

dlbs Mini-Challenge-Titel

Namen der Teammitglieder

13. Februar 2025

1 Allgemeine Bemerkungen zur Aufgabenstellung

Wählt eine Fragestellung aus den Gebieten Bild- bzw. Signalverarbeitung, welche ihr in dieser Mini-Challenge bearbeiten möchtet. Die Fragestellung soll durch Objekterkennung, Bildsegmentierung, Instanzsegmentierung oder Signalvorhersage mittels Deep Learning implementiert lösbar sein. Die Programmiersprache kann frei gewählt werden. Das Lösungsvorgehen der Mini-Challenge basiert auf den Artikeln:

- How to Avoid Machine Learning Pitfalls: A Guide for Academic Researchers [[Lon23](#)]
- A Recipe for Training Neural Networks [[Kar19](#)]

1.1 Zusammenarbeit

Die Mini-Challenge-Arbeit wird in 2-er-Teams abgegeben. Ausnahmen wie Einzelarbeiten sind mit den DS Fachexpert:innen zu besprechen. Bei der Abgabe muss für gewisse markierte Fragestellungen ausgewiesen werden, wer diese Teilaufgabe durchführte. Die Studierenden dürfen für diese Mini-Challenge mit anderen Teams zusammenarbeiten, aber jedes Team reicht eine einzigartige Abgabe ein.

1.2 Zeitplanung und Termine

- Meilenstein 1: Aufgabenstellung
- Meilenstein 2: 2-3 individuelle Check-Ins
- Meilenstein 3: Demo/Präsentation der Mini-Challenge
- Meilenstein 4: Finale Abgabe der schriftlichen Dokumentation

Die Termine sind auf Spaces im Kalender ersichtlich.

FIXME: Grafik mit Semesterwochen und Meilensteinen ala Robot Challenge

FIXME: Highlight: Ein Report ist typischerweise zwischen 8-12 Seiten und sollte 15 Seiten nicht übersteigen

1.3 Abgaben - Deliverables

Dokumentiert eure Lösung der Mini-Challenge in geeigneter Form inkl. abschliessender Reflexion (in Deutsch oder Englisch). Die Daten müssen nicht abgegeben werden. Der Lösungsweg muss aber nachvollziehbar sein und ausgewählte Daten müssen visualisiert sein. Falls vorhanden, den Link zu den Daten angeben.

- Aufgabenstellung: Fragestellung, Datenlage und Methoden/Vorgehen zusammenfassen (siehe [2](#)) und in einem der Check-Ins mit DS Fachexpert:in besprechen.
- Report, Overleaf bevorzugt - Overleaf Projekt mit Pro-Features kann bei DS Fachexpert:in beantragt werden. Zu lange Textpassagen geben einen Abzug.
- Ggf. Folien eurer Präsentation
- GitRepository mit Quellcode - inkl. HTML Export eurer Notebooks. Hinweis: Wenn ein Modell nicht trainiert, gibt es keine Punkte.

1.4 Demo

Demo/Präsentation: Recap Fragestellung, Demo und Diskussion der wichtigsten Resultate und Schlussfolgerungen; pro Team 10 Minuten mit anschließenden Q&A. Die Demo wird gestoppt, wenn ihr zu lange präsentiert. Online Teilnahme auf Anfrage möglich. Teilnahme an anderen Präsentationen ist erwünscht, aber nicht obligatorisch.

2 Aufgabenstellung

2.1 Fragestellung

TASK: Beschreibt die Ausgangslage und Anwendungsdomäne eurer Fragestellung. Formuliert dazu explizit Hypothese(n), Forschungsfragen oder Ziele. Hier soll das WAS eurer Mini-Challenge beschrieben werden. Ca. 100-200 Wörter.

2.2 Datenlage

TASK: Wählt einen Datensatz mittels welchem die Fragestellung gelöst werden kann. Es können Bilder (2D, 3D), Videos, Signale aus Sensordaten, Punktwolkendaten, Audiodaten oder ähnlich verwendet werden. Es kann ein öffentlich verfügbarer Datensatz, ein eigener Datensatz sowie ein Datensatz aus einem Geschäftsbetrieb sein. Die Daten müssen nicht mit den DS Fachexpert:innen geteilt werden, sofern der Lösungsweg nachvollziehbar ist. Da wir in dieser Mini-Challenge ein supervised Problem lösen, sollen Ground Truth Daten bzw. Labels vorhanden sein oder mit geringem Aufwand generiert werden können. Bemerkung: Für diese Mini-Challenge kann mit einem «zu kleinen» Datensatz gearbeitet werden. In einem solchen Falle, soll dieser Punkt entsprechend diskutiert werden. Beschreibt die Datenlage in ca. 100-200 Wörtern.

2.3 Methoden/Vorgehen

TASK: Beschreibt und diskutiert die Wahl eurer Lösungsansätze (Baseline, DL Architekturen, Evaluation) gemäss eurer Fragestellung. Gebt eine Übersicht eurer verwendeten Methoden und Parameter. D.h., warum habt ihr welche Methode bzw. welches Vorgehen gewählt? Ca. 300-500 Wörter. Die folgenden Teilkapitel sollen direkt in einem Notebook beantwortet werden.

3 Resultate

Zusammenfassung

TASK: Zusammenfassung der Fragestellung, Datenlage, ML Approaches und wichtigsten Resultate inkl. Kennzahlen. Ca. 200 Wörter.

3.1 Datenanalyse

FIXME: Highlight: von hier aus bitte nur in NOTEBOOKS *TASK:*

- *Führt eine dlbs spezifische Explorative Datenanalyse anhand dieser drei Schritte durch [Kar19] und diskutiert pro Person 1 Schritt in 2-3 Sätzen:*
 - *Prüft die Daten mittels visueller Inspektion*
 - *Erfasst ein qualitatives Verständnis der Daten*
 - *Analysiert die Daten quantitativ (mittels Code)*
- *Befolgt für die Datenanalyse Richtlinien von [Lon23]:*
 - *Do take the time to understand your data*
 - *Don't look at all your data*
 - *(Do make sure you have enough data)*
 - *Do talk to domain experts*
 - *Do survey the literature*
 - *Do think about how your model will be deployed*
 - *Don't allow test data to leak into the training process*

3.2 Setup einer Baseline

TASK:

- *Wählt ein einfaches Modell als Baseline, geeignete Metriken für die Evaluation der Netzwerke und diskutiert die Wahl inkl. Parameter. Die Baseline soll trainiert/generiert, analysiert und visualisiert werden (Loss, Predictions etc.).*
- *Befolgt und diskutiert pro Person 1 ausgewählter weiterer Aspekt aus [Kar19] in 2-3 Sätzen.*
- *Befolgt für die Baseline Richtlinien von [Lon23]:*
 - *Do use an appropriate test set*
 - *Do use a validation set*
 - *Do save some data to evaluate your final model instance*
 - *Don't use accuracy with imbalanced data sets*
 - *Do evaluate a model multiple times*
 - *Don't do data augmentation before splitting your data*
 - *Don't allow test data to leak into the training process*

3.3 Overfit

TASK:

- *Fokussiert auf den Loss des Trainingssets*
- *Fügt schrittweise zusätzliche Komplexitäten ein und dokumentiert diese*
- *Lernt ein Modell nicht, soll zuerst dieses Problem gelöst werden (schrittweise Komplexitäten steigern), bevor weitergefahren wird*
- *Wählt eine Modell-Architektur zur Lösung eurer Fragestellung aus und führt ein Overfitting gemäss [Kar19] durch. Wählt pro Person jeweils ein unterschiedlicher Aspekt von [Kar19] aus, welchen ihr implementiert bzw. durchführt und in 2-3 Sätzen diskutiert.*
- *Befolgt für das Overfitting Richtlinien von [Lon23]:*
 - *Don't use inappropriate models*
 - *Do try out a range of different models*
 - *Do evaluate a model multiple times*
 - *Don't do data augmentation before splitting your data*
- *Optionale Bonusaufgabe: Vergleicht und evaluiert eine zusätzliche DL Architektur oder ein zusätzliches Modell*

3.4 Regularize

TASK:

- *Gewinnt an Validierungsgenauigkeit durch Verzicht auf gewisse Trainingsgenauigkeit*
- *Arbeitet pro Person individuell mit eurem Modell weiter und führt pro Person 1-2 Regularisierungsexperiment(e) gemäss [Kar19] durch. Wählt idealerweise die gleichen Regularisierungsexperimente. Diskutiert eure Resultate pro Regularisierungsaspekt in 2-3 Sätzen.*
- *Vergleicht eure Modelle in geeigneter Weise und befolgt dabei die Richtlinien von [Lon23]:*
 - *Do evaluate a model multiple times*
 - *Don't assume a bigger number means a better model*
 - *Do use statistical tests when comparing models*
 - *Do correct for multiple comparisons*
 - *Don't always believe results from community benchmarks*

- *Do be transparent*
- *Do report performance in multiple ways*
- *Do be careful when reporting statistical significance*
- *Do look at your models*
- *Wählt eines eurer Modelle als bestes aus und diskutiert die Entscheidungsfindung in 2-3 Sätzen.*

3.5 Tune

TASK:

- *Führt nun gemeinsam mit dem ausgewählten Modell fort. Und führt eine der folgenden Tuning-Varianten durch.*
 - *Random search*
 - *Grid search*
 - *Hyper-param optimization*
- *Befolgt für das Tuning Richtlinien von [Lon23]:*
 - *Do evaluate a model multiple times*
 - *Do optimise your model's hyperparameters*
 - *Do be careful where you optimise hyperparameters and select features*

3.6 Optional: Squeeze the Juice Out of It

TASK:

- *Wählt eine Methode aus dem Bereich:*
 - *Ensemble models*
 - *Train longer*
- *Befolgt dafür Richtlinien von [Lon23]:*
 - *Do consider combinations of models*
 - *Do evaluate a model multiple times*

3.7 Wichtigste Resultate

TASK: Wählt eure wichtigsten Resultate und beschreibt diese in ca. 400-500 Wörtern. Verwendet ggf. Abbildungen. Z.B. besonders gute vs. schlechte Modellausgaben.

4 Diskussion

TASK:

- *Diskutiert eure Resultate explizit in Bezug auf eure Fragestellung(en) in ca. 500-600 Wörtern. Wie gut beantwortet euer Modell eure Fragestellung? Was bedeuten die Fehler/Genauigkeiten für euren Anwendungsfall? Mit Diskutieren ist gemeint, Vor- und Nachteile, Chancen und Risiken eures Lösungsweges bzw. der Resultate spezifisch für euren Anwendungsfall zu besprechen. Wie sind die Resultate der Baseline im Vgl. zu den optimierten Modellen? Welche Fälle funktionieren gut, welche nicht? Lässt sich die Fragestellung mit den gewählten Modellen und Daten beantworten?*
- *Befolgt für die Diskussion die Richtlinien von [Lon23]:*
 - *Don't generalise beyond the data*
 - *Don't always believe results from community benchmarks*
 - *Do be transparent*

5 Reflexion

TASK: Reflektiert abschliessend in ca. 150-200 Wörtern. Was sind die Schlussfolgerungen für die spezifische Fragestellung? Was ist in dieser Mini-Challenge gut gelaufen? Was würdet ihr das nächste Mal anders machen? Was würdet ihr ggf. in der Aufgabenstellung ändern? Fokus auf fachlicher Reflexion.

6 Code

TASK: Zugang zu aufgeräumtem und mit README versehenem Git-Repository. Die Abgabe darf keine auskommentierten Codestellen enthalten (ausser pip install statements). Bemerkung zu Code Kommentaren: Code Kommentare sind hilfreich, wenn sie erklären, weshalb eine Entscheidung getroffen wurde oder weshalb eine Rechnung in der bestimmten Art und Weise ausgeführt wurde oder gemäss welcher Quelle etwas implementiert wurde. In der Praxis wird jedoch bei der Code-Entwicklung oft vergessen, die Code-Kommentare anzupassen. Deshalb sind Kommentare oft obsolete. Auskommentierter Code wird in der Praxis typischerweise nicht getestet und später weiss man nicht mehr, ob dieser noch funktioniert oder nicht. Die Empfehlung von mir lautet deshalb: So viele Kommentare wie nötig, aber so wenige wie möglich. Grundsätzlich sollte der Code so geschrieben sein, dass er selbsterklärend ist.

Literatur

- [Kar19] Andrej Karpathy. A recipe for training neural networks, 2019.
- [Lon23] Michael A. Lones. How to avoid machine learning pitfalls: a guide for academic researchers, 2023.