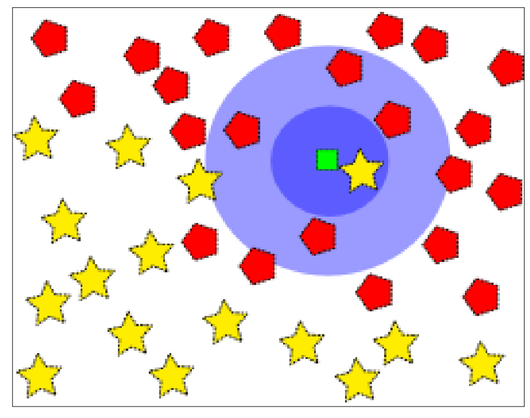
FACHHOCHSCHULE BRANDENBURG
FACHBEREICH INFORMATIK UND MEDIEN

Maschinelles Lernen
Überwachtes Lernen von Objektbeschreibungen aus Beispielbildern

Leischker, Müller, Quehl, Söhnel



Bisher war es den Lebewesen vorbehalten zu „denken“. Manchen mehr, anderen weniger. Aber es gibt schon seit einiger Zeit Algorithmen, die dieses Privileg versuchen zu imitieren und erahnen lassen, wie uns Roboter stets ähnlicher werden.

Aufgabenstellung

Ziel war die Entwicklung eines Programms, das mit Hilfe eines aufgenommenen Kamerabildes bestimmte Gegenstände, in diesem Fall sind es Obst und Gemüse, klassifizieren kann.

Zunächst waren für das Programm alle Gegenstände unbekannt. Durch Training mit Beispielen, sollten nach und nach die Objekte zuverlässig erkannt werden.

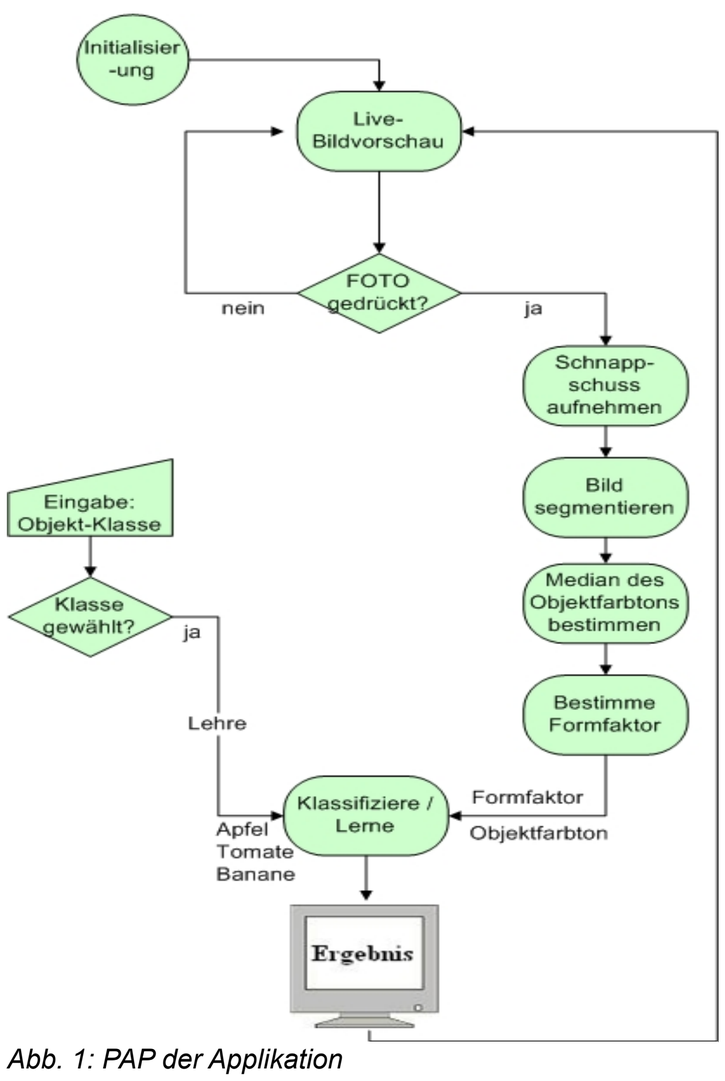


Abb. 1: PAP der Applikation

Hauptprogramm & GUI

Zunächst kann von dem Benutzer über die GUI ein Schnappschuss des im linken Fenster angezeigten Live-Bildes gemacht werden. Das Bild wird nun vom Hauptprogramm an die Bildverarbeitungskomponente geschickt, die daraus die für den Klassifikator benötigten Merkmale extrahiert. Jener entscheidet dann, um welche Objekt-Klasse es sich vermutlich handelt. Das rechte Fenster der GUI zeigt das Ausgabebild der Bildverarbeitung. Die Textbox zeigt die extrahierten Merkmale und den Klassifikationsvorschlag. Letzterer kann über die rechts-stehenden Buttons entweder bestätigt, korrigiert oder verworfen werden. Bestätigte bzw. korrigierte Ergebnisse werden danach zum Training wieder an den Klassifikator geleitet.

Bildverarbeitung

Nachdem die Bildverarbeitungskomponente das aufgenommene Foto von dem Hauptprogramm erhalten hat, werden verschiedene Objektmerkmale bestimmt. Der Formfaktor des Objekts und der Median seiner Farbtöne (HSI-Raum) schienen hier eine sinnvolle Wahl zu sein.

Hierbei wurde die kostenlose Bildverarbeitungs-Bibltiothek OpenCV eingesetzt. Außerdem wurde die Entscheidung getroffen, dass Schatten nicht als Teil des Objektes betrachtet werden sollen, unter anderem deshalb wurde als Randbedingung ein dunkler, homogener Hintergrund definiert. Für die Objektextraktion und Farbtonbestimmung ist eine Segmentierung des Schnappschusses nötig und damit eine Trennung von Objekt und Hintergrund; hierbei sei erwähnt, dass stets von genau einem aufgenommenen Objekt ausgegangen wird. Nachdem das Originalbild mit Hilfe eines Gauß-Filters geglättet wurde, erfolgt die Segmentierung mit dem Floodfill-Algrorithmus und der maximalen Farbdifferenz von 4 als Zuordnungskriterium. Zur Eliminierung von Ausreißern wird das segmentierte Binärbild anschließend mehrmals mit einem Medianfilter bearbeitet.

Bei der Bestimmung des Farbtonmedians, wird der Schnappschuss in den HSI-Raum konvertiert. Das segmentierte Binärbild wird nun zur Abtastung des Hue-Kanals benutzt, um sicher zu stellen, dass nur die Farbtöne des Objekts in die Median-Bestimmung einfließen.

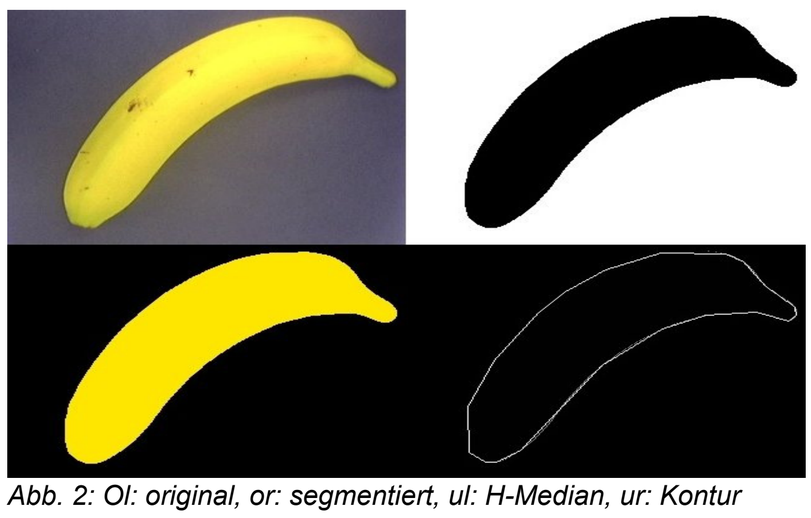


Abb. 2: Ol: original, or: segmentiert, ul: H-Median, ur: Kontur

Die Klasse FindFormFactor bekommt das segmentierte Binärbild übergeben. Konkret wird vom ersten Teil der Bildverarbeitungskomponente die Methode

```
findConture(IplImage *src ,double *f);
```

aufgerufen, wobei src der Zeiger auf das segmentierte Binärbild und f ein Zeiger auf die Variable für den Formfaktor ist. Das Programm sucht mittels der OpenCV-Methode

```
cvFindContours(binaryImage, storage, &
contours, sizeof(CvContour));
```

nach einer Kontur - eben einem Objekt. Wenn der Algorithmus mehr als eine Kontur findet, befinden sich wahrscheinlich mehrere Objekte im Bild. Dies führt zu einem weiteren Bearbeitungsprozess, welcher das Bild invertiert oder weitere Filter auf das Bild anwendet (Canny, Gauß). Sollte selbst dann ein Formfaktor außerhalb eines realistischen Bereiches (Formfaktor f in $1 < f < 100$) auftreten, wird -1 zurückgegeben.

Umsetzung der Intelligenz

Als Klassfikator dient der kNearestNeighbour-Algorithmus, da er relativ einfach überwachtes Lernen ermöglicht. Er beginnt nun im Training die Werte als Punkte in einem Koordinatensystem fest zu legen. Da als charakteristische Merkmale in diesem Fall der Formfaktor und der mittlere Farbton gewählt wurden, ist es zweidimensional. Eine Umsetzung auf n Merkmale und entsprechend n Dimensionen ist aber auch vorstellbar und wie im Folgenden erläutert, auch empfehlenswert.

Nachdem die ersten Punkte gefunden wurden, sucht der Klassifikator für jedes neue Wertepaar die Umgebung nach Nachbarn ab. Der Klassifikator vergleicht die Entfernungen und weist dem Objekt die Klasse, bzw. den Namen, des Objektes mit dem geringsten Abstand zu.

Der Klassifikator gibt nun sein Ergebnis an die Oberfläche zurück. Zunächst einmal wird dem Benutzer ein Vorschlag unterbreitet, was für ein Objekt es sein könnte. Er kann ihn bestätigen oder verwerfen. Auch ein neues Foto verwirft die berechneten Werte, ohne sie neu in das Koordinatensystem einzutragen. Im Laufe einiger Trainingsobjekte sollte das Programm zuverlässig Gegenstände erkennen können. Wenn Objekte in Form und Farbe sehr ähnlich sind, stößt diese Implementierung an ihre Grenzen. Das Problem kann gelöst werden indem man, wie schon erwähnt, weitere Werte zufügen und ein mehrdimensionales Koordinatensystem nutzt.

Zusammenfassung & Ausblick

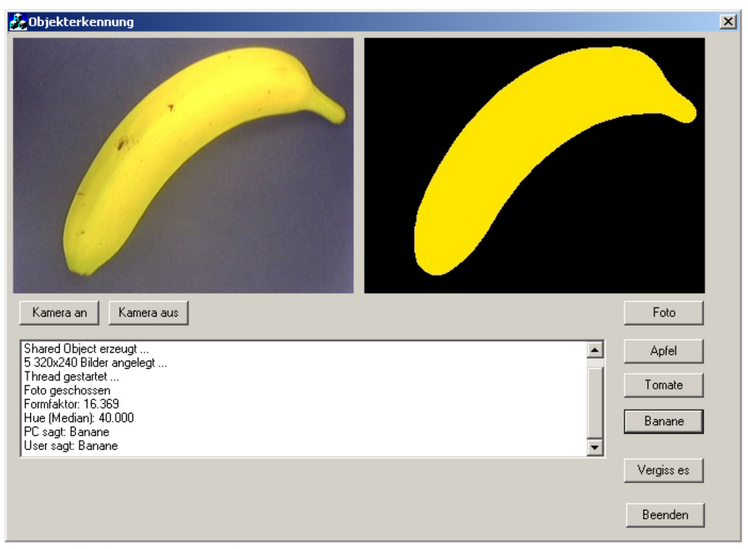


Abb. 3: Screenshot der Anwendung

Im Rahmen des Projekts ist eine Applikation entstanden, die die geforderten Eigenschaften prototypisch umsetzt. Die Klassifikation unbekannter, in min. 1 von 2 Merkmalen (Form, Farbe) unterscheidbarer Objekte kann erlernt werden.

Für die Zukunft ist eine Erweiterung um andere Klassifikatoren, die Einbeziehung weiterer Merkmale zur Erhöhung der Unterscheidbarkeit und eine Verbesserung der Segmentierung hinsichtlich der Randbedingungen vorstellbar. Wichtiger ist jedoch die Verbesserung der generellen Stabilität.

Betreuer: Dipl.-Inform. I. Boersch

Abgabe: 22.03.2007