

Eberhard Karls Universität Tübingen

Mathematisch-Naturwissenschaftliche Fakultät

Wilhelm-Schickard-Institut für Informatik

Dokumentation Praktikum Bildverarbeitung

Automatisches ausschneiden von Bildern aus Fotoalben

Mariella Dreißig, Tom Eckardt, Stefan Bergmann

11. Dezember 2018

Betreuer

Andreas Karge, Prof. Andreas Schilling

Mariella Dreißig, Tom Eckardt, Stefan Bergmann:

Automatisches ausschneiden von Bildern aus Fotoalben

Dokumentation Praktikum Bildverarbeitung

Eberhard Karls Universität Tübingen

Zusammenfassung

Inhaltsverzeichnis

1	Einleitung	5
2	Hintergrundentfernung	6
3	Randentfernung	7
4	Gesichtserkennung	8
4.1	Cascade Classifier	9
5	Auswertung	11
5.1	Hintergrundentfernung	11
5.2	Randentfernung	11
5.3	Gesichtserkennung	11
6	Bedienungsanleitung	12
	Literaturverzeichnis	13

1 Einleitung

2 Hintergrundentfernung

3 Randentfernung

4 Gesichtserkennung

Die Aufgabenstellung beinhaltet, in den extrahierten Fotos Gesichter zu erkennen. Hier muss die Gesichtserkennung (in der Fachliteratur "Face Detection") von der Wiedererkennung von bereits identifizierten Gesichtern (in der Fachliteratur "Face Recognition") unterschieden werden. In der vorliegenden Arbeit geht es ausschließlich darum, die Position von Gesichtern in den eingescannten Fotos zu erkennen und nicht, bereits erkannte Gesichter wiederzuerkennen oder zuzuordnen ¹.

In der klassischen Bildverarbeitung wurden lange einfache 2D-Verfahren verwendet, um Gesichter in Bildern zu erkennen ([1], [6]). Beim Template Matching werden beispielsweise Vorlagen (von Teilen) von Gesichtern erzeugt um diese dann im Anwendungsfall mit den Zielbildern zu vergleichen. Dieses Verfahren ist nicht sehr robust, besonders bei sich verändernden Lichtverhältnissen, aber gleichzeitig extrem rechenaufwändig.

Eine einfachere Methode der Gesichtserkennung in der klassischen Bildverarbeitung ist die Nutzung von geometrischen Merkmalen in den Gesichtern. Hierbei werden beispielsweise Grauwerte in einem bestimmten Bereich des Bildes aufsummiert oder Gradientenoperatoren zur horizontalen und vertikalen Kantenerkennung angewandt. So kann die Ausrichtung, die relative Lage und ähnliche Merkmale der Nase, der Augen und des Mundes identifiziert und zusammen als Gesicht klassifiziert werden.

Doch auch diese Methode ist rechen- und zeitintensiv. Deswegen wird häufig eher auf komplexere Methoden zur Gesichtserkennung gesetzt, wie beispielsweise das Elastic Bunch Graph Matching, das eine robuste Waveletanalyse nutzt ([7]) oder die auf der Hauptkomponentenanalyse basierende Eigenfaces-Methode ([4]). Weiterentwicklungen dieser beiden Methoden werden heute noch eingesetzt, obwohl die Ansätze schon vergleichsweise alt sind.

Im Folgenden wird ein populärer Ansatz vorgestellt, der in den vorliegenden Arbeit verwendet wurde.

¹ Aufgrund der geringen Datenmenge und der schlechten Qualität der alten Fotos ist es nicht möglich, Gesichter wiederzuerkennen.

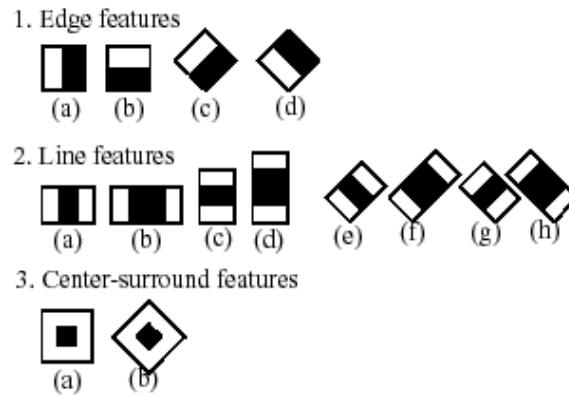


Abbildung 4.1: Alle vom Cascade Classifier Algorithmus genutzten Haar-ähnlichen Features zur Objekterkennung. Zur Gesichtserkennung werden nur fünf Features genutzt: 1(a), 1(b), 2(a), 2(c) und eine Kombination aus zwei 1(a) Features, die zusammen ein schachbrettartiges Muster ergeben.

4.1 Cascade Classifier

Die OpenCV Programmbibliothek beinhaltet vorimplementierte Funktionen zur Gesichtserkennung. Sie basiert auf einem 2001 entwickelten Ansatz zur Objekterkennung in Bildern ([5], [2]). Es ist ein trainierter Classifier, der einfache geometrische Merkmale im Bild erkennt und auf deren Basis die gewünschten Objekte erkennen kann ([3]).

Die Methode nutzt sogenannte Haar-ähnliche Features, mit denen grundlegende Strukturen (Helligkeitsunterschiede) im Bild erkannt werden. Im Bild 4.1 werden alle Features aufgelistet, die der Classifier nutzen kann. Für die Gesichtserkennung werden in der Praxis aber nicht so viele genutzt, sondern nur einige wenige. Mit diesen können allgemeine Eigenschaften von menschlichen Gesichtern repräsentiert werden. So ist zum Beispiel die Nase heller als die Bereiche um die Augen, das entspricht dem Feature 2(a) in Bild 4.1.

Nachdem die zu nutzenden Features ausgewählt wurden, wird ein Integralbild erzeugt, das in der Lage ist, die Features in konstanter Zeit auszuwerten. Dadurch wird eine relativ geringe Laufzeit erzielt.

In der Trainingsphase wird der Classifier mit Positiv- und Negativbeispielen von dem zu erkennenden Objekt trainiert, in diesem Fall mit Gesichtern ². Dieser Vorgang wird mit verschiedenen Arten des AdaBoostings beschleunigt und verbessert. Dadurch werden die besten Features für die Klassifizierung identifiziert.

²In der OpenCV Bibliothek ist sowohl ein Trainer als auch ein Detector enthalten. Der Detector ist vortrainiert kann direkt zur Gesichtserkennung verwendet werden. Wenn man genug Daten zur Verfügung hat, kann man aber auch einen selbst trainieren.

4 Gesichtserkennung

Danach wird der Classifier in mehreren Stufen auf die Bilder angewandt ("cascade"), in denen das Objekt identifiziert werden soll. Jeder Classifier funktioniert dabei wie ein Filter, der die negativen Regionen aus dem Bild herausfiltert. Die verbleibenden Bereiche werden Stück für Stück durch immer komplexere Classifier gegeben, bis sie entweder aussortiert werden oder das Objekt (ein Gesicht) identifiziert wurde. Somit hat der Cascade Classifier die Struktur eines Entscheidungsbaumes und Rechenzeit wird gespart. Die Methode ist so robust, dass das Objekt in verschiedenen Größen in dem Bild erkennen kann, unabhängig von der Größe der Bilder, mit der der Classifier trainiert wurde.

Wenn die OpenCV Methode `detectMultiScale` ein Gesicht im Input-Bild erkennt, gibt es die Position in x- und y-Koordinaten und dessen Abmessungen (Höhe, Breite) zurück. In dem vorliegenden Programm werden zwei verschiedene Classifier auf die Bilder angewandt: einmal frontale Gesichter und einmal Gesichter im Profil. Die Rückgabewerte der Funktionen werden dafür genutzt, die Gesichter auszuschneiden und Dopplungen zu erkennen.

5 Auswertung

5.1 Hintergrundentfernung

5.2 Randentfernung

5.3 Gesichtserkennung

6 Bedienungsanleitung

Literaturverzeichnis

- [1] Kees, Benjamin: *Gesichtserkennung*, 2012. <https://www.algoropticon.de/Dateien/Seminararbeit{ }Gesichtserkennung.pdf>.
- [2] Lienhart, Rainer, Alexander Kuranov, Vadim Pisarevsky und M R L Technical Report: *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection*. Technischer Bericht, 2002. <http://www.multimedia-computing.de/mediawiki//images/5/52/MRL-TR-May02-revised-Dec02.pdf>.
- [3] OpenCV: *Haar Feature-based Cascade Classifier for Object Detection*. <https://docs.opencv.org/2.4/modules/objdetect/doc/cascade{ }classification.html{#}grouprectangles>, besucht: 2018-12-09.
- [4] Turk, Matthew und Alex Pentland: *Eigenfaces for Recognition*. Journal of Cognitive Neuroscience, 3(1):71–86, 1991. <https://doi.org/10.1162/jocn.1991.3.1.71>.
- [5] Viola, Paul: *Rapid Object Detection using a Boosted Cascade of Simple Features*. In: *Conference on Computer Vision and Pattern Recognition*, 2001. <http://wearables.cc.gatech.edu/paper{ }of{ }week/viola01rapid.pdf>.
- [6] Wächter, Carsten und Stefan Römer: *Gesichtserkennung I - Biometrische Systeme*, 2001. <http://www.informatik.uni-ulm.de/nl/Lehre/WS01/HS-Biometrische-Systeme/ausarbeitungen/Gesichtserkennung{ }1{ }block.pdf>.
- [7] Wiskott, Laurenz, Norbert Kr und Christoph Von Der Malsburg: *Face Recognition by Elastic Bunch Graph Matching*. Intelligent Biometric Techniques in Fingerprint and Face Detection, (11):355–396, 1999. <http://www.face-rec.org/algorithms/ebgm/wisfelkrue99-facerecognition-jainbook.pdf>.