Tests

The program was tested on numerous texts, English and Hebrew, typed and handwritten. Test were used first to determine the functionality of the program and second to tune it to optimal results.

Functionality tests:

The functionality tests involved running every function of the program. On several inputs, and checking if it worked. Functions were tested on windows 7 and Mono.

Rotation:

- Expected result: Page rotated to 0˚.
- Actual result: works on Windows and Mono

Threshholding:

- Expected result: Grayscale image turned to black & white only
- Actual result: works on Windows and Mono

Noise cleaning:

- Expected result: Noise in the form small dots is removed
- Actual result: works on Windows and Mono (Sometimes this creates problems in the characters that need to be recognized)

Line straightening:

- Expected result: Handwritten lines straightened.
- Actual result: works on Windows and Mono

New printed font learning:

- Expected result: New font learned from picture
- Actual result: works on Windows and Mono

Learning fonts from a list:

- Expected result: The selected fonts learned
- Actual result: works on Windows and Mono

Deploying dataset:

- Expected result: Every member of the dataset is deployed in image form to the folder ims
- Actual result: works on Windows and Mono

Network training:

- Expected result: Network trained until the chosen error, stops training, and saves the trained network.
- Actual result: works on Windows and Mono

Handwriting pages print:

- Expected result: Prints the handwriting learning pages
- Actual result: works on Windows and Mono

Handwriting pages learning:

- Expected result: Gets the filled out and scanned handwriting learning pages, and learns the font from them and saves it with the chosen name.
- Actual result: works on Windows and Mono

Create all fonts script:

- Expected result: 10 preselected fonts are learned
- Actual result: works on Windows and Mono

Train all fonts:

- Expected result: All fonts in the program trained to their respective errors
- Actual result: works on Windows and Mono

Client:

- Expected result: Client connects to the server, checks what fonts it needs, and downloads them from the server. All settings and UI work.
- Actual result: works on Windows and Mono

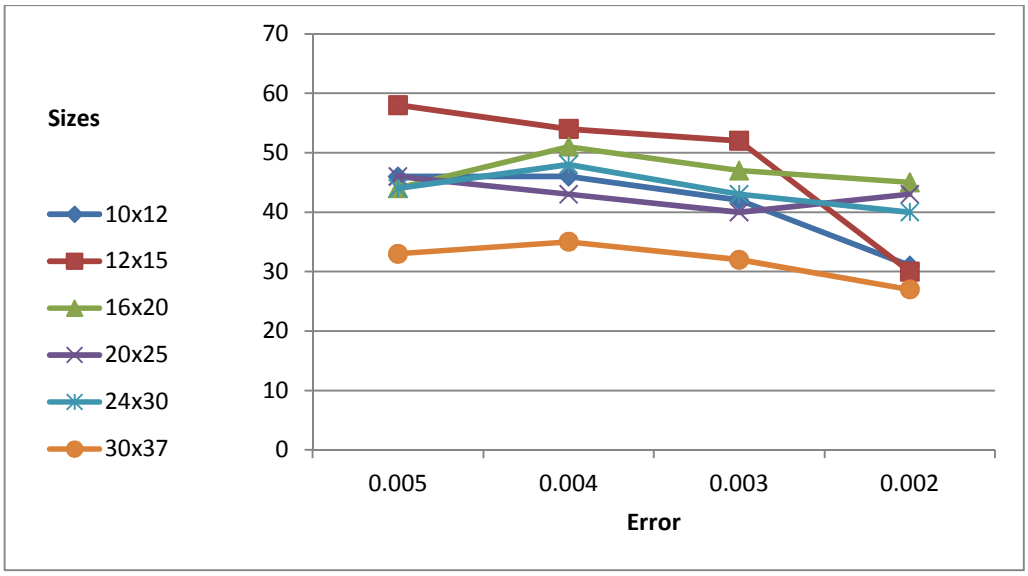
Server:

- Expected result: Server works, and is able to connect to client. All settings and UI work.
- Actual result: works on Windows and Mono

Tuning tests:

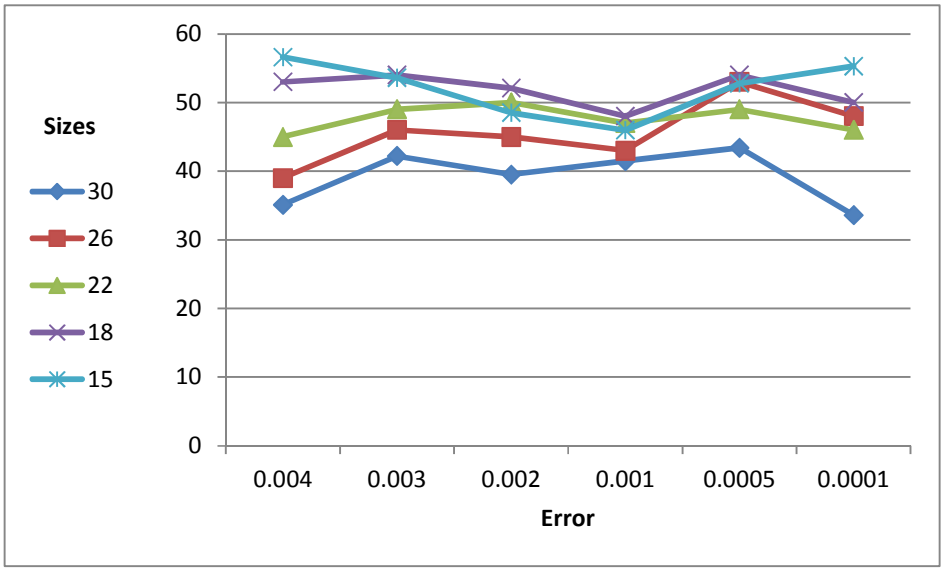
The tuning tests were meant to find the optimal settings for the character image creation (bounds, slant correction, resizing etc.), optimal character image size (Handwritten and printed), optimal error to train to, optimal number of neurons , and number of hidden layers, and show improvement of the text recognition over time.

Handwritten size and error:



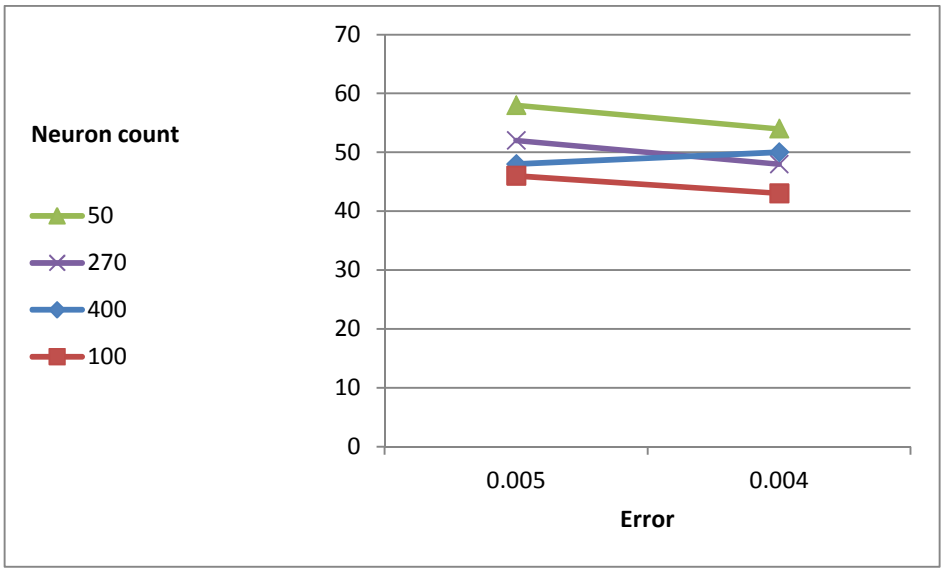
The optimal size was 12x15 and error of 0.005.

Typed size and error



The optimal size was 15 and an error of 0.005

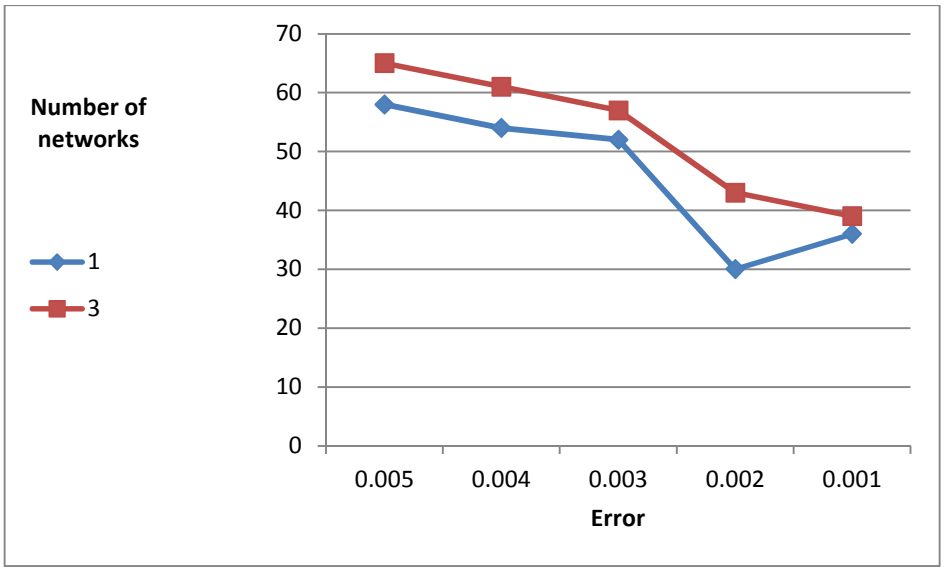
Hidden layer neuron count:



The optimal number was 50 neurons.

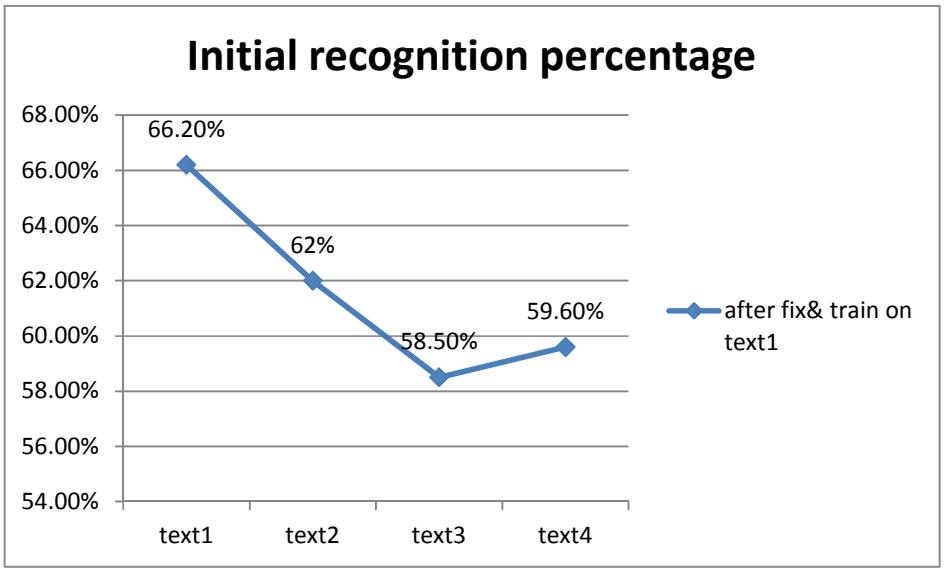
Number of hidden layers: 2 hidden layers were tried but training wasn't good enough and the results were poor, therefore 1 layer is used.

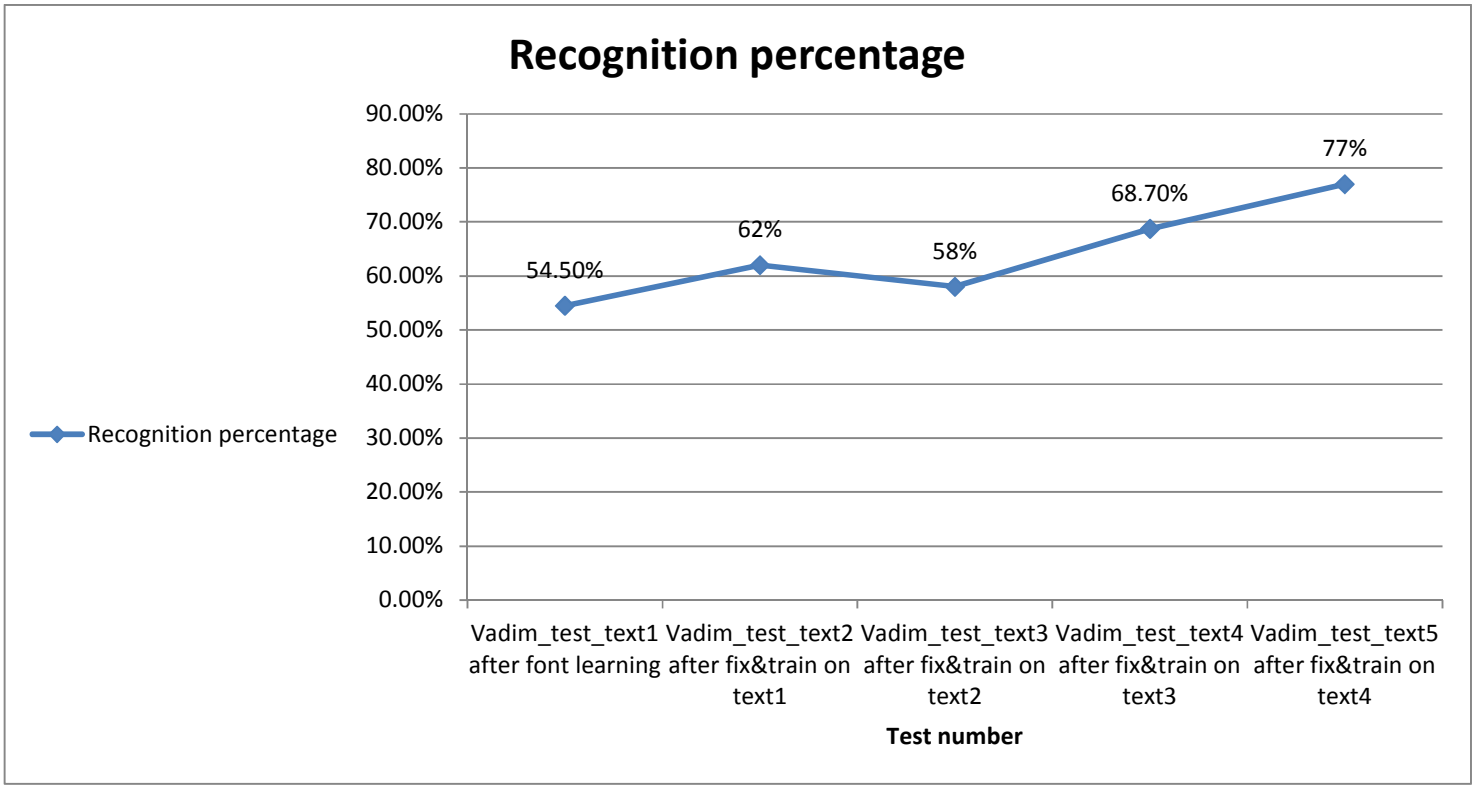
Number of neural networks working synchronously:



There is a clear advantage to using 3 networks. This advantage only grows as the networks learn more.

Progression on different texts:





As can be seen as more texts are learned and recognized, and error corrected the recognition of new texts is improved.