

Zahlenerkennung in der Informatik



1. Einleitung

Im Rahmen einer besonderen Lernleistung (BLL) habe ich mich selbstständig um die programmatische Umsetzung der Zahlenerkennung gekümmert. Dabei habe ich großen Wert auf die Umsetzung eigener Ansätze gelegt und es daher vermieden weitere Ansätze aus diversen Quellen zu recherchieren, um die Eigenleistung auf ein Maximum zu bringen. Die Ergebnisse werden zeitweise auf askcenter.ddns.net/BLL verfügbar sein. Auf Anfrage kann auch die eigene Handschrift getestet werden. Kontaktdaten finden sie im Impressum.

Zur Erstellung des Programms habe ich meine Vorkenntnisse in der Programmiersprache PHP genutzt.

2. Der Arbeitsprozess

→ 2.1 Ideenfindung:

Der Ideenfindungsprozess verlief zunächst chaotisch, da mir zwar bewusst war, dass ich eine besondere Lernleistung im Fachbereich Informatik erbringen wollte, allerdings noch nicht wusste, über welches Thema es sich handeln sollte.

Die erste Idee war es das Thema der Audioanalyse zu wählen. Dieses Feld ist allerdings sehr reichhaltig an verschiedenen Nutzungsmöglichkeiten verschiedenster Analyseformen und somit zu groß um es als ganzes zu behandeln. Dazu kommt die hohe Komplexität die mit einigen Teilen der Audioanalyse verbunden ist. Vor allem durch das Zusammenkommen verschiedener Fachbereiche der Mathematik, des Ingenieurwesens und natürlich der Informatik auf einem überschulischen Niveau fiel die Wahl hin zu einem anderen Thema.

Durch den Vorschlag des Fachlehrers eine Zahlenerkennung zu programmieren wandte ich mich von der Audioanalyse ab und hin zur Auswertung von Bildern, beziehungsweise dem erkennen geometrischer Formen aus rasterisierten Grafiken.

→ 2.2 Lösungsansätze:

0. Meine erste Idee war es aus den einzelnen Pixeln Formen werden zu lassen, da sich Zahlen auf die simpelsten Geometrischen Formen wie Geraden, Kurven und Kreise herunterbrechen lassen. Dies erwies sich allerdings als Trugschluss. Somit wurde dieser Ansatz verworfen.

1. Durch den Fachlehrer angeregt entstand dann ein neuer Ansatz, der durch die Zählung der Pixel in verschiedenen Bereichen einer Zahl die Differenzierung ermöglichen sollte. Nach der Ausarbeitung dieser Idee zeigte sich allerdings, dass ihre Effektivität mit einer Erkennungsrate von nur etwa 30% bei vom Computer geschriebenen Zahlen und 14% bei handschriftlich geschriebenen Zahlen im Anbetracht der Tatsache, dass die Wahrscheinlichkeit dafür, dass man eine Zahl richtig errät bei 10% liegt, sehr niedrig war. Daher musste ein neuer Ansatz her.

2. Mein nächster Einfall war es, die Schwarz-Weiß-Wechsel einer Zahl zu zählen und somit determinieren zu können, welche Zahl nun dargestellt wird. Ab etwa der Hälfte der Umsetzung haben meine Tests eine bessere Rate als Ansatz 1. geliefert. Allerdings waren sie immer noch nicht zufriedenstellend.

3. Einige Überlegungen während der Ausarbeitung der verschiedenen Ansätze und letztendlich auch der Idee des nullten Ansatzes brachten mich letztendlich zu meiner finalen Idee. Bei der Überlegung was unsere arabischen Zahlen identifizierbar macht, kam mir eine Siebensegmentanzeige in den Sinn. Nach diesem Vorbild habe ich versucht die Zahlen auf diese Form herunterzubrechen und anschließend hauptsächlich mithilfe der Schwarz-Weiß-Wechsel in einzelnen auf drittel aufgeteilten Bereichen, auch mit bestimmten Aspekten des ersten Ansatzes, zu analysieren.

→2.3 Testumgebung:

Für den Test meiner Ideen war es nötig grafische Darstellungen für jeden meiner Ansätze anzufertigen. Glücklicherweise konnte ich dadurch, dass ich mit PHP gearbeitet habe, alle Funktionen von HTML und CSS nutzen, sodass es ein leichtes war, mir meine Ansätze grafisch logisch darzulegen.

→2.4 Umsetzung:

Die Umsetzung der Ideen in eine Applikation verlief immer nach einem bestimmten Prinzip. Zuerst wurden alle Zahlen in eine Art Oberkategorie für die jeweiligen Ansätze gebracht. Diese Gruppierungen entstanden bei der Differenzierung der einzelnen Ziffern jedes mal ganz automatisch. Zunächst musste ich nach grundlegenden und großen Unterschieden suchen, die es mir ermöglichen würden größere Zahlengruppen aus dem Pool unterscheiden zu können. Dann habe ich in diesen Gruppen untereinander nach Unterschieden gesucht. Das Prinzip habe ich so

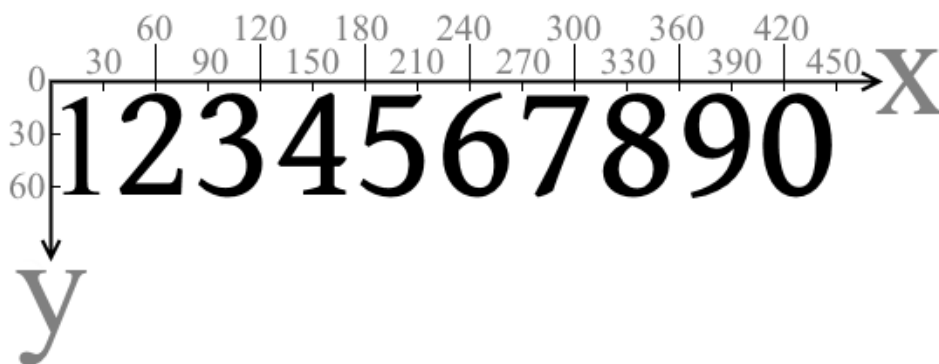
lange verfolgt, bis nur noch die Unterscheidung zwischen 2 verschiedenen Zahlen nötig war, um diese dann via Ausschlussverfahren differenzieren zu können. Je nach Font oder Handschrift konnte es allerdings passieren, dass Zahlen zu Beginn der falschen Kategorie zugeordnet wurden. Daher mussten noch andere Merkmale der Zahl in Betracht gezogen werden um sie von der Gruppe zu unterscheiden.

3. Erklärung und Darstellung

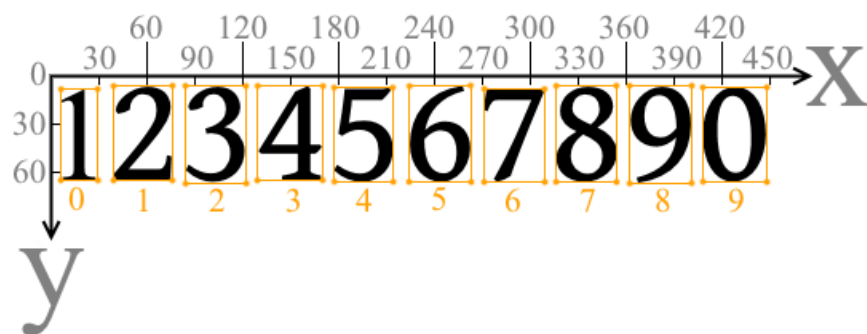
→3.1 Akquirierung der Bilddaten:

Zunächst war es wichtig, die Bilder in einem gut zu bearbeitenden Format vorliegen zu haben. Ich habe mich für ein 2-Dimensionales Array mit X- & Y-Koordinaten als Indizes bestehend aus Nullen für schwarz und Einsen für alle anderen Farben entschieden (**Image Array**). Für jede einzelne Nummer die abgebildet ist habe ich die Dimensionen in einem weiteren Array gespeichert (**Dimensions Array**). Für die Verfolgung des ersten Ansatzes brauchte ich noch ein weiteres Array, welches die Anzahl der schwarzen Pixel in den neun gleichgroßen Feldern speichert (**Pixel Count Array**). Ein weiteres Array aus der Anzahl von Schwarz-Weiß-Wechseln war für den zweiten und dritten Ansatz notwendig (**BWChanges Array**).

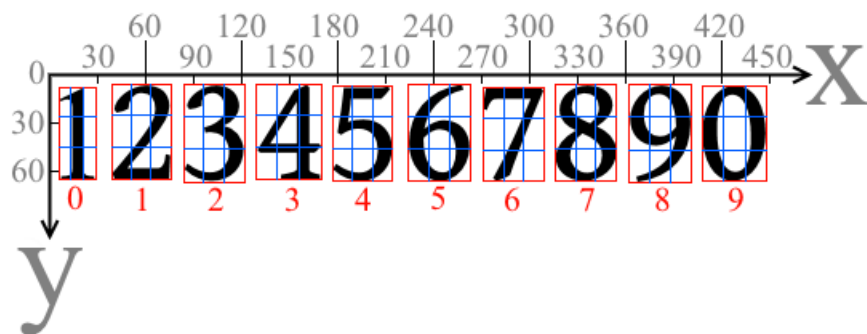
Image Array

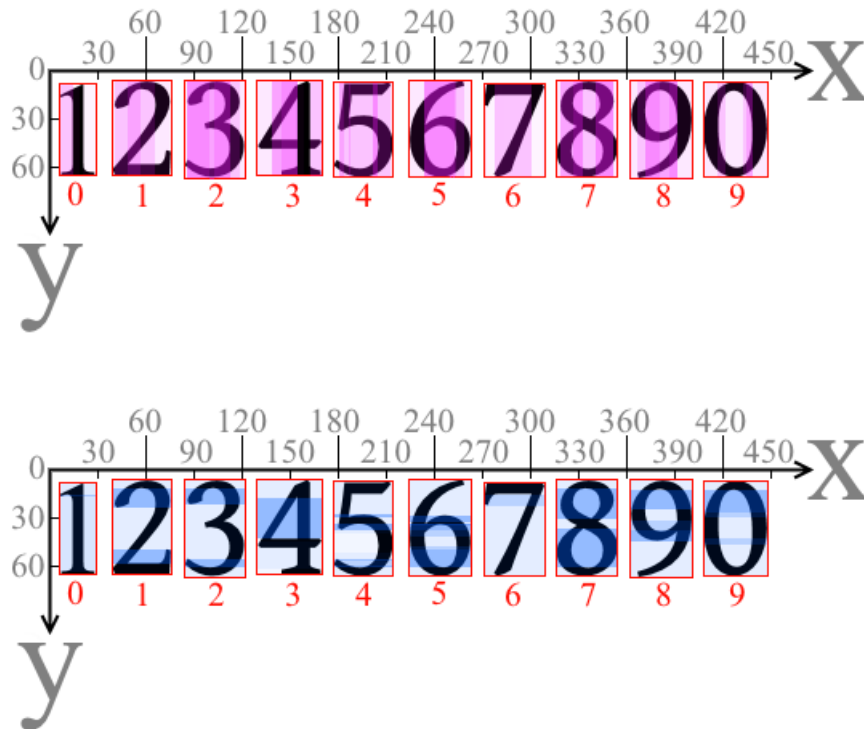


Dimensions Array



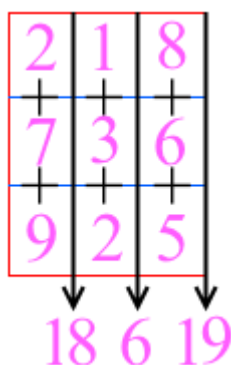
Pixel Count Array





→3.2 Ansatz 1: Pixel zählen

Bei dem Vergleich zwischen den Verschiedenen oben dargestellten Zahlen im Pixel Count Array lässt sich ein Muster erkennen, welches die Zahlen 0, 6, 8, 9 und 5 in einer Gruppe zusammenschließt:

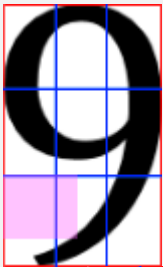
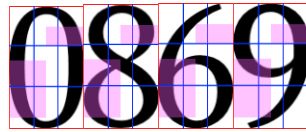


Die Pinken Zahlen stehen für einen beliebigen prozentualen Anteil von schwarzen Pixeln in den einzelnen Bereichen. Bei dem Vergleich zwischen der linken, mittleren und rechten Spalte fällt bei diesen 5 Zahlen auf, dass die linke und rechte Spalte jeweils mehr schwarze Pixel beinhalten als die mittlere.

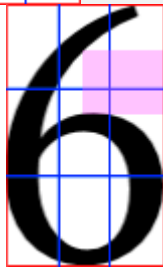
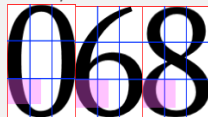
Die nächsten Differenzierungen zwischen den Zahlen erfolgt auf individuellem Level.



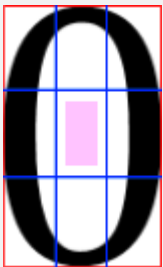
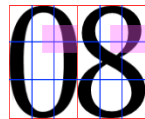
Hier sollen die Pixel im markierten Bereich gezählt werden. Solange es keine Überschreitung des Grenzwertes von jeweils 1% gibt wird angenommen, dass die Zahl eine 5 ist.



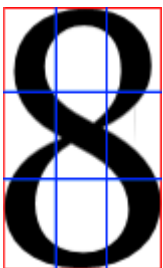
Falls die Zahl nicht als 5 erkannt wurde, wird ein neuer Bereich überprüft. Da die Zahlen 0, 6 und 8 noch aus dieser Gruppe übrig sind ist die einzige Zahl, die den Grenzwert von 1% nicht überschreitet, die 9.



Da die Zahl 5 bis jetzt schon erkannt worden wäre kann ein neuer Bereich überprüft werden, der mit hoher Wahrscheinlichkeit auch für eine 5 gelten würde.

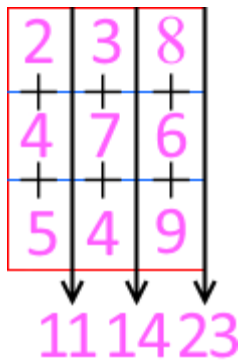


Die einzige Unterscheidung die noch fehlt ist somit die zwischen 0 und 8. Genau wie die male davor werden auch hier die schwarzen Pixel im Markierten Bereich gezählt und überprüft, ob sie mehr als 1% der Gesamtanzahl ausmachen.



Falls bis jetzt keine der Zahlen aus der Gruppe erkannt wurden kann per Ausschlussverfahren festgestellt werden, dass die Zahl eine 8 sein muss.

Ähnlich wie zuvor lässt sich ein bestimmtes Muster erkennen, welches die Zahlen 2, 3 und 4 in eine Gruppe einordnet:



Hier lässt sich erkennen, dass die Anzahl der in der linken Spalte vorkommenden schwarzen Pixeln niedriger ist, als die der beiden anderen. Die mitte enthält zwar mehr schwarze Pixel als die linke, hat allerdings auch weniger Schwarzanteil als die rechte.

Die weitere Differenzierung erfolgt auch hier erneut auf individueller Ebene.



Erneut wird in dem markierten Bereich geprüft, ob die Anzahl der schwarzen Pixel über 1% liegt. Da die neue Gruppe jetzt aus 3 anderen Zahlen besteht, müssen auch nur diese auf diese Weise unterschieden werden.



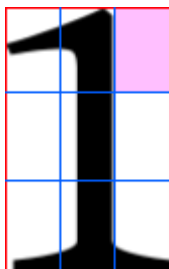
Dieses mal werden Schwarz-Weiß-Wechsel überprüft, was auch das erste mal war, dass diese Idee aufkam. Das Bild wird von links nach rechts zeilenweise abgefahren. Bei 3 Farbwechseln wird angenommen, dass die Zahl eine 4 ist.



Durch das Ausschlussverfahren lässt sich erneut bestimmen, dass die Zahl eine 2 sein muss.

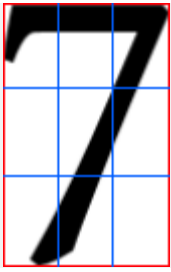


Die Zahlen die noch übrig bleiben sind somit 1 und 7.



Diese lassen sich gut durch das Überprüfen des links abgebildeten Bereichs unterscheiden. Bei einer Zählung der Anzahl der schwarzen Pixel wird überprüft, ob diese mehr als 1% des gesamten Bildes ausmachen.





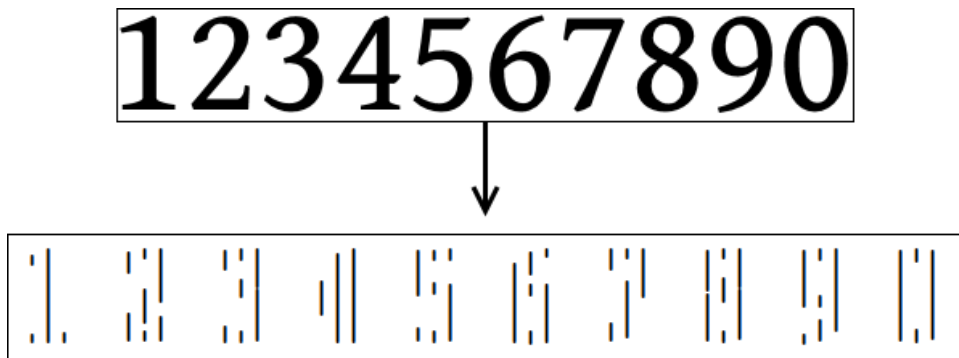
Somit ist die letzte noch übrige Zahl eine 7.

Diverse Überprüfungen sind auch für andere Gruppenkonstellationen eingebaut, damit keine Zahl, falls sie versehentlich in eine falsche Gruppe eingeordnet wird, unerkant bleibt.

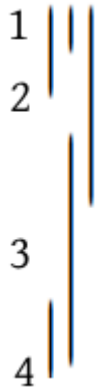
Das Gute an diesem Ansatz ist es natürlich, dass der Algorithmus diesen Font mit einer Wahrscheinlichkeit von 100% erkennt. Er funktioniert allerdings kaum bis zu gar nicht für andere Schriftarten, weshalb er verworfen wurde.

→ 3.3 Ansatz 2 & 3: Schwarz-Weiß-Wechsel & Hohlräume

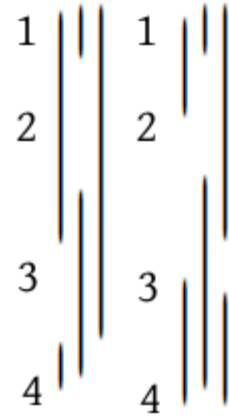
Zunächst war es wichtig die Zahlen auf ihre Grundform herunter zu brechen. Nach dem Vorbild der Siebensegmentanzeige sollten die Zahlen vereinfacht werden:



Dazu werden in jedem vertikalen Drittel die schwarzen Pixel überprüft. Sobald ein Pixel in einem Drittel auf einer Höhe Y erkannt wird, wird diese Höhe in dem Bereich als schwarz verzeichnet. Hiervon ausgehend werden nun erneut Gruppen erstellt. Das mittlere Drittel bietet dafür die besten Voraussetzungen, da es dort die meisten Unterschiede gibt. Die Zählung der Farbwechsel erfolgt ab dem ersten schwarzen Bereich.



Die Erste überprüfte Gruppe besteht aus einer 7, einer 2 und einer 9, wobei 2 und 9 nur dabei sind, um sie bei unzureichender Zeichnung immer noch zu erkennen. In dieser Gruppe wird davon ausgegangen, dass es insgesamt 4 Schwarz-Weiß-Wechsel im mittleren Bereich gibt.



Für die weitere Unterscheidung wird nun geprüft, ob der Teil des zweiten Wechsels einen Hohlraum darstellt. Dazu wird im linken und rechten Drittel der Bereich zwischen den Farbwechseln von 1 zu 2 und von 2 zu 3 nach Lücken überprüft. Falls diese Bereiche keine Lücken aufweisen, wird davon ausgegangen, dass die Zahl eine 9 ist.

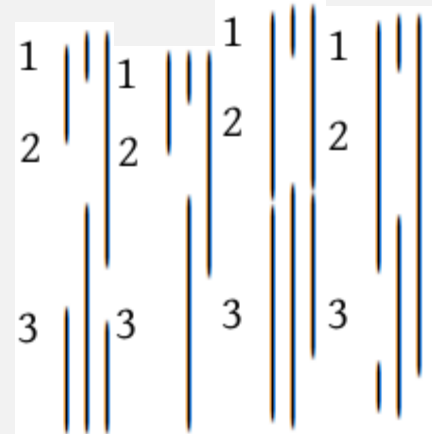


Um die verbleibende Unterscheidung zwischen 2 und 9 zu machen wird im letzten viertel aller 3 Bereiche überprüft, ob in allen 3 ein schwarzer Teil vorhanden ist. Falls dies der Fall ist, wird angenommen, dass es sich um eine 2 handelt. Falls nicht, wird eine 7 angenommen.

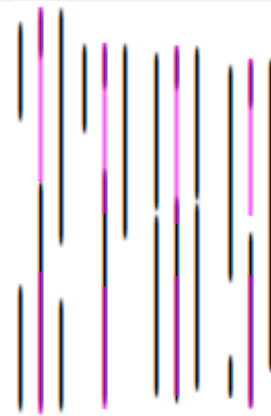




Bei einer Anzahl von 3 Schwarz-Weiß-Wechseln kommt im Idealfall nur die Zahl 0 in diese Gruppe. Durch undeutliche Zeichnung der Zahlen kommt es allerdings oft vor, dass die Zahlen 2, 7, 8 und 9 auch zu dieser Gruppe gezählt werden, wodurch diese auch in dieser Gruppe berücksichtigt werden müssen.



Als erstes wird die 0 aus der Gruppe herausgefiltert. Indem überprüft wird, ob ein Wechsel von weiß zu schwarz im unteren Drittel der mittleren Spalte vorliegt, sowie ein Farbwechsel in der oberen Hälfte existiert, wird sicher gestellt, dass es sich um eine 0 handelt. Wie man in den Darstellungen sehen kann, ist dies auch nur bei der 0 mit einer relativ hohen Wahrscheinlichkeit der Fall.



Um sicher zu gehen, dass es sich auch um eine 0 handelt wird zudem überprüft, ob es einen Hohlraum zwischen den beiden schwarzen Teilen der mittleren Spalte gibt. Dies ist allerdings auch bei den Zahlen 8 und 9 in diesem Beispiel zutreffend, weswegen diese Methode lediglich zur weiteren Unterscheidung zwischen einer 2 oder einer 7 dient.

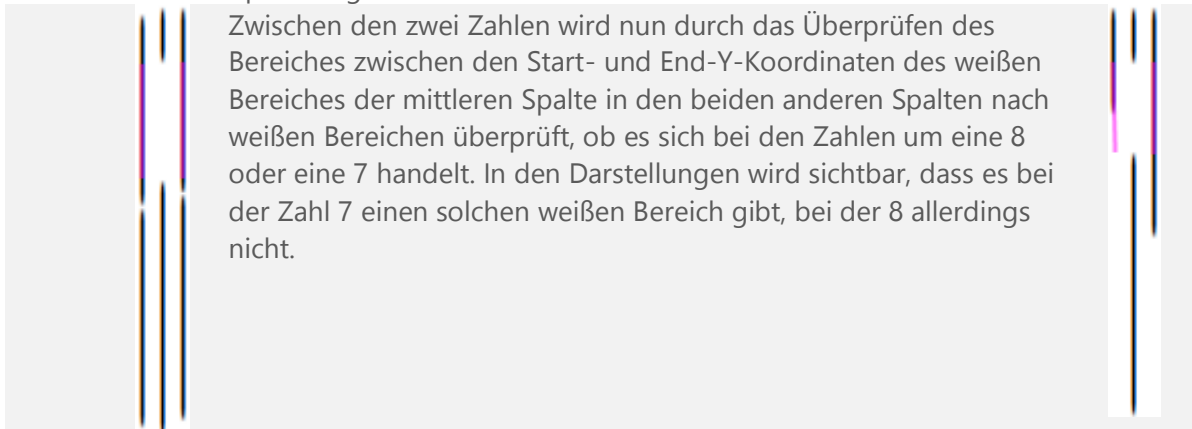


Als nächstes werden kleinere Gruppen aus insgesamt 2 Zahlen gebildet. Es wird deutlich, dass die Unterscheidungen in dieser Gruppe aus 5 Zahlen relativ schwer fällt, da die Zahlen auf verschiedene Arten ungenau gezeichnet sein könnten, sodass man, selbst wenn man wüsste, dass es sich um eine dieser Zahlen handelt, die als Ausnahme in diese Gruppe fallen, nicht sagen kann, ob sie wegen der fehlenden Lücke im unteren oder oberen Bereich dort gelandet sind. Daher, und da es sich hier um einen absoluten Ausnahmefall handelt wird von der häufigsten Form für eine ungenau gezeichnete Zahl ausgegangen und basierend darauf überprüft, ob der zweite



schwarze Bereich oberhalb des zweiten Drittels in der mittleren Spalte beginnt.

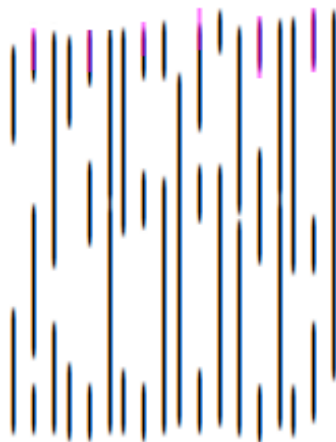
Zwischen den zwei Zahlen wird nun durch das Überprüfen des Bereiches zwischen den Start- und End-Y-Koordinaten des weißen Bereiches der mittleren Spalte in den beiden anderen Spalten nach weißen Bereichen überprüft, ob es sich bei den Zahlen um eine 8 oder eine 7 handelt. In den Darstellungen wird sichtbar, dass es bei der Zahl 7 einen solchen weißen Bereich gibt, bei der 8 allerdings nicht.



Um die zweite Kleingruppe aus zwei Zahlen zu unterscheiden wird die Anzahl der schwarzen Pixel in der oberen und unteren Hälfte der mittleren Spalte verglichen. Falls die Anzahl in der oberen Hälfte größer ist, als die Anzahl in der unteren wird angenommen, dass es sich um eine 9 handelt. Falls nicht, wird eine 2 angenommen.



Die nächste Gruppierung besteht aus allen Zahlen, die mehr als 5 Schwarz-Weiß-Wechsel in der mittleren Spalte aufweisen. Dazu gehören die Zahlen 2, 3, 5, 6, 8 und 9. Die restlichen Zahlen, die aufgrund undeutlicher Zeichnung in eine der da vorigen Gruppen eingeteilt hätten werden können und nicht zu dieser Gruppe gehören, gehören dementsprechend zur letzten Gruppe an. Diese beinhaltet demnach nur noch die Zahlen 1 und 4.



Um sicher zu stellen, dass es sich tatsächlich um eine dieser Zahlen handelt wird sicherheitshalber überprüft, ob der erste Farbwechsel innerhalb des ersten Neuntels des mittleren Bereiches stattfindet.

Die darauf folgende Unterscheidung der Zahlen verläuft hauptsächlich mithilfe der Überprüfung nach Hohlräumen, beziehungsweise der Überprüfung nach Teilweise vorhandenen Hohlräumen oder C-Formen.



Zunächst wird zwischen der 9 und der 8 unterscheiden, die auch gleich durch diese Unterscheidung aus der Gruppe fallen. Die 8 ist die einzige Zahl die sowohl Oben, als auch Unten in sich geschlossen ist. Die neun hingegen ist die einzige Zahl, die Oben zwar in sich geschlossen, Unten allerdings offen ist.



Die 6 und in Ausnahmefällen die 2 sind die einzigen, die Oben in sich offen, unten allerdings geschlossen sind. Sie werden durch das Überprüfen nach weißen Bereichen zwischen der Hälfte und zwei Dritteln in der linken Spalte unterschieden. Falls es dort einen weißen Bereich gibt, ist es eine 2.



Bei den restlichen 3 Zahlen lässt sich auch schnell feststellen, dass sie weder im unteren, noch im oberen Bereich in sich geschlossen sind, sondern in den jeweiligen Bereichen nur teilweise, beziehungsweise nur von einer Seite geschlossen sind.

Also wird als nächstes überprüft, ob es C-Förmige partielle Hohlräume gibt. Bei einer 5 wäre also im oberen Bereich ein solcher Hohlraum mit Öffnung nach rechts und im unteren Bereich einer mit Öffnung nach links, bei einer 3 wären beide nach links hin geöffnet und bei einer 2 wäre der obere nach links und der untere nach rechts hin geöffnet.

1
2
3

Um die letzte Gruppe, bestehend aus 1 und 4 zu unterscheiden werden die Schwarz-Weiß-Wechsel, hier ebenfalls beginnend ab dem ersten schwarzen Bereich, in der linken Spalte gezählt. Sobald die Anzahl größer als 2 ist, wird eine 1 als Zahl angenommen. Sobald die Anzahl kleiner ist, wird überprüft, ob der zweite Schwarz-Weiß-Wechsel unterhalb der Hälfte liegt. Falls nicht, wird auch hier eine 1 angenommen, falls schon, eine 4.

1
2

4. Tests

→4.1 Schlecht gescannte Handschrift ohne vorgaben

[0]
=> 0
[1]
=> 4
[2]
=> 2
[3]
=> 3
[4]
=> 4
[5]
=> 2
[6]
=> 6
[7]
=> 1
[8]
=> 8
[9]
=> 9

Richtige: 7

[0]
=> 0
[1]
=> 4
[2]
=> 1
[3]
=> 9
[4]
=> 4

Richtige: 6

[5]
=> 5
[6]
=> 6
[7]
=> 3
[8]
=> 8
[9]
=> 9

[0]
=> 0
[1]
=> 1
[2]
=> 2
[3]
=> 3
[4]
=>

N.A.

[5]
=>

N.A.

[6]
=> 6
[7]
=> 1
[8]
=> 8
[9]
=> 9

[0]
=> 8

[1]
=> 4

[2]
=> 4

[3]
=> 9

[4]
=> 4

[5]
=>

N.A.

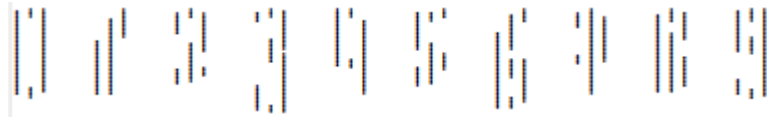
[6]
=> 2

[7]
=> 1

[8]
=> 1

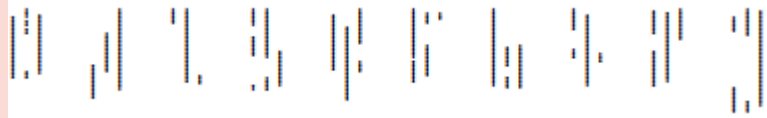
[9]
=> 9

0 1 2 3 4 5 6 7 8 9



Richtige: 7

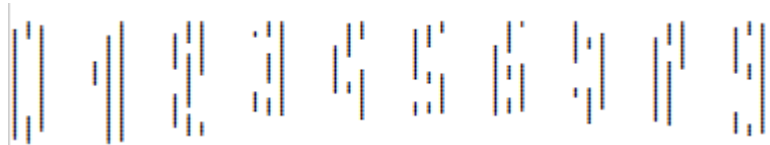
0 1 2 3 4 5 6 7 8 9



Richtige: 2

[0]
=> 0
[1]
=> 4
[2]
=> 9
[3]
=> 3
[4]
=>
N.A.
[5]
=> 5
[6]
=> 6
[7]
=> 7
[8]
=> 8
[9]
=> 9

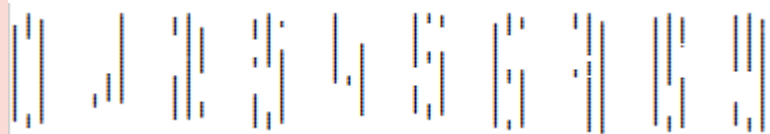
0 1 2 3 4 5 6 7 8 9



Richtige: 7

[0]
=> 0
[1]
=> 4
[2]
=> 2
[3]
=> 9
[4]
=> 4
[5]
=> 5
[6]
=> 6
[7]
=> 2
[8]
=> 9
[9]
=> 9

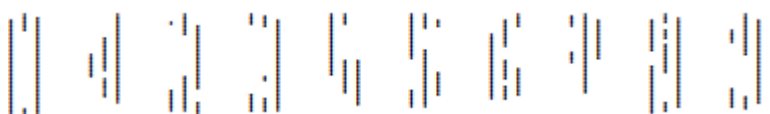
0 1 2 3 4 5 6 7 8 9



Richtige: 6

[0]
=> 0
[1]
=>
N.A.
[2]
=> 2
[3]
=> 3
[4]
=>
N.A.
[5]

0 1 2 3 4 5 6 7 8 9



Richtige: 6

=> 7
[6]
=> 6
[7]
=> 1
[8]
=> 8
[9]
=> 9

Richtige

41

Getestete

70

Prozentual Richtige

58,57%

→4.2 Computergeschriebene Zahlen

[0]
=> 0
[1]
=> 1
[2]
=> 2
[3]
=> 3
[4]
=>
N.A.
[5]
=> 5
[6]
=> 6
[7]
=> 7
[8]
=> 8
[9]
=> 9

0123456789

0 1 2 3 4 5 6 7 8 9

Richtige: 9

[0]
=> 0
[1]
=> 1
[2]
=> 2
[3]
=> 9
[4]
=> 4
[5]
=> 5
[6]
=> 6
[7]
=> 7

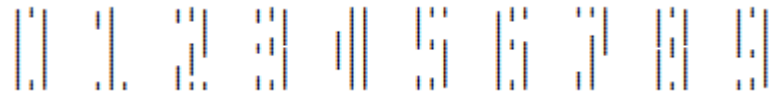
0123456789

0 1 2 3 4 5 6 7 8 9

Richtige: 9

[8]
=> 8
[9]
=> 9

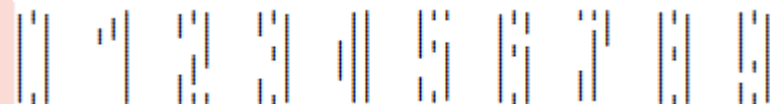
0123456789



Richtige: 10

[0]
=> 0
[1]
=> 1
[2]
=> 2
[3]
=> 3
[4]
=> 4
[5]
=> 5
[6]
=> 6
[7]
=> 7
[8]
=> 8
[9]
=> 9

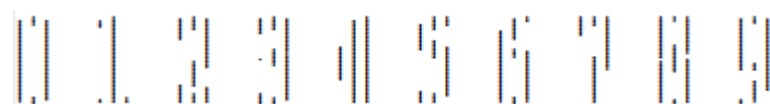
0123456789



Richtige: 10

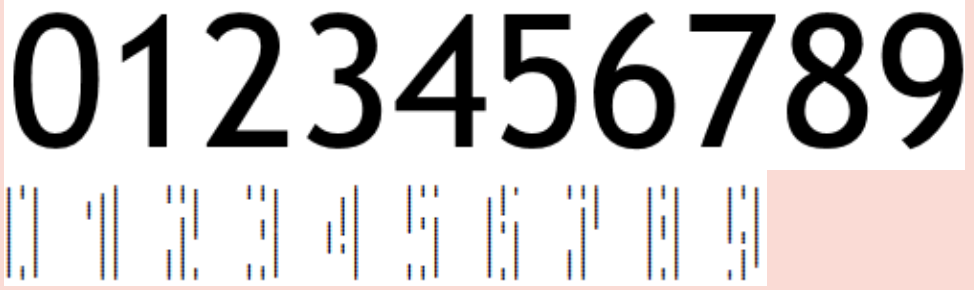
[0]
=> 0
[1]
=> 1
[2]
=> 2
[3]
=> 3
[4]
=> 4
[5]
=> 5
[6]
=> 6
[7]
=> 7
[8]
=> 8
[9]
=> 9

0123456789



Richtige: 10

[0]
=> 0
[1]
=> 1
[2]
=> 2
[3]
=> 3
[4]
=> 4

[5] => 5	
[6] => 6	
[7] => 7	
[8] => 8	
[9] => 9	
[0] => 0	
[1] => 1	
[2] => 7	
[3] => 3	
[4] =>	
N.A.	
[5] => 5	
[6] => 6	
[7] => 7	
[8] => 8	
[9] => 9	
Richtige	56
Getestete	60
Prozentual Richtige	93,33%

5. Fazit

Insgesamt ist das Ziel der Erkennung von Computergeschriebenen Zahlen für die Standardisierten Fonts zum Großteil geglückt. Da das die Hauptanforderung ist bin ich relativ zufrieden. Die Funktionalität mit verschiedenen skalierbaren Zahlen, solange sie nicht bis zur Unkenntlichkeit geschrumpft sind, ist zudem zufriedenstellend. Ich habe mir allerdings ein besseres Ergebnis für die Handgeschriebenen Zahlen erhofft. Auch wenn keine Scanfehler aufgetreten wären, hätte sich das Ergebnis nur minimal verändert.

Die Erfahrung hat mir sehr geholfen und wird auch bei Zukünftigen Projekten hilfreich sein, da sich das Gelernte nicht nur auf Zahlen spezialisiert, sondern auch auf struktureller Ebene von großer Bedeutung ist.