minimum gradient direction given by G_{min}. For example, if the minimum gradient is in the horizontal direction the mean will be calculated as $(I_4 + I_5)/2$.

Another fuzzy membership function GL (Low Gradient), with $c_1 = 40$ and $c_2 = 80$, is introduced for determining which method should be used,

$$U_{GL}(x) = \begin{cases} 1 & x \leq c_1, \\ (c_2 - x)/(c_2 - c_1) & c_1 < x < c_2, \\ 0 & x \geq c_2. \end{cases}$$

Fundamentally this second level in the WFBF incorporates, components of the traditional mean and median filters. We take weights for both mean and median to be simply $U_{GL}(x)$ and $U_{GL}= 1 - U_{GL}(x) = U_{GH}$, respectively.

Then the reconstructed value is obtained from,

$$I_{recons} = U_{GL}(x) \times I_{mean} + U_{\overline{GL}} \times I_{median} \qquad (5)$$

This filter has been shown to be capable of removing noise from images corrupted by impulse noise up to35%, see details in [1]. The fuzzy rule base in theWFBF filter was created using human reasoning. A simple iterative search can be undertaken on all possible (2^{16} = 65, 536) fuzzy rule bases to find the best rule base. The Mean Square Error (MSE) between the uncorrupted image and the enhanced version of the corrupted image was used as the criteria to determine the quality of the image.

A 512× 512 image corrupted by 35% impulse noise was used and the minimum MSE obtained was 94.73. This MSE was lower than that obtained by theWFBF which shows that it is possible to find a ‘better’ WFBF by a direct search of all the possible knowledge bases.

In this paper we propose to develop a second layer which has as output the reconstructed pixel valueI_{recons}which can be immediately input into Equation (3).

MULTI-LAYERED FUZZY IMAGE FILTER

Multi-Layered Fuzzy System Design

A multi-layered fuzzy logic system is now designed to remove impulse noise in similar manner to the WFBF.This particular structure, shown in Figure 3, has twoFKBs. The first layer has 4 input intensity differences given by Equation 1 and an output variable *out* $\in\{0, 1\}$representing whether a selected pixel isnoisy or not as described in Section 3. The fuzzy membershipfunction for the output variable *out* is as givenin Section 3. The number of rules in the first FKB is2^4 = 16. The second FKB has 3 inputs; they are:

(i) Minimum gradient (as calculated in the previous section) fuzzified using 2 given fuzzy sets.

(ii) Median of the pixels in the 5×5 sliding window fuzzified using 11 triangular fuzzy sets shown inFigure 4.

(iii) Mean in the direction of the minimum gradient (as calculated in WFBF) fuzzified using the 11triangular fuzzy sets.

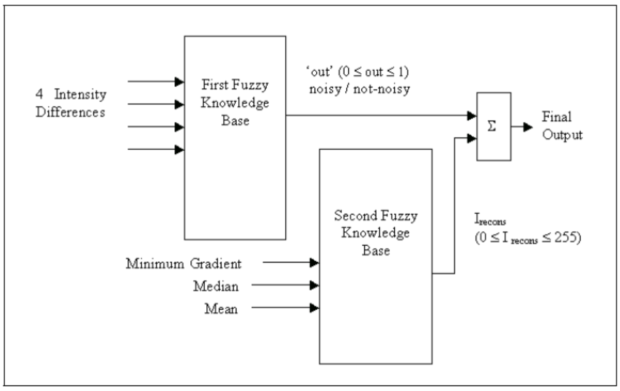

Figure 3. Multi-Layered Fuzzy Image Filter

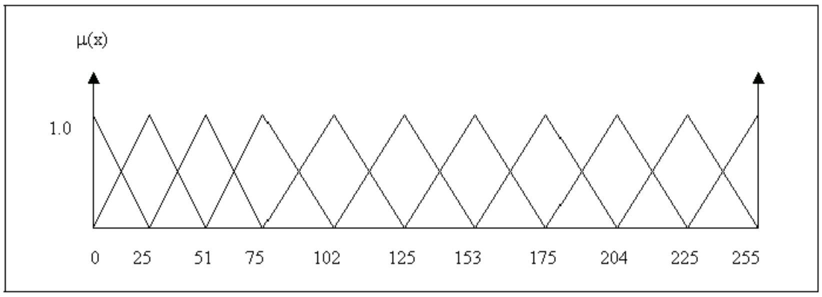

Figure 4.Fuzzy sets for fuzzification of median andmean values

Each fuzzy set in Figure 4 can be described by the following equation where *x* is the value of the variableto be fuzzified, *L* is the 'left corner' of a fuzzy set, *C* is the 'center' of a fuzzy set and *R* is the 'right corner'of a fuzzy set.

$$\mu(x) = \begin{cases} (x-L)/(C-L) & L < x \leq C \\ (R-x)/(R-C) & C < x < R \\ 0 & else \end{cases}$$

The output of the second layer is taken as the 'reconstructed' integer value of the pixel ($I_{recons} \in \{0,..,255\}$). The number of rules in the second FKB is $2^1 \times 11^2 = 242$. The final output of the enhanced pixel is calculated according to Equation 3.

4.2 Evolutionary Learning of the Fuzzy Rule Bases

We show now how to apply an EA to learn the fuzzy rules in the two knowledge bases. Each individual string in the evolutionary population is to uniquely represent the multi-layered structure. Thisis achieved as follows. In the first knowledge base each fuzzy rule is uniquely defined by the consequent part, it being represented by either a 0 or 1 depending on whether the output variable *out* belongs to the fuzzy set VL or VH respectively. In similar manner eachfuzzy rule in the second knowledge base is defined by its consequent. The two fuzzy rule bases can therefore be represented as a string of M = 258 consequents,

$$\underset{\sim}{x}_k = \{a_1, \cdots, a_{16}, a_{17}, \cdots, a_{258}\},$$

where $a_j \in \{0, 1\}$ for $j = 1, \ldots, 16$, are the consequents for the rules in the first knowledge base and $a_j \in \{0, 255\}$ for $j = 17, \ldots, M$, are the consequents for the rules in the second knowledge base. Each such string forms an individual in the evolutionary population and a possible solution to finding the"best" fuzzy filter. The population at generation t,

$$P(t) = \{\underset{\sim}{x}_k \;:\; k = 1, \cdots, N\},$$

where N = 120 is the number of individuals in the population, the population size. We define the fitness for each individual f_k, as the MSE between the uncorrupted image and the enhanced version of the 512× 512 'Lena' image corrupted by 35%. The initial generation was formed by placing random elements in each individual string.

```
if (flip(p_{M}) == true)
{ if (pv  ==  0)
    {mv = pv + rand2limit(1,128);}
   else f (pv == 255)
    {mv = pv - rand2limit(1,128);}
   else
    {
     if (flip(0.5) == true)
       {mv = pv + rand2limit(1,128);}
     else
       {mv = pv - rand2limit(1,128);}
    }
} else
   {mv = pv;}
if (mv <  0 ) { mv =  0 ; }
if (mv > 255) { mv = 255; }
```

Arithmetic crossover with different parameter (α) valueswas used as shown by the following code:

```
double alpha1 = rand0_1();
double alpha2 = rand0_1();
for (i = 0; i < length_of_chromosome; i++) {
     c1[i] = round(alpha1 * p2[i]
                + (1 - alpha1) * p1[i]);
     c2[i] = round(alpha2 * p2[i]
               + (1 - alpha2) * p2[i]); }
```

It was found that for thismulti-layered structure the EA converged around the 280thgeneration to a fuzzy rule base which gave a minimumMSE value of 63.95. The algorithm was re-run with different random initializing seeds achieving minimum MSEaround the same value.

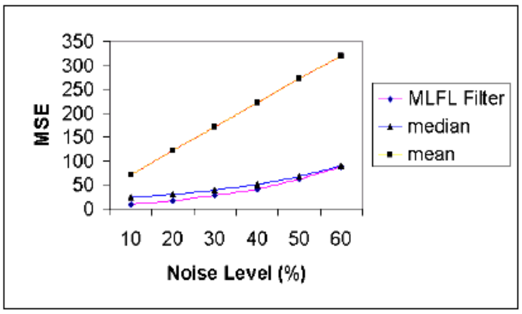

Figure 5. Graph showing the MSE vs Noise Percentage

5. RESULTS AND DISCUSSION

This filter was implemented on a ‘Lena’ test image having 35% corruption and size 512 × 512pixels to reduce thecomputational time required and then tested on a 1024× 1024 ‘Lena’ image. The results for the 1024× 1024 image areshown in Figures 6 and 7 for corruption levels of 35% and 45% and the MSE values obtained for corruption levelsranging from 10% to 60% are shown in Figure 5.

It is observed that this new MLFL system has sufficient knowledge learnt from enhancing a 512× 512 image corruptedby 35% impulse noise, to remove noise as high as 45% from the 1024× 1024 image.

Figure 6. (a) Corrupted image (35%) (b) Restoredimage - MLFL Structure

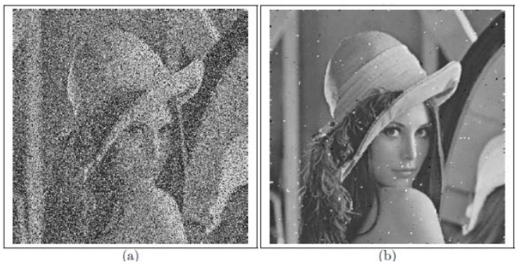

Figure 7. (a) Corrupted image (45%) (b) Restored image - MLFL Structure

6. CONCLUSION

In this paper we have presented a multi-layered fuzzy imagefilter based on the weighted fuzzy blend filter. Therules in the layers have been learnt directly from data usingan appropriately defined evolutionary algorithm withmodifications to the basic mutation and crossover operators.It has been shown that this new fuzzy image filterperforms better than conventional mean and median filters. Indeed the MLFL system having been learnt by enhancinga reduced 512× 512 'Lena' image corrupted by35% impulse noise, is able to remove noise as high as 45%in a 1024× 1024 'Lena' image.The filter is seen to preserve intricate features of the imagewhile removing heavy impulse noise whereas the conventionalmean and median filters fail in this context even at low corruption levels. That is, the granularity is well maintained in the enhanced images obtained through the fuzzy processing which is not reflected through the plots of MSE values for median filter and the MLFL structure as shown in Figure 5.The learning of fuzzy rules in a fuzzy image filter with a true hierarchical fuzzy logic structure where the output ofthe first layer is fed in to the second layer to obtain an 'improved' final output, is being currently studied.

7. REFERENCES

[1] MAO. X, STONIER, R. J and THOMAS, P. (2001) "An Effective Fuzzy Image Filter", Proceedings of the InternationalConference on Computational Intelligence for Modelling Control and Automation (CIMCA),M. Mohammadian (Ed), Las Vegas, pp. 280-285.

[2] STONIER, R.J. and ANVER, M.M.(2002) "A Multi-Layered Fuzzy Image Filter for Removing Impulse Noise",accepted for presentation at and publication in the Proceedings of the International Conference on IntelligentSystems (ICIS), Boston.

[3] MASOUD MOHAMMADIAN AND R.J. STONIER (1995) "Adaptive Two Layer Fuzzy Logic Control of a MobileRobot System", Proceedings of IEEE Conference on Evolutionary Computing, Vol.1, Perth WA Australia, pp. 204-209.

[4] Mohammadian, M and Stonier, R. J. (1998) "Hierarchical Fuzzy Control", Proceedings of the 7th International conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, Paris, pp. 621-629.