University of Waterloo

Winter 2015 – ECE457B Computational Intelligence
ECE Dept, University of Waterloo
Instructor: Fakhri Karray

Numerical Character Recognition

Group #4: Mark Berger, Sahar Haider, Naveen Narayanan, Gordon Zhou

Abstract and Overview:

- Extracting information from images is a common problem, for example: scanning passports, identifying license plates, cashing cheques, filling out forms etc.
- Use Optical Character Recognition (OCR) to extract characters from images using automated computer algorithms
- Recognize Arabic numerals from 0 – 9 by separating individual digits

Problem Formulation:

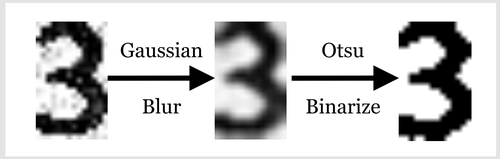
- Aiming to solve the problem of identifying different fonts; both serif and sans-serif variants
- Character recognition main stages:
  1) Character segmentation
  2) Character recognition
- Factors that need to be accounted for in character segmentation include:
  - Size of text
  - Image noise

Proposed Solution:

- Two alternatives using computational intelligence
  1) Fuzzy Logic
  2) Artificial Neural Networks
- Fuzzify pixels, apply patterning rules:
  - Left-right symmetry
  - Top-bottom symmetry
  - Pixel density
  - Pixel width
  - Number of pixels in the center of the image
- Artificial Neural Network
  - Train the ANN using a number of different fonts and characters
  - Multi-Layer Perceptron with Back Propagation Learning algorithm
  - Image input is in the form of a 2-dimensional array
  - The array is flattened, and each entry is passed as an input into the neural network
  - The output is a floating point value indicating closeness to the specific target

Tools, Algorithms:

- Python 2.7
- OpenCV, an open source computer vision library
  - Turning images into gray scale
  - Gaussian blurring to reduce the noise
  - Otsu's method to binarize the image into black and white using local thresholds



- Simple algorithm to separate lines and characters based on continuous white pixels
  - Fonts with poor kerning results in poor separation
- SciPy and NumPy, packages for scientific computing
  - Used for manipulating multi-dimensional arrays

Algorithms

Fuzzy Logic

- Generalized bell membership function to fuzzify input parameters
- Use different fuzzy rules, for example pixel density, symmetry, to de-fuzzify the fuzzified input

Artificial Neural Network

- Back propagation algorithm to train data
- Learning rate of 0.2
- Max error of 0.2
- 10 individual neural networks for each character
- 2 hidden layers of 442 nodes each

Experiments, Analysis:

Experiment Setup

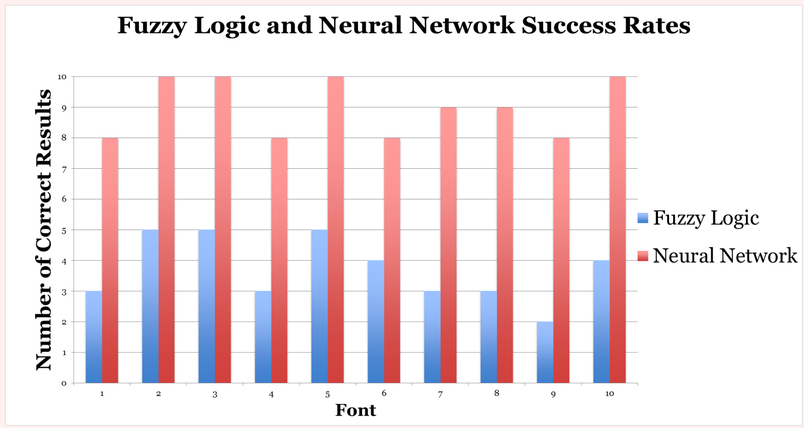
- Image file with several different fonts including both serif and sans-serif fonts

Evaluation Metrics

- Number recognition accuracy using Fuzzy Logic and ANN
- The solutions are evaluated using trained and random data

Analysis

	Fuzzy Logic	ANN
Trained Fonts Success Rate	78.90%	100%
Random Fonts Success Rate	36.47%	94.11%



Conclusion:

- Number of training sets were used to train the ANN algorithm, and multiple different rules were used for fuzzy inferencing
- Based on the results and analysis of both the proposed solutions, it was found the ANN was more reliable for the purpose of numerical character recognition
- Training the neural network was computationally expensive, however once the weights were determined recognition was quicker
- The fuzzy logic patterns need to be more developed in order to successfully differentiate between several fonts