Testing

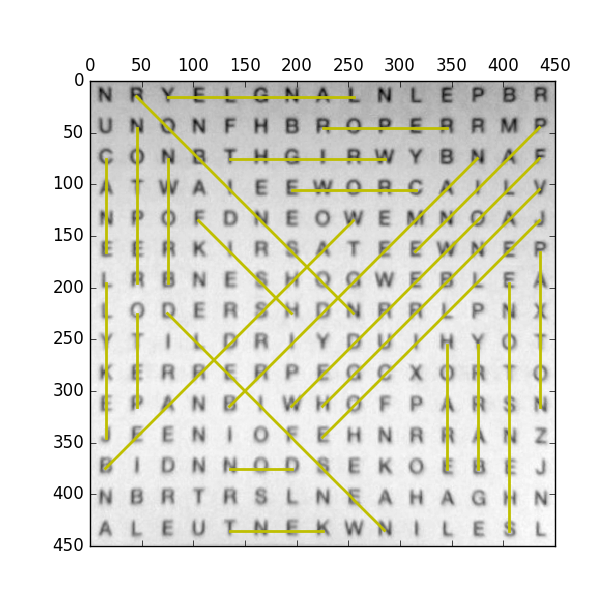
# Trial 1

Without dimensionality reduction, the classifier only classified 1 letter incorrectly. The wordsearch function, being able to find the best matches for words that aren’t exact matches, manages to draw every line on the image correctly.

**Input:**

wordsearch(test1, words, train\_data, train\_labels)

**Output:**

224 letters were correctly labelled which is about 99% of the letters.

Searching for barry

Line from index 192 to index 132

Searching for beardshaw

Line from index 180 to index 68

Searching for bridgeman

Line from index 154 to index 42

Searching for brown

Line from index 92 to index 32

Searching for cane

Line from index 30 to index 75

Searching for crowe

Line from index 55 to index 51

Searching for don

Line from index 186 to index 184

Searching for fish

Line from index 63 to index 111

Searching for flowerdew

Line from index 44 to index 156

Searching for hoare

Line from index 131 to index 191

Searching for jekyll

Line from index 165 to index 90

Searching for jellicoe

Line from index 74 to index 172

Searching for kent

Line from index 217 to index 214

Searching for langley

Line from index 8 to index 2

Searching for nesfield

Line from index 219 to index 107

Searching for paine

Line from index 29 to index 85

Searching for paxton

Line from index 89 to index 164

Searching for peto

Line from index 151 to index 106

Searching for repton

Line from index 91 to index 16

Searching for robinson

Line from index 1 to index 113

Searching for roper

Line from index 22 to index 26

Searching for shenstone

Line from index 223 to index 103

Searching for vanbrugh

Line from index 59 to index 157

Searching for wright

Line from index 39 to index 34

Solved

24 out of 24 found correctly.

# Trial 2

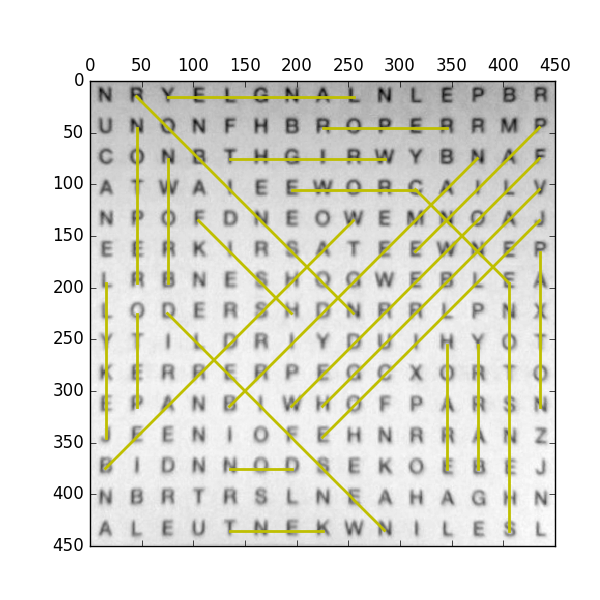
With dimensionality reduction, the classifier classified 8 letter incorrectly. The wordsearch function draws the line for “cane” in the wrong place, every other line was drawn in the right place.

**Input**:

wordsearch(test1, words, train\_data, train\_labels, True)

**Output:**

Solving test1 With Reduction

217 letters were correctly labelled which is about 96% of the letters.

Searching for barry

Line from index 192 to index 132

Searching for beardshaw

Line from index 180 to index 68

Searching for bridgeman

Line from index 154 to index 42

Searching for brown

Line from index 92 to index 32

Searching for cane

Line from index 55 to index 103

Searching for crowe

Line from index 55 to index 51

Searching for don

Line from index 186 to index 184

Searching for fish

Line from index 63 to index 111

Searching for flowerdew

Line from index 44 to index 156

Searching for hoare

Line from index 131 to index 191

Searching for jekyll

Line from index 165 to index 90

Searching for jellicoe

Line from index 74 to index 172

Searching for kent

Line from index 217 to index 214

Searching for langley

Line from index 8 to index 2

Searching for nesfield

Line from index 219 to index 107

Searching for paine

Line from index 29 to index 85

Searching for paxton

Line from index 89 to index 164

Searching for peto

Line from index 151 to index 106

Searching for repton

Line from index 91 to index 16

Searching for robinson

Line from index 1 to index 113

Searching for roper

Line from index 22 to index 26

Searching for shenstone

Line from index 223 to index 103

Searching for vanbrugh

Line from index 59 to index 157

Searching for wright

Line from index 39 to index 34

Solved

23 out of 24 found correctly.

# Trial 3

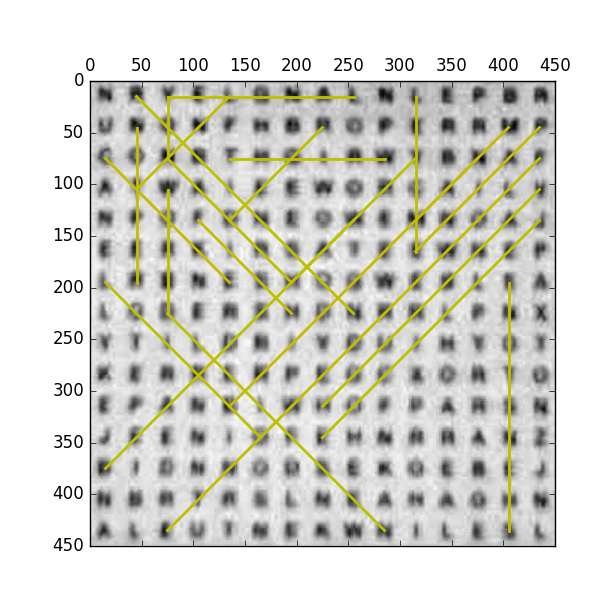
Without dimensionality reduction, the classifier classified 129 letters incorrectly. The wordsearch function, being able to find the best matches for words that aren’t exact matches, manages to draw half of the lines in the correct place.

**Input:**

wordsearch(test2, words, train\_data, train\_labels)

**Output:**

Solving test2

96 letters were correctly labelled which is about 42% of the letters.

Searching for barry

Line from index 96 to index 40

Searching for beardshaw

Line from index 180 to index 68

Searching for bridgeman

Line from index 154 to index 42

Searching for brown

Line from index 96 to index 32

Searching for cane

Line from index 70 to index 28

Searching for crowe

Line from index 30 to index 94

Searching for don

Line from index 2 to index 32

Searching for fish

Line from index 63 to index 111

Searching for flowerdew

Line from index 44 to index 156

Searching for hoare

Line from index 156 to index 212

Searching for jekyll

Line from index 85 to index 10

Searching for jellicoe

Line from index 74 to index 172

Searching for kent

Line from index 4 to index 46

Searching for langley

Line from index 8 to index 2

Searching for nesfield

Line from index 219 to index 107

Searching for paine

Line from index 29 to index 85

Searching for paxton

Line from index 91 to index 16

Searching for peto

Line from index 22 to index 64

Searching for repton

Line from index 170 to index 90

Searching for robinson

Line from index 1 to index 113

Searching for roper

Line from index 47 to index 107

Searching for shenstone

Line from index 223 to index 103

Searching for vanbrugh

Line from index 59 to index 157

Searching for wright

Line from index 39 to index 34

Solved

12 out of 24 found correctly.

Timing

The search function called inside the body of the wordsearch function only adds tuples where the third part is non-zero. The third part of each tuple represents how many letters are matched with the word if a line is drawn between the indices represented by the first and second parts of the tuple. So when wordsearch comes to combining all lists of tuples from the horizontal, vertical and diagonal searches for sorting, it does not need to bother with tuples that have a third part equal to zero because we know they’d be at the end of the sorted list.

If the search function finds a tuple that represents an exact match i.e. when the number of matching letters is equal to the length of the word, instead of adding it to the list of tuples and go through finding more tuples, the return list will consist of (0,0,0) at index 0 with the tuple representing an exact match at index 1.

The wordsearch function arbitrary searches horizontally then vertically and finally diagonally. If the list returned from the horizontal search has (0,0,0) as the item at index 0 it means the item at index 1 is an exact match and the function skips vertical and diagonal searching. Likewise, if (0,0,0) is detected as the item at index 0 in the vertically found list, diagonal searching is skipped.