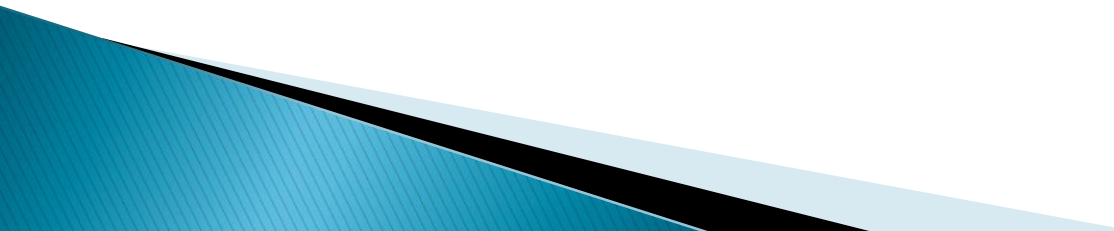


# Rozpoznawanie cyfr pisanych ręcznie

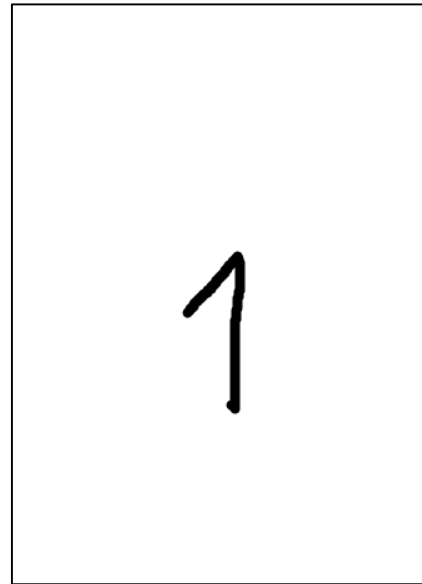
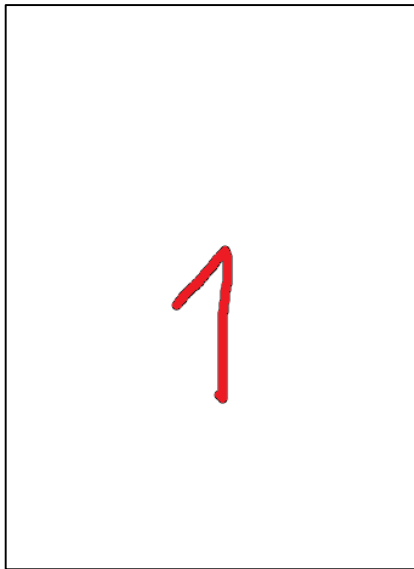
Marcin Okularczyk  
Dawid Kuczyński

# Założenia projektu

- ▶ Projekt oparty na sieci neuronowej
  - ▶ Program po wczytaniu pojedynczej cyfry, pisanej odręcznie, jako bitmapy jako wynik podaje co to za cyfra
  - ▶ GUI
- 

# Działanie programu

- ▶ Program po wczytaniu bitmapy przy użyciu filtru dokonuje konwersji na obraz czarno biały dla lepszego kontrastu



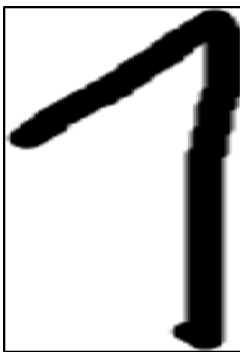
# Działanie programu

- ▶ Program znajduje czarne krawędzie cyfry, na tej podstawie przycina bitmapę aby pozdbyć się białych marginesów



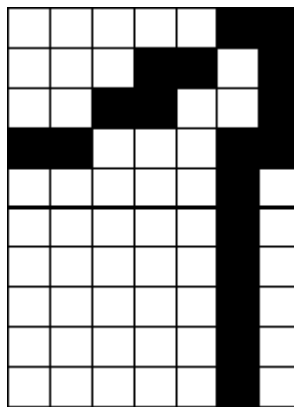
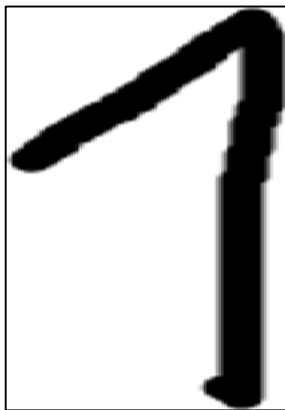
# Działanie programu

- ▶ Następnie program skaluje bitmapę do formatu 7x10 pikseli



# Działanie programu

- ▶ Teraz tworzymy macierz 7x10 i odwzorowujemy w niej wcześniej przygotowaną bitmapę



0	0	0	0	0	1	1
0	0	0	1	1	0	1
0	0	1	1	0	0	1
1	1	0	0	0	1	1
0	0	0	0	0	1	0
0	0	0	0	0	1	0
0	0	0	0	0	1	0
0	0	0	0	0	1	0
0	0	0	0	0	1	0

# Działanie programu

- ▶ Naszą macierz przekształcamy na wektor uczący, dla tego przykładu będzie następujący:

Dane wejściowe	Dane wyjściowe
0000011000110100110011100011000001000000100000010000001000000100000010	100000000

# Sieć neuronowa

- ▶ Dane wejściowe (macierz  $7 \times 10$ ) i wyjściowe (10 cyfr) determinuje nam sieć z jedną warstwą ukrytą, sieć będzie następująca

$$70 - X - 10$$

X – ilość neuronów w warstwie ukrytej (zostanie ustalona doświadczalnie)



# Sieć neuronowa

- ▶ Na warstwie wyjściowej otrzymamy wyniki dla każdego neuronu z prawdopodobieństwem na ile jest to cyfra, którą zna ten konkretny neuron

# Uczenie sieci neuronowej

- ▶ Uczenie metodą propagacji wstecznej błędów

# Przykładowe wywołanie programu

