

Praktikumsprotokoll

Cailin Guan

30. Juni 2017

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbst andig und ohne fremde Hilfe verfasst, und dass nur die angegebenen Quellen und Hilfsmittel verwendet wurden. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keineranderen Prüfungsbehörde vorgelegt.

Hannover, 30. Mai 2017

Unterschrift

Inhaltsverzeichnis

1	Einleitung	3
2	Rahmenbedingungen	4
2.1	Art, Inhalt und Umfang dem Thema	4
2.2	Vorstellung des Projekt: StanDAP-Herb	5
3	Aufgaben der Tätigkeit	6
3.1	Aufgabenzuteilung	6
3.2	Aufgabenbeschreibung	6
3.2.1	Microsoft Visual Studio	6
3.2.2	OpenCV	6
3.2.3	GUI: Web-Lokal-Steuerung	7
3.2.4	XML-Analysierung	7
3.2.5	Fuzzy Search	7
3.2.6	Rest-Service	7
3.2.7	Auto-Cut	7
3.2.8	HSV-Farbraume	8
3.2.9	Segmentierung	9
3.2.10	SVM-Klassifikator	10
3.2.11	Automatischer Objekterkennung	12
3.2.12	Json	13
4	Zusammenfassung	14
4.1	Erfahrungsgewinnen	14
4.2	Auswirkungen auf die eigene Berufsvorstellung	14
4.3	Bewertung der eigenen Leistungen	14
4.4	Rückmeldung des Arbeitgebers	14
4.5	Zusammenfassung der Tätigkeit	14

Einleitung

Vom September 2014 bis Mai 2017 arbeite ich in Teilzeit(33%) an der Hochschule Hannover im Forschungsprojekt „Standard Daten Akquisitionsprozess“, Abkürzung „stanDAP-Herb“ mit Visual Studio und C/C++/C#. Dazu gehört die Entwicklung automatisierter Erkennung von Handschriftlichen Herbar - Belegen. Unter der Berücksichtigung bestehender Datenbanksystem und die Entwicklung einer funktionalen webbasierten, anwendungsorientierten Bedienoberfläche. Meinen Aufgaben zählen die digitalen Bildverarbeitung mit OpenCV Bibliothek und Programmierung, z.B GUI Oberfläche zur Web-Lokal-Steuerung, Objekterkennung mit Features, Klassifikation von multiple Objekten, Maschine Learning und Datenanalysieren wie XML und Json.

Ich habe am „3+1“ Austausch- und Kooperationsprogramm der Hochschule Hannover und der Partner Zhejiang University of Science and Technology in Hangzhou teilgenommen, und kam im Rahmen dieses Programms im Herbst 2013 nach Deutschland und belegte im WS13/14 einige englischsprachige Veranstaltung. Meine Praktikum und Bachelorarbeit absolvierte ich in diesem Forschungsprojekt „stanDAPHerb“. Danach habe ich einen Chance von Prof. Dr. Karl-Heinz Steinke bekommen, in diesem Projekt weiter zu arbeiten.

Das Thema „Bildverarbeitung“ interessiert mich seit erstem Besuchen der Vorlesung „Digitale Bildverarbeitung“. Deshalb freute ich mich sehr, als ich die Arbeitsstelle bekam, und ich so die Möglichkeit bekam, im Rahmen dieser Teilzeitbeschäftigung arbeite ich mit zwei andere Studenten im Team zusammen. Dabei habe ich Berufserfahrung gesammelt und selbstständiges, eigenverantwortliches Arbeiten gelernt.

Rahmenbedingungen

2.1 Art, Inhalt und Umfang dem Thema

Das Projekt gehört zu Forschungsprojekt, die Grund, warum es gemacht wird, lautet, bisher werden die Metadaten von Herbarbelegen manuell in Sammlungsdatenbanken eingegeben, aber zunehmend werden Bilderfassungsverfahren eingesetzt, die auch die Nachprüfbarkeit der online verfügbaren Metainformation sichern. Das Standardverfahren soll nun so weit wie möglich die manuelle Metadatenerfassung ersetzen oder ergänzen. Bildverarbeitungssoftware erkennt Objekte auf dem digitalisierten Herbarbeleg und klassifiziert sie. Die Textobjekte werden mit Hilfe von Text - Mining Algorithmen in strukturierte Information überführt. Bei Handschriften wird versucht, den Autor zu erkennen. Im Projekt wird vorhandene Software evaluiert, unter Bildung von standardisierten Interfaces weiterentwickelt und in eine übergreifende offene Softwarearchitektur auf Grundlage etablierter IT-Standards integriert. Abschließend wird das Verfahren hinsichtlich seiner Anforderungen als Standard formuliert und hinsichtlich seiner Anwendung dokumentiert. Das Verfahren adressiert einen großen Bereich naturwissenschaftlicher Sammlungen, allein in Deutschland liegen ca. 22 Millionen Herbarbelege vor, weltweit über 500 Mio [3].

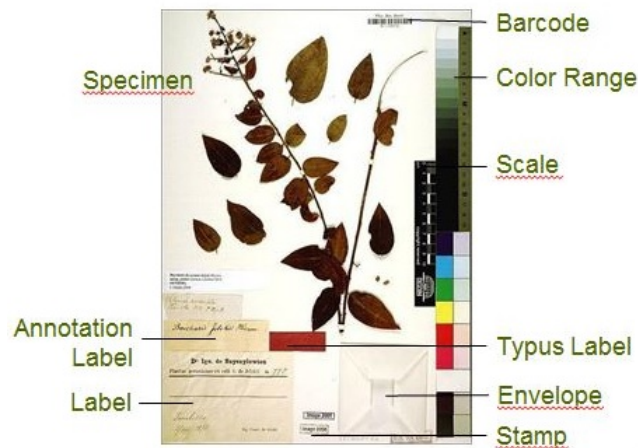


Abbildung 2.1: Beispiel zu digitalem Bild [4], auf Herbarbelegen werden Metadaten wie Artname, Fundort und - datum, Sammler, Katalognummern etc. mit Etiketten, Barcodes usw. flächig sichtbar auf den Bogen gebracht und damit im Foto oder Scan abgebildet.

2.2 Vorstellung des Projekt: StanDAP-Herb

Das Projekt, Standard Daten Akquisitionsprozess - Ein standardisierter und optimierter Prozess zur Erschließung von digitalen Herbarbelegen, ist durch DFG Deutsche Forschungsgemeinschaft, Literaturversorgung und Information / Erschließung und Digitalisierung gefördert, dauert insgesamt 3 Jahre lang, ab Juli 2014 bis Juni 2017. Das Ziel von diesem Projekt ist, einen softwarebasierten Standardprozess für die Extraktion von Metadaten von digitalen Herbarbelegen zu entwickeln und dokumentieren. Die Partnern von diesem Projekt sind *Botanic Garden and Botanical Museum Berlin*, *Fraunhofer IOSB - Karlsruhe* und *Hochschule Hannover*.

Aufgaben der Tätigkeit

3.1 Aufgabenzuteilung

Im Rahmen des Projekt werden die Aufgaben von vier Mitarbeitern geteilt, meine Tätigkeiten im Team sind eine GUI Oberfläche zu entwickeln, um die Bilder und Metadaten zu steuern, XML Daten von OmniPage OCR Software und zu analysieren und Json Daten von gefundenen Objekten zu erzeugen. Einen kleinen Server zu bauen, um Die Lerndateien automatisch auszuschneiden. Einen SVM Klassifizierung - Server für Multiple Objekte zu bauen, und das rote Typus zu erkennen. Für ganze Server Hunderttausend Lernbilder zu sortieren.

3.2 Aufgabenbeschreibung

Für weitere Details von meinen täglichen Aufgaben werden in der folgenden Abschnitten genauer beschrieben.

3.2.1 Microsoft Visual Studio

Visual Studio ist eine für verschiedene Hochsprachen integrierte Entwicklungsumgebung, z.B. Visual Basic .Net, C, C++, C++/CLI, C++/CX, C#, F#, SQL Server, TypeScript und Python sowie HTML, JavaScript und CSS. Visual Studio ermöglicht es dem Programmierer, sowohl native Win32/Win64-Programme als auch Anwendungen für das .NET Framework zu entwickeln[1]. Wir haben die Version 2008 und 2010 in Win32-Programme genutzt.

3.2.2 OpenCV

OpenCV (Open Source Computer Vision Library) ist eine freie Programmierschnittstelle, sie ist mit verschiedenen Algorithmen für die Bildverarbeitung und Maschinelles Sehen zur Akademie und Kommerz unter einer BSD-Lizenz. Sie hat C++, C, Python und Java Oberflächen und unterstützt Windows, Linux, MacOS, Android und IOS. Wir verwenden die Version 1.1, 2.1 und 2.4.10 für unsere Tätigkeiten.

3.2.3 GUI: Web-Lokal-Steuerung

3.2.4 XML-Analysierung

3.2.5 Fuzzy Search

3.2.6 Rest-Service

3.2.7 Auto-Cut

Es gibt viele Objekte auf eine Herb-beleg, z.B. Barcode, Stampe, Typus... , die wir erkennen sollen. Eine Seite kann das ganze Bild in Server (OCR) laufen, um Informationen zu extrahieren, andere Seite, es ist auch wichtig, die einzeln Objekt in Server behandeln, damit kann die Anerkennungsquote zu erhöhen. Deswegen brauchen wir riesige Menge Ausschneiden von den Objekten. Um das Ausschneiden durchzuführen, müssen wir überlegen, wie es machen kann. Natürlich können wir das Ausschneiden manuell durch Paint oder GIMP erledigen, aber es ist offensichtlich, zeitaufwändig. Anstatt die Objekte aus Herb-Beleg selbst per Hand auszuschneiden, ist es nötig, ein kleines Programm zu entwickeln, damit kann man sehr effizient im kurzen Zeitraum viel Objekte erhalten. Bild ist durch das Programm hochgeladen worden, und durch Maus das Objekt abzugrenzen, z.B. bei einem Barcode klicken wir mit der linken Maustaste auf den oben links Punkt und ziehen wir bei gedrückter Maustaste bis zum unten rechts Punkt, danach lassen wir die Maustaste los. Damit bekommen wir die Koordinaten von den beiden Punkten, schneiden die Rectangle, die die Punkte bezüglich ist, aus.



Abbildung 3.1: Objekte Belege

Speichern sollen wir auch berücksichtigen, um die Objekte später direkt zum An-

lernen genutzt zu werden. Brauchen wir Sortierung während Speichern, z.B. bei dem Objekt Barcode, drucken wir die Tastatur „3“, damit das Barcode würde automatisch in der dritten Datei gespeichert werden. Die ausführlichen Informationen von „Objekt-Tastatur-Speicherdatei“ sind in folgende Tabelle gegeben, sehen Sie Abbildung 3.2.

Objekt	Tastatur
Blocktext	„1“
Brief	„2“
Barcode	„3“
Handschrift	„4“
Pflanzen	„5“
Stampe	„6“
Negativ	„7“

Abbildung 3.2: Objekt-Tastatur-Speicherdatei

Der Tabelle zeigt genau bei welchem Objekt sollen wir welche Tastatur drucken, um das Objekt richtig zu speichern.

3.2.8 HSV-Farbraume

Der HSV-Farbraum ist ein Farbraum etlicher Farbmodelle, bei denen man kann die Farbe mit Hilfe des Farbwerts (englisch hue), der Farbsättigung (saturation) und des Hellwerts (oder der Dunkelstufe) (value) definieren (Abbildung 3.3) [6]. Mit diesen Objekten von Typus habe ich diesen Farbraum Modell wegen ihrer speziellen Farbe, normalerweise sind die immer rot, angewendet.

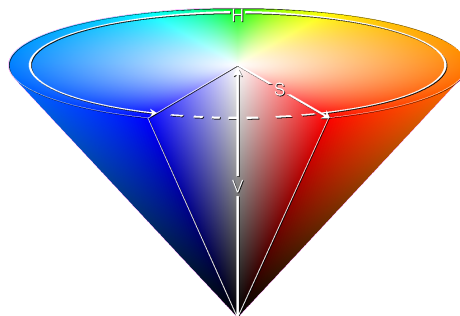


Abbildung 3.3: HSV-Farbraum als Kegel

3.2.9 Segmentierung

Ermittlung von Rectangle ist der Prozess, ein Bild in multiple Segmentation zu trennen. Dieser Prozess dient dafür, Analysierung von Bild und Extraktion von Merkmal zu vereinfachen.[5] In unsere Situation ist das wichtige Merkmal von Segmentation vertikal Kante in dem Bild, der Merkmal kann ausgenutzt werden und die Regionen, die gar keine vertikale Kante haben, zu eliminieren. Bevor Finden der vertikale Kante, sollen wir die mögliche Rausch mit Gaussian blur möglich viele entfernen. Um die vertikale Kante zu finden, haben wir uns die Sobel Filter entschieden. Weil mit diese Funktion können wir selbst Richtungsderivat definieren. Nach Sobel Filter bieten wir eine Threshold Funktion, um das Bild zum Binary zu transformieren. Und danach durch Morphologische Operation können wir den Spalt zwischen Kante eliminieren und die andere nötige Kante verbinden. In unsere Situation liefert es bei der Dimension 41x30 ein gutes Ergebnis. Aber wir lassen das veränderbar, weil für uns die Bilder immer Variante ist, mit verschiedenen Leuchtdichte, verschiedenen Lokalisation der Objekten, manchmal sehr eng miteinander, manchmal gibt es genug Abstand zueinander, und natürlich mit verschiedenen Klarheit von den Objekten, deswegen hat die Dimension von morphologischer Operation sehr groß Beeinflusse, zu groß oder zu klein würden die Objektregionen nicht mehr finden können. Beispielsweise die Abbildung 3.4).

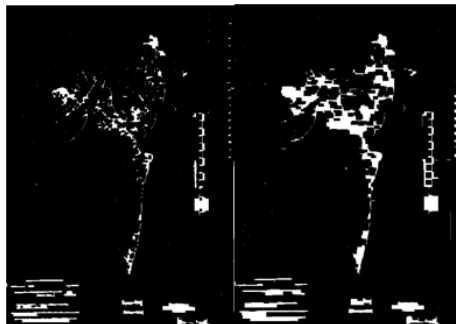


Abbildung 3.4: Verschneide Bilder zu vergleichen

Links Bild mit Dimension 41x30 ist für Stampe besser, aber recht Bild ist mit Dimension 80x45 ist für Mus.Bot.Berol besser. Man kann nicht nur mit einer bestimmten Dimension alles gewonnen. Konturen sollen dann mit dem resultierenden Bild gefunden werden. Damit erhalten wir aber viele Müll, deswegen sollen wir die Segmentation überprüfen. Mit definiertem Seitenverhältnis, In

unsere Situation ist es 3.8 mit 40% Fehlertoleranz, können wir Stampe, Barcode, Mus.Bot.Berol alle ausfinden. Die Richtung der Segmentation ist auch berücksichtigt, d.h. die in beide Richtungen liegende Segmentation können herausgefunden werden. Für Typus ist nichts anderes als Stampe, Barcode, außerdem von Beginnen werden die HSV genutzt, weil der Typus ist alles rot oder ähnlich rot. Danach mit 5x5 Dimensionen morphologische Operation, und Seitenverhältnis 2.5 mit 40 Fehlertoleranz. Die alle gefundenen möglichen Objektregionen würden auch sortiert, und parallel muss gespeichert werden. Z.B. Tastatur „0“ bezüglich auf nicht interessierte Objektregion, Tastatur „1“ bezüglich auf Stampe, Tastatur „2“ bezüglich auf Mus.Bot.Berol, und Tastatur „3“ bezüglich auf Barcode.

3.2.10 SVM-Klassifikator

SVM ist eine Support Vector Machine, die als Klassifikator(vgl. Klassifizierung) und Regressor (vgl. Regressionsanalyse) dient. Eine SVM unterteilt eine Menge von Objekten so in Klassen, dass um die Klassengrenzen herum ein möglichst breiter Bereich frei von Objekten bleibt, sie ist ein sogenannter Large Margin Classifier (engl. „Breiter-Rand-Klassifikator“). SVMs können sowohl zur Regression als auch zur Klassifizierung verwendet werden[2]. SVMs sind keine herkömmliche Maschinen zu verstehen. Die bestehen nicht aus greifbaren Bauteilen, sondern handeln sie sich um ein rein mathematisches Verfahren der Mustererkennung, das in Computerprogrammen umgesetzt wird. Der Namens- teil Machine weist dementsprechend das Herkunftsgebiet der Support Vector Machines, das maschinelle Lernen. In unsere Programm verwenden wir SVMs nur zur Klassifizierung, die darstellte Abbildung 3.5 zeigt, wie sollen wir die Objekte erkennen. In dem SVM Programm definieren wir drei wichtige Funktionsschritte, unser System zu trainieren.

Die alle gespeicherten Objektregionen, sollen wir entscheiden, dass ob die gefundene Objektregion eine Stampe oder ein Barcode ist. Um dies zu erledigen, brauchen wir jetzt SVM Algorithmus. Bevor der Klassifikation muss der Klassifikator mit Label trainiert werden, sog. Offline Training. In unsere Situation, wegen der vielen varianten Bilder brauchen wir eine ausreichende Menge Daten zu trainieren. D.h. je größer unsere Datenbank ist, umso kriegen wir besser Ergebnis. Deswegen brauchen wir zu mindesten ein paar tausend Herb-Beleg für Preprozessor und Segmentierung. SVM ist inhärent ein binärer Klassifikator, in unsere Situation, für Stampe, Barcode, Mus. Bot. Berol. bedürfen wir 4 Klasse, wir nutzen lineare Klassifikation. Es gibt viele Möglichkeiten zu trainieren,

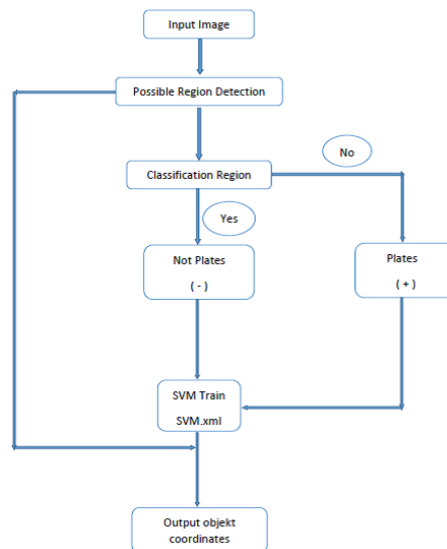


Abbildung 3.5: SVM Programm : Dieses ganze Prozesses beziehen sich auf das Programm sind Sobel Filter, Threshold Operation, Morphologic Operation, Find Contours, SVM Klassifikation, SVM Training.

z.B. one-versus-rest(OVR SVMs), one-versus-one (OVO SVMs). Wir haben uns OVR SVMs entschieden, weil für uns ist es sehr schnell und unser System ist nicht kompliziert. Unser Prozess beruht sich auf die Bildpixels, sehen Sie in folgendem Bild(Abbildung 3.6).



Abbildung 3.6: Positive Bilder und negative Bilder

Klasse Stampen gegen die andere, Stampe werden mit positivem Label identifiziert, die andere wird mit negativem Label identifiziert. Barcode, Mus.Bot.Berol werden auch so trainiert. Dann die ganzen trainierten Daten werden in eine XML Datei gespeichert.

Nach dem Bekommen der trainierten Daten müssen wir jetzt die SVM Parametern definieren, um das SVM Algorithmus zu verwenden. In OpenCv ist es

einfach, dass wir können einfach CvSVMParam direkt anrufen und die Parameter eingeben. Hier könnten wir auch die Parameter automatisch mal trainieren, um die besten Parametern zu erzielen. Mit eingegebenen Parameters kann die Klassifikator trainiert werden. Danach ist es bereit zu Prognose von möglichen Objektregionen mit Predict Funktion. Falls wir positive Resultat erhalten, ist die Objektregion interessiert uns, negative Resultat, dann interessiert uns nicht. Somit können wir klassieren alle gefundenen möglichen Objektregionen mit der Rückmeldung von Prognose. Nur die mit dem positiven Prognose Objektregionen und auch deren Koordinaten werden gespeichert(Abbildung 3.6).



Abbildung 3.7: Resultat von SVM Programm: Die Abbildung zeigt, wie das Resultat von SVM Programm im originalen Bild aussieht, die mit grünen Rechteangle eingehüllten Objekte sind die gefundenen Objektregionen. Links ist für Typus, rechts ist für Stampe, Barcode, Mus.Bot.Berol.. Die gespeicherten Koordinaten werden weiter in andere Programm geliefert.

3.2.11 Automatischer Objekterkennung

Um die Koordinaten auf Herb-Beleg zu erhalten, automatische Erkennung von den Objekten ist in der erste Linien. Dieses Programm ist genau dafür gesorgt. Es beinhaltet Ermittlung von Objekt(Preprozessor), und Klassifikation von Objekte. Das Ziel von Ermittlung dem Objekt ist es, die möglichen Objektregionen in ganzen Bildframe zu ermitteln. Falls eine Rectangle wird ermittelt, die Rectangle Segment wird in den zweite Schritt überträgt, mit SVM(Support Vector Machine) Algorithmus zu entscheiden, ob die Rectangle richtig ist. Durch das Programm können die Koordinaten von Objekt automatisch geliefert werden. Die Koordinaten können später in andere Programm eingeben, um die Region automatisch zu ausschneiden.

3.2.12 Json

Zusammenfassung

- 4.1 Erfahrungsgewinnen**
- 4.2 Auswirkungen auf die eigene Berufsvorstellung**
- 4.3 Bewertung der eigenen Leistungen**
- 4.4 Rückmeldung des Arbeitgebers**
- 4.5 Zusammenfassung der Tätigkeit**

Literatur

- [1] URL: https://de.wikipedia.org/wiki/Visual_Studio.
- [2] URL: https://en.wikipedia.org/wiki/Support_vector_machine.
- [3] *Hier ist ein Eingang für diesen Projekt*. URL: http://wiki.bgbm.org/standapherb/index.php/Main_Page.
- [4] Agnes Kirchhoff. *Herbarbeleg_Objekte*. [Online; accessed 18 August 2015]. 2015. URL: http://wiki.bgbm.org/standapherb/index.php/File:Herbarbeleg_Objekte.JPG.
- [5] Birmingham [u.a.] Packt Publ. *Mastering OpenCV with practical computer vision projects*. 2012.
- [6] Wiki. *HSV-Farbraum*. URL: https://de.wikipedia.org/wiki/HSV-Farbraum#/media/File:HSV_cone.png.