README

Pflichtenheft: KI zur Erkennung von echten und KI-generierten Bildern

1. Projektübersicht

1.1 Projektziel

Entwicklung eines Machine-Learning-Modells, das zwischen echten (von Menschen erstellten) und KI-generierten Bildern unterscheiden kann. Technologien die Beispielsweise verwendet werden können: CNN-Klassifizierung, Anomalieerkennung.

1.2 Anwendungsbereich

- Erkennung von Deepfakes und synthetischen Bildern (z. B. generiert durch ChatGPT, DALL·E, MidJourney)
- Mögliche Einsatzgebiete: FakeNews-Erkennung, Qualität von Generierten Bildern bestimmen

---

2. Anforderungen

2.1 Funktionale Anforderungen

ID	Anforderung	Beschreibung
01	Datensammlung	Beschaffung eines ausgewogenen Datensatzes mit echten und KI-generierten Bildern.
02	Datenvorverarbeitung	Bereinigung, Normalisierung und Augmentierung der Bilddaten.
03	Modellauswahl	Auswahl eines geeigneten ML/DL-Modells (z. B. CNN, Vision Transformer, Hybridmodell).
04	Modelltraining	Training des Modells mit Trainings- und Validierungsdaten.
05	Evaluierung	Bewertung der Modelleistung mittels Metriken (Accuracy, Precision).
06	Inferenz	Bereitstellung einer Methode zur Klassifikation neuer Bilder (Echt vs. KI-generiert).

2.2 Nicht-funktionale Anforderungen

ID	Anforderung	Beschreibung
01	Performance	Das Modell soll eine Accuracy von mindestens 85% auf Testdaten erreichen.
02	Skalierbarkeit	Das Modell sollte auf neuen Datensätzen anpassbar sein.
03	Laufzeit	Möglichst geringe Rechenzeit.
04	Robustheit	Das Modell sollte gegen kleine Bildmanipulationen robust sein.

---

3. Projektplanung

3.1 Meilensteine

Meilenstein	Beschreibung
M1	Datensatzbeschaffung & -aufbereitung
M2	Modellauswahl & -implementierung
M3	Training & Hyperparameter-Optimierung
M4	Evaluierung & Dokumentation
M5	Präsentation & Abschluss

---

4. Technische Spezifikationen

4.1 Tools & Frameworks

- **Programmiersprache:** Python
- **Bibliotheken:** Pytorch
- **Datenverarbeitung:** Pandas, NumPy
- **Visualisierung:** Matplotlib, Django (für einen Webserver)

4.2 Hardware

- **Training:** GPU-Unterstützung (z. B. Google Colab, lokale GPU)

---

5. Risikoanalyse

Risiko	Auswirkung	Gegenmaßnahme
Unausgewogener Datensatz	Schlechte Generalisierung	Datenbalance prüfen, Augmentierung
Overfitting	Gute Trainings-, schlechte Testperformance	Regularisierung, Dropout, Cross-Validation
Hardware-Limitationen	Lange Trainingszeiten	Cloud-Ressourcen nutzen (Colab, AWS)
KI-generierte Bilder werden immer realistischer	Modell veraltet schnell	Aktuelle Datensätze verwenden

---

6. Abnahmekriterien

- Das Modell erreicht eine Accuracy ≥ 85% auf einem separaten Testset.
- Die Dokumentation ist vollständig (Code, Trainingsprotokolle, Evaluierung).
- Eine Demo (z. B. Jupyter Notebook) liegt vor.
- Visualisierung auf einem Django-Server

---

7. Projektabschluss

- **Präsentation** der Ergebnisse (Metriken, Herausforderungen, Learnings)
- **Code- & Dokumentationsabgabe** (GitHub-Repository)
- **Reflexion** über mögliche Verbesserungen