

# KI1: Einführung in die künstliche Intelligenz

Sommersemester 2023

Nachklausur



Die vorgefertigten Codeteile liegen im aus den Übungen bekannten Gitlab unter [https://git-kik.hs-ansbach.de/2022\\_wise\\_ki1/ki1\\_nachklausur](https://git-kik.hs-ansbach.de/2022_wise_ki1/ki1_nachklausur). Der Projektname darf beim Fork in die Gruppe <benutzername>\_and\_sg nicht geändert werden. Alternativ kann das Projekt vor Klausurbeginn per USB Stick oder auf Papier bezogen werden.

Zur Beantwortung der Theoriefragen ist die Datei `abgabe.py` zu verwenden. Die Programmieraufgaben sind im jeweiligen Code zu bearbeiten.

Aufgaben:

1. Fülle die allgemeinen Angaben in `abgabe.py` unter Aufgabe1\_Organisatorisches korrekt aus. [1 Punkt]
2. Nenne zwei Beispiele, wie KI beim Autofahren unterstützen kann. [2 Stichpunkte, 2 Punkte]
3. Was unterscheidet Supervised und Unsupervised Learning? [Ca. 2 Sätze, 2 Punkte]
4. Ein alter Spionagesatellit nimmt ein Farbbild der Erdoberfläche in einem Bereich von 2km x 2km auf. Auf dem Foto sollen Strukturen ab einer Größe von 10m unterschieden werden können. Wie viele Werte muss ein Bild der Kamera an Bord (ohne Kompression) mindestens beinhalten? [Zahl und optional Rechnung, 2 Punkte]
5. Die Messung der Temperaturen in einem Raum ergibt folgende Werte:  
[17, 22, 27, 19]  
Normiere diese Werte auf den Wertebereich zwischen 0 und 1. [Zahlen und optional Rechnung, 4 Punkte]

6. Featureoptimierung: Tabelle 6 zeigt die Ergebnisse von sechs Messungen M1 bis M6 mit vier Features F1 bis F4. Der Algorithmus um die Daten zu verarbeiten kann nur mit drei Features umgehen. Angenommen die gezeigten Daten sind repräsentativ für die gesamte Datenmenge: Welches Feature sollte nicht verwendet werden? [1 Punkt]

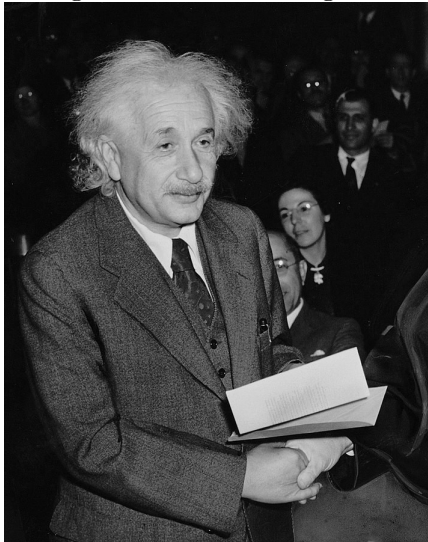
Tabelle 6: Features F1 bis F4 für die Messungen M1 bis M6

	F1	F2	F3	F4
M1	9	0.8	0.77	-0.55
M2	2	0.5	0.77	-1.1
M3	11	0.12	0.77	-0.1
M4	3	0.8	0.77	-0.36
M5	16	0.4	0.77	-0.77
M6	6	0.24	0.77	-0.57
...	...	...	...	...

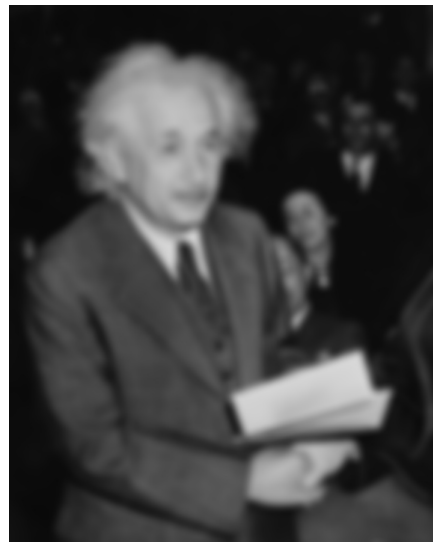
7. Begründe die Antwort von Aufgabe 6. [Ein Grund, 1-3 Sätze, 2 Punkte]
8. Welche Alternative zum vollständigen Verwerfen eines Features gäbe es noch, um den Algorithmus in Aufgabe 6 mit drei Features verwenden zu können? Was macht der alternative Ansatz anders? [Ca. 2 Sätze, 2 Punkte]

9. Abbildung 9 zeigt ein Foto und das Ergebnis einer Vorverarbeitung. Welche in der Vorlesung besprochene Vorverarbeitung kommt dafür in Frage? [Ein Stichpunkt, 1 Punkt]

Abbildung 9: Vorverarbeitung



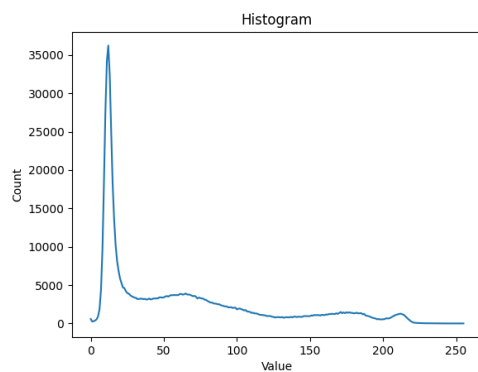
Originalbild



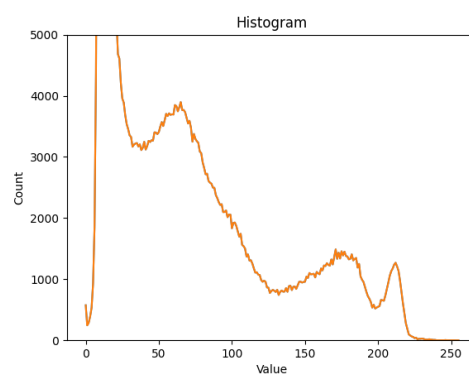
Vorverarbeitet

10. Begründe die Antwort aus Aufgabe 9. [Ca. ein Satz, 1 Punkt]
11. Abbildung 11 zeigt das Histogramm des Originalbild aus Abbildung 9. Was bedeuten lokale Maxima (z. B. bei den Values um 20 und um 60) in so einem Histogramm? [Ca. zwei Sätze, 2 Punkte]

Abbildung 11: Histogramm



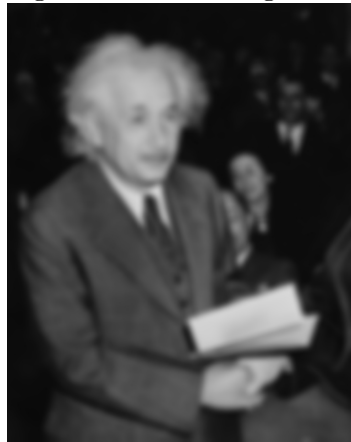
Ganzes Histogramm



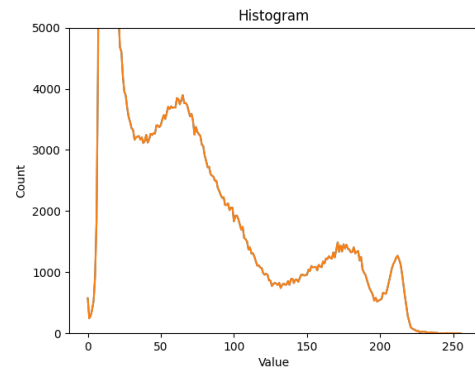
Zoom

12. Abbildung 12 zeigt die Ergebnisse einer Binarisierung durch verschiedene Schwellwerte. Ordne die Bilder den Schwellwerten 40, 130, 145 und 195 zu. [3 Punkte]

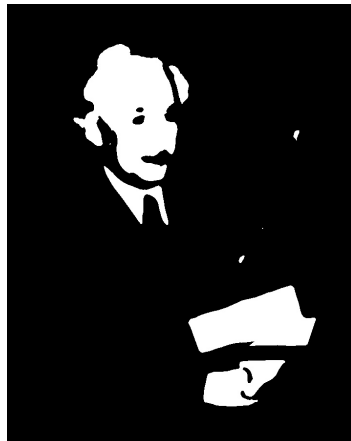
Abbildung 12: Binarisierung durch Schwellwerte zur Segmentierung



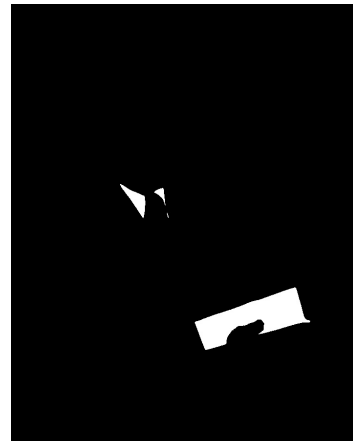
Input



Zoom des Histogramms



A



B



C



D

13. Tabelle 13 zeigt Confusion Matrizen zweier Klassifikatoren A und B. Berechne die True-Positive-Rate für Klassifikation A. [1 Punkt]

Tabelle 13: Confusion Matrizen für Klassifikatoren

	0 pred	1 pred
0	70	30
1	60	140

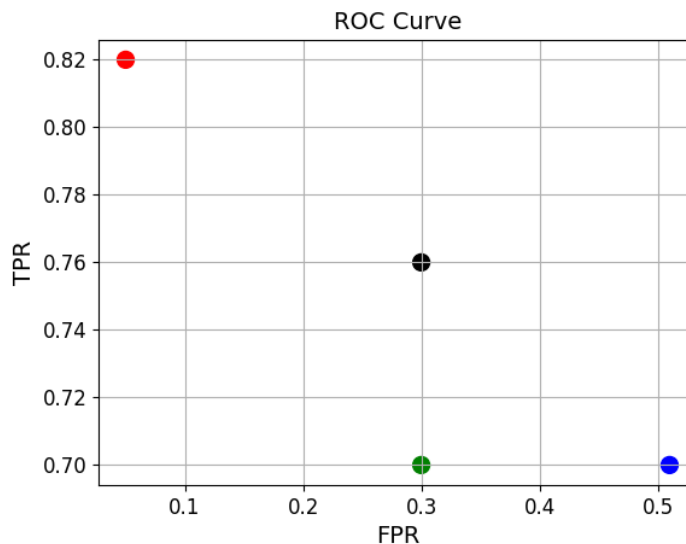
A

	0 pred	1 pred
0	49	51
1	60	140

B

14. Berechne die False-Positive-Rate für Klassifikation A aus Tabelle 13. [1 Punkt]
15. Abbildung 15.2 zeigt die Receiver Operating Characteristic Kurve für die Klassifikatoren aus Tabelle 13 und zwei zusätzliche Ergebnisse der Klassifikatoren C und D. Ordne die Klassifikatoren A und B den entsprechenden Farben in der ROC Kurve zu. [2 Punkte]

Abbildung 15: ROC-Curve der Klassifikatoren A und B sowie zwei unbekannter Klassifikatoren C und D.



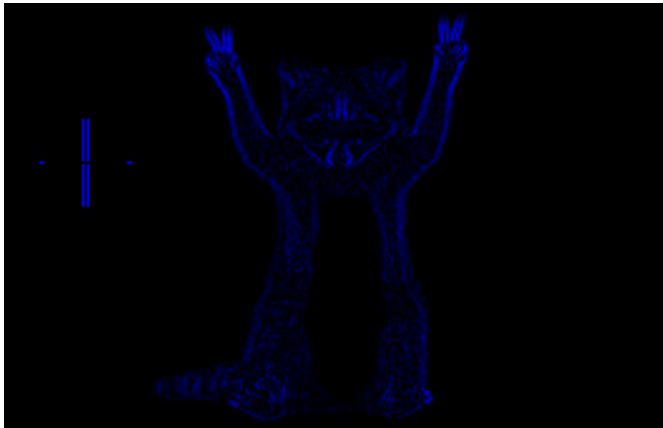
16. Welche Klassifikation in Abbildung 15 ist die beste? [Farbe, 1 Punkt]
17. Programmieren: Das Program soll Kanten auf einem Bild in horizontaler Richtung berechnen. In der Datei *prog\_main.py* ist der Programmablauf vorgegeben. Die Datei *prog\_given* enthält eine Hilfsfunktion. Diese beide Dateien müssen und dürfen nicht verändert werden.

In der Datei *prog\_own.py* müssen drei Funktionen vervollständigt werden.

- *edge\_detection* implementiert eine Kantendetektion in horizontaler Richtung. Relevante Teile des Skripts sind im Unterverzeichnis *./doc* zu finden. Es

soll der "symmetrische" Stencil (mit 0.0 in der Mitte) verwendet werden. Die anderen Funktionen können auch ohne diese Teilaufgabe implementiert werden! [10 Punkte]

- *blue\_edges* überschreibt den Bildinhalt. Der blaue Farbkanal soll auf die Werte der detektierten Kanten gesetzt werden. Die anderen Farbkanäle sollen überall auf 0.0 gesetzt werden. [4 Punkte]
- *save\_final\_image* speichert das Bild mithilfe geeigneter Bibliotheksfunktionen als "hopefully\_blue\_edges.png" ab. [3 Punkte]



Für volle Punktzahl muss jeder Code ohne Fehler importierbar und ausführbar sein. Die Abgabe des Source Codes soll wie aus den Übungsaufgaben bekannt als Commit im geforkten Gitlabprojekt erfolgen [1 Punkt].

Alternativ kann auf Wunsch per USB Stick oder in Papierform abgegeben werden.

Viel Erfolg!

Bearbeitungszeit 60 Minuten