

Übungablatt 4 zur Vorlesung ‘Einführung in das maschinelle Lernen’

Aufgabe 4: Bayesian Image Classification

Kai Uhrig
Roger Braun
Sebastian Rheinacker

15. Juni 2010

1 a

Die Funktion **generateTData.m** erstellt die Daten des Features I für alle gewöhnlichen Apfel und Bananenbilder. Dazu wird die Funktion **featureI.m** verwendet, die wiederum Unterfunktionen aufruft um die jeweiligen Bilder einzulesen, zu Segmentieren und die Farbinformation des segmentierten Bereiches zu gewinnen. Die Feature Daten sind die Differenz aus mittlerem Rot-Wert und dem Grauwert, der im wesentlichen aus den mittleren Werten für Rot, Gelb und Blau besteht. Die verwendete Formel stammt von Wikipedia.de.

2 b

Training und Verwendung des Klassifikators wird von der Funktion **performClassification.m** durchgeführt. Die aus a) benötigten Daten werden durch einen Aufruf von **generateTData.m** erzeugt. Als user Eingabe ist ein Integer-Wert erforderlich der kleiner oder gleich der maximalen Anzahl der Äpfel bzw. Bananen Daten ist (14). Anhand dieser Eingabe wird ein Trainingsdatensatz und ein Labeling erzeugt. Diese Datensätze werden verwendet um den Klassifikators, bei dem es sich um einen Naiven Bayes Klassifikator handelt, zu trainieren. Das Training erfolgt durch den Aufruf der **trainClassifier.m** Funktion. Die Klassifikation eines nicht gelabelter Featureinformation erfolgt durch **classifyData.m**.

Bis dato war Kenntnis über die Parameter μ und σ^2 nicht erforderlich. Um die Charnoff und Bhattacharyya zu bestimmen müssen allerdings zunächst diese Parameter abgeschätzt werden. Die verwendete Schätzmethode ist Maximum Likelihood, mittels der Funktion **maximumLikelihood.m**, die wiederum **unvariantLikelihood.m** aufruft. Das Ergebnis der Schätzung entspricht der in Matlab integrierten Funktion **mle**.

Zur Bestimmung der Fehlerschranken werden die Funktionen **calcChernoffBound.m**, **chernoffBound.m** und **chernoffBound2.m** benötigt. Dabei wird mittels **chernoffBound.m** das β^* berechnet. Dies geschieht über die in Matlab integrierte Funktion **fminbnd**. Da für $k(\beta)$ das Maximum bezüglich β zu

bestimmen ist, muss der entsprechende Ausdruck invertiert werden. Für die Prior-Wahrscheinlichkeiten wurde 0.5 angenommen, was auch den zugrundeliegenden Daten (Bilder) entspricht.

Die Berechnung der Bhattacharyya-Schranke geschieht über `chernoffBound2.m`, da hier β^* nicht bestimmt werden muss, sondern als 0.5 angenommen werden darf.

2.1 Exkurs: Fehlerschranken bei zwei Klassen

Intuitiv wird eine Entität derjenigen Klasse zugeordnet, der es mit der größten Wahrscheinlichkeit nach angehört. Da die Basis dieser Entscheidung allerdings auf Wahrscheinlichkeiten begründet ist, muss davon ausgegangen werden dass Fehler bei der Zuordnung auftreten können. Um ein möglichst gutes Ergebnis zu gewährleisten ist die Wahrscheinlichkeit für das Auftreten eines Fehlers zu minimieren (siehe Duda et al. Kapitel 2.1).

Chernoff und Bhattacharyya können in diesem Fall bestimmt werden, da es sich um eine Klassifikation mit genau zwei Klassen handelt und die Daten Normalverteilt sind. Die Chernoff-Schranke wird über das Minimum von $e^{-k(\beta)}$ bestimmt, was einem Maximieren von $k(\beta)$ bezüglich β entspricht. Die Stelle an der $e^{-k(\beta)}$ minimal wird beschreibt die unterste Fehlerschranke. Der Vollständigkeit halber ist dieser Wert noch mit den Prior Wahrscheinlichkeiten gemäß der von Chernoff aufgestellten Ungleichung zu multiplizieren.

Die Bhattacharyya-Schranke stellt eine Variante der Chernoff-Schranke dar, die auf Kosten der Genauigkeit die Berechnung erleichtert. Damit ist Bhattacharyya eine schwächere Aussage über die Fehlerwahrscheinlichkeit.

Die Fehlerschranken alleine sind allerdings nicht das Maß aller Dinge und geben alleine keine verlässliche Aussage über die Qualität eines Klassifikationsmodells. Beispielsweise besitzt das Modell auf Basis von je 4 Bildern (der ersten vier), vergleichsweise niedrige Fehlerschranken, klassifiziert jedoch (da das Labeling ja bekannt ist) schlecht und würde damit gegenüber anderen Modellen mit einer wesentlich höheren Accuracy, aufgrund der Fehlerschranken überschätzt werden.

Ebenfalls kritisch zu betrachten ist, dass Chernoff und Co zwar für nicht Normalverteilte Ergebnisse verwendet werden können, aber in diesem Fall nicht aussagekräftig sind (Duda et al. Kapitel 2 Seite 32). Die vorliegenden Histogramme geben rein optisch keine verlässliche Aussage, ein möglicher Test, z.B. Chi-Quadrat wurde nicht durchgeführt. Rein intuitiv ist die Tatsache dass der gute Freddy aber gerne mal die Sorten mischt (grün und rot) ein starkes Indiz für nicht Normalverteilte Daten und damit sind die Fehlerschranken vergebene Liebesmüh. Sinnvoller (und einfacher) wäre es den tatsächlichen Klassifikationsfehler zu berechnen.

2.2 Beispielplot

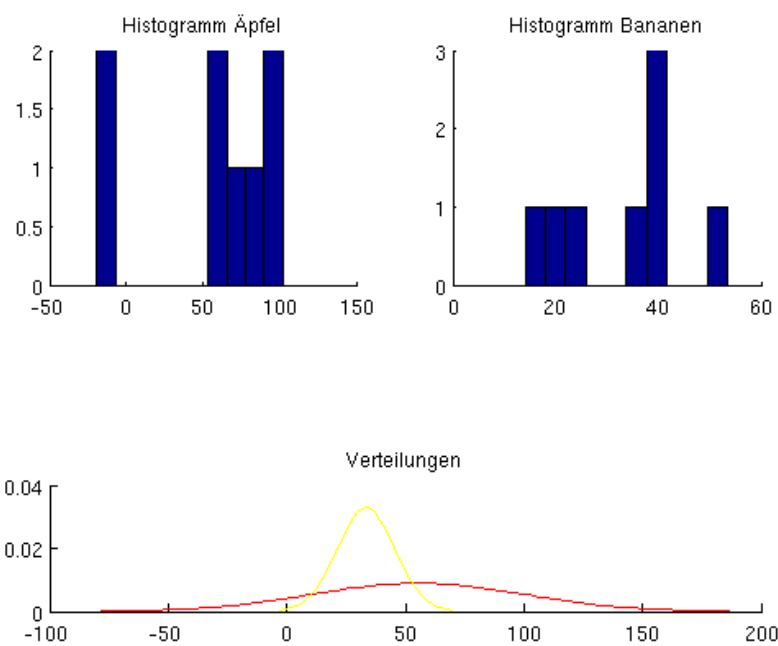


Abbildung 1: Training des Klassifikators für die jeweils 8 ersten Bilder. Die Accuracy der Klassifikation liegt bei 100%. Charnoff liegt bei 0.0451 und Bhattacharyya bei 0.0556.

3 c

Um nur jedes 10te Pixel zu übertragen bzw. zu betrachten sind nur geringfügige Änderungen am Code vorzunehmen. Durch eine Modifikation der Schleifenbedingung der inneren For-Schleife in **splitSegments2.m** wird erreicht, dass für jede Zeile nur das 1,11,21...usw pixel betrachtet wird. Genauer gesagt wird dadurch das Bild auf circa 10% seine Spalten reduziert. So wird zwar nicht exakt jedes 10. Pixel betrachtet (dazu müsste man einfach in den Schleifen mitzählen), aber so wird eine wesentlich geordnetere Reduktion der Bandbreite gewährleistet. Man erhält also keinen Pixelsalat, sondern ein quasi in der Breite runterskaliertes Bild. Die Ergebnisse der Fehlerschranken liegen geringfügig unter den Fehlerschranken bei voller Bandbreite. Die Klassifikation zeigt auf den ersten Blick keine Veränderungen. Als Fazit lässt sich also sagen, dass die Reduktion der Bandbreite keinen nennenswerten Qualitäts-Tradeoff darstellt und die eingesparte Rechenkapazität durchaus gerechtfertigt erscheint. Interessant zu prüfen wäre an dieser Stelle in wiefern dadurch allerdings eine Zeitersparnis erreicht wird. Mehr Aufschluss bringt ein Vergleich der Posterior Wahrscheinlichkeiten. Diese liegen sehr dicht beieinander. Die maximale Differenz beträgt 0.0033 und die mittlere Differenz liegt bei 0.0013, bei einer Standardabweichung von 0.0012. Damit ist die Reduktion der Bandbreite offensichtlich unabhängig von der Qualität des Modells.

Klassifikator1		Klassifikator2	
Apfel	Banane	Apfel	Banane
0.9967	0.0033	0.9966	0.0034
0.9988	0.0012	0.9989	0.0011
1.0000	0.0000	1.0000	0.0000
0.9800	0.0200	0.9833	0.0167
0.9914	0.0086	0.9914	0.0086
1.0000	0.0000	1.0000	0.0000
0.6388	0.3612	0.6404	0.3596
0.6099	0.3901	0.6109	0.3891
0.2481	0.7519	0.2468	0.7532
0.2393	0.7607	0.2404	0.7596
0.2149	0.7851	0.2134	0.7866
0.2218	0.7782	0.2227	0.7773
0.2107	0.7893	0.2126	0.7874
0.3613	0.6387	0.3583	0.6417
0.4704	0.5296	0.4727	0.5273
0.2358	0.7642	0.2392	0.7608

4 d

Das Kreisfeature liefert bei der Erkennung geringfügig schlechtere Ergebnisse als der Rotanteil, allerdings wird der Killerapfel richtig erkannt. Wichtig für die Erkennung ist in diesem Fall besonders die richtige Segmentierung und damit die Wahl des richtigen Thresholds. (Siehe Abbildung 3)

Die Reduktion der Datenmenge wird mit der Matlab-Methode **downsample** vorgenommen, das Graubild wird also praktisch in x-Richtung gestaucht.

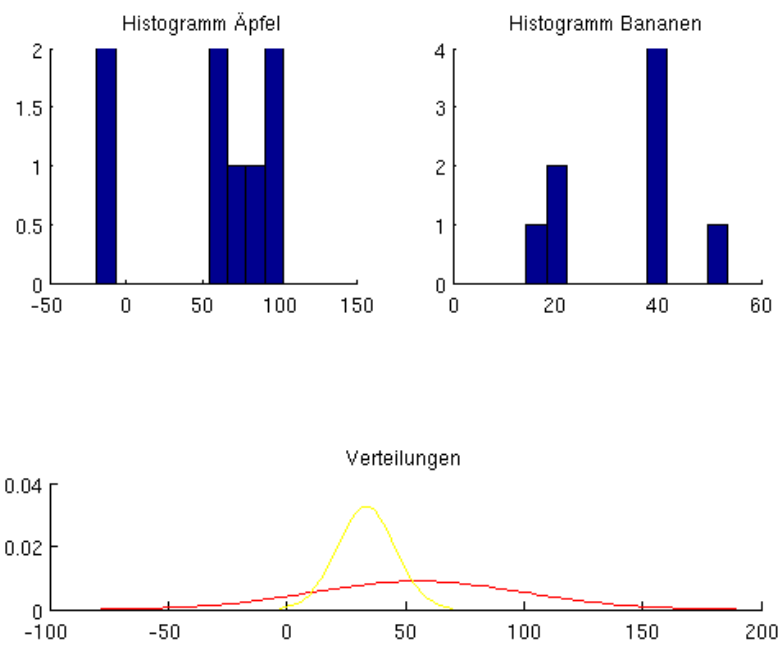


Abbildung 2: Training des Klassifikators für die jeweils 8 ersten Bilder. Die Accuracy der Klassifikation liegt bei 100%. Charnoff liegt bei 0.0437 und Bhattacharyya bei 0.0540.

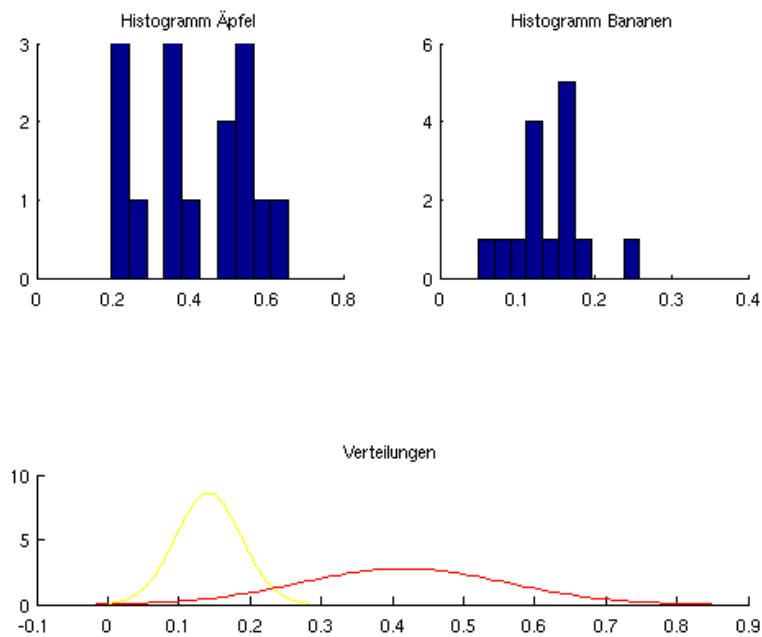


Abbildung 3: Training des Klassifikators für alle Bilder. Die Accuracy der Klassifikation liegt bei 87%. Charnoff liegt bei 0.4157 und Bhattacharyya bei 0.4197.

Trotz dieses offensichtlichen Nachteils liefert der reduzierte Ansatz sogar bessere Ergebnisse. (Siehe Abbildung 4)

5 e

Beide Features zusammen ergeben wie erwartet ein relativ gutes Ergebnis. Der Killerapfel wird richtig zugeordnet.

Die reduzierte Variante macht nur noch einen Fehler. Die fehlenden Informationen wirken sich also anscheinend nicht sehr stark auf das Erkennungsergebnis aus.

6 f

Siehe `modifyClassifier.m`.

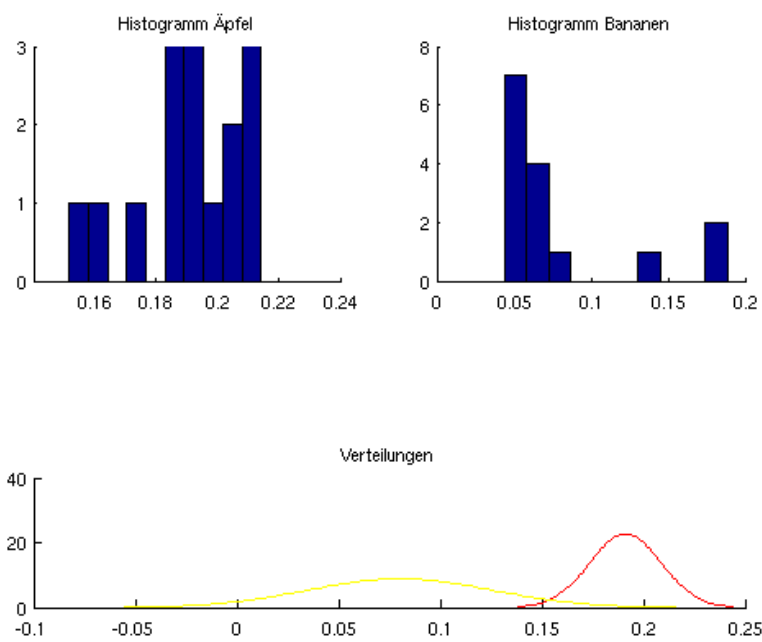
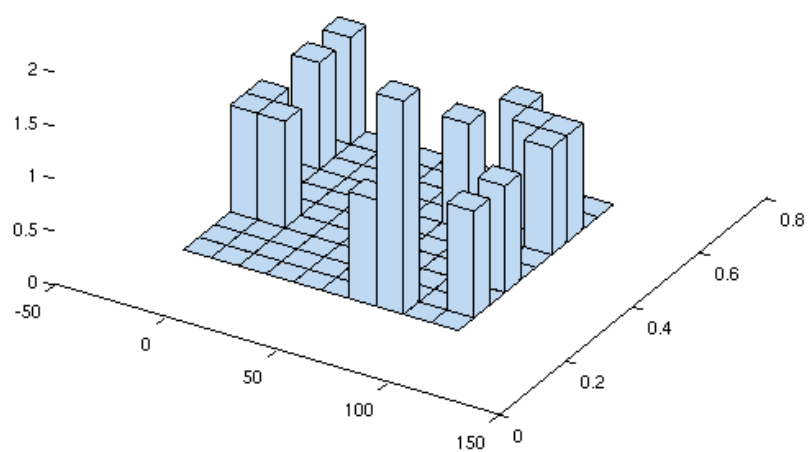
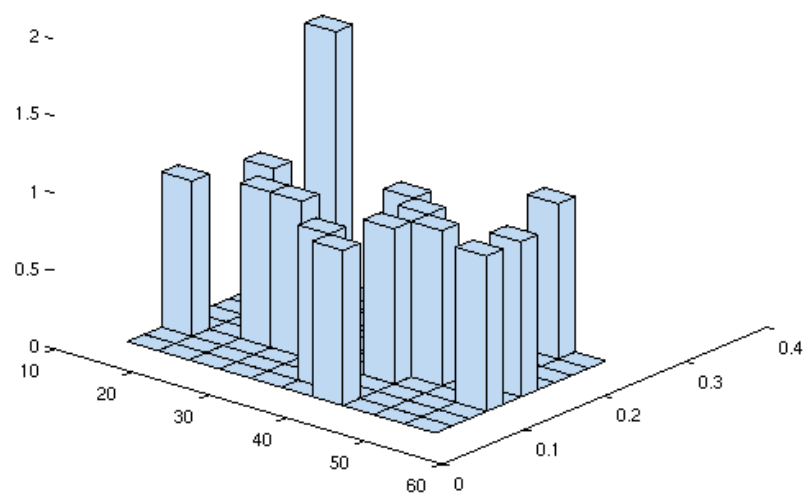


Abbildung 4: Training des Klassifikators für alle Bilder. Die Accuracy der Klassifikation liegt bei 90%. Charnoff liegt bei 0.4493 und Bhattacharyya bei 0.4510.

Histogramm Äpfel



Histogramm Bananen



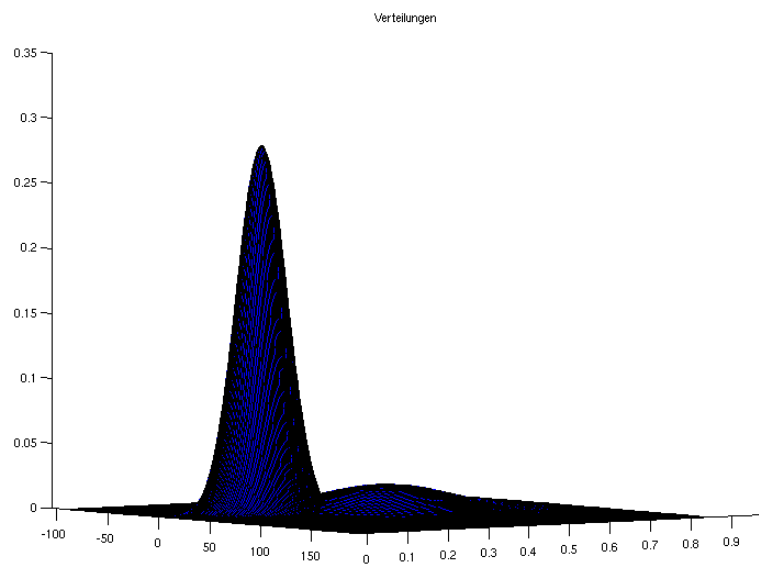
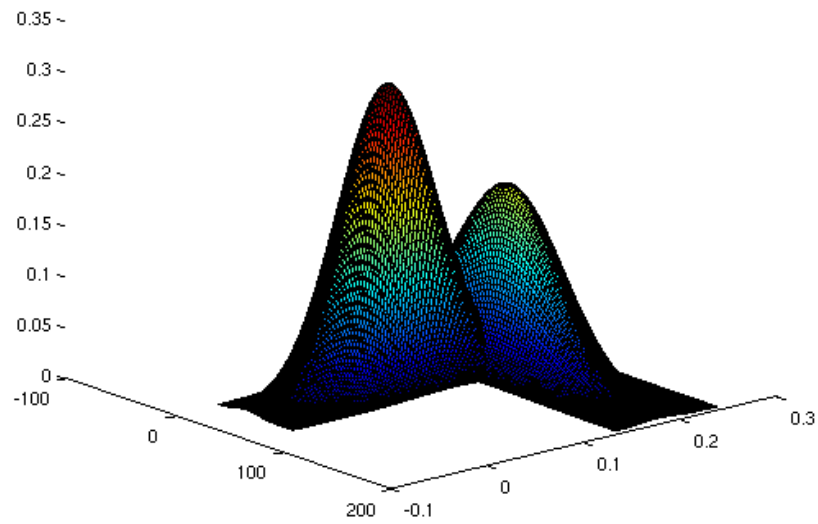
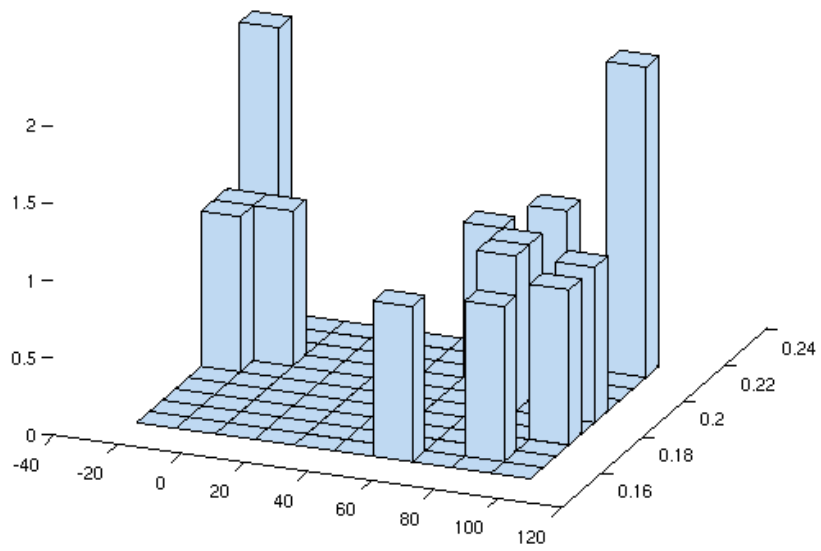


Abbildung 5: Training des Klassifikators für alle Bilder. Die Accuracy der Klassifikation liegt bei 93%.

Verteilungen



Histogramm Äpfel



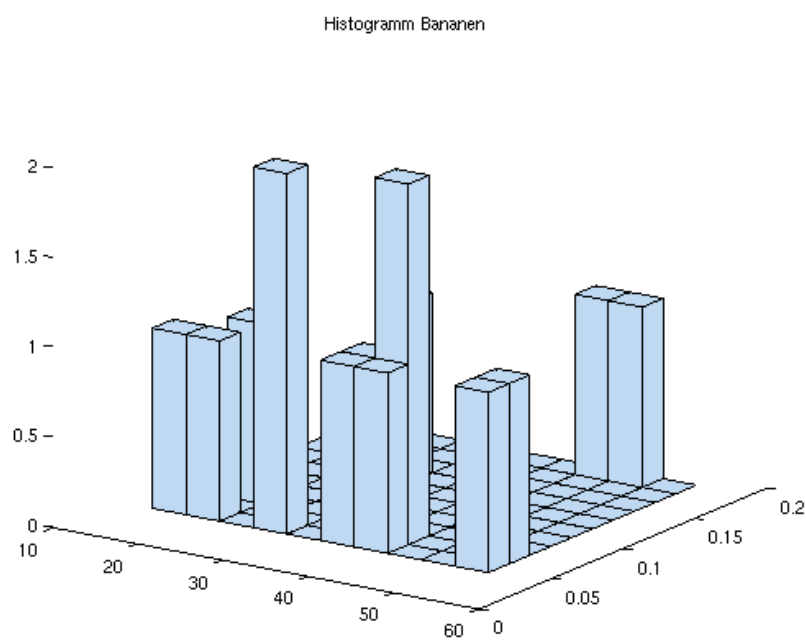


Abbildung 6: Training des Klassifikators für alle Bilder. Die Accuracy der Klassifikation liegt bei 97%.