Projektseminar I4: 'Angewandte Informationswissenschaft'

Andreas Schwanitz - 1945397

23. September 2016

Inhaltsverzeichnis

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 Genaugkeit 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 verwenden.

4.2.1 Berechnung in der Software

Hier möchte ich nun genau erklären, wie ich auf die von mir errechneten Werte komme.

Ich Vergleiche jeden Hashwert des Testvideos mit jedem Hashwert aus der Datenbank. Beim Vergleich von 2 Hashwerten wird ein Differenzwert als Grenzwert für die Hammingdistanz genutzt. Die Hammingdistanz gibt die Anzahl der Unterschiede zwischen den beiden Hashwerten an.

Ich Vergleiche nun alle Werte und zähle wie oft ein Treffer gelandet wurde. Danach berechne ich den prozentualen Anteil, der von den zu testenden Werten gefunden wurde.

Dieses Verfahren wiederhole ich mit einem Grenzwert der Hammingdistanz von 0-19 um einen Überblick der Werte zu erhalten.

5 Probleme

Während des laufenden Projektes gab es mehrere Probleme, die ich hier auflisten möchte:

- Inkompatibilitäten von Videoformaten bzw. mit ffmpeg
- Probleme beim Multithreading

Diese Liste enthält zwar nur wenige Punkte, diese sind aber sehr umfangreich wie auch komplex und haben mich im Projekt viel Zeit gekostet. Ich erwähne dies, da man es am Code nicht erkennen kann, da dieser das Projekt über fast unverändert geblieben ist.

6 Auswertung

Für meine Auswertung habe ich folgende Filme in der Datenbank gespeichert:

- Mulan 1

Als Testmuster sind folgende 4 verschiedenen Youtube-Videos zum Einsatz gekommen:

- MULAN I'll Make a Man Out of You (HQ)
- MULAN Reflection (HQ)
- Mulan 2 Shang Dies HD (BQ)
- Pocahontas - Listen With Your Heart (HQ)

Alle Videos wurden mit den Einstellungen 1, 2 und 10 Fps bzw. Frame per Second verarbeitet und es wurde jeweils eine Datenbankdatei gespeichert um den Zugriff für die Analyse zu beschleunigen.Als Vergleich diente die Hauptdatenbank mit dem Film Mulan 1, der ebenfalls mit 10 Fps verarbeitet wurde. Die Tabellen mit den genauen Werten für jedes Video ist als PDF beigelegt.

6.1 Warum diese Testvideos?

Die Testvideos sind keineswegs zufällig ausgewählt, sondern haben alle eine spezielle Eigenschaft, die ich nutze um etwas zu testen.

Die ersten 2 Videos sind aus dem Film Mulan 1, der die Datenbank bildet. Mit diesen Testmustern teste ich ob eine Erkennung mittels der gleichen Basis möglich sind.

Das dritte Video ist eine Szene aus dem Film Mulan 2, der sich nicht in der Datenbank befindet, aber auf Grund des gleichen Zeichenstils eine Ähnlichkeit besitzt, die ich nutze um zu testen, wie genau die Datenbank ist.

Das vierte Video ist ebenfalls eine Szene aus einem Zeichentrickfilm. Dieses mal ist es der Film Pocahontas, der zwar auch nicht in der Datenbank vorhanden ist, aber als Zeichentrickfilm auch eine Art Ähnlichkeit hat, die wichtig zu testen ist.

6.2 Auffälligkeiten?

Gab es bei der Analyse der Ergebnisse Dinge, die mir aufgefallen sind?

Ja! es gab Dinge, die mir aufgefallen sind:

- Der Grenzwert ist ab dem Wert 12 sinnlos, da dann keine Unterscheidung mehr möglich ist.
- Die Fps-Anzahl bei der Analyse des Videos ist zu vernachlässigen, da oftmals keine Verbesserung der Erkennung möglich ist.
- Der gesuchte Grenzwert der Hammingdistanz für die Analyse scheint bei 6-7 zu liegen, da es dort einen Unterschied von 35-45 Prozent zwischen den richtig erkannten und den anderen Videos gab.

- Bis zum Grenzwert der Hammingdistanz beim Wert 6, liegen die Videos, die nicht in der Datenbank vorhanden sind immer bei 0 Prozent, während die anderen Videos auch dort eine Erkennung verzeichnen.

6.3 Fazit

Als Fazit kann man sagen, dass dieses kurze Projekt sein Ziel in diesem begrenzten Fall erreicht hat. Wir wissen nun, dass wir eine sehr schnelle Analyse mit 1 Fps ohne Verluste durchführen können, was Zeit und Ressourcen spart und kennen nun auch den in unserem Fall ermittelten Grenzwert, der natürlich bei anderen Daten wieder geprüft werden sollte, aber trotzdem ein Anhaltspunkt für weitere Analysen bietet.

Wie es mit einer größeren Datenbank aussieht kann man nur spekulieren, da man aber an den fremden Testvideos gesehen hat, dass eine sehr genaue Trennung bzw. Erkennung möglich ist, bin ich zuversichtlich.