Gesellschaft für Informatik e.V. (GI) GI. (Hrsg.): INFORMATIK 2021,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2021 1077

Technische Herausforderungen bei der Umsetzung von Uploadfiltern

Erkennen von 15 Sekunden Video oder Audio

Martin Steinebach [1]

Abstract: Uploadfilter sind derzeit Gegenstand der öffentlichen Diskussion. Dabei wird in erster Linie über ihren erwünschten Einsatz und ihre Wirkung gesprochen, aber kaum über ihre technischen Hintergründe. Wir beleuchten daher die Verfahren, die Uploadfilter ermöglichen und vergleichen rechtliche Vorgaben mit technischen Eigenschaften. Insbesondere betrachten wir die Vorgabe, dass Ausschnitte von Audio und Video von mehr als 15 Sekunden Länge erkannt werden sollen und zeigen auf, welchen Interpretationsspielraum diese Vorgabe mit sich bringt. Denn für die Performanz der Verfahren ist es von großer Bedeutung, ob die 15 Sekunden zusammenhängend sein müssen oder nicht.

Keywords: Urheberrecht; Robuste Hashverfahren

1 Motivation

Unter dem Begriff ”Uploadfilter” werden Systeme verstanden, die digitale Werke beim Hochladen auf die Plattform eines Onlinedienstes untersuchen und basierend auf dem Untersuchungsergebnis eine Entscheidung über die nachfolgende Verfahrensweise mit diesem Inhalt treffen. Lösungen wie ContentID des Onlinedienstes YouTube sind sicher die bekanntesten Vertreter von Uploadfiltern, die bereits lange in der Praxis verwendet werden. Ihre Aufgabe ist es, urheberrechtlich geschütztes Material zu erkennen, welches auf einer Internetplattform angeboten werden soll. Dazu werden die zu filternden Materialien entweder im Vorhinein in einer geeigneten Form in einer Datenbank hinterlegt oder bei einer Urheberrechtsverletzung auf der Plattform im Nachhinein der Datenbank hinzugefügt. Solche Lösungen regieren nur auf inhaltlich identische Kopien, ein Konzept von Ähnlichkeit wird hier nicht verfolgt: Ein geschütztes Musikstück wird erkannt, nicht aber ähnlich klingende Werke oder andere Werke des gleichen Musikers.

Von der Aufgabe her ähnlich sind andere Filterverfahren wie beispielsweise das Erkennen und Blockieren von erotischen Inhalten in einem Firmennetz oder in sozialen Medien oder das Unterdrücken von Formulierungen mit Beschimpfungen in einem Forum.

[1] Fraunhofer SIT, MSF, Rheinstrasse 75, 64295 Darmstadt, Deutschuland martin.steinebach@sit.fraunhofer.de

Technisch ähnlich zu Uploadfiltern sind Methoden zum Monitoring von Werbevideos, die automatisiert Fernsehkanäle überwachen, um zu prüfen, ob bezahlte Werbeslots auch tatsächlich mit den Werbeinhalten bestückt wurden oder Verfahren zum automatisierten Erfassen von Musikstücken, die bei einem Radiosender gespielt wurden, um Abrechnungen oder Hitlisten zu erstellen.

1.1 Rechtliche Vorgaben

In dieser Arbeit soll der öffentlich bekannte Stand der Technik mit den aktuellen rechtlichen Vorgaben [RS20] verglichen werden. Grundlage dafür ist § 10 UrhDaG-E. Hier werden die Grenzen für "geringfügige Nutzung geregelt"[2]: Erlaubt sind 15 Sekunden Audio oder Video, Bilder und Grafiken mit einer Dateigröße bis zu 125kB und Texte bis zu 160 Zeichen.

Dabei soll der Fokus auf Audio und Video liegen. Die erlaubte Spieldauer von bis zu 15 Sekunden bedeutet im Umkehrschluss, dass ein Uploadfilter erkennen soll, dass eine Verwendung von mehr als 15 Sekunden vorliegt. Nicht klar geregelt ist allerdings, ob die Spieldauer zusammenhängend sein muss oder auch beispielsweise vier unterschiedliche Ausschnitte aus dem gleichen Werk zu je vier Sekunden bereits erkannt werden müssen. Dies führt zu sehr unterschiedlichen Einschätzungen hinsichtlich der technischen Umsetzbarkeit abhängig von der Interpretation dieser Vorgabe. Diese wird im Abschnitt "Herausforderungen" ausführlich diskutiert.

Beachtet werden muss, dass nach § 7 UrhDaG (1) nur ein *bestmöglicher* Ansatz gefordert ist. Auch deshalb ist eine kritische Betrachtung des Machbaren wichtig, um "bestmöglich" einschätzen zu können.

1.2 Inhalte

Ein wichtiger Aspekt bei der Umsetzung von Uploadfiltern ist die Frage, wie viele Inhalte erkannt werden müssen bzw. wie groß die Menge ist, die potentiell erkannt werden soll. Prinzipiell ist die Obergrenze hier die Summe aller Inhalte, die bekannt sind und auf denen Rechte angemeldet werden können. Als Näherung soll hier ein Blick auf die Sammlungen geworfen werden, die von großen Anbietern verfügbar gemacht werden.

Im Audiobereich bietet amazon music 40 Millionen Musikstücke an, Spotify 35 Millionen und Apple Music 45 Millionen [3]. Es sind also zumindest 45 Millionen Musikstücke potentiell zu erkennen, wenn Apple die Sammlungen von amazon und Spotify vollständig abdeckt.

Filme sind weniger präzise abzuschätzen, da hier die Anbieter oft das Angebot wechseln. Bei amazon prime video stehen 20.000 Filme zur Auswahl, bei netflix 2.800 und bei google play

[2] Gesetzentwurf der Bundesregierung: Entwurf eines Gesetzes zur Anpassung des Urheberrechts an die Erfordernisse des digitalen Binnenmarktes

[3] https://www.telegraph.co.uk/technology/0/best-music-streaming-services-apple-music-spotify-amazon-music

13.000[4]. Hier ist allerdings die Summe aller Filme, die potentiell erkannt werden müssen, deutlich höher. Die Filmdatenbank IMDb kennt beispielsweise gut 7,5 Millionen Filme[5]. Hinzu kommen noch Serien, die ebenfalls eine große Menge von Inhalten produzieren, sowie beispielsweise Fernsehsendungen, Musikvideos und Dokumentationen.

2 Technik

Uploadfilter sind eine verkürzte Bezeichnung für komplexe Systeme, in denen das eigentliche Filtern nur eine von vielen Komponenten ist. Entsprechende Lösungen müssen ja neben der Analyse beispielsweise auch die Umsetzungen der Reaktionen auf das Ergebnis der Analyse bereitstellen. Das Wiedererkennen bzw. Re-Identifizieren von Inhalten geschieht entweder über sogenannte robuste Hashverfahren oder über Verfahren zur Extraktion von Merkmalen ("Features") . Entsprechende Verfahren erzeugen eine kompakte Darstellung des Inhalts und speichern diese in einer Datenbank. Zum Prüfen wird dann vom Medium mit derselben Methode ein weiterer Hash oder ein Merkmalsvektor errechnet und mit der Datenbank verglichen. Diese Methoden sind schnell und einfach zu berechnen, weiterhin weisen sie niedrige Fehlerraten auf. Entsprechende Verfahren sind in erster Linie für Multimedia Daten bekannt und werden für Bild, Video und Ton in zahlreichen Anwendungen eingesetzt. Aber auch für Texte sind robuste Hashverfahren bekannt. Ein Uploadfilter besteht aus einer Reihe von Komponenten. Abstrakt müssen dabei zumindest die folgenden vorhanden sein:

Referenzdatenbank: Anhand dieser Referenzdaten entscheidet der Uploadfilter, wie seine Reaktion ausfallen soll. Üblich ist hier eine Liste von Werken, in der die Fälle gespeichert sind, auf die der Uploadfilter mit weiteren Maßnahmen reagieren soll.

Entscheidungsverfahren Im Kern des Uploadfilters muss immer die Frage geklärt werden, ob ein eingehendes Datum (also beispielsweise ein Bild, ein Text oder ein Video) in einer Referenzdatenbank hinterlegt ist. Parametrierbar ist hierbei, wie tolerant das Verfahren gegen Abweichungen zwischen dem vorliegenden Datum und der Referenz ist.

Reaktionsmechanismus Auf Basis der Entscheidung des Einordnungsverfahrens muss der Uploadfilter in der Lage sein, eine abhängige Handlung durchzuführen. Diese Reaktion kann in Abhängigkeit vom Einsatzszenario wieder vielfältig ausfallen. Zur Vermeidung von Urheberrechtsstreitigkeiten wird ein Inhalt vielleicht blockiert oder dem Rechteinhaber und dem Uploader zur Klärung übergeben.

3 Erkennen von Inhalten

Es gibt unterschiedliche Ansätze, die sich mit der Einordnung von Inhalten befassen, also im Kontext eines Uploadfilters die Aufgabe übernehmen, über einen eingehenden Inhalt

[4] https://streaming-geraete.de/filme-netflix-amazon-disney
[5] https://de.wikipedia.org/wiki/Internet_Movie_Database

eine Entscheidung zu treffen. Dabei unterscheiden sich die Verfahren für verschiedene Inhalte deutlich. Eine Lösung für Bilder ist nicht ohne weiteres auf Texte übertragbar. Generell geht es um das Wiedererkennung von Inhalten. Es wird angenommen, dass ein Inhalt bereits bekannt ist, z.B. aus einer früheren Untersuchung. Er soll re-identifiziert und mit einigen in einer Datenbank gespeicherten Informationen darüber abgeglichen werden. Diese kann beispielsweise durch kryptographische (üblich bei Texten) oder robuste Hashes (üblich bei Medien wie Audio, Video und Bild) geschehen. Es muss betont werden, dass es bei dieser Aufgabe darum geht, die Inhalte selbst zu identifizieren und nicht darum, weitere Informationen daraus zu gewinnen. So zählt die Aufgabe, in einem Bild eine abgebildete Person zu erkennen, wie z.B. in [Go14] besprochen, nicht zur hier diskutierten Re-Identifikation.

In den folgenden Abschnitten gehen wir auf eine Reihe von Methoden ein, mit denen Wiedererkennen und Klassifizierung umgesetzt werden können. Diese sind eher beispielhaft zu sehen. Ziel ist es zu zeigen, wie unterschiedlich eine Einordnung erfolgen kann.

3.1 Kryptographische Hash-Funktionen

Kryptographische Hash-Funktionen (siehe z.B. [Ka96]) sind ein Primitiv der Sicherheitsprotokolle mit vielen Anwendungen, die in der IT-Sicherheit schon sehr lange bekannt sind [Da87]. Sie berechnen Hash-Werte fester Länge aus Informationen beliebiger Länge. Sie müssen eine Reihe von Anforderungen erfüllen, unter anderem Effizienz und Kollisionsresistenz und die Eigenschaften einer Einwegfunktion aufweisen. Diese Eigenschaften führen dazu, dass kryptographische Hash-Funktionen nur dazu geeignet sind, identische Kopien eines Werkes zu erkennen. Sobald auch nur minimale Änderungen an der Datei auftreten, die die Informationen speichert, ist der Hash ein vollständig anderer. Dazu genügt es, die Datei mit einem verlustbehafteten Kompressions-Algorithmus wie JPEG für Bilder oder h.264 für Videos zu speichern.

3.2 Robuste Hash-Funktionen

Es sind mehrere robuste oder wahrnehmungsbezogene Hashes für verschiedene Medientypen bekannt, die unterschiedliche Robustheitsgrade bieten. Da es zu viele Algorithmen gibt, um sie hier zu erwähnen, empfehlen wir Erhebungen wie die von Haouzia et al. [HN08] oder Neemila und Singh [NS14]. Es existieren auch Methoden für Audio [HKO01]- und Videostreams [OKH01] sowie für Textdaten [St13]. Auch kommerzielle Lösungen wie Shazam sind bekannt, die auf entsprechende Verfahren setzen[6].

Robuste Hash-Funktionen extrahieren wahrnehmungsrelevante Merkmale aus Multimedia-Inhalten zu Identifikationszwecken. Sie müssen eine Reihe von Anforderungen erfüllen. Die wichtigsten sind:

[6] https://www.heise.de/ct/artikel/Wie-Shazam-Songs-erkennt-4192471.html

- Unterscheidung: Wahrnehmbar unterschiedliche Stücke von Mediendaten sollen unterschiedliche Hash-Werte haben.
- Robustheit: Die robusten Hash-Werte sollen eine gewisse Wahrnehmungsinvarianz aufweisen, d.h. zwei Mediendaten, die für einen durchschnittlichen Zuschauer/Zuhörer hinsichtlich seiner Wahrnehmung ähnlich sind, sollen auch ähnlich sein.
- Sicherheit: Die Merkmale müssen Angriffe überstehen, die direkt auf die Merkmals-Extraktion und nachfolgende Verarbeitungsschritte abzielen. Ähnlich wie bei kryptographischen Hash-Funktionen müssen die robusten Hash-Werte gleichmäßig auf alle möglichen Mediendaten verteilt und paarweise statistisch unabhängig für zwei Mediendaten sein, die sich in der Wahrnehmung unterscheiden.

Die Zuverlässigkeit robuster Hashverfahren bei der Wiedererkennung ist hoch. So weist beispielsweise das Verfahren für Bilder aus [SLY12] in dem dort durchgeführten Test eine Falsch-Positiv-Rate von 0% und eine Falsch-Negativ-Rate von 0.2% auf. Ein Verfahren für Videodaten [LFS20] kommt bei einer ausgeglichenen Optimierung auf beide Raten auf eine Falsch-Positiv-Rate von 1% bei einer Falsch-Negativ-Rate von 5%. Bei dem Audioverfahren AudioID [Al01] wird eine Erkennungsleistung von über 95% auch nach starken Veränderungen angegeben, teilweise werden auch 100% Erkennung erreicht. Das liegt an den Merkmalen, die zur Berechnung der Hashbits verwendet werden, und an dem Umstand, dass die Hashes nicht identisch sein müssen, um zwei Werke als gleich anzusehen. Hier wird mit Distanzen gearbeitet, häufig mit der Hamming-Distanz, die misst, wie viele Bits zweier Hashes sich unterscheiden. Die Distanz, die erlaubt ist, um zwei Werke als gleich anzusehen, ist dann der Schwellwert des Verfahrens. Dieser wird in absoluten Werten (x Bits von n Bits) oder in Prozent (x% von n Bits) angegeben.

Um einen besseren Eindruck eines Hashverfahrens zu haben, stellen wir hier noch eine vereinfachte Variante des Hashes aus [Al01] vor. Der Hash wird gebildet, indem 16 Frequenzbänder aus einem Abschnitt des Audios gebildet und ihre Veränderung über 16 Zeitabschnitte betrachtet werden. Ein Hashbit wird gesetzt, wenn die Summe der Energie im Frequenzband F zum Zeitpunkt t und im drüber liegenden Frequenzband $F + 1$ zum Zeitpunkt $t + 1$ größer ist als die Summe der Energie im Frequenzband $F + 1$ zum Zeitpunkt t und im darunter liegenden Frequenzband F zum Zeitpunkt $t + 1$. So wird der 16x16 Hash aus Abbildung 2 (links) für die drei Sekunden Audio aus Abbildung 1 gebildet. Der Hash daneben ist aus dem gleichen Audio nach einem Hochpassfilter, der das Audio unter 80Hz abschneidet, gebildet. Er unterscheidet sich nur an sechs Stellen vom ersten Hash, ist also robust gegen die Filterung.

Um eine längere Passage zu hashen, werden mehrere Hashes in einer überlappenden Sequenz gebildet. Abbildung 3 zeigt eine solche Sequenz. Hier sind 50 Zeilen dargestellt, die zusammen 10 Sekunden Audio darstellen. Die einzelnen Zeilen stehen dabei jeweils für einen Ausschnitt von 0.2 Sekunden, die 256 Spalten entsprechend den Hashes der

Matrixdarstellung aus Abbildung 2, nur dass diese nun als Folge hintereinander aufgeführt werden und nicht mehr in der ursprünglichen 16x16 Matrix.

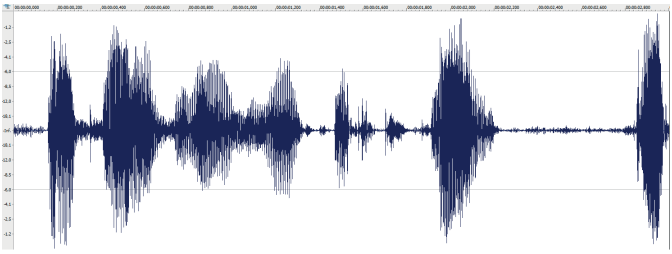

Abb. 1: Amplituden Plot von drei Sekunden Audio. Die X-Achse stellt den Zeitverlauf dar, die Y-Achse die Energie des Audiosignals.

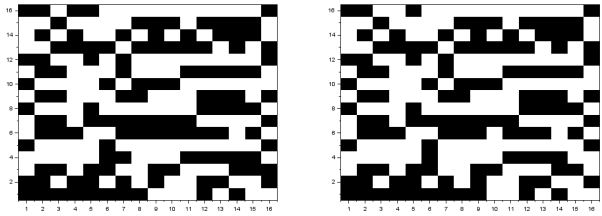

Abb. 2: Links: Hash des Originals, Rechts: Hash nach Hochpassfilter bei 80Hz. Die Hammingdistanz der Hashes liegt bei 6.

4 Herausforderung

Wie gut sind nun die rechtlichen Vorgaben technisch mit dem bekannten Stand der Technik umzusetzen? In diesem Abschnitt sollen die öffentlich verfügbaren Kennzahlen, die wissenschaftlichen Publikationen entstammen, mit den Anforderungen verglichen werden. Von in Unternehmen eingesetzten Verfahren sind entsprechende Daten nicht bekannt, allerdings werden Erfahrungen von Anwendern auf Foren und Blogs verbreitet. Hier wird davon berichtet[7], dass ContentID auf Ausschnitte ab fünf Sekunden Spiedauer reagiert. Sollte das die Grenze sein, unter der eine Erkennung nicht möglich ist, so wären die erlaubten 15 Sekunden nur dann eine technisch umsetzbare Grenze, wenn die Nutzer die verwendeten Inhalte auf maximal drei Ausschnitte aufteilen würden. Würden 16 Sekunden eines Videos in vier Teile zu je 4 Sekunden geteilt werden, würde die Erkennung scheitern. Um die Herausforderungen zu diskutieren, die sich durch die Uploadfilter und die Vorgaben ergeben, verwenden wir ein abstraktes Modell eines robusten Hashverfahrens, wie weiter oben beschrieben. Wir fokussieren uns dabei der Einfachheit halber auf Audiohashes. Die Herausforderungen, auf die eingegangen werden soll, sind folgende:

[7] https://www.eff.org/de/wp/unfiltered-how-youtubes-content-id-discourages-fair-use-and-dictates-what-we-see-online

Abb. 3: Plot von 50 Hashes aus einem Audio von 10 Sekunden Spieldauer

- Datenmengen: Wie groß werden die Datenbanken der Hashes, wenn eine gegebene Anzahl an Werken erkannt werden soll?
- Erkennungsleistung: Wie kann das Verhältnis zwischen zu erkennender Spieldauer abhängig von der erlaubten Fragmentierung zu der Wahrscheinlichkeit einer Erkennung durch das Verfahren abgeschätzt werden?
- Suchverfahren: Welche Möglichkeiten zu einer effizienten Suche nach übereinstimmenden Stellen können umgesetzt werden und welche Auswirkungen haben diese?

4.1 Datenmengen

Robuste Hashverfahren erfordern eine im Vergleich zu den bekannteren kryptographischen Hashverfahren einen größeren Speicherplatz und sind linear abhängig von der Spieldauer von Audio und Video. Das führt bei umfangreichen Sammlungen zu großen Datenbanken. Verwendet man den bekannten robusten Audiohash von Haitsma et al. für alle Musikstücke, die Apple Music anbietet, so gilt Folgendes: In ihrer Veröffentlichung [HKO01] beschreiben die Autoren ein Verfahren, welches in Schritten von 12,5 Millisekunden einen Hash von 32 Bit erzeugt. Pro Minute entspricht dies 4800 Hashes. Nimmt man eine Spieldauer von durchschnittlich 3 Minuten pro Musikstück an, so ergeben sich bei der genannten Sammlung 135 Millionen Minuten Spieldauer. Die ergibt 648 Milliarden Hashes zu 32 Bit, was 2,35 Terabyte Speicherplatz entspricht. Auch andere Verfahren im Audiobereich, die in der Literatur benannt werden, führen zu vergleichbaren Datenbankgrößen. In Tabelle 1 sind die Angaben von Özer et al. [Oz05] und Allamanche et al. [Al01] eingearbeitet.

Tab. 1: Beispiele für die Datenbankgrößen verschiedener Verfahren bei 45 Millionen Musikstücken.

	Hash/s	Bit per Hash	DBS (TB)
Haitsma	80	32	2.00497352
Özer	78	16	0.97742459
Allamanche	43	128	4.31069306

Die Menge an Stücken hat auch eine direkte Auswirkung auf die Erkennungsleistung, die in der Literatur nur selten ausführlich diskutiert wird. Eine Ausnahme ist hier [Al01]. Dort wird beschrieben, dass eine Erweiterung der Referenzdatenbank von 1.000 auf 15.000 Titel zur Folge hatte, dass die Anzahl der Merkmale, die als Hash verwendet wurden, viermal so groß sein musste, um eine vergleichbare Erkennung zu erreichen. Ohne diese Erweiterung wäre die Verwechslungsgefahr durch den geringen Merkmalsraum zu groß. Im nächsten Schritt[8] wurde die Referenzdatenbank von 15.000 auf 90.000 Titel erweitert, ohne die Merkmale weiter anzupassen. Hier bleib die Erkennungsleistung vergleichbar. Nicht abgeleitet werden kann allerdings, wie sich eine Datenbank von 45 Millionen Stücken auf die Erkennung oder die notwendigen Merkmale auswirkt. Hier kann eine Häufung von Fehlerkennungen auftreten, die eventuell weitere Verbesserung am Verfahren erfordert, was wiederum zu größeren Datenbanken führen kann. Das Verfahren von Haitsma et al. gibt eine Datenbank von 10.000 Stücken an [HK02]. Auch hier ist nicht abschätzbar, wie sich eine deutlich größere Menge von Musikstücken auf die Erkennung auswirkt.

4.2 Erkennungsleistung

Die Frage, wie groß die Herausforderung ist, eine Nutzung eines fremden Inhalts von über 15 Sekunden Spieldauer zu erkennen, ist abhängig davon, wie diese verwendet werden. Liegen beispielsweise die 15 Sekunden am Stück vor, so ist eine Erkennung möglich, wenn man die Angaben in der Literatur betrachtet: So wird in [Al01] von Testsequenzen von 20 Sekunden gesprochen. In [HK02] werden nur ”kurze Ausschnitte” erwähnt. Das Verfahren selbst basiert auf Hashes von 3 Sekunden Länge, es kann aber nicht direkt gefolgert werden, dass dies die Länge der Ausschnitte war, da zur Erkennung eventuell mehrere Hashes notwendig sind. In diesem Kontext sind sowohl die falsch-negativen, also nicht erkannte Werke, als auch die falsch-positiven, also Werke, die angezeigt werden, obwohl sie nicht verwendet wurden, von Bedeutung. In der Literatur wird oft nur von der Erkennungsleistung im Sinne der richtig-positiven gesprochen, was dementsprechend erlaubt, auf die falsch-negativen zu schließen.

[8] http://virtualgoods.tu-ilmenau.de/2003/Robust_Audio.ppt

4.2.1 Falsch-Positive

Das führt aber auch dazu, dass ein wichtiges Problem im Betrieb eines Uploadfilters schwer abschätzbar ist: Wie oft kommt es vor, dass ein Uploadfilter auf ein hochgeladenes Medium fälschlicher Weise reagiert? Theoretisch müsste diese Wahrscheinlichkeit verschwindend gering sein. Dies wird auch in [HK02] entsprechend dargestellt, wo eine falsch-positiv Rate in der Größenordnung von $10^{-}20$ errechnet wird. Nehmen wir die Verfahren aus Tabelle 1 und gehen von nur 3 Sekunden Audiolänge aus. Das Verfahren von Haitsma würde dann $3 * 60 * 32 = 5760$ Bit verwenden. Das bedeutet, wir haben 2^{5760} mögliche Hashsequenzen. Auch wenn nur 75% der Bits übereinstimmen müssen, was hier als Erkennungsgrenze erwähnt wird, so ist die Anzahl der möglichen Kombinationen noch immer extrem hoch. Bei kryptographischen Hashes bewegt man sich in deutlich niedrigeren Bereichen, gebräuchlich sind 160 bis 512 Bits. Trotzdem treten bei kryptographischen Hashfunkionen nur sehr selten Kollisionen auf, während bei den robusten Hashverfahren immer wieder falsch-positive entdeckt werden. Ein Bildhashverfahren auf Basis von Helligkeitsblöcken z.B. hat durchaus falsch-positiv-Raten im Prozentbereich bei einer Größe von 256 Bit und muss für eine bessere Leistung besonders optimiert werden [SLY12].

Der Grund dafür ist ein grundsätzlicher: Wie bereits oben beschrieben, arbeiten robuste Hashverfahren anders wie kryptographische. Da sie leichte Änderungen akzeptieren, führen sehr ähnliche Eingaben auch zu ähnlichen Hashes. Bei kryptographischen Hashfunktionen ist dies anders, hier führen schon kleine Veränderung zu maximalen Unterschieden im Hash. Das führt dazu, dass robuste Hashverfahren zwar in der Theorie durch ihre Größe auf einen großen Zahlenraum abbilden, in der Praxis dieser Zahlenraum aber nicht ausgenutzt wird, da die Eingaben nicht zufällig generiert sind, sondern Werke darstellen, die sich oft in gewissen Maßen ähnlich sind. Ein Musikgenre wird unter anderem durch Ähnlichkeiten in Rhythmus, Geschwindigkeit und Tonalität definiert. In Genres wie HipHop mit dem bekannten Amen-Break oder in House mit dem Grundrhythmus von vier Schlägen mit dem Roland 909 Drumcomputer treten solche Ähnlichkeiten ganz deutlich hervor. Dadurch kommt es zu häufigeren Kollisionen zwischen zwei Hashes. Wie hoch diese Wahrscheinlichkeit genau ist, kann nur durch Evaluierungen festgestellt werden. Dass diese Fragestellung noch immer aktuell ist, zeigt beispielsweise [SBL20], wo falsch-postive diskutiert werden, allerdings keine falsch-negative.

Die Wahrscheinlichkeit, dass sich kurze Passagen von beispielsweise einer Sekunde Spieldauer in zwei Stücken ähneln, ist deutlich größer als bei 15 Sekunden. Schon durch wenige Takte, in denen ein Rhythmus ohne Begleitung spielt, kann hier ein Alarm ausgelöst werden. Solche Stellen treten in manchen Genres häufig auf. Das gleiche gilt aber auch für klassische Musik: Für einen Hash kann ein lang gehaltener Ton einer Tonart schnell zu Verwechslungen führen. Das bedeutet, dass eine zu erkennende Stückelung in kurze Passagen durch einen Uploadfilter zu vermehrten falsch-positiven Treffern führen kann.

4.2.2 Falsch-Negative

Da die primäre Aufgabe eines Uploadfilters liegt darin, einmal in der Datenbank hinterlegte Werke wiederzuerkennen. Liegen Teile eines in der Datenbank bekannten Werkes in einem Upload vor, sollte er diese also erkennen. Gelingt ihm dies nicht, liegt eine falsch-negative Entscheidung vor. In der Praxis können Algorithmen Werke desto besser erkennen, je mehr zusammenhängendes Material ihnen zur Prüfung vorliegt. Das liegt daran, dass nicht jede Sekunde eines Werkes gleich gut erkannt werden kann. Teile sind vielleicht durch eine Nachbearbeitung besonders gestört, andere haben schon grundlegend die Eigenschaft, nur schlecht für den Hash geeignet zu sein: Der Algorithmus aus [HK02] basiert beispielsweise darauf, Änderungen in benachbarten Frequenzen und Zeitabschnitten abzubilden. Liegen aber keine Änderungen vor, ist die Energie in den betreffenden Frequenzen über längere Zeit statisch, so entsteht ein schwacher Hashwert, der leicht gestört werden kann. Solche Stellen sind üblicher Weise nur kurz und führen dann nicht zu Problemen, wenn ausreichend besser geeignete Stellen zur Verfügung stehen.

Unter anderem deshalb ist es für einen robusten Hash deutlich einfacher, eine robuste Wiedererkennung zu erreichen, wenn 15 Sekunden Material am Stück vorliegen, als wenn der Hash mehrere Ausschnitte von wenigen Sekunden erkennen soll. Würden beispielsweise 20 Sekunden verwendet, die aber in 10 Stücke zu je zwei Sekunden aufgeteilt sind, und sind dann drei Stücke für eine Erkennung ungeeignet, dann werden nur noch 14 Sekunden erkannt und der Uploadfilter gibt eine falsch-negative Entscheidung aus. Eine vergleichsweise umfangreiche Betrachtung der Fehlerraten wird in [Oz05] erstellt. Hier liegen die Erkennungsraten bei ca. 99%, die Fehlalarme bei ca. 1%. Relativiert werden muss die Evaluierung leider durch die geringe Anzahl von nur 1550 Teststücken.

Auch wenn der Fokus der Betrachtungen hier auf Audio liegt, gibt es die gleichen Überlegungen und Eigenschaften auch bei Hashes für Videodaten. So werden in [LM12] Analysen der Abhängigkeit für falsch-positive und falsch-negative für mehrere Hashverfahren durchgeführt. Selbst bei schwachen Veränderungen gilt für alle Verfahren: Eine niedrige Wahrscheinlichkeit, ein Video nicht zu erkennen, wird immer mit einer hohen Chance erkauft, einen Fehlalarm auszulösen. So liegt bei einer falsch-negativen Einschätzung von ca. 10% die Fehlalarmrate für den LRTA (low-rank tensor approximations)- Algorithmus schon bei 20%.

4.3 Suchverfahren

Ein robuster Hash erfordert andere Suchstrategien wie ein kryptographischer Hash oder auch allgemein ein Datenbankeintrag. Da der robuste Hash nur ähnlich zu einem anderen sein muss und diese Ähnlichkeit durch die Hamming Distanz berechnet wird, sind bei der Suche zahlreiche Operationen notwendig. In der Praxis wird der Hash des hochgeladenen Datums errechnet und dann mit allen Hashes in der Datenbank verglichen. Dieser Vergleich

zählt die Bit-Positionen, an denen Such-Hash und Datenbankeintrag nicht übereinstimmen. Die Anzahl dieser Fehlpositionen ist die Hamming Distanz. Bei der großen Menge von Hashes wie in Tabelle 1 dargestellt, ist eine Suche rechenaufwändig. Dementsprechend müssen Optimierungen erfolgen, wie bereits in [Al01] dargestellt. Hier werden besonders prägnante Passagen der Such-Hashes identifiziert und zuerst nur nach diesen gesucht, was nur einen Verlust bei der Erkennung im Bereich eines Prozentes mit sich bringt. Auch in [WSY14] werden Methoden zum schnellen Suchen vorgestellt und auch hier ist der Verlust an erkannten Bildern nur gering im Vergleich zum Gewinn an Geschwindigkeit.

Wie sich die zu erwartenden großen Datenbanken auf die Leistung der Verfahren auswirken, lässt sich schwer abschätzen. Offensichtlich ist allerdings, dass ein Kompromiss zwischen Erkennungsleistung und Performanz notwendig ist. Ein praktikabler Uploadfilter wird bei der Suche eine Optimierung einsetzen müssen, um mit vertretbarem Aufwand Werke zu suchen. Diese werden deutlich stärker ausfallen müssen als in der Literatur diskutiert wird, da die Datenbanken in der Praxis um mehrere Größenordnungen umfangreicher ausfallen als die bei den Evaluierungen eingesetzten.

5 Diskussion

Der vorhergehende Abschnitt zeigt, dass eine Umsetzung eines Uploadfilters eine Reihe signifikanter Herausforderungen mit sich bringt, die teilweise wissenschaftlich nur schwer abgeschätzt werden können. Dies gilt vor allem für die Frage, wie die Erkennung von mehr als 15 Sekunden Material gehandhabt werden soll. Sind die 15 Sekunden am Stück zu erkennen, kann dies mit dem Stand der Technik gut abgebildet werden, wenn robuste Hashverfahren oder Inhaltsmerkmale eingesetzt. Die Fehlerraten sollten hier akzeptabel sein. Kryptographische Hashverfahren sind für den Einsatz in Uploadfilter bei Medien nicht akzeptabel. Mit ihnen könnten zwar die effizienten Ansätze umgesetzt werden, sowohl was Berechnungs- als auch Suchaufwand betrifft, allerdings ist bei ihnen ein Umgehen trivial.

Bei einer Verteilung der 15 Sekunden auf mehrere kürzere Stücke ist mit einem signifikanten Anstieg an Fehlern zu rechnen. Hier muss dann entschieden werden, ob fehlerhafte Ablehnungen oder Fehlalarme die größere Problematik mit sich bringen. Darauf basierend müssen die zur Erkennung eingesetzten Algorithmen optimiert werden.

Aber auch die schiere Menge an Daten, die durchsucht werden muss, wird in der Praxis eine Herausforderung sein. Dass dies technisch möglich ist, zeigen Lösungen wie ContentID von Google. Dahinter steht allerdings ein Aufwand, der für Plattformbetreiber erheblich sein kann. Dies gilt insbesondere, wenn die Rechteinhaber all ihre Werke in die Referenzdatenbanken das Uploadfilters einstellen können, um deren Verwendung erkennen zu können. Um die Uploadfilter in die Praxis einzuführen empfiehlt es sich, die Datenbanken schrittweise zu befüllen und hier mit Grenzen zu arbeiten, die beispielsweise auf der Verbreitung der Werke basieren, wenn davon ausgegangen werden kann, das bekannte und aktuelle Werke häufiger verwendet werden. So kann die Infrastruktur kontinuierlich wachsen. Auch eine Regelung,

die die 15 Sekunden anfangs nicht segmentiert und im Laufe der Zeit die Erkennung immer kürzere Segmente fordert, würde einer realistischen Entwicklung der Technologie entgegenkommen.

Bei der Suche nach geeigneten Technologien werden auch immer wieder Anbieter der Eingangs erwähnten verwandten Anwendungen wie Werbemonitoring oder das Erkennen von Musikstücken im Radio. Diese Technologien haben die gleichen Wurzeln, allerdings eine deutlich abweichende Ausgangslage. Prüft beispielsweise ein Unternehmen, ob seine Werbespots im Fernsehen gezeigt werden, dann ist die Referenzliste nur sehr kurz. Wahrscheinlich hat das System höchstens eine zweistellige Anzahl von Inhalten, die es erkennen muss. Gleichzeitig ist ein Werbespot meist 30 Sekunden lang und nicht unterbrochen oder von anderen Inhalten überlagert. Das bedeutet, dass eine in diesem Umfeld erfolgreiche Technologie nicht zwingend auch für einen Uploadfilter geeignet sein muss.

Förderhinweis

Diese Forschungsarbeit wurde vom Bundesministerium für Bildung und Forschung (BMBF) und vom Hessischen Ministerium für Wissenschaft und Kunst (HMWK) im Rahmen ihrer gemeinsamen Förderung für das Nationale Forschungszentrum für angewandte Cybersicherheit ATHENE unterstützt.

Literaturverzeichnis

[Al01] Allamanche, Eric; Herre, Juergen; Hellmuth, Oliver; Froeba, Bernhard; Kastner, Throsten; Cremer, Markus: Content-based Identification of Audio Material Using MPEG-7 Low Level Description. In: ISMIR. 2001.

[Da87] Damgård, Ivan Bjerre: Collision free hash functions and public key signature schemes. In: Workshop on the Theory and Application of of Cryptographic Techniques. Springer, S. 203–216, 1987.

[Go14] Gong, Shaogang; Cristani, Marco; Loy, Chen Change; Hospedales, Timothy M: The re-identification challenge. In: Person re-identification, S. 1–20. Springer, 2014.

[HK02] Haitsma, J.; Kalker, T.: A Highly Robust Audio Fingerprinting System. In: ISMIR. 2002.

[HKO01] Haitsma, Jaap; Kalker, Ton; Oostveen, Job: Robust audio hashing for content identification. In: International Workshop on Content-Based Multimedia Indexing. 2001.

[HN08] Haouzia, Adil; Noumeir, Rita: Methods for image authentication: a survey. Multimedia tools and applications, 39(1):1–46, 2008.

[Ka96] Katz, Jonathan; Menezes, Alfred J; Van Oorschot, Paul C; Vanstone, Scott A: Handbook of applied cryptography. CRC press, 1996.

[LFS20] Liu, Huajian; Fach, Sebastian; Steinebach, Martin: Motion vector based robust video hash. Electronic Imaging, 2020(4):218–1, 2020.

[LM12] Li, Mu; Monga, Vishal: Robust video hashing via multilinear subspace projections. IEEE transactions on image processing, 21(10):4397–4409, 2012.

[NS14] Neelima, Arambam; Singh, Kh Manglem: A short survey on perceptual hash function. ADBU Journal of Engineering technology, 1, 2014.

[OKH01] Oostveen, Job C; Kalker, Ton; Haitsma, Jaap: Visual hashing of digital video: applications and techniques. In: Applications of digital image processing XXIV. Jgg. 4472. International Society for Optics and Photonics, S. 121–131, 2001.

[Oz05] Ozer, Hamza; Sankur, Bulent; Memon, Nasir; Anarim, Emin: Perceptual audio hashing functions. EURASIP Journal on Advances in Signal Processing, 2005(12):1–14, 2005.

[RS20] Raue, Benjamin; Steinebach, Martin: Uploadfilter - Funktionsweisen, Einsatzmöglichkeiten und Parametrisierung. Zeitschrift für Urheber- und Medienrecht, 64(5):355–364, 2020.

[SBL20] Son, Heui-Su; Byun, Sung-Woo; Lee, Seok-Pil: A Robust Audio Fingerprinting Using a New Hashing Method. IEEE Access, 8:172343–172351, 2020.

[SLY12] Steinebach, Martin; Liu, Huajian; Yannikos, York: Forbild: Efficient robust image hashing. In: Media Watermarking, Security, and Forensics 2012. Jgg. 8303. International Society for Optics and Photonics, S. 83030O, 2012.

[St13] Steinebach, Martin; Klöckner, Peter; Reimers, Nils; Wienand, Dominik; Wolf, Patrick: Robust hash algorithms for text. In: IFIP International Conference on Communications and Multimedia Security. Springer, S. 135–144, 2013.

[WSY14] Winter, Christian; Steinebach, Martin; Yannikos, York: Fast indexing strategies for robust image hashes. Digital Investigation, 11:S27–S35, 2014.