Sebastian Keppler
AI-Master

Dokumentation Computer Vision

Objekt-Klassifikation in Qt mit der Durchschnittsfarbe und OpenCV-SIFT

1. Einleitung

Das Programm bietet die Möglichkeit, Objekte nach deren Durchschnittsfarbe oder mit Hilfe von SIFT zu klassifizieren. Geschrieben wurde das Programm mit Hilfe des C++ Frameworks Qt, sowie der OpenCV-Bibilothek. Die Anwendung wird zunächst mit einen Bilder-Set anhand einer der oben genannten Methoden trainiert. Anschließend kann ein oder mehrere unbekannte Test-Bilder in die Anwendung geladen werden, um diese zu klassifizieren. Die Methode der Klassifizierung entspricht der OneVsAll-Klassifizierung in der Trainungs- und Test-Daten verschieden sind. Die Ergebnisse können dann in Form einer Confusion-Matrix, der Overall CorrectionRate und einer Liste der eingegeben Test-Bilder und der vorhergesagten Gruppe evaluiert werden.

2. Vorgehensweise

Wie in der Einleitung bereits beschrieben, bietet die Anwendung zwei Möglichkeiten Bilder zu klassifizieren. Ein Weg geht über die Durchschnittsfarbe des Bildes, der andere mit Hilfe des SIFT-Algoritmus. Nachfolgend sollen beide Vorgehensweisen näher erläutert werden.

2.1 Durchschnittsfarbe

Zunächst wird anhand eines Trainingsbilder-Sets die Gruppen gebildet. Aus jedem Bild wird aus allen Pixel der Durchschnitt für jeden RGB-Kanal ermittelt. Für jedes Bild wird zunächst geprüft ob es das erste seiner Gruppe ist, falls ja wird eine neue Gruppe angelegt. Jede Gruppe speichert für sich die durchschnittlichen RGB-Werte aller zur Gruppe zugehörigen Bilder.

Vom jedem Bild wird also ein durchschnittlicher Rot-, Grün- und Blau-Wert ermittelt. Diese 3 Werte werden anschließend zu den Gruppen RGB-Werten hinzuaddiert und zum Schluss durch die Anzahl der Bilder in der Trainingsgruppe geteilt.

Wenn jetzt ein unbekanntes Bild zum klassifizieren in die Anwendung hineingegeben wird, werden hier zunächst wieder die 3 durchschnittlichen RGB-Werte des Bildes ermittelt. Diese Werte werden miteinander wieder addiert und zum Schluss durch 3 geteilt, um einen Gesamtfarbdurchschnitt des Bildes zu haben.

Anschließend wird für jede Gruppe auch ein Gesamtfarbdurchschnitt berechnet und ein Betrag zwischen der Durchschnittsfarbe des Bildes und der Gruppe gebildet. Dieser Betrag entspricht dem Euklidischen Abstand von dem Testbild zur Gruppe.

$$distance = |TestMeanColor - GroupMeanColor|$$

Dieser Betrag wird für jede Gruppe berechnet. Die Gruppe mit der die kleinste Distanz berechnet wurde, ist dann die Gruppe in der das Testbild klassifiziert wird.

2.2 SIFT

Für die Realisierung der Klassifikation mithilfe des SIFT-Algorithmus wurde die freie Bibilothek OpenCV verwendet. Nebendem SIFT wurden auch noch die Funktionalität eines Bag of Features(BoF) und einer Support Vector Machine (SVM) aus OpenCV verwendet, die nachfolgende noch erklärt werden.

Zunächst werden mithilfe von SIFT für alle Bilder eines Training-Sets die Features, also markante Bildpunkte, bestimmt. Mit den Features und des Bildes wird ein sogenannter Descriptor erstellt, dieser speichert die Informationen zu den Features eines Bildes. Für jedes Bild wird ein solcher Descriptor angelegt und anschließend in einem Array gespeichert.

Es wird dann ein BoF(In OpenCV auch BoW genannt) erstellt und diesem das Array mit den Descriptoren übergeben. Das BoF versucht anschließend alle Features mithilfe von K-Means zu clustern. Die Anwendung erlaubt es einzustellen wieviele Features pro Cluster gespeichert werden sollen. Als Ergebnis wird ein Vokabular von Features aller Bilder erstellt. Jeder Cluster aus dem Vokabular wurde anschließend zu einer Gruppe zugeteilt.

Zum Schluss der Trainingsphase wird eine SVM für jede Gruppe erstellt und die SVM mit den Clustern aus dem Vokabular „trainiert“. Da jede Gruppe ein SVM besitzt wird die ensprechende SVM mit positiven Beispielen aus dem Cluster und negativen aus anderen Clustern trainiert die entsprechend markiert wurden.

Wenn nun ein unbekanntes Bild zum klassifizieren in die Anwendung eingegeben wird, ermittelt SIFT hier zunächst auch wieder die Features des Bildes. Für diese Features wird geschaut ob sich Übereinstimmungen im Vokabular finden lassen und ein entsprechendes Feature-Histogramm wird erstellt.

Die SVM jeder Gruppe ermittelt nun die Distanz des Feature-Histogramms zur jeweiligen Gruppe. Die niedrigste Distanz entspricht dann der bestimmten Gruppe.

3. Tests

Für die Tests wurden 2 Bilder-Sets zur Verfügung gestellt.

Das erste Set beinhaltet insgesamt 250 Bilder mit 50 Gruppen, jede Gruppe besitzt 5 Bilder und jeweilgen Bilder der Gruppe sind sich alle sehr ähnlich.

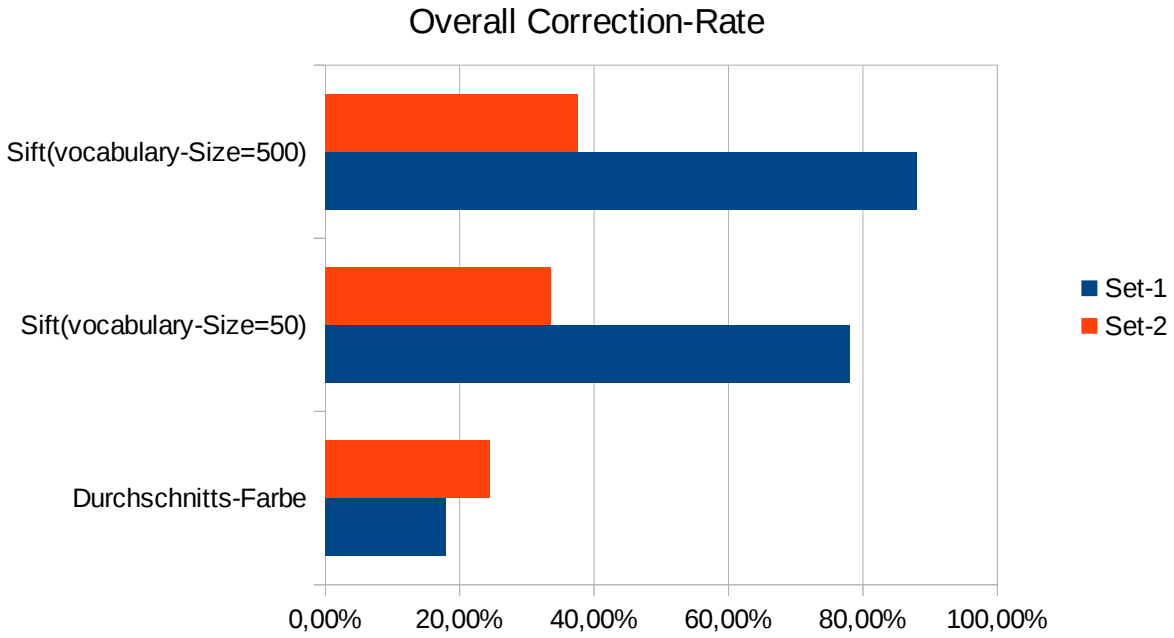
Das zweite Set besteht aus 720 Bildern mit 26 Gruppen. Hier variert die Größe der Gruppen von 15 bis 40 Bilder pro Gruppe. Die Bilder innerhalb einer Gruppe unterscheiden sich deutlich stärker als es im ersten Set der Fall war.

Um die OneVsAll-Methode anzuwenden wurden aus jedem Set einige Bilder herausgenommen, damit diese nicht beim Training verwendet werden. Diese Bilder wurden anschließend als „unbekannte“ Bilder für die Tests verwendet. Im Falle des ersten Sets wurde aus jeder Gruppe 1 Bild zum Testen genommen. Im zweiten Set bestand das Test-Set aus 3-6 Bilder je Gruppe.

Somit bestand das erste Bilder-Set aus insgesamt 200 Trainingsbildern und 50 Testbildern. Das zweite Set bestand aus 577 Trainingsbildern und 143 Testbildern.

Der Gruppen-Name ist im Dateinamen der Bilder enthalten gewesen.

4. Auswertung



4.1 Durchschnittsfarbe

Betrachtet man die Auswertung fällt für die Durchschnittsfarbe auf, dass das zweite Set, welches als schwiergier eingestuft wurde, ein höheres Ergebnis erzählt als das einfachere erste Set. Dies kann daran liegen, dass in allen Bildern alle Pixel miteinbezogen werden also auch der Hintergrund. Die Hintergrund-Pixel aus dem ersten Set sind alle relativ homogen von der Farbe her über alle Gruppen hinweg. Somit ist die Gesamtdurchschnittsfarbe jedes Bildes doch relativ ähnlich im ersten Set. Im zweiten Set besitzen die Bilder einer Gruppe oft einen ähnlichen Hintergrund der sich von Gruppe zu Gruppe aber widerrum unterscheidet. Als Beispiel hier kann man die Gruppe „fireworks“ nehmen wo relativ viele schwarze Hintergrundpixel vorhanden sind, während in anderen Gruppen der Hintergrund heller ist.

4.2 SIFT

Die Klassifikation mit SIFT wurde 2mal getestet, einmal mit einer Vokabulargröße von 50 und einmal mit 500. Ein größeres Vokabular lässt die Trainingszeit deutlich ansteigen, bis hin zu mehreren Minuten oder auch Stunden. 50 ist hier von der Anwendung der kleinste einzustellende Wert.
Mit einer Vokabelgröße von 50 ist, im Gegensatz zur Durchschnittsfarbe, ein deutlicher Anstieg der Treffergenauigkeit im ersten Set zu beobachten. Hier werden Werte von fast 80% erreicht. Im zweiten Set werden immerhin noch 34% erreicht. Bei einer Vokabelgröße von 500 lässt sich nochmal ein höhere Genauigkeit in beiden Sets feststellen. Hier erreicht das erste Set nun eine Genauigkeit von 88% und das zweite Set steigt auf 38%.

Bei der Klassifikation mit SIFT konnte man feststellen, dass einzelne Gruppen besser erkannt wurden als andere. So wurden die Gruppen „doors“, “fireworks“ und „foods“ mit einer Genaugigkeit >90% erkannt, während Gruppen wie „churces“, „constructions“ oder „roses“ garkeinen Treffer erzielt hatten. Mögliche Ursachen könnten sein, dass die Bilder innerhalb einer Gruppe sich doch noch stärker unterscheiden als in anderen Gruppen und das vielleicht auch nicht genügend Trainingsbilder vorhanden waren (die Bilder je Gruppe varierten von 15 bis 30). Möglicherweise ließe sich das Ergebnis auch noch verbessern wenn eine größere Vokabulargröße verwendet wird. Desweiteren ist zu sagen das die Klassifikation mit SIFT jedes mal Varianzen haben kann, da SIFT sich die interessanten Features innerhalb des Bildes eher zufällig aussucht. Deswegen kann die CorrectionRate mit jeder neuen Klassfikation leicht anders sein.

5. Auswertung

Der Versuch hat gezeigt das komplexere Verfahren eine bessere Genauigkeit liefern bei der Klassifikation von Bildern. Die Ergebnisse ließen sich bestimmt noch weiter verbessern, wenn die Verfahren um zusätzliche Arbeitsschritte ergänzt werden würden. Dies könnte z. B. durch eine Objekt-Segmentierung geschehen um die Hintergrundpixel herauszufiltern. Das Ergebnis ist auch von der Wahl der Trainingsdaten abhängig. Ein größeres Set aus Trainingsbildern könnte dementsprechend eine höhere Genauigkeit ergeben. Auch in der Parametrisierung ließen sich noch Justierungen vornehmen. Die Anwendung bietet hier jetzt nur 1 Parameter beim der Klassifikation mit SIFT an, jedoch hat der SIFT von OpenCV noch wesentlich mehr Parameter die sich noch anpassen ließen, um ein besseres Ergebnis zu erzielen.

6. Quellen

http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html [13.02.2016]

http://docs.opencv.org/2.4/modules/features2d/doc/object_categorization.html [13.02.2016]

http://www.morethantechnical.com/2011/08/25/a-simple-object-classifier-with-bag-of-words-using-opencv-2-3-w-code/ [13.02.2016]

https://github.com/royshil/FoodcamClassifier [13.02.2016]

http://www.codeproject.com/Articles/619039/Bag-of-Features-Descriptor-on-SIFT-Features-with-O [13.02.2016]