

# Interactive Systems

## Bildklassifikation / Learning from images

WiSe 2017/18

Ziel dieses Übungsblattes ist es sich mit zwei Methoden der Bildklassifikation auseinanderzusetzen.

### Aufgabe 1: Support Vector Machines (5 Punkt)

Eine Support Vector Machine ist ein *supervised* Lernalgorithmus und dient häufig als Klassifikator zur Unterteilung von Objekten in unterschiedliche Klassen. Wie in der Vorlesung besprochen, kann das auch sehr effektiv für hochdimensionale Daten eingesetzt werden. D.h. in unserem Fall sind die hochdimensionalen Daten die Deskriptoren der Bilder (die Sammlung der Bildfeatures). Die SVM muss zunächst auf dem mitgelieferten Bilddatensatz trainiert werden. Dafür müssen Sie ähnlich wie in Hausaufgabe 2.3 Bildfeatures extrahieren und diese dann als Zeilenvektoren untereinander in eine Trainingsmatrix ( $X_{train}$ ) schreiben (siehe Abbildung 1). Zu jedem Zeileneintrag in  $X_{train}$  gibt es dann auch einen korrespondierenden Zeileneintrag im Labelvektor  $y_{train}$ . Auf einigen Testbildern, sollen Sie dann testen, ob das funktioniert. Den Beispieldatensatz finden Sie im .zip.

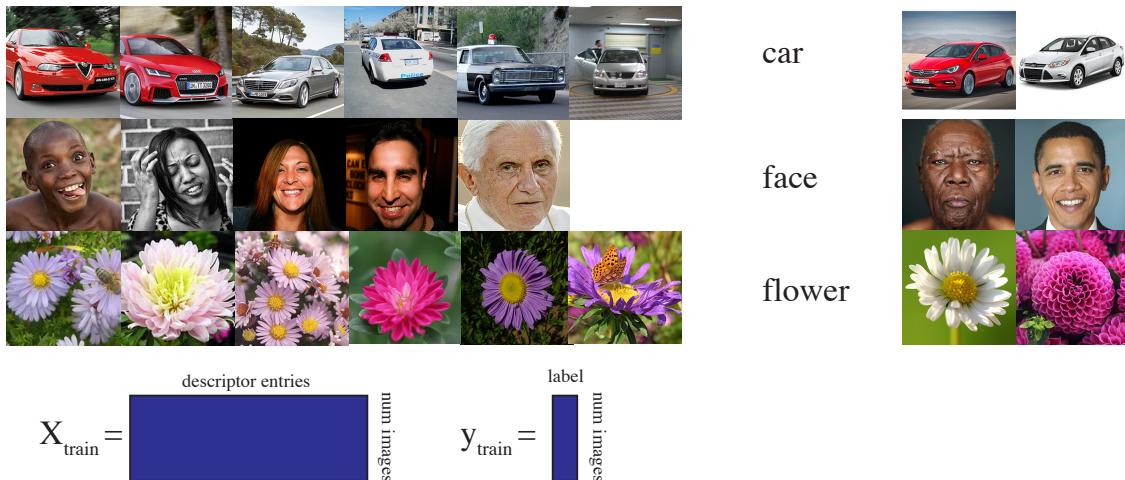


Abbildung 1: (Links) Trainingsbilder der drei Kategorien. (Rechts) Testbilder die nach dem Training in die korrekten Kategorien eingeordnet werden sollten.

Folgen Sie für die Implementierung den Hinweisen und Anweisungen im Source Code. In diesem Fall handelt es sich um ein sehr sehr kleines Beispiel. Sie können auch gern eine eigene Datenbank aufbauen bzw. erweitern, müssen dann aber wahrscheinlich die Parameter der SVM, Anzahl der Keypoints bzw. Größe der Features anpassen. Als Ergebnis sollten alle Trainingsbilder korrekt klassifiziert werden können.

### Aufgabe 2: Deep Learning / Convolutional Neural Networks (5 Punkt + 1 Zusatzpunkt)

Der zweite Ansatz zur Bildklassifikation basiert auf der besprochenen erweiterten Art von tiefen Neuronalen Netzen, den sog. Convolutional Neural Networks (CNN). D.h. Featurevorgaben wie SIFT o.ä. gibt

es hier nicht, sondern diese werden gelernt. Für diese Aufgabe greifen wir auf die sogenannten *Fashion MNIST* Datensatz von Zalando zurück (<https://github.com/zalandoresearch/fashion-mnist#get-the-data>). Dieser besteht aus 60k Trainingsbildern und 10k Testbildern aus 10 unterschiedlichen Kategorien. Die Bilder haben jeweils eine Größe von 28x28px. Ihre Aufgabe ist es zwei unterschiedliche Neuronale Netzwerkarchitekturen zu erstellen, die in ihrer Genauigkeit mindestens über 85% Klassifizierungsgenauigkeit kommen.

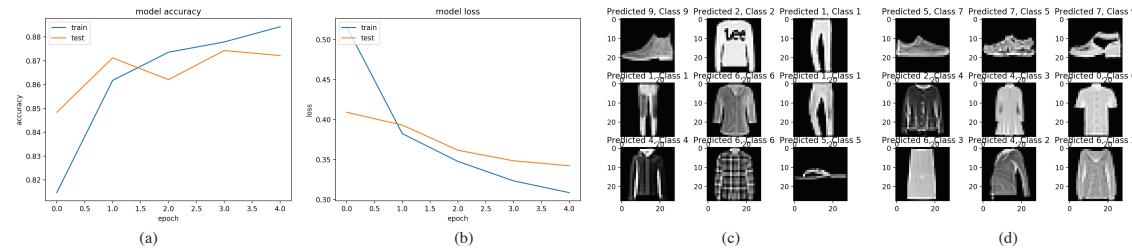


Abbildung 2: (a) Genaugigkeit der Klassifikationsergebnisse mit steigender Anzahl von Epochen. (b) Fehler der Klassifikationsergebnisse mit steigender Anzahl von Epochen. (c) Beispiele richtiger Klassifikation. (d) Beispiele falscher Klassifikation.

Implementieren Sie dafür bitte die Funktion `load_model` in `cnn_model.py` **nach der Vorgabearchitektur in Abbildung 3** und erstellen Sie zusätzlich ein weiteres Model (Neuronales Netz) mit einer anderen Architektur. Diese sollte in Struktur, Kernelgröße etc unterscheiden. Versuchen Sie eine Architektur zu finden, die die gegebenen Architektur in der Vorhersagegenauigkeit übertrifft (**ein Zusatzpunkt**). **Hinweis:** Das Trainieren von CNNs kann sehr lange dauern. Lassen Sie das Netz deshalb zu Beginn erst eine Epoche trainieren, um abschätzen zu können, ob das Training fehlerfrei klappt. Sie sollten zwischen 10-50 Epochen trainieren. Lesen Sie sich selbstständig in die keras.io Dokumentation ein, um die Aufgabe zu implementieren. **Hinweis:** Nutzen Sie für das Model bitte die *Cross-Entropy*-Kostenfunktion: `model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])`. Sie können die Vorgabeimplementierung gern nach ihren Vorstellungen umbauen, wenn ihnen das leichter fällt. Die Abgabe der beiden Architekturen sollte aber durch die Änderung eines Parameters möglich sein.

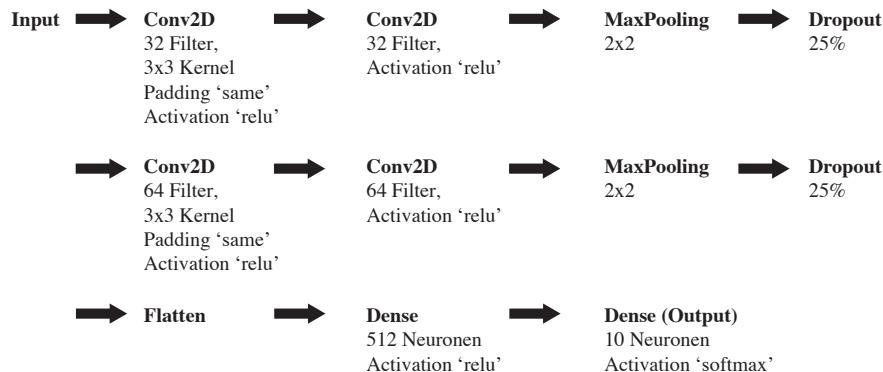


Abbildung 3: Zu implementierende Architektur eines Convolutional Neural Networks.

**Abgabe:** Die Abgabe soll via Moodle zum angegebenen Zeitpunkt erfolgen. Geben Sie bitte jeweils nur eine einzige .zip-Datei mit den Quellen Ihrer Lösung ab. Verspätete Abgaben werden mit einem Abschlag von 3 Punkten je angefangener Woche Verspätung belegt.