

# uebungszettel02

May 10, 2019

Inhaltsbasierte Bild- und Videoanalyse - SoSe 19

Übungszettel 2

Abgabe: So. 12.05.2018 - 23:59 Uhr

Hinweis: Alle Lösungen sollen in einem IPython Notebook realisiert werden, wobei Teilaufgaben und Zwischenergebnisse ausgegeben bzw. visualisiert werden sollen. Die Übungszettel sollen in Gruppen von 3 Personen abgegeben werden. Die Datei soll nach folgendem Muster benannt werden: uebung01 {nachname1} {nachname2} {nachname3}.ipynb

Für eine Einleitung in IPython siehe z.B.: <http://cs231n.github.io/ipython-tutorial/>

## 0.1 Aufgabe 1 - Farbräume, Farbmodelle (2+1,5+2 Punkte)

Farbmodell	Zweck	Bereich
RGB		
HSV		
HSB		
YCbCr		

a) Zu welchem Zweck wurden die Farbmodelle RGB, HSV, HSB, YUV und YCbCr eingeführt und in welchen (technischen oder Anwendungs-)Bereichen werden sie verwendet?

b) Geben Sie für die Farbmodelle RGB, HSV, YCbCr jeweils die Kodierung für Magenta, Pink und ein mittleres Grau an. Farbmodell

Magenta  
Pink  
Mittleres Grau  
RGB  
HSV  
YCbCr

c) Setzen Sie Transformationsmatrizen für die Umwandlung aus dem YUV in den YCbCr Farbraum und umgekehrt auf.

## 0.2 Aufgabe 2 - Evaluation (3+0,5+1+1,5+1,5+1+1 Punkte)

Gegeben sei ein Modell A, welches voraussagen soll, mit welcher Wahrscheinlichkeit auf einem gegebenen Bild eine Katze zu sehen ist. Im folgenden werden Arrays mit je 5000 Einträgen geladen. Jeder Eintrag bezieht sich dabei auf ein Bild in einem Datensatz X. Das Array `gt_x` enthält die korrekten Klassen (groundtruth): 1 für "Katze" und 0 für "keine Katze". Das Array `probs_a_x` enthält die vom Modell A vorausgesagten Wahrscheinlichkeiten für "Katze".

```
[ ]: # -*- coding: utf-8 -*-
      %matplotlib inline
      import numpy as np
      import pickle as pkl

      # Datasets
      gt_x, gt_y = pkl.load(open('data.pkl', 'rb'), encoding='latin1')

      # Predictions
      probs_a_x, probs_b_x, probs_rand_x, probs_rand_y = pkl.load(open('predictions.
      ↪pkl', 'rb'), encoding='latin1')
```

a) Das Modell wird als binärer Klassifikator benutzt. Nehmen Sie dazu an, dass ein Bild als „Katze“ klassifiziert wird, wenn das Modell eine Wahrscheinlichkeit  $> 0.7$  ausgibt. Berechnen Sie zunächst die Werte für *true positives* (*tp*), *false positives* (*fp*), *true negatives* (*tn*) und *false negatives* (*fn*). Beurteilen Sie nun die Qualität des binären Klassifikators indem sie die folgenden GütemaSSe berechnen:

- Recall
- Precision
- Accuracy

```
[ ]: 
```

b) Wie könnte ein Modell auf einfache Weise einen Recall von 100% erreichen?

c) Beschreiben Sie den Vorteil des *F-measure* und berechnen Sie diesen.

```
[ ]: 
```

d) Die Precision-Recall Kurve ist ein zweidimensionaler Graph, in dem der *Recall* auf der X-Achse und die *Precision* auf der Y-Achse abgetragen werden. Mit dieser Kurve kann die Performance eines Klassifizierers visuell beurteilt und ein geeigneter Grenzwert gefunden werden. Plotten Sie die Recall-Precision Kurven von Modell A und B auf Datensatz X und vergleichen sie die Plots. Berechnen Sie auch die *AUC* (area under curve). *Hinweis*: Sie dürfen die Bibliothek `sklearn.metrics` benutzen.

```
[ ]: from sklearn.metrics import precision_recall_curve, auc
      import matplotlib.pyplot as plt
```

e) Nehmen Sie an, dass das Modell benutzt wird, um dem Benutzer eine Liste von Katzenbildern zu präsentieren. In der Retrieval-Liste werden Katzenbilder weiter oben als solche ohne Katzen zurückgeben. Um die Güte des Retrieval-Ergebnisses zu messen, berechnen Sie die *average precision* (AP) auf (1) den top-100, (2) top-500 und (3) auf allen Bildern.

[ ]:

f) Betrachten Sie nun einen weiteren Datensatz Y, der ebenfalls Bilder von Katzen enthält. Auch hierfür befinden sich die Groundtruth-Werte im IPython Notebook (gt\_y). Plotten Sie die Verteilungen der Klassen in X und Y. Was fällt auf?

[ ]:

g) Berechnen Sie die *AP-Werte* der Ergebnisse (probs\_rand\_x und probs\_rand\_y) eines zufälligen Klassifizierers auf den Datensätzen X und Y. Welchen Einfluss hat die Verteilung der Daten auf die AP?

[ ]: