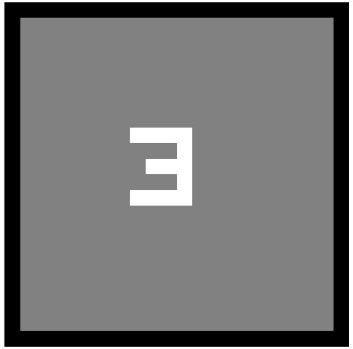
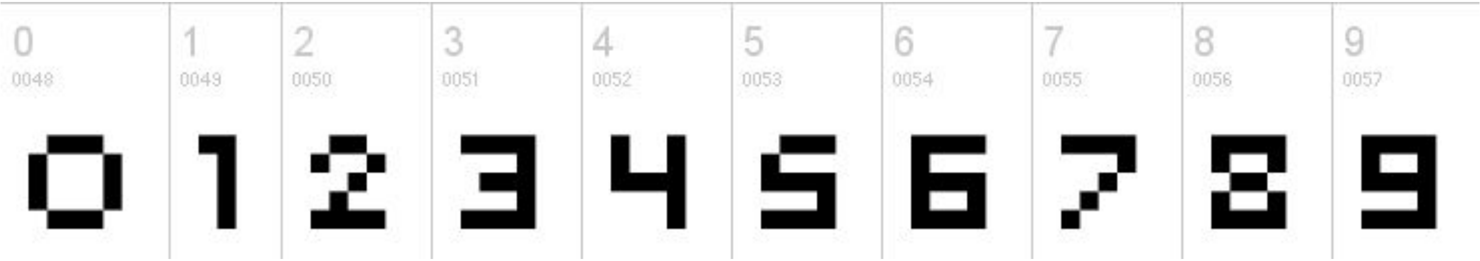
Classification of Digits

Riley Tuttle

Goals

- Develop classifier for digits set in a 5x5 frame
- Systematically remove information from assumptions





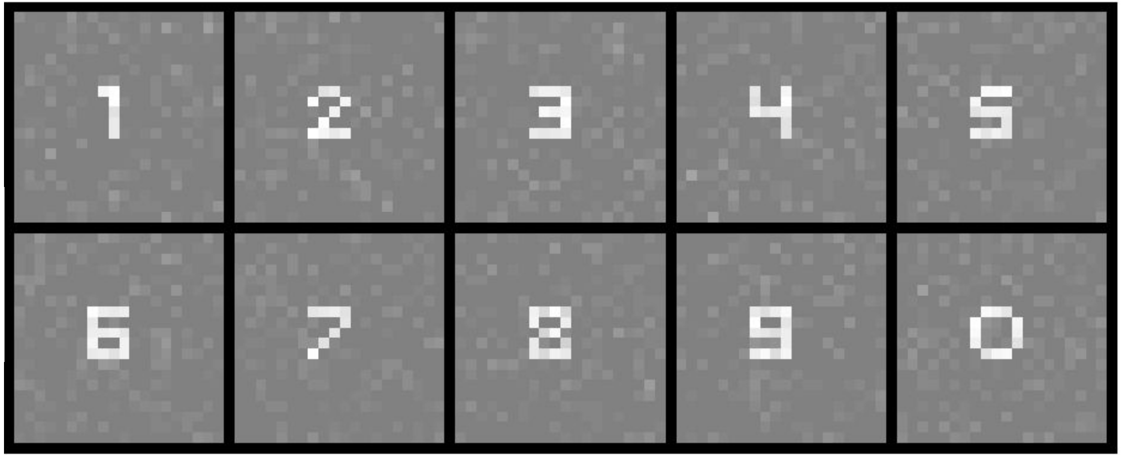
Digits in noise

- Color is grayscale (0,255)
- Ideal digit color is 127
- Becomes a Minimum distance receiver.
- Choose Hi for which:

$$T_i(x) = \sum_{n=1}^{N} x[n]s_i[n] - \frac{1}{2}\epsilon_i$$

is maximum

- Pe = 0



Digits in noise, unknown location

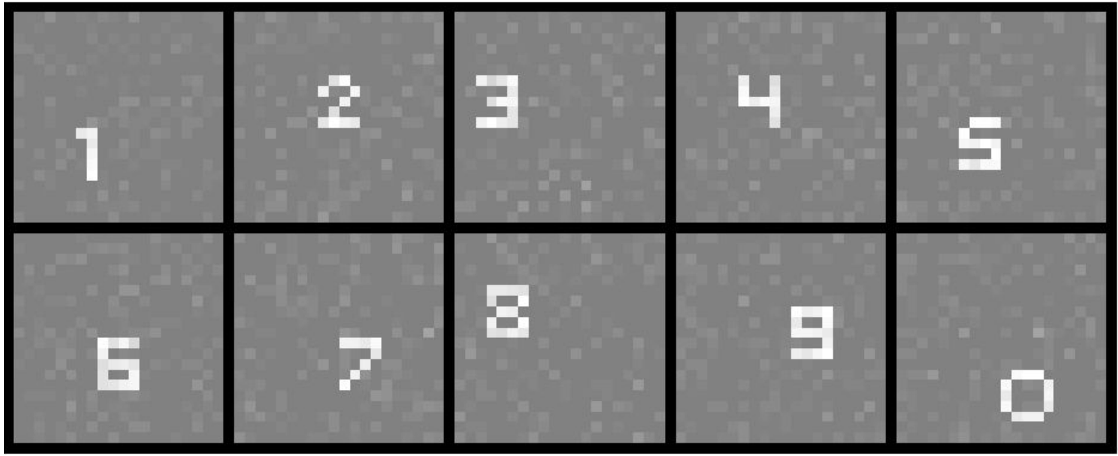
- Becomes minimum distance receiver with unknown arrival time in 2 dimensions
- decide Hi for which:

$$T_i(x) = \max_{n_0, m_0 \in 1,16} \sum_{n=n_0, m=m_0}^{n=N-1, m=M-1} x[n] s_i[n,m] - \frac{\epsilon_i}{2}$$

$$N, M = 20$$

is maximized

- Pe = .0132
- 66 out of 5000



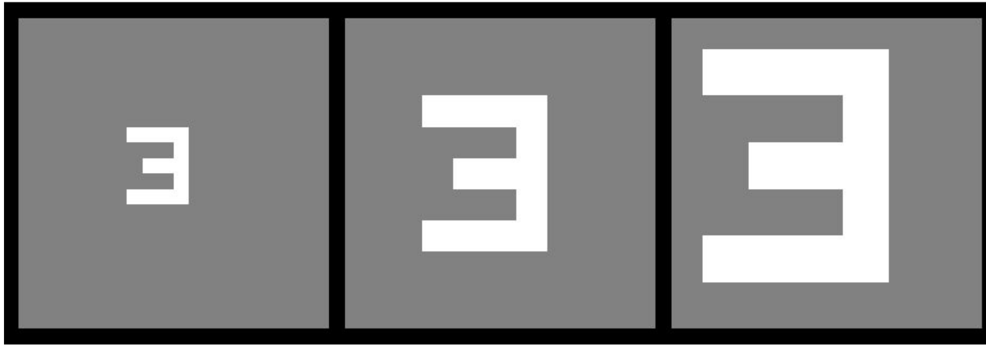
Unknown size

- unknown size is not the same as amplitude because the signal changes.
- same detector as before but optimized over a different parameter
- choose Hi for which:

$$T_i(x) = \max_{s \in 1,3} \sum_{n=0}^{N-1} x[n] s_i[n, s] - \frac{\epsilon_i}{2}$$

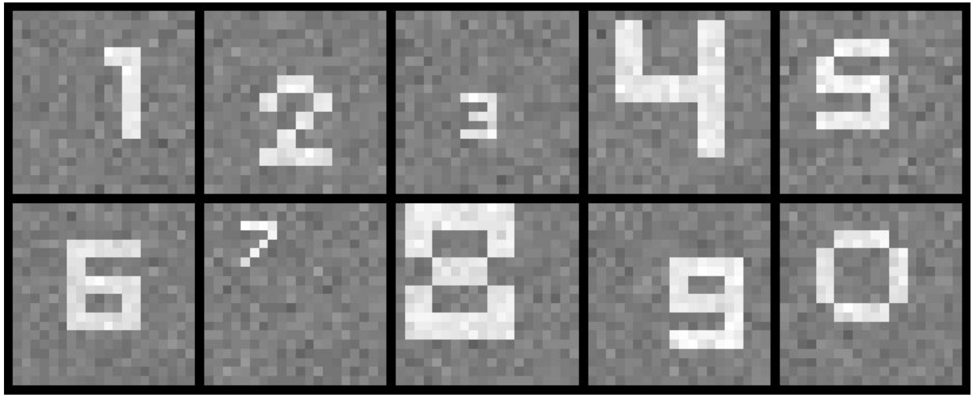
is maximum

- Pe = 0
- Errors are small because we sometimes are increasing the energy to noise ratio of the signal



Unknown size and location

- Again a minimum distance receiver optimized over the unknown parameters
- Pe = .0022
- 11 out of 5000



Unknown size, location, rotation

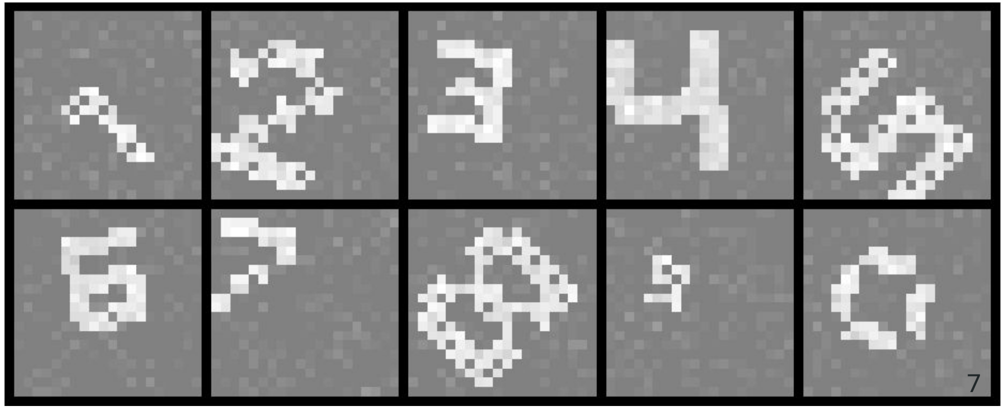
- As before a minimum distance receiver optimized over unknown parameters
- choose Hi for which:

$$T_i(x) = \max_{n_0,m_0 \in 1,16 \;\; s \in 1,3 \;\; r \in -\frac{\pi}{4},\frac{\pi}{4}} \sum_{n=n_0,m=m_0}^{n=N-1,m=M-1} x[n]s_i[n,m,s,r] - \frac{\epsilon_{i,s,r}}{2}$$

$$N, M = 20$$

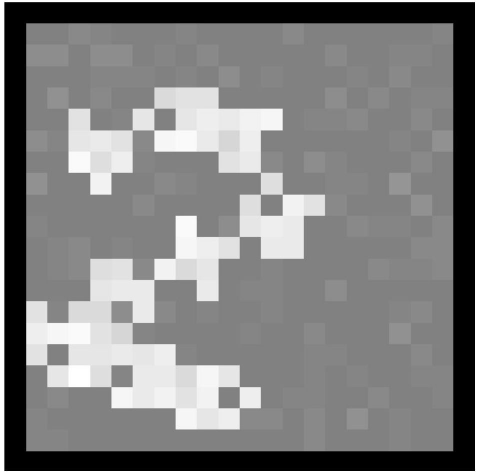
is maximum.

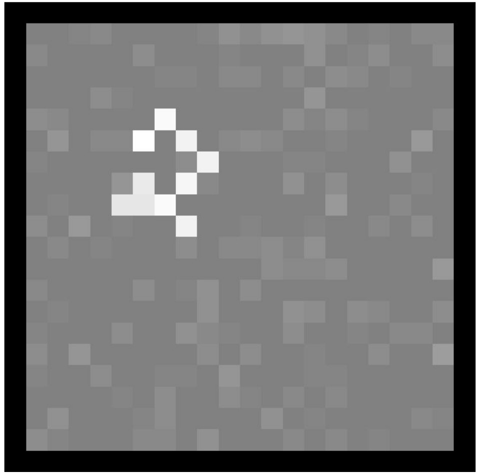
- Pe = .0076
- 38 out of 5000



Note on translation, scaling, and rotation processes

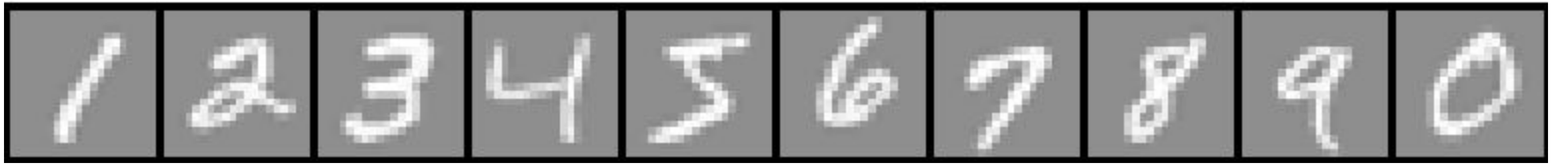
- Do not take into account interpolation
- Some rotations will come out better than others
- Using the same functions to generate the data and then detect makes the error very low.





Written digits

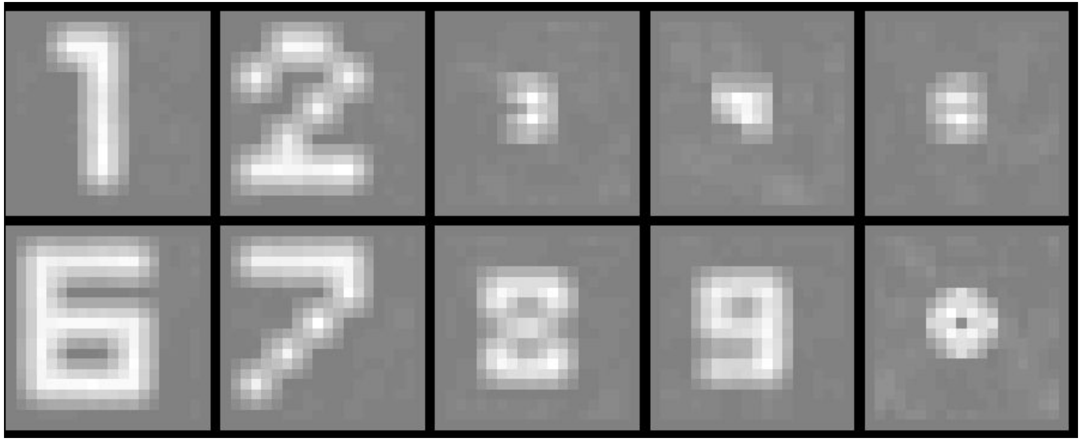
- Low error isn’t really that accurate
- The problems weren’t really different
- Use the same detector from previous slide on a different data set.
- Pe = .6884
- Pretty good compared to random classifier.



Written digits cont.

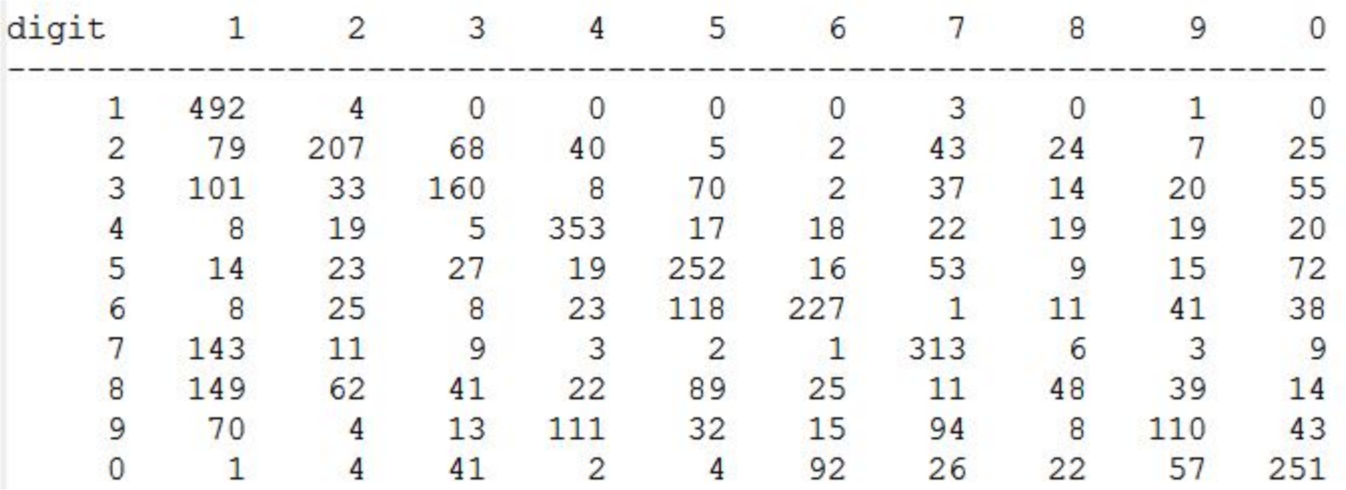
- Notice that written digits have smoother edges.
- blur the templates to make them closer the written digits.

- Simple 3x3 pixel averager



Written digits cont.

- After using the detector with blurred templates.
- Pe = .5174
- much better than random classification
- even pretty good compared to previous Pe = .6884



```
digit      1      2      3      4      5      6      7      8      9      0
---------------------------------------------------------------------------
    1    492      4      0      0      0      0      3      0      1      0
    2     79    207     68     40      5      2     43     24      7     25
    3    101     33    160      8     70      2     37     14     20     55
    4      8     19      5    353     17     18     22     19     19     20
    5     14     23     27     19    252     16     53      9     15     72
    6      8     25      8     23    118    227      1     11     41     38
    7    143     11      9      3      2      1    313      6      3      9
    8    149     62     41     22     89     25     11     48     39     14
    9     70      4     13    111     32     15     94      8    110     43
    0      1      4     41      2      4     92     26     22     57    251
```

Written digits cont.

- I think that I could improve the performance even more if I fixed the translation, scaling, and rotation processes to include interpolation.
- Maybe added a shrinker/stretcher process to account for slender digits like ‘8’.
- Obviously working with higher resolution images would help
- At that point the calculations would take prohibitively long.
    - a relatively small data set of 5000 images took about 1 hour to do the calculations on