

## Leser für handgeschriebene Zahlen mit Hilfe eines Neuronalen Netzes

---

### Aufgabenstellung:

Es sollen mit Hilfe von JavaNNS Neuronale Netze (Multilayer Perzeptron) zum Lesen handschriftlicher Zahlen konstruiert und trainiert werden. Die trainierten Netze sind in ein C-Programm zu übersetzen (snns2c.exe) und in ein C++-Programm zu integrieren.

### Beschreibung:

1. Es ist ein geeignetes Neuronales Netz (Multilayer-Perceptron) zu konstruieren. Dabei soll durch Auswahl und Erprobung verschiedener Netzkonfigurationen (Anzahl und Auslegung der Hidden-Layer) und Trainingsverfahren (Backpropagation-Momentum, Resilient-Propagation, o.ä.) ein möglichst gutes Netz erzielt werden.

Für das Training und den Test stehen folgende Pattern-Dateien zur Verfügung:

- a) Numbers\_Train.pat (Zeichenfenster 10 x 16)
- b) Numbers\_Test.pat (Zeichenfenster 10 x 16)

2. Das Netz ist in das vorbereitete C-Programm zu integrieren.

- a) Das trainierte Netz abspeichern (xyz.net).
- b) xyz.net umbenennen in "numbers.net"
- c) snns2c.exe mit numbers.net als Parameter aufrufen.  
--> es wird numbers.c und numbers.h erzeugt.
- d) In numbers.h folgendes ändern:  
--> alle float durch double ersetzen,  
---> die Funktionsdeklaration ergänzen um: `extern "C"`  

`extern "C" int numbers(double *in, double *out, int init);`
- e) In numbers.c alle float durch double ersetzen.
- f) numbers.c und numbers.h in den C++-Projektordner kopieren.
- g) Die beim Training verwendete Größe der Zeichenmaske muss im C-Programm eingestellt werden

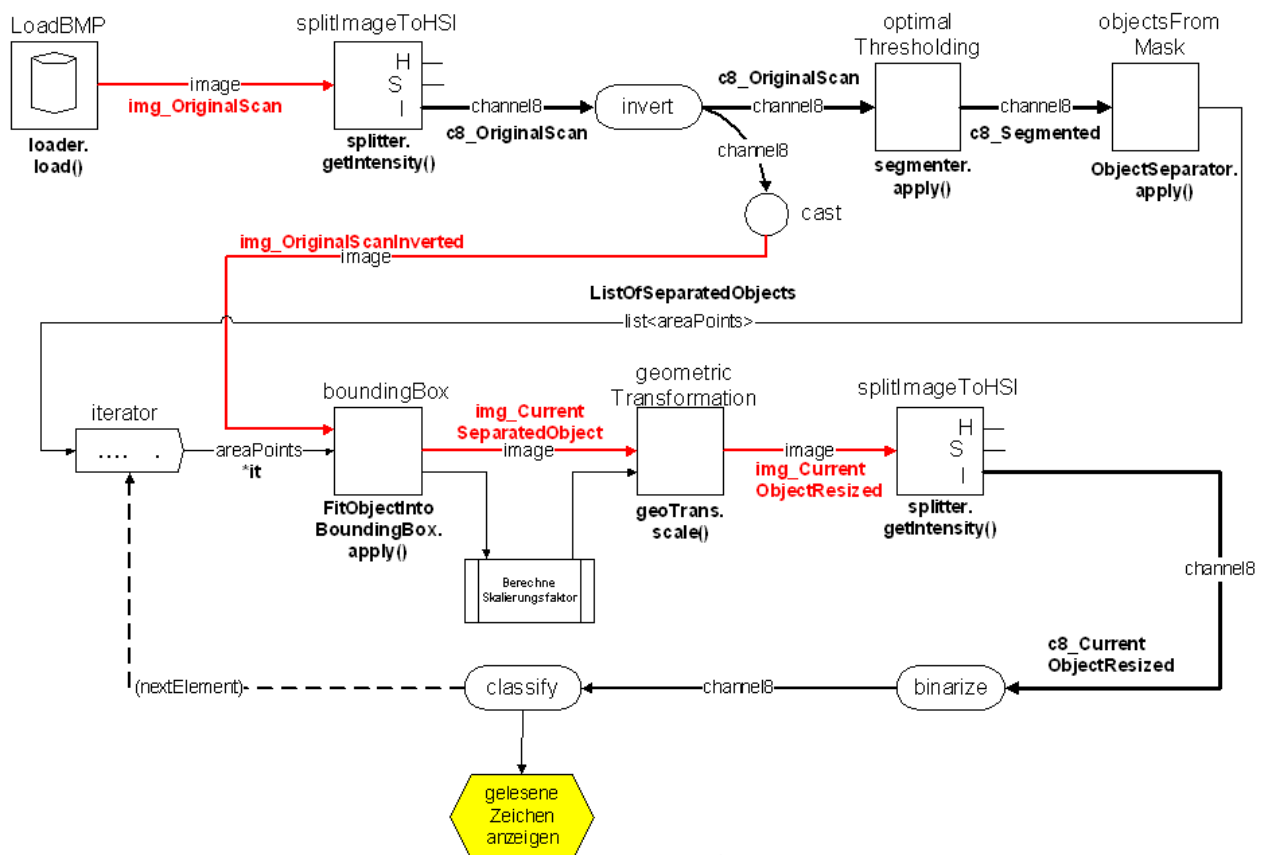
```
/* Program Constants */
const int    CHARMASK_X = 10;
const int    CHARMASK_Y = 16;

const int    NUMBER_OF_OUTPUT_CLASSES = 10;
```

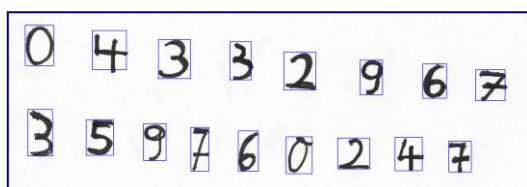
## Leser für handgeschriebene Zahlen mit Hilfe eines Neuronalen Netzes

### Arbeitsweise des LTILib-Programms:

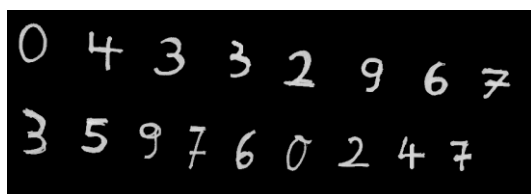
Die verschiedenen Verarbeitungsschritte des Programms sowie die jeweiligen Bildnamen und Datentypen sind in der folgenden Abbildung schematisch dargestellt:



eingescanntes Bild (mit Segmenten)



invertiertes Bild



segmentiertes Zeichen



reskaliertes und binarisiertes Zeichen (10 x 16)



**Leser für handgeschriebene Zahlen  
mit Hilfe eines Neuronalen Netzes**

---

**Dokumentieren Sie zu den getesteten Netzen/Verfahren die folgenden Informationen:**

Anzahl und Art der Hidden Layer:    30 / 20    (2 Hidden Layer mit 30 und 20 Neuronen)

Trainingsalgorithmus:                    Backpropagation / BP Momentum / Resilient Propagation

Trainingszyklen:                        z.B. 600

Restfehler (Sum  $e^2$ ) für *Train.pat*:    z.B. 0.5

Restfehler (Sum  $e^2$ ) für *Test.pat*:    z.B. 10.0

Anzahl fehlerhaft gelesener Zeichen  
auf *Numbers.bmp* :                        z.B. 0

Anzahl fehlerhaft gelesener Zeichen  
auf *Numbers\_difficult.bmp* :            z.B. 10

Besonderheiten:                        z.B. „lineare Aktivierungsfunktion für 2. Hidden Layer“