Dokumentation zum Python-Skript: Texterkennung und Klassifizierung von DD-Charakterbögen

Jannis Kerz

8. März 2025

1 Einleitung

Dieses Python-Skript wurde entwickelt, um Texte aus gescannten oder digitalen DD-Charakterbögen (Dungeons Dragons) zu extrahieren und automatisch in Kategorien wie *Attribute* und *Kampfwerte* zu klassifizieren. Das Skript nutzt `easyocr` für die Texterkennung (OCR) und ein neuronales Netzwerk auf Basis von `TensorFlow` für die Klassifizierung. Obwohl der Text erfolgreich erkannt wurde, war die Genauigkeit der Erkennung nicht ausreichend, um die extrahierten Daten direkt weiterzuverarbeiten.

2 Funktionsweise

Das Skript besteht aus den folgenden Hauptkomponenten:

1. **Texterkennung (OCR)**: Mit `easyocr` wird der Text aus Bildern oder PDF-Dateien extrahiert. Die Erkennung war grundsätzlich erfolgreich, jedoch führten Ungenauigkeiten dazu, dass der extrahierte Text nicht direkt verwendet werden konnte.

2. **PDF-zu-Bild-Konvertierung**: Falls die Eingabe eine PDF-Datei ist, wird diese mithilfe von `PyMuPDF` (fitz) in Bilder umgewandelt. Diese Bilder werden dann an die OCR-Komponente weitergeleitet.

3. **Textklassifizierung**: Ein einfaches neuronales Netzwerk, basierend auf `TensorFlow`, klassifiziert die extrahierten Texte in die Kategorien *Attribute* oder *Kampfwerte*. Das Modell verwendet eine Bidirektionale LSTM-Architektur, um die Texte zu verarbeiten.

4. **Modellverwaltung**: Das trainierte Modell kann gespeichert und später wiederverwendet werden, um Zeit bei der erneuten Verarbeitung zu sparen.

3 Probleme und Einschränkungen

Während der Entwicklung traten folgende Probleme auf:

- **Begrenzte Trainingsdaten**: Das Modell wurde nur mit zwei Beispiel-Charakterbögen trainiert, was zu einer unzureichenden Genauigkeit führte.
- **Installation von Abhängigkeiten**: Die Installation von `easyocr` und `PyMuPDF` erforderte zusätzliche Tools wie `Poppler`, was zu Komplikationen führte.
- **Ungenauigkeit der Texterkennung**: Obwohl der Text erkannt wurde, war die Genauigkeit nicht ausreichend, um die extrahierten Daten direkt weiterzuverarbeiten. Dies führte dazu, dass die Klassifizierung fehlerhaft war und der erkannte Text manuell nachbearbeitet werden musste.

4 Nicht umgesetzte Ideen

Aufgrund von zeitlichen Einschränkungen konnten folgende Ideen nicht umgesetzt werden:

- **Erweiterung der Trainingsdaten**: Das Modell hätte mit mehr Charakterbögen trainiert werden müssen, um die Genauigkeit zu verbessern.
- **Verbesserung der Texterkennung**: Es wäre sinnvoll gewesen, die Texterkennung durch die Verwendung von zusätzlichen OCR-Tools oder die Nachbearbeitung der erkannten Texte zu verbessern.
- **Erweiterung der Klassifizierung**: Die Klassifizierung könnte um weitere Kategorien erweitert werden, um mehr Informationen aus den Charakterbögen zu extrahieren.

5 Fazit

Das Skript zeigt das Potenzial zur automatischen Texterkennung und Klassifizierung von DD-Charakterbögen. Allerdings war die begrenzte Anzahl an Trainingsdaten und die ungenaue Texterkennung erhebliche Hindernisse für die Genauigkeit des Modells. In zukünftigen Arbeiten sollte die Texterkennung verbessert und das Modell mit mehr Daten trainiert werden, um die Genauigkeit zu erhöhen. Zudem könnte die Klassifizierung um weitere Kategorien erweitert werden, um den Nutzen des Skripts zu erhöhen.

6 Quellcode

Der vollständige Quellcode des Skripts ist auf GitHub verfügbar.