

Prof. Dr. J.M. Zöllner

Nikolai Polley, Marcus Fechner
E-Mail: nikolai.polley@kit.edu

Maschinelles Lernen I: Grundverfahren

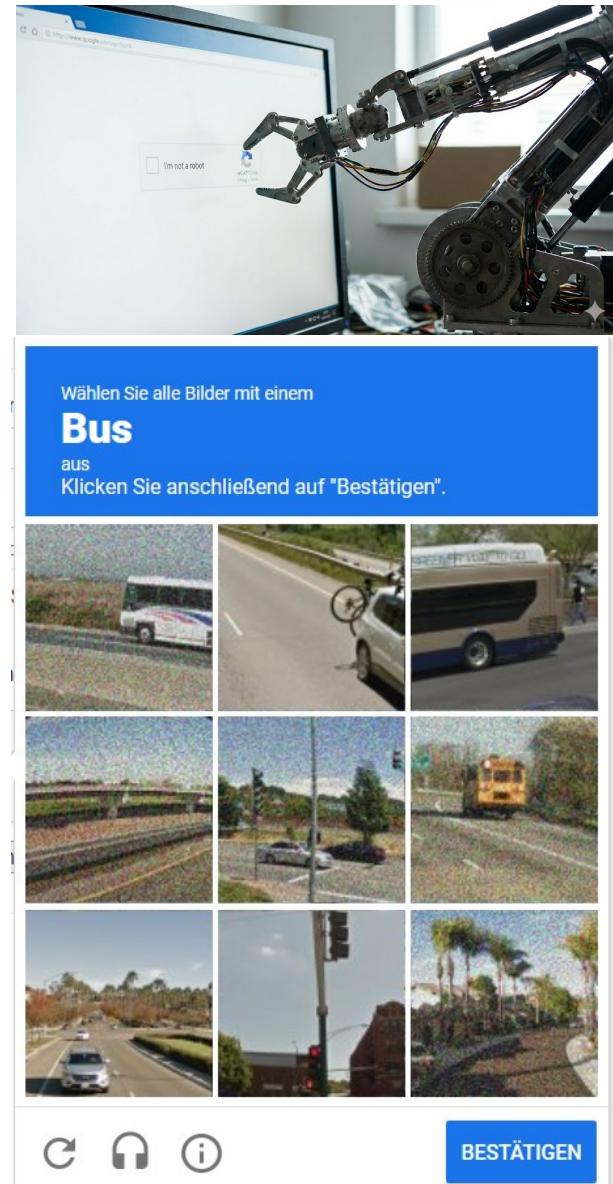
Wintersemester 2025/2026

Abgabe - ML1

(Bearbeitung in maximal 3er Gruppen in der Zeit vom 22.12.2025 bis zum 08.02.2026)

Lernziele: Praktische Anwendung des ge-
lerten Wissens

- Warum ist maschinelles Lernen für diese Aufgabe sinnvoll?
- Welche maschinellen Lernverfahren eignen sich?
- Was ist der Unterschied zwischen Trainingsdaten, Validierungsdaten und Testdaten?
- Welche Frameworks erleichtern das Programmieren?
- Welche Techniken erleichtern das Lernen?
- Welche Techniken beschleunigen das Lernen?



1 Überblick

Der Turing-Test, 1950 von Alan Turing entwickelt, soll den Unterschied zwischen Menschen und Maschinen erkennen. CAPTCHA ist die Kurzform von completely automated public Turing test to tell computers and humans apart. CAPTCHAs werden verwendet, um zu verhindern, dass Computerprogramme Dienste ausführen können, die nur von Menschen ausführbar sein sollen. Meist wird eine Aufgabe gestellt, die von Menschen leicht zu lösen ist, aber Maschinen vor hohe Herausforderungen stellt.

In reCAPTCHA v2 von Google werden u.a. einem Benutzer mehrere Bilder zur Auswahl gegeben, die einer vorgegebenen Klasse zugeordnet werden sollen. Es bestand allgemein die Annahme, dass Maschinen nicht in der Lage sind, solche Klassifizierungen korrekt vorzunehmen. Sie sollen in dieser Abgabe allerdings ein maschinelles Lernverfahren erstellen, das dennoch in der Lage ist, diese Klassifizierung durchzuführen. Es soll gezeigt werden, dass die Bildklassifikation durch moderne ML-Methoden keine gute Basis eines Turing-Tests mehr ist.

2 Organisatorisches

Die Aufgabe startet am 22.12.2025 und darf bis zum 08.02.2026 um 23:55 Uhr bearbeitet werden. Sie dürfen die Aufgabe in Gruppen von bis zu drei Personen bearbeiten. Jede Gruppe darf nur ein Ergebnis abgeben, z.B. darf eine 3er Gruppe nur ein Ergebnis abgeben.

Sie können sich in der Klausur bis zu einem Notenschritt (0,3/0,4) verbessern. Das entspricht bis zu 3 Bonuspunkten. Im Regelfall gibt die Klausur 60 Punkte und unsere Notenskala vergibt alle 3 Punkte eine bessere Note. Sollte sich die Bewertung in zukünftigen Klausuren ändern, wird die Punkteverbesserung analog dazu verändert. Es besteht somit die Möglichkeit, 1, 2 oder 3 Punkte durch diese Abgabe zu erhalten. Sollten Sie die Klausur nicht bestehen, werden keine Bonuspunkte auf die erreichten Punkte addiert.

Alle Mitglieder in der Gruppe erhalten die gleiche Anzahl von Bonuspunkten. Es wird garantiert, dass der Bonus in der Klausur im Wintersemester 2025/2026 und im Sommersemester 2026 gilt. Es wird nicht garantiert, dass der Bonus in den folgenden Semestern (WS 2026/2027 ...) angewendet wird. Alle Boni, die in früheren Semestern erworben wurden, werden berücksichtigt.

Die Abgabe muss über Ilias erfolgen. Dort können Sie bei „Team verwalten“ Ihre bis zu zwei Gruppenmitglieder anhand ihres u****-Kürzels hinzufügen.

Sollten Sie zu zweit oder zu dritt die Aufgabe bearbeiten, müssen Sie untereinander einen „Teamführenden“ auswählen, der Sie in Ilias in das Team aufnimmt und die Abgabe für alle Teilnehmenden in Ilias abgibt.

Es ist nicht möglich, nach der Abgabefrist einem Team zugeordnet zu werden. Stellen Sie daher sicher, dass ihr Teamföhrer Sie in das Ilias Team hinzugefügt hat und die erforderlichen Dokumente hochgeladen hat.

ist nicht erlaubt, die Aufgabe in einer Dreiergruppe zu bearbeiten und in Ilias drei getrennte, aber identische Einzelabgaben einzureichen. Dies hat den Hintergrund, dass unsere Plagiatssoftware in diesem Fall Ihre Abgaben als Plagiat erkennen würde.

Sollte eine Gruppe mehr als 3 Mitglieder besitzen, wird ihre Abgabe nicht gewertet. Es ist nicht erlaubt, unbeteiligte Studierende in ein Team aufzunehmen.

Bei organisatorischen oder inhaltlichen Fragen zu der Abgabe soll das Ilias-Forum verwendet werden.

3 Datensatz

In Ilias ist ein Cloud-Anbieter verlinkt, auf dem zwei .zip Dateien hochgeladen sind: *train_val.zip*, *test.zip*. Sie sollen als Datensatz für diese Aufgabe ausschließlich diese beiden .zip Dateien verwenden. Es ist nicht erlaubt, zusätzliche Daten von CAPTCHAs zu sammeln, um so den Datensatz zu vergrößern. Klassische Transfer-Learning- und Bildaugmentierungs-techniken sind aber natürlich erlaubt.

3.1 TrainVal

Die erste Datei ist *train_val.zip*. Sie enthält 3000 .png Bilder von echten gesammelten reCAPTCHA v2 Aufgaben. Die Bilder haben teilweise unterschiedliche Auflösungsgrößen. Diese 3000 Bilder sind jeweils in eine von 12 Klassen unterteilt. Der Name des Ordners repräsentiert jeweils die Klasse/Label des Bildes. Achtung: Der Datensatz ist nicht balanciert. Sie haben z.B. deutlich mehr Bilder in der Klasse „Car“ als in der Klasse „Chimney“. Sie sollen diese gelabelten Bilder verwenden um Ihr maschinelles Lernverfahren zu trainieren. Aus diesem Ordner sollen Sie einen Trainingsdatensatz und

einen Validierungsdatensatz erstellen. Die konkrete Aufteilung von Training und Validierung ist dabei Ihnen überlassen. Sollten Sie PyTorch benutzen, eignet sich die Klasse ImageFolder um aus dieser Datenstruktur einen Datensatz zu erstellen. Die Funktion random_split könnte daraus einen Trainings- und einen Validierungsdatensatz erstellen.

3.2 Test

Die zweite Datei ist `test.zip`. Sie enthält 8730 .png Bilder. Für diese Bilder erhalten Sie keine Labels. Sie können allerdings davon ausgehen, dass alle Bilder in die gleichen 12 bekannten Klassen aus dem TrainVal Datensatz einsortierbar sind. Sie können außerdem davon ausgehen, dass die Verteilung der Klassen im Testdatensatz ähnlich dem TrainVal Datensatz ist. D.h. auch in dem Testdatensatz sind deutlich weniger Bilder von „Chimney“ als von „Car“.

4 Aufgabe und Bewertung

Ihre Aufgabe ist es, ein maschinelles Lernverfahren zu erstellen, welches diesen Testdatensatz möglichst akkurat klassifiziert. Für diese Klassifikation dürfen alle vorhandenen maschinellen Lernverfahren verwendet werden. Dies beinhaltet auch Verfahren, die nicht in der Vorlesung gelehrt werden. Auch eine manuelle Feature-Erstellung (z.B. SIFT mit anschließendem ML Verfahren wäre denkbar. Mindestens eine Komponente Ihres Programmes muss ein lernbares Verfahren sein, welches ausschließlich auf den Daten aus dem Trainval Datensatzes trainiert wurde.

Für die Bewertung verwenden wir ausschließlich die Genauigkeit/Accuracy auf den Testdaten. Sie allein entscheidet, wie viel Klausurbonus Sie erhalten.

Genaugkeit auf Testdaten	Bonuspunkte in Klausur
80%	3 Bonuspunkte (ein Notenschritt)
75%	2 Bonuspunkte
70%	1 Bonuspunkt

5 Abgabedokumente

Sie müssen in Ilias mindestens drei **separate** Dateien abgeben. Sie sollen **keine** gebündelte .zip Datei abgeben.



5.1 Textuelle Erklärung

Das erste Dokument soll eine .txt Textdatei sein, in der Sie kurz ihr Programm und ihre Vorgehensweise erklären. Sie soll u.A. Folgendes enthalten

- Welches ML-Verfahren haben Sie gewählt.
- Einen groben Überblick darüber, wie Sie trainiert haben.

Es muss keine ausführliche Erklärung geschrieben werden. 10 Sätze reichen aus. Diese Datei wird nicht bewertet.

5.2 Prädiktion des Testdatensatzes

Das zweite Dokument enthält Ihre Prädiktionen des Testdatensatzes in .csv Format. In Ilias wird eine Dummy .csv bereitgestellt. Diese können Sie inspizieren, um die geforderte Formatierung besser zu verstehen. Ihre Datei sollte die gleiche Formatierung wie diese Dummy Datei haben. Ihre Datei muss nach folgender Struktur aufgebaut werden:

- Die erste Zeile muss folgendes enthalten:
ImageName,Bicycle,Bridge,Car,Chimney,Crosswalk,Hydrant,Motorcycle,Other,Palm,Stair,Traffic Light

Es dürfen keine Leerzeichen zwischen den Kommas hinzugefügt werden. Die Spalten dieser Zeile dienen als Überschriften aller folgenden Zeilen.

- Die Zeilen 2 bis 8731 enthält die Prädiktionen Ihres maschinellen Lernverfahrens für den gesamten Testdatensatz. Als erstes wird immer der Name des Bildes in die Zeile geschrieben, z.B. 3000.png. Sie dürfen nicht den ganzen Pfad dieser Datei angeben. C:\Users\Musterstudierende\ML-Abgabe\3000.png wäre beispielsweise nicht erlaubt. Nach dem Namen des Bildes wird die Prädiktion des Netzwerkes für alle Klassen durch Kommas getrennt geschrieben. Achten Sie darauf, dass die Reihenfolge der Prädiktionen mit der ersten Zeile mit den Klassen übereinstimmt (alphabetisch sortiert). Sie sollen die Logits (direkte Ausgabe ihres Verfahrens) oder den Softmax ihrer Logits angeben. Die zweite Zeile welche Logits verwendet könnte folgendermaßen aussehen:

03000.png,-0.50063723,-0.6047857,-0.17692138,4.687087,-0.38999337,-0.9977547,-0.45563334,-0.4828421,0.08075903,-0.43307126,-0.43522504,-0.7576522

Auch hier sind keine Leerzeichen oder andere Abweichungen erlaubt.

Es ist auch erlaubt, Wahrscheinlichkeitswerte anzugeben, sodass die Summe aller Prädiktionen 1 ergibt.

03000.png,0.02941782,0.07778217,0.07384536,0.43320315,0.00789948,0.12801697,0.01816502,0.00768259,0.07211915,0.03236570,0.09052128,0.02898125

Sie dürfen diese Werte nicht runden. Schreiben Sie immer alle Nachkommastellen. Außerdem müssen kontinuierliche Werte für jede Klasse angegeben werden. Es ist nicht erlaubt nur die Klasse mit der höchsten Prädiktion anzugeben wie z.B.:

03000.png,0,0,0,1,0,0,0,0,0,0,0,0

Eine Ausgabe der oberen Art ist nicht erlaubt. Sollten Sie z.B. SVM oder Entscheidungsbäume verwenden, müssen sie dies möglicherweise manuell aktivieren/deaktivieren. Beispielsweise hat das SVM bei scikit-learn den Parameter probability der auf true gesetzt werden müsste. Anschließend müssten die Prädiktionswerte auf den Testdaten mit predict_proba anstelle von predict erstellt werden. Analog müsste auch bei einem Entscheidungsbaum die Funktion predict_proba verwendet werden.

Sollte in ihrer Datei dennoch in sehr seltenen Fällen so eine Zeile auftreten, dann ist es erlaubt. Dies sollte aber nahezu nie vorkommen.

Alle Bilder des Testdatensatzes sollen in sortierter Reihenfolge prädiziert werden. D.h. Zeile 2 beginnt mit 3000.png, Zeile 3 beginnt mit 3001.png, Zeile 4 beginnt mit 3002.png,... und Zeile 8731 beginnt mit 11730.png. Sie dürfen keine zusätzlichen Prädiktionen (von z.B. Trainings oder Validierungsdaten) abgeben. Es darf maximal eine .csv Datei hochgeladen sein. Sie können im Ilias eine bereits hochgeladene .csv Datei durch eine neue .csv Datei ersetzen. Um diese .csv Datei zu erzeugen, kann die Prädiktion aller Testdaten in einem Pandas DataFrame gespeichert werden. Anschließend können Sie mithilfe der Funktion to_csv("result.csv", index=False) daraus eine .csv Datei erstellen.

5.3 Code

Das dritte (und falls notwendig vierte, fünfte ...) Dokument enthält Ihren Code, den Sie für diese Abgabe erstellt haben. Sollten Sie mehrere Dateien für die Abgabe verwendet haben, geben Sie bitte alle verwendeten Dateien ab. Eine explizite Dokumentation des Codes ist für Sie selber hilfreich, wird allerdings nicht von der Übungsleitung vorausgesetzt. Es gibt keine Voraussetzungen wie Ihr Code auszusehen hat. Sie können jede Programmiersprache und ML-Framework verwenden. Es ist auch Ihnen überlassen ob sie Jupyter Notebooks .ipynb oder z.B. python .py Dateien abgeben. Der Code selbst wird nicht explizit bewertet. Code aus Internetrecherchen darf verwendet werden, sollte die dazugehörige Lizenz dies erlauben.

Code generiert von Large Language Models wie ChatGPT o.ä. darf ebenfalls verwendet werden. Hier empfehlen wir allerdings, sich den Code dann auch von ChatGPT erklären zu lassen, um einen Lerneffekt zu erzeugen.

Inzwischen existieren Webseiten und Tools, bei denen ganze Datensätze hochgeladen werden können und diese dann automatisiert das Training vornehmen. Solche „All-In-One“ Tools sind nicht erlaubt. Ihr Code muss selber trainieren, und nicht einfach nur API-Calls vornehmen.

Code von anderen Gruppen darf nicht verwendet werden und kann durch das Plagiatserkennungsprogramm erkannt werden.

6 Webseite

Die Übungsleitung stellt Ihnen die Webseite <https://kit-ml1.streamlit.app> zur Verfügung. Die Nutzung dieser Webseite ist freiwillig. **Die Webseite ersetzt nicht die Abgabe der Dateien im Ilias.** Diese Website kann allerdings helfen die aktuelle Qualität des trainierten Modells einzuschätzen.

Zuerst sollen die u**** Kürzel aller Teammitglieder auf das vorgesehene Feld auf der Webseite eingegeben werden. Die Reihenfolge der u**** Kürzel ist hierbei egal. Mindestens ein Feld muss ausgefüllt sein. Bei 3er Gruppen sollen alle drei Felder ausgefüllt sein. Anschließend ist es möglich, Ihre generierte .csv Datei mit den Prädiktionen auf den Testdaten hochzuladen. Die Website verarbeitet diese Datei und erfüllt anschließend folgende Funktionalität:

- a) Sie visualisiert die abgegebene .csv Datei in einer tabellarischen Ansicht.
- b) Sie überprüft ob das Format den .csv Ansprüchen entspricht.
- c) Sie berechnet die Genauigkeit Ihrer abgegebenen Prädiktion auf den Testdaten.
 - Falls die Genauigkeit sehr niedrig ist (<30%) wird außerdem eine Fehlermeldung ausgegeben. Dies kann Rückschlüsse auf Fehlerquellen bieten.
 - Sollte die Genauigkeit über 30% liegen wird ausgegeben, dass Ihr Modell besser als Zufall ist.
- d) Sie berechnet die Konfusionsmatrix, den Recall, Precision und F1-Score Wert ihrer .csv Datei und visualisiert diese Werte für jede Klasse
 - Diese Werte sind nur für einen besseren Überblick der Performance ihres Verfahrens gedacht. Sie fließen nicht in die Bewertung ein.
- e) Für jede Klasse wird ein Bild aus dem Testdatensatz genommen. Es wird ermittelt, welche Prädiktion ihr Programm für dieses Bild getätigt hat. Es werden die Logits und die normalisierte Ausgabe für alle Klassen dargestellt. Zusätzlich wird analysiert bei welcher Klasse der höchste Wert prädiziert wurde. Dieser Wert wird normalisiert mit dem Klassennamen zurückgegeben.
- f) Anhand der Konkatenation ihrer u**** Kürzel wird ein Pseudonym als Teamname generiert (zufälliges Tier). Dieser Teamname und Ihre Genauigkeit werden in einem Leaderboard gespeichert. Das Leaderboard wird anschließend visualisiert. Dort können Sie sehen, wie gut Ihre Abgabe in Relation zu der Übungsleitung und anderen Kommilitonen ist. Anschließend wird Ihnen Ihr aktueller Platz in diesem Leaderboard mitgeteilt. In die Bewertung geht nur die Genauigkeit Ihrer Abgabe ein. Sie müssen sich keine Sorgen machen, dass z.B. nur die 10 besten Teams den vollen Bonus erhalten.
 - Der aktuelle Stand des Leaderboards kann auch hier eingesehen werden <https://kit-ml1-leaderboard.streamlit.app/>

6.0.1 Verwendung der Website und Botting

Die Website soll Ihnen helfen Fehler in Ihrem Code zu finden. Wir wollen vermeiden, dass Sie viel Arbeit und Zeit in die Abgabe stecken aber keine Bonuspunkte erreichen, da z.B. die Ausgabe falsch formatiert war. Über Ilias lässt sich so eine Funktionalität nicht anbieten. Bitte erstellen Sie keine Bots o.Ä. die automatisiert im Studententakt Ihre .csv Datei auf dieser Website hochlädt. Dies kann zu einer Überlastung der Seite führen und verhindern, dass Ihre Kommilitonen die Seite verwenden können.

6.0.2 Datenschutz

Für das Deployment und Hosting der Website wird Streamlit verwendet. Die Datenschutz- und Sicherheitsbestimmungen finden Sie hier <https://streamlit.io/privacy-policy> und <https://streamlit.io/security>. In der Konfigurationsdatei wurde festgelegt, dass keine Nutzerstatistiken gesammelt werden. Streamlit verwendet Google Server und dementsprechend gelten auch deren Datenschutzbestimmungen. Ihre u**** Kürzel werden mit einem modernen Verschlüsselungsverfahren verschlüsselt gespeichert. Den Schlüssel dafür besitzt ausschließlich die Übungsleitung. Sollten Sie dennoch nicht wollen, dass Ihr u**** Kürzel auf einer nicht-KIT Webseite verwendet wird, können Sie in das dazugehörige Feld eine beliebige 5-stellige Buchstabenkombination eingeben. Damit das Leaderboard Ihnen immer den gleichen Teamnamen zuweist, sollten Sie allerdings dann immer die gleiche Buchstabenkombination verwenden. Falls Sie mit diesen Datenschutzregeln nicht einverstanden sind, benutzen Sie bitte nicht die Website sondern ausschließlich das Ilias.

7 Tutorials und Hilfestellungen

Der Programmcode, der für diese Abgabe geschrieben werden muss orientiert sich zwar an den Übungen, geht allerdings weiter hinaus, als bisher gezeigt. Da es einen Bonus für die Klausur gibt, wird erwartet, dass nicht nur bereits gesehnes angewendet wird, sondern auch darüber hinaus recherchiert wird. Die meisten Probleme lassen sich mithilfe von Suchmaschinen und LLM-Verfahren wie ChatGPT lösen. Hier allerdings ein paar URLs, die man sich anschauen kann:

- PyTorch
 - Beginner Tutorial v. a. das Kapitel über neuronale Netze und CNNs
 - Verwendung von Optimierern und dynamischen Lernraten findet sich [hier](#)
 - Speichern und laden eines Modells
 - Alles über unterschiedliche Layer, Aktivierungsfunktionen etc. findet sich [hier](#)
 - Bereits vorhandene neuronale Netze finden Sie [hier](#)
 - Tutorial über Datasets und Dataloader und volle Dokumentation [hier](#)
- SkLearn (falls Sie kein neuronales Netz verwenden wollen)
 - Einen großen Überblick über alle überwachten Verfahren findet sich [hier](#)
- bwJupyter

Sollten Sie ein neuronales Netz verwenden, dann ist ein Training auf einer GPU sehr sinnvoll. Sollte man keine GPU in seinem privaten Laptop/PC besitzen kann man auf bwJupyter ausweichen. bwJupyter wurde vom KIT und der Uni Stuttgart für die Lehre entwickelt und stellt begrenzte Ressourcen zur Verfügung. Versuchen Sie nach Möglichkeit, unter 6 GB VRAM zu bleiben (kann mit `!nvidia-smi` überprüft werden).

Das Notebook und der Datensatz liegen hier im `__shared` Ordner. Achtet dabei, dass der `__shared` ordner im Read-only-Modus ist, d.h. das vorliegende Notebook muss aus dem ordner in Ihr oberliegendes Verzeichnis kopiert werden, um die `result.csv` zu generieren und Ihre Änderungen in der Datei abzuspeichern.

- Google Colab Alternativ lassen sich auch Cloud Anbieter mit kostenlosen GPUs wie Google Colab verwenden.
 - Tutorial wie man Dateien in Google Colab hochlädt und wieder herunterlädt.

```
* from google.colab import files
uploaded = files.upload()
```

Wenn dieser Code verwendet wird, muss nach jedem Neustart des Notebooks, die Dateien neu hochgeladen werden. Dies kann relativ lange dauern, da die zwei `.zip` Dateien 400MB groß sind.

* Es bietet sich an die zwei `.zip` Dateien in Google Drive hochzuladen. Noch besser ist es, die Dateien auf dem Laptop/PC zu entpacken und die Ordner in Google Drive hochzuladen. Dann müssen die `.zip` Dateien nicht in Drive entpackt werden. Anschließend kann mit

```
from google.colab import drive
drive.mount('/content/drive')
```

immer auf diese Dateien zugegriffen werden.

- Der Befehl `!unzip Pfad` entpackt `.zip` Dateien.