

Projektseminar I4: 'Angewandte Informationswissenschaft'

Andreas Schwanitz - 1945397

11. August 2016

Inhaltsverzeichnis

1	Allgemein	2
1.1	Was wird gemacht?	2
2	Video	2
2.1	Quellvideos	2
2.1.1	Welche Videos bilden die Quelle?	2
2.1.2	Welche Qualität haben die Videos?	2
2.2	Prüfvideos	2
2.2.1	Welche Videos werden zur Prüfung verwendet?	2
2.2.2	Welche Qualität haben die Videos?	2
3	Software	3
3.1	Welche Bestandteile hat die Software?	3
3.2	Welche Tools werden benutzt?	3
3.2.1	Welche Programmierungsumgebung?	3
3.2.2	Welcher CodeStyle?	3
3.2.3	Welche Plugins?	3
3.2.4	Welche externe Bibliotheken?	3
3.3	Wie sieht der Programmfluss aus?	4
3.3.1	Bei der Erstellung der Datenbank	4
3.3.2	Bei der Analyse der Videos?	4
3.4	Welche Methoden werden benötigt?	4
3.5	mögliche Probleme/Lösungen	4
3.5.1	hoher Rechenaufwand	4
3.5.2	große Datenmengen	4
3.6	optionale Erweiterungen	5
3.7	Der Testalgorithmus	5

4	Evaluation	5
4.1	Was wird evaluiert?	5
4.2	Wie wird evaluiert?	6

Zusammenfassung

1 Allgemein

1.1 Was wird gemacht?

Es wird eine Software ohne Gui entwickelt, die verschiedene Videos an Hand einer Datenbank erkennen bzw. analysieren kann.

2 Video

2.1 Quellvideos

2.1.1 Welche Videos bilden die Quelle?

Die Quelle bilden Zeichentrickfilme von Walt-Disney. Zunächst werden folgende Filme verwendet:

- Mulan 1

2.1.2 Welche Qualität haben die Videos?

Ich werde möglichst hochauflösende Videos als Quellmaterial verwenden, sofern es mir möglich ist. Damit sind HD Videos in einer Auflösung von 1080p oder 720p gemeint. Dies ist wichtig, da dieses Material die Basis für die Datenbank und somit für einen möglichen Vergleich mit anderen Videos bildet.

2.2 Prüfvideos

2.2.1 Welche Videos werden zur Prüfung verwendet?

Zur Prüfung werden entweder selbst zusammengeschnittene Videos verwendet oder Videos von Youtube, welche eine oder verschiedene Szenen eines Filmes aus den Quellfilmen enthält.

2.2.2 Welche Qualität haben die Videos?

Die Qualität wie die Auflösung kann hier stark schwanken, was aber erwünscht ist, da wir die Erkennungsrate von möglichst unterschiedlichen Videos testen wollen.

3 Software

3.1 Welche Bestandteile hat die Software?

Die Software hat folgende Bestandteile:

- Einlesen der Videodatei
- Verarbeiten der Videodatei
- Kommunikation mit einer Datenbank
- Vergleich der Videodateien

3.2 Welche Tools werden benutzt?

3.2.1 Welche Programmierumgebung?

- IntelliJ IDEA 2016

3.2.2 Welcher CodeStyle?

- GoogleStyle

3.2.3 Welche Plugins?

- Gradle (für Bibliotheken)
- JUnit (zum Testen)
- SLF4J (zum Loggen)
- Checkstyle (zur Codeüberprüfung)
- FindBugs (zum auffinden von Bugs)

3.2.4 Welche externe Bibliotheken?

- ffmpeg (Video-Bibliothek)
- phash (Original Hash-Bibliothek)
 - Hash-Bibliothek in Java

3.3 Wie sieht der Programmfluss aus?

3.3.1 Bei der Erstellung der Datenbank

1. Einlesen der Videodatei im Hauptprogramm
2. Übergabe der Videodatei an die Video-Bibliothek
3. Erstellung von Snapshots mittels der Video-Bibliothek
4. Einlesen der Snapshots im Hauptprogramm
5. Erstellung von Hashs mittels der Hash-Bibliothek aus den Snapshots
6. Speicherung der Hashs in der Datenbank

3.3.2 Bei der Analyse der Videos?

1. Einlesen der Videodatei im Hauptprogramm
2. Übergabe der Videodatei an die Video-Bibliothek
3. Erstellung von Snapshots mittels der Video-Bibliothek
4. Einlesen der Snapshots im Hauptprogramm
5. Erstellung von Hashs mittels der Hash-Bibliothek aus den Snapshots
6. Vergleich der Hashs mit den Werten aus der Datenbank

3.4 Welche Methoden werden benötigt?

- Eine Methode zum Analysieren eines Videos

3.5 mögliche Probleme/Lösungen

3.5.1 hoher Rechenaufwand

Ein hoher Rechenaufwand tritt bei der Verarbeitung von Videodateien mit großen Auflösungen auf. Eine mögliche Lösung wäre eine Reduzierung dieser auf eine niedrigere Auflösung.

3.5.2 große Datenmengen

Große Datenmengen treten bei einer genauen Analyse eines Videos auf. Bei der Erstellung der Datenbank braucht man möglichst genaue Daten, wodurch die Datenmengen relativ groß sind. Eine Lösung wäre die Genauigkeit zu verringern, wodurch aber auch die Erkennungsrate leidet.

3.6 optionale Erweiterungen

Hier werden Erweiterungsmöglichkeiten gezeigt, die eventuell noch eingebaut werden könnten.

- Erweiterung des Datenbestandes der Datenbank
- Nutzung einer SQL-Datenbank
- Erstellen einer Gui
- Performanceverbesserungen
 - Threading/Parallele Verarbeitung
- genauere Evaluation
 - mehr Testfälle
 - genauere Auswertung

3.7 Der Testalgorithmus

Der Testalgorithmus führt folgende Aufgaben hintereinander aus.

- Datei in Java laden
- neuen Prozess für ffmpeg erstellen und eine Java Thread erstellen
- ffmpeg den Befehl zum auslesen der Bilder ausführen lassen
- mittels Java die Bilder abfangen und in einem anderen Thread hashen
- die ermittelten Hashs cachen
- die Hashs in einer Datenbank hinterlegen / abfragen

4 Evaluation

4.1 Was wird evaluiert?

Ich werde die Qualitätsstufen der Analyse evaluieren, um ein Optimum zu finden, welches eine maximale Erkennungsrate und zudem eine zeitlich möglichst kurze Rechenzeit ermöglicht.

4.2 Wie wird evaluiert?

Bei der Analyse soll man die Anzahl der Snapshots einstellen können, die pro Sekunde erstellt und somit auch überprüft werden. Mehr Snapshots bedeuten mehr Rechenaufwand, aber auch mehr Genauigkeit. Deswegen werde ich mit verschiedenen Werten experimentieren und die Ergebnisse auswerten. Als Standard werde ich zunächst 10fps (Bilder pro Sekunde) bei den Quelldaten.