

COMPUTER AND ROBOTVISION
(PRÜFUNGSNUMMER: 135031)

BRIEFMARKENERKENNUNG

GEORG JAHN, MARTIN HAAG*

25. Januar 2022

Eingereicht bei PROF. DR. DIETER MAIER

*195XXX, 194980, gjahn@stud.hs-heilbronn.de, mahaag@stud.hs-heilbronn.de

Inhaltsverzeichnis

List of Abbreviations	III
Abkürzungsverzeichnis	IV
Management Summary	V
1 Einleitung	1
1.1 Motivation	1
1.2 Ziel der Arbeit	1
1.3 Vorgehensweise	1
2 Klassische Bildverarbeitung	3
2.1 Übersicht	3
2.2 Beispiel ohne Komplikationen	3
2.3 Mögliche Probleme	3
2.4 Programmablaufplan	3
3 Neuronales Netzwerk	8
3.1 Architektur und Vorgehen	8
3.2 Implementierung und Auswertung	8
4 Fazit	9
Literaturverzeichnis	10

List of Abbreviations

Abkürzungsverzeichnis

ABK: ABKÜRZUNG

Management Summary

...

1 Einleitung

1.1 Motivation

Bildverarbeitung ist kein neues Feld der Forschung mehr und auch neuronale Netzwerke sind bereits seit vielen Jahren Subjekt von ständiger Weiterentwicklung. Da diese beiden Fachgebiete durchaus gut miteinander harmonieren haben wir uns entschieden, in unserem Projekt eine Aufgabe die beides benötigt zu bearbeiten. Im Rahmen unseres CRV Projektes haben wir deshalb die Unterscheidung von Briefmarken in gestempelte und ungestempelte automatisiert, durchgeführt. Wenn auch unsere Arbeit sehr unwahrscheinlich kommerzielle Anwendung findet, ist es doch geeignet um an einem Tag der offenen Tür oder einem Studientag demonstriert zu werden.

1.2 Ziel der Arbeit

Das Ziel der Arbeit war es, mit Hilfe eines Teils "klassischer" Bildverarbeitung und eines neuronalen Netzwerkes Bilder von gestempelten von Bildern mit ungestempelten Briefmarken zu unterscheiden. Ursprünglich war der Plan mit Bildern von ganzen Briefen zu arbeiten. Der Vorteil bei diesem Ansatz wäre, dass die Überlappung des Stempels über Briefmarke und Brief es wohl für das Netzwerk einfacher gemacht hätte die Klassen gestempelt/nicht gestempelt zu unterscheiden. Auf Grund von Datenschutzgesetzen waren wir jedoch nicht in der Lage, einen ausreichend großen Datensatz zusammen zu tragen. Deshalb haben wir uns entschieden, eine alte selbst gesammelte Briefmarkensammlung abzufotografieren und somit zu digitalisieren.

Dieser selbst generierte Datensatz ist zwar auch nicht groß genug um ein ernsthaftes Training mit einem NN durchzuführen, reicht aber für einen "Proof of concept"

1.3 Vorgehensweise

Unser Vorgehen teilt sich in zwei Teilschritte auf:

- "klassische" Bildverarbeitung:

Dabei werden die Bilder so vorverarbeitet, dass sie als Input für das NN dienen können. Das wird mit Hilfe von helligkeitsbasierte Binarisierung und einigen morphologischen Filtern erreicht. Außerdem kommen noch Schritte zum drehen des Bildes zum Einsatz. Das Endergebnis dieser Bildverarbeitungsschritte ist ein rechteckiger Bildausschnitt, der nur die Briefmarke enthält.

- Klassifizierung mit einem NN:

In diesem Bereich der Arbeit wird der vorverarbeitete Datensatz für das Training und die Evaluierung des NN aufbereitet. Außerdem wird das NN selbst hier erstellt. Da wir die Technik des Transferlearning genutzt haben, wurde hier nur ein vortrainiertes Model herausgesucht und das Outputlayer ausgetauscht.

2 Klassische Bildverarbeitung

2.1 Übersicht

Das folgende Schaubild zeigt, welche Schritte vorgenommen werden und in welcher Reihenfolge. Dafür wurde ein Beispiel gewählt, bei dem keine Komplikationen auftreten.

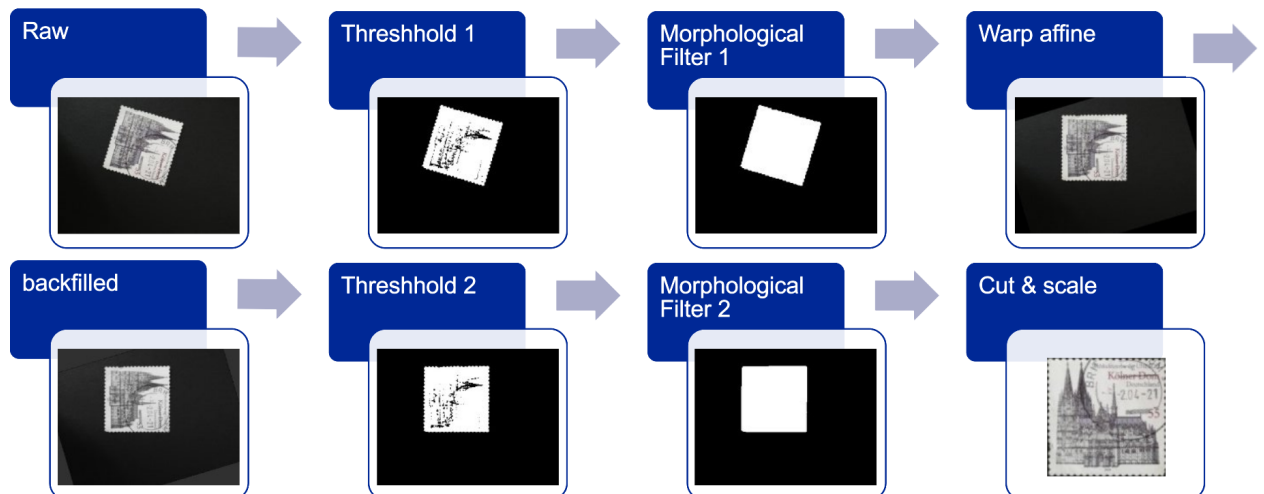


Abb. 2.1: Übersicht über die Schritte der Bildvorverarbeitung

2.2 Beispiel ohne Komplikationen

Im Folgenden wird der Ablauf der klassischen Bildverarbeitung an einem Beispiel, das ohne weiteren Aufwand funktioniert, genauer erläutert.

2.3 Mögliche Probleme

In diesem Abschnitt sollen aufgetretene Probleme geschildert und die gefundenen Lösungen gezeigt werden.

2.4 Programmablaufplan

Die beiden Bilder auf der rechten Seite sind Eingang und Ausgang des ersten Schrittes (Raw \rightarrow Threshold1). Das Raw Bild ist dabei ein RGB Bild, auf dem die Briefmarke in zufälliger Orientierung zu sehen ist. Der Hindergrund ist matt und schwarz-grau. Der Übergang zu Threshold1 wird mit einer Otsu Binarisierung bewerkstelligt. Es wird also in ein Graustufenbild übergegangen. Je nach Motiv der Briefmarke sind im Binärbild jedoch noch viele schwarze Stellen auf der Briefmarke. Dadurch resultiert die Marke nicht in einer einzelnen großen Kontur, sondern in vielen kleinen. Da wir jedoch erstmal das Bild gerade drehen wollen, bräuchten wir dafür eine einzige große, möglichst rechteckige, Kontur.



Abb. 2.2: Übergang zu Threshold1

Die Abbildung 2.3 zeigt das Ergebnis des zweiten Schrittes des "klassischen" Teiles. Hier sind keine dunklen Stellen mehr auf der Fläche der Briefmarke. Das wurde durch die Verwendung eines morphologischen Filters zur Erweiterung und Schließung von Konturen erreicht. Es wurde dafür eine Quadratische Matrix mit Größe 11x11 verwendet. Bei einer Auflösung von 320x240 ist das ganz stattlich.

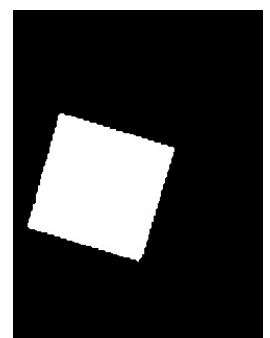


Abb. 2.3: Übergang zu morph. Filter1

Die Abbildung 2.4 zeigt, wie gut die Kontur aus dem Binärbild nun die Briefmarke repräsentiert. Der blaue Rahmen aus dem oberen Teil der rechten Abbildung wird mit der opencv Methode `minAreaRect` aus der Kontur des Binärbildes gewonnen. Mit dem Verdrehwinkel dieses Rechtecks kann nun das ursprüngliche Bild ausgerichtet werden. Dies geschieht mit Hilfe der `warpAffine` Funktion. Um diese zu nutzen muss vorher noch eine 2D Rotationsmatrix erstellt werden. Dafür werden wie Werte des Rechtecks verwendet. Das gerade gedrehte Bild ist im unteren Teil von Abbildung 2.4 zu erkennen.

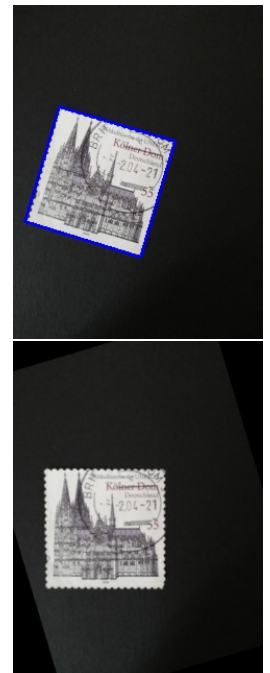


Abb. 2.4: Rechteck, gerade gedrehtes Bild

Aus dem gerade gedrehten Bild der Abbildung 2.4 kann nun wieder ein Binärbild erstellt werden. Wenn man darauf wieder morphologische Filter anwendet, kann ein Binärbild wie in Abbildung 2.5 erreicht werden. Das Rechteck aus dieser Kontur ist nun geeignet, um die Briefmarke aus dem gerade gedrehten Farbbild auszuschneiden. Das ausgeschnittene Bildstück hat natürlich von Marke zu Marke andere Größenverhältnisse. Die Briefmarken haben schließlich nicht nur unterschiedliche Formate, sondern sind auch nicht alle aus der perfekt gleichen Distanz fotografiert worden. Da das NN eine bestimmte Inputgröße erfordert, muss hier also noch was geändert werden.



Abb. 2.5: Übergang zu morph. Filter2

Wie in der Abbildung 2.6 zu sehen ist, können gerade gedrehte Bilder nach der Binarisierung durch ein ungünstiges Motiv mehrere zureichend Große Konturen enthalten. Das Kriterium, das darüber entschieden hat, ob die Kontur eine potentielle Briefmarke ist, war vorher nur die Fläche. Demnach würden hier drei Bilder für das NN ausgeschnitten werden. Um das zu verhindern, wird hier auf eine Hierarchiebedingung zurück gegriffen. Diese lautet, dass nur Konturen ohne Parentkontur weiter genutzt werden. Im Beispiel rechts fallen deshalb die blau und die rot markierten Konturen weg und nur die richtige, grüne, bleibt übrig.

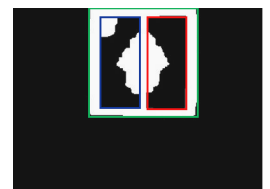


Abb. 2.6: Binarisierung unzureichend

Ein weiteres Problem ist in Abbildung 2.7 zu erkennen. Hier sind durch die Rotation große schwarze Flächen in den Ecken entstanden. Zur Binarisierung wird die Otsu Methode verwendet, der Schwellwert zur Binarisierung wird also automatisch und abhängig vom Bild und dessen Histogramm festgelegt. Normalerweise ist der Hintergrund dunkler als die Briefmarke. Durch die dunklen Ecken wird aber noch eine dritte, noch dunklere Fläche eingeführt. Dadurch verschiebt sich der Schwellwert in eine mittlere Helligkeitsstufe und Teile des Hintergrunds werden mit weiß binarisiert (oberer Teil der Abbildung). Um dieses Problem zu beheben wurden die dunklen Stellen mit dem Mittelwert der Helligkeit des Originalbildes nachgefärbt. Dadurch wird der Schwellwert deutlich schwächer beeinflusst und die Binarisierung kann die Briefmarke wieder korrekt herausfiltern.

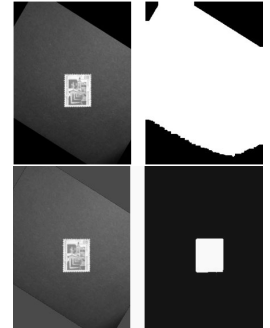


Abb. 2.7: Darstellung des Binärisierungsproblems

Das letzte Problem, das hier betrachtet werden soll, ist in Abbildung 2.8 zu sehen. Auch wenn es mit dem menschlichen Auge schwer zu erkennen ist, ist das Bild sehr ungleichmäßig beleuchtet. Dadurch ist der ursprünglich schwarze Hintergrund in großen Bereichen sogar dunkler als die Briefmarke. Dadurch kann die Briefmarke gar nicht erst gerade gedreht werden. Im Vergleich zum vorher gegangenen Problem entstehen die Komplikationen also nicht erst durch während der Verarbeitung (wie z.B. die schwarzen Ecken) sondern die Ursache dafür ist eine schlechte Qualität im Originalbild. Dieser Fall ist jedoch im ganzen Datensatz nur einmal aufgetreten. Da wir entschieden haben, dass dieser Verlust verkraftbar ist wurden hier keine weiteren Anstrengungen unternommen um das letzte Bild auch noch zu retten.

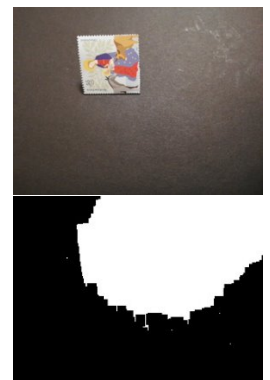


Abb. 2.8: Darstellung des Binärisierungsproblems

Die Abbildung 2.9 stellt eine Zusammenfassung des ganzen Kapitels 2 in Form eines Programmablaufplans dar. In der Abbildung wird zwar nur auf den Teil der gestempelten Bilder eingegangen, der Teil der ungestempelten folgt jedoch dem gleichen Ablauf. Die Unterschiede sind, dass die Liste der Originalbildpfade anders ist und die als gut befundenen Bilder in ein anderes Verzeichnis gespeichert werden.

Der PAP zeigt, dass das Programm in zwei ineinander geschachtelten Schleifen abläuft. In der äußeren Schleife wird über die Liste der Bilder iteriert. Sollte eine Kontur mit ausreichender Größe durch die Binarisierung entstehen wird daraufhin in die zweite Schleife übergegangen. In dieser wird über die gefundenen Konturen nach der zweiten Binarisierung und Dilation iteriert. Zuerst wird dann die Fläche jeder Kontur abgefragt. Sollte die ausreichend sein ist die nächste Bedingung die Hierarchiebedingung, die das Problem aus 2.6 löst. Wenn beide Bedingungen erfüllt wurden wird der Bildausschnitt dem Benutzer angezeigt. Sollte dieser den Ausschnitt für gut befinden kann er mit der "g" Taste der Tastatur die Briefmarke abspeichern. Mit der "b" Taste werden diverse Zwischenergebnisse in Form von Bilddateien in einen separaten Ordner abgelegt. Dies erleichtert die Fehlersuche, ist aber für die Funktion des Programms unwichtig. Sollte das Bild weder gut noch ein interessanter Fehler sein kann mit einer beliebigen Taste mit dem nächsten Bild fortgefahren werden. Zusätzlich zur Anzeige der Bilder gibt eine Konsolenausgabe auch noch Aufschluss über die Anzahl der Schleifendurchläufe (Anzahl bearbeiteter Bilder und Konturen). Durch die beiden Schleifen und das zweimalige Binarisieren ist der Code zwar nicht besonders performant, aber gut nachvollziehbar. Da für jede Marke eine Benutzereingabe erforderlich ist, ist es egal das der Code nicht besonders effizient programmiert ist. Solange es dem Benutzer noch flüssig vorkommt wird praktisch keine Zeit verloren, denn die Wartezeit auf die Eingabe macht die meiste Zeit aus.

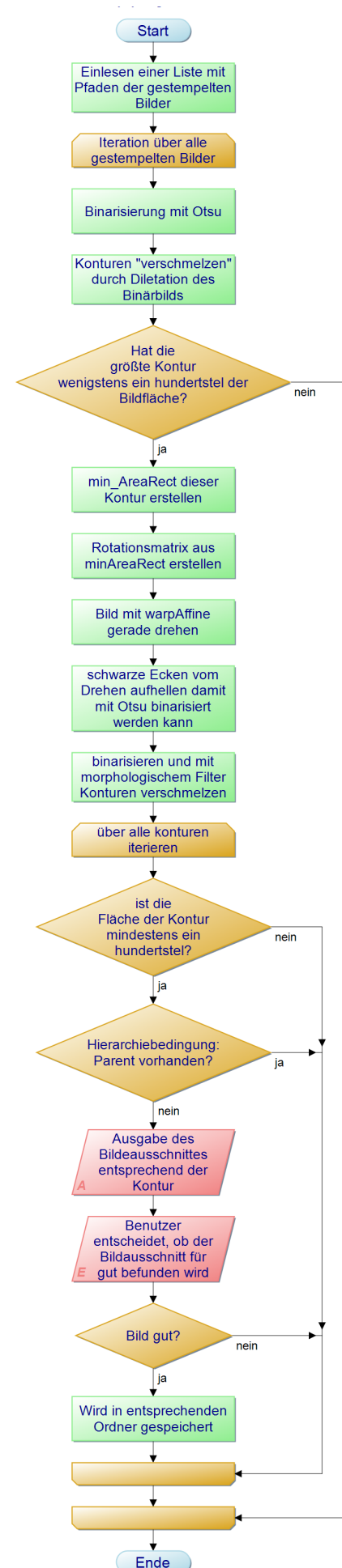


Abb. 2.9: Programmablaufplan

3 Neuronales Netzwerk

Beim verwendeten NN handelt es sich um das MobileNetV2. Es kann für supervised learning Probleme zum Klassifizieren verwendet werden.

3.1 Architektur und Vorgehen

3.2 Implementierung und Auswertung

Die Abbildung 3.1 zeigt, welche Schritte zur Implementierung in Python nötig waren. Zuerst ist natürlich wieder das Einlesen der Daten durchzuführen. Mit Hilfe einer assoziativen Liste (Dictionary) werden die Listen, die die Pfade zu den einzelnen Bildern enthalten, den beiden Klassen 0 und 1 zugewiesen. 0 steht in diesem Fall für gestempelt und 1 für ungestempelt. Als nächstes wird der Datensatz zufällig durchgemischt und dann in Verhältnis eins zu drei aufgeteilt. Der dreiviertelste Teil wird nachher zum Training verwendet. Mit dem kleineren Teil wird die Auswertung des Ergebnisses durchgeführt. Danach werden die Farbwerte aller Kanäle durch 255 geteilt, um den Wertebereich auf 0-1 zu reduzieren. Abschließend werden die Bilder des Datensatzes auf die Größe 224x224 geändert. Das NN kann schließlich nur mit Bildern gleicher Größe arbeiten. Damit ist die Arbeit am Datensatz abgeschlossen.

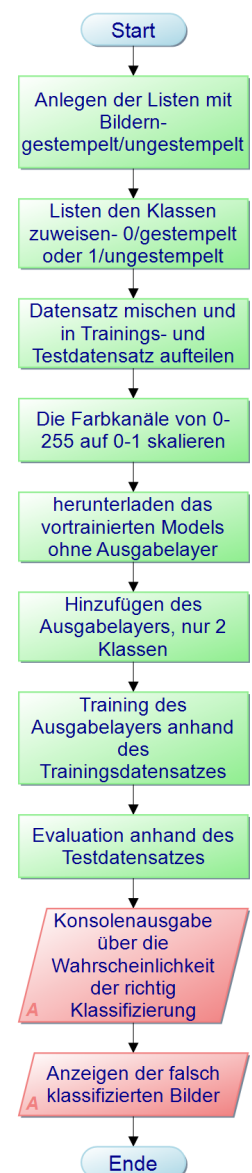


Abb. 3.1: Programmablaufplan NN

4 Fazit

...

Literaturverzeichnis

- [1] Manuel René Theisen: *Wissenschaftliches Arbeiten: Technik – Methodik – Form*;
15. Auflage; Vahlen; München 2011; ISBN 978-3-8006-3830-7
- [2] Hochschule Heilbronn; <http://www.hs-heilbronn.de/>; abgerufen am 14.08.2010