‘Optical Character Recognition using Artificial Neural Networks’

Conference Paper · January 2007

1 author:

Parthasarathy Subashini
Avinashilingam University

127 PUBLICATIONS 309 CITATIONS

See profile

Some of the authors of this publication are also working on these related projects:

- E-CO DAR, Solar Decathlon Africa (SDA) 2019 View project
- My Ph.D Research work View project
Optical Character Recognition using Artificial Neural Networks

“Much of the world’s information is held captive in hard-copy documents. OCR systems liberate this information by converting the text on paper into electronic form.”
- S.V. Rice, G. Nagy, & T. A. Narker [5]

Kurt Alfred Kluever (kurt@kloover.com)
Department of Computer Science
Rochester Institute of Technology
OCR

- Definition: the process of translating images of handwritten, typewritten, or printed text into a format understood by machines

- Why? allows for reduced storage size, editing, indexing, searching, etc.
Steps of OCR

- Character segmentation
  - Based on vertical projections (or VP diffs)
  - Connected component analysis
- Feature extraction
  - Side profiles
  - Line adjacency graphs
  - Vertical/Horizontal projections
- Character recognition => ANN
Improving OCR

• Improve the scanning technique
  • Higher dpi
  • Scan in grayscale or color

• Improve pre-processing
  • noise removal
  • skew correction

• Use a single “OCR-friendly” font

• Domain specific knowledge
Artificial Neural Networks

• Why ANNs?
  • can deal with fuzzy data
  • can learn over time

• Steps:
  • Character segmentation
  • Feature extraction
  • Input: feature vectors
  • Output: character class
Avi-Itzhak et al., 1995

- High Accuracy Optical Character Recognition Using Neural Network with Centroid Dithering
- Idea: Simple is better!
  - Input = every pixel of the scanned character
  - Output = highest indicates character class
- Experiments with multi-font + multi-size
- Near perfect accuracy
Pre-Processing

- Remove noise via thresholding
- Normalize via scaling to 50x50 pixels
- Center character on centroid

- Convert 2D array to a vector (concatenation)
- Feed into the input nodes of ANN
Network Topology

Input Layer → Hidden Layer → Output Layer

- Input Layer: 1, 2, ..., 2500
- Hidden Layer: 1, ..., 100
- Output Layer: 1, 2, ..., 94
Training the ANN

• Generate one example character for each of the 12 different fonts

• Centroid dithering: creates many “different” images from this single input character
  • Shift character around in [-2,+2] window
  • Allows for width variations in character strokes

• 8,650,000 iterations => MSE=2×10^-6
Testing the ANN

• Test samples varied in font and size
• Assumes perfect segmentation of cleanly printed characters
• Doesn’t account for l vs 1 vs | across fonts
  • Needs surrounding context
  • Ex: “32₁₀” vs. “SCIENCE” vs. “isRed || isBlue”
• Recognized 100% of the 347,712 test samples
Ramirez et al., 1996 [7]

- On Structural Adaptability of Neural Networks in Character Recognition
- Idea: A fixed structured network is bad
  - Based on idea by T.C. Lee’s PhD dissertation [3]
  - Change weights, structure, and learning rate of the network
- Uses vertical + horiz run lengths as features
- Result: Fixed structure network outperforms the structure adapting network
Structure Adaptability

- An optimal solution requires a suitable number of neurons in the hidden layer

- If error stabilizes but is still greater than desired, insert another neuron

- If a neuron is a redundant element or a non-functioning element, get rid of it

- Adapt the learning rate to accelerate convergence (high at first, low towards end)
Results

1 = Fixed structure  2 = Adaptable structure
Mani et al., 1997 [4]

- Application of Artificial Neural Network Model for Optical Character Recognition

- Idea: Use histograms as feature vectors
  - Very similar to Avi-Itzhak et al.’s approach
  - Uses image projections as feature vectors

- Result: Not as good as Avi-Itzhak’s method
  - Noisy data: 70%
  - Clean data: 99%
Image Projections

[2]

Applying this rule ... method in [591. A minimum of the projection is located
and the projection value noted. The sum of the differences
Neves et al., 1997 [6]

• A Multi-Font Character Recognition Based on its Fundamental Features by Artificial Neural Networks

• Idea: Use a complex feature vector to recognize multi-font capital letters

  • Use curvature, line slope, space, line interconnection, relative distance between two lines, and other topological and geometrical features

• 17 features in total
Feature Example

- Number of line intersections at three horizontal crossing lines (V1,V2,V3) drawn at 30%, 50%, and 80% of overall character height.
Network Topology

Camada de saída
(1 neurônio)

Camada intermediária
(12 neurónios)

Camada de entrada
(17 neurónios)
Results

• 99.3% accuracy on the fonts it was trained on
• Can also recognize unknown fonts semi-reliably (3 fonts 100%, 2 fonts ~70%)
• Developing features for feature vectors is difficult and requires human interaction
References


