
DIPLOMARBEIT

Herr Ing.
David Eder

video-detective

Ein Tool zur Kontrolle von Video-Files
und Workflow-Beschleunigung

2017

DIPLOMARBEIT

video-detective

Autor:
Herr Ing. David Eder

Studiengang:
Technische Informatik

Seminargruppe:
KT12wWA-F

Erstprüfer:
Prof. Dr.-Ing. Olaf Hagenbruch

Dipl. Ing. (FH) Matthes Nagel

Zweitprüfer:
Dipl.-Wirt. Ing. (FH) Gregor Moretti

video-detective

author:

Herr Ing. David Eder

course of studies:

Technical computer science

Seminar group:

KT12wWA-F

First examiner:

Prof. Dr.-Ing. Olaf Hagenbruch

Dipl. Ing. (FH) Matthes Nagel

Second examiner:

Dipl.-Wirt. Ing. (FH) Gregor Moretti

Graz, Februar 2017

Bibliografische Angaben:

Eder, David:

Video-detective – Ein Tool zur Kontrolle von Video-Files und Workflow-Beschleunigung – 2017. – 82 S.

Mittweida, Hochschule Mittweida (FH), University of Applied Sciences,
Fakultät Angewandte Computer- und Biowissenschaften, Diplomarbeit, 2017

Referat:

Die vorliegende Diplomarbeit befasst sich mit der Analyse, Kontrolle und Optimierung der Videoaufzeichnung bei LinkedIn Austria. Anhand der Problemstellung wurde ein Tool (video-detective) programmiert. Daten zu sammeln und zu analysieren ist das Hauptziel des video-detective. Anhand der Analyse wird der gesamte Workflow der Videokontrolle beschleunigt und vereinfacht. Auf die Bedienbarkeit wird besonders Wert gelegt. Für eine schnelle Übersicht werden die gesammelten Daten grafisch dargestellt. Ein weiteres Ziel dieser Arbeit ist, fehlerhafte Video-Files so schnell wie möglich zu erkennen und richtige Gegenmaßnahmen einzuleiten. Die Kontrolle der Files soll möglichst im Hintergrund stattfinden und den Dozenten nicht bei den Aufzeichnungen stören.

Inhalt

Abbildungsverzeichnis:	III
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VI
Glossar	1
1 Übersicht	3
1.1 Motivation	3
1.2 Problemstellung.....	3
1.3 Zielsetzung	4
1.4 LinkedIn-Qualitätskriterien eines Video-Trainings	5
1.5 Kapitelübersicht.....	7
2 Stand der Technik	9
2.1 Alternative Software zur Kontrolle von Video-Settings	9
2.2 Eingeschlagener Weg zur Realisierung.....	13
3 Programmiersprache grundlegende Überlegungen	14
3.1 Vor- und Nachteile der einzelnen Programmiersprachen	17
3.2 Die Gewählte Programmiersprache.....	18
4 Grundlegende Struktur und Vorgehensweise	20
4.1 Musskriterien	20
4.2 Wunschkriterien.....	21
4.3 Abgrenzungskriterien.....	21
4.4 Notwendige Funktionen und Teilfunktionen.....	21
4.5 Hauptfunktionen und Teilfunktionen des video-detective (Skizze)	23
5 Grundlegende Programmierung media_info	28
5.1 MediaInfo grundlegende Funktionen	29
5.2 Einbinden der MediaInfo-Funktionen in C#	30

6	FFmpeg grundlegende Funktionen	35
6.1	FFmpeg – Bedeutung im video-detective	35
6.2	FFmpeg-Programmierung im video-detective	36
7	Die automatisierte Indexerstellung im video-detective	39
8	Funktionen und Aufbau des video-detective	41
8.1	Der linke Teil des video-detective.....	43
8.2	Der rechte Teil des video-detective (deaktivierter Zustand) Control-Tab.....	50
8.3	Indexerstellung eines Video-Trainings.....	52
8.4	Import und Export der XML-Datei im video-detective	54
8.5	Production-Tab	57
8.6	video audio split Funktion im video-detective	58
9	Fehlermeldung und Fehlerbehebung des video-detective	60
9.1	Beispiel für ein Attention-E-Mail (Fehlermeldung)	62
9.2	Beispiel für ein Attention-Dokument (Fehlermeldung)	63
10	Zusammenfassung und Ausblick	64
10.1	Ergebnisse	64
10.2	Ausblick.....	65
Anlage A	66	
Anlage B	68	
Anlage C	69	
Anlage D	70	
Literaturverzeichnis	71	

Abbildungsverzeichnis:

Abbildung 1: Grundlegende Funktionen und Teilfunktionen des video-detective	23
Abbildung 2: Use Case Diagram	25
Abbildung 3: MedialInfo in das Projekt einbinden	30
Abbildung 4: MedialInfo in Visual Studio einbinden	31
Abbildung 5: Verweismanager Fenster geöffnet	31
Abbildung 6: Visual Studio Projektmappen-Explorer Verweis MedialInfo	32
Abbildung 7: Visual Studio Projektmappen-Explorer MedInfo.dll	32
Abbildung 8: MedialInfo im Einsatz, Auflösung	33
Abbildung 9: MedialInfo im Einsatz, Bitrate	34
Abbildung 10: FFmpeg im Projekt einbinden	36
Abbildung 11: FFmpeg im Einsatz, vor dem Split	37
Abbildung 12: FFmpeg im Einsatz, nach dem Split	38
Abbildung 13: Office Verweise im Projekt hinzufügen	39
Abbildung 14: video-detective Startoberfläche	41
Abbildung 15: video-detective Startoberfläche linke Seite	43
Abbildung 16: video-detective Startoberfläche Ordner hinzufügen	44
Abbildung 17: video-detective Startoberfläche Ordner wurde hinzugefügt	45
Abbildung 18: Attention Dokument Auswahl	46
Abbildung 19: Attention Email Auswahl	46
Abbildung 20: video-detective ein Pfad aktiviert	47
Abbildung 21: video-detective gesamte Kontrolle aktiviert	48
Abbildung 22: video-detective Startoberfläche rechte Seite	50
Abbildung 23: video-detective Startoberfläche Content Tab	51
Abbildung 24: video-detective content tab index Optionen	53
Abbildung 25: video-detective Content Tab, nach der Kontrolle	53
Abbildung 26: Exportierter Index	54
Abbildung 27: video-detective Feedback-Export-Funktion	54
Abbildung 28: video-detective Export-Funktion im Einsatz	55
Abbildung 29: video-detective exportiertes XML-File	56
Abbildung 30: video-detective Import XML-File	56
Abbildung 31: video detective rechts Production Tab	57

Abbildung 32: video detective rechts Video Audio Split Funktion	58
Abbildung 33: video detective rechts video audio split Funktion im Einsatz	58
Abbildung 34:	
video detective rechts video audio split Funktion Status Anzeige.....	58
Abbildung 35:	
video detective rechts video audio split Funktion, generierter Ordner	59
Abbildung 36: Outlook Attention Email Darstellung	62
Abbildung 37: Attention Dokument Darstellung	63
Abbildung 38:	
video detective gesamte Oberfläche im gestarteten Modus, Control Tab	66
Abbildung 39:	
video detective gesamte Oberfläche im gestarteten Modus, Content Tab	66
Abbildung 40:	
video detective gesamte Oberfläche im gestarteten Modus, Production Tab.....	67

Tabellenverzeichnis

Tabelle 1 Qualitätskriterien LinkedIn	7
Tabelle 2 Vorteile und Nachteile Alternativer Software Lösungen	12
Tabelle 3 Vorteile und Nachteile verschiedener Programmiersprachen	17
Tabelle 4 Entscheidungsmatrix Programmiersprache	18

Abkürzungsverzeichnis

AAC	Advanced Audio Coding
PCM	Puls-Code-Modulation
tsc	TechSmith Screen Capture Codec
avi	Audio Video Interleave
DLL	Dynamic Link Library
FLV	Flash Video
bat	batch
wav	Waveform
dgv	datagridview
Kbps	kilobyte per second
HTML	Hypertext Markup Language
XML	eXtensible Markup Language
UML	Unified Modeling Language

Glossar

Content-Team:

Das Content-Team bei LinkedIn Austria ist für den gesamten Inhalt (die Themen) der Trainings verantwortlich. Die Videos müssen kontrolliert und in strukturierter Form der Postproduktion übergeben werden.

Postproduktion:

Die Postproduktion ist für die Bearbeitung der Videos verantwortlich. Das bedeutet, alle Störgeräusche und sonstigen Versprecher sind herauszuschneiden.

Trainings:

Als Trainings bezeichnen wir unsere Lernvideos (Tutorials), die von unseren Trainern (Experten, Professoren, Autoren, ...) aufgezeichnet werden.

Studiomanager:

Der Studiomanager kümmert sich um die Bereitstellung der benötigten Hard-/Software, damit sie für die Aufzeichnungen rechtzeitig zur Verfügung stehen. Zusätzlich ist er für den direkten technischen Support unserer Trainer verantwortlich.

Videoproofing:

Als Videoproofing wird der Prozess der Videokontrolle bezeichnet. Dieser Prozess beinhaltet die genaue Untersuchung der Videos auf Versprecher sowie Bildfehler.

Index:

Der Index ist eine in Excel erstellte Sammlung an Trainingsinformationen. Unter anderem beinhaltet er Videobeschreibungen, Schnittenweisungen, Korrekturen sowie Länge der Videos. Der Index wird vorzugsweise schon vor dem Aufenthalt des Trainers mit Informationen (Trainingsinhalt, geschätzte Länge, gewünschte Gliederung) befüllt. Das Content-Team nutzt den Index als Kommunikationsschnittstelle zwischen Trainer und Content-Manager. Die Postproduktion wiederum verwendet den Index schlussendlich als Schnittenweisung.

Aufzeichnungsroutine:

Als Aufzeichnungsroutine wird der gesamte Ablauf einer Trainingsaufzeichnung bezeichnet. Diese Routine beinhaltet die technischen Aspekte wie Soundkarte, Kompressor und Mikrofon sowie den gesamten Ablauf einer Aufnahme (Videogliederung, Namensgebung, Indexerstellung, ...).

Trainer:

Als Trainer bezeichnen wir in unseren Unternehmen Doktoren, Professoren und Experten, die zur jeweiligen Thematik Trainings aufzeichnen. Die Trainer kommen aus allen Herrenländer und zeichnen in den dafür vorgesehenen Studios auf.

1 Übersicht

Im einleitenden Kapitel werden Motivation, Problemstellung sowie Qualitätskriterien und Zielsetzung dieser Diplomarbeit erörtert. Ein kurzer Überblick zu den einzelnen Kapiteln dieser Arbeit rundet dieses Kapitel ab.

1.1 Motivation

Bei LinkedIn Austria werden täglich Video-Trainings (siehe Glossar) in verschiedenen Sprachen (Deutsch, Englisch, Französisch, Spanisch, Japanisch) produziert. Im Durchschnitt entstehen pro Woche ca. 30 Trainings mit einer Laufzeit von je fünf Stunden. Die Aufnahme der Trainings beziehungsweise der Lernvideos erfolgt in speziell dafür ausgestatteten Aufnahmestudios mit sehr sensibler Audio- und Videotechnik.

Die Dozenten zeichnen ihre Trainings im Studio auf. Die Aufnahmen werden intern vom Content-Team (siehe Glossar) überprüft und anschließend in der Postproduktion optimiert. Für die Aufnahmen ist ein definierter Zeitrahmen vorgesehen. Insofern ist eine zeitnahe Überprüfung sehr wichtig. Mit der Videoüberprüfung beschäftigt sich die vorliegende Arbeit.

1.2 Problemstellung

Aufgrund der großen Menge der Videos und der sensiblen Audiotechnik kommt es mitunter zu fehlerhaften Aufzeichnungen. Dies kann verschiedene Gründe haben:

- Ausfall der Soundkarte
- Falsche Einstellungen
- Neue/fehlerhafte Hardware
- Mangelnde Kompatibilität
- Hohe Auslastung
- Stromausfall
- Verbindungsprobleme

Solche Fehler müssen vermieden werden, um die hohen Qualitätsstandards von LinkedIn Austria einhalten zu können.

Ein weiterer Punkt ist, dass es für die Dozenten aufgrund des straffen Zeitplans sehr schwierig ist, eine große Anzahl an Videos komplett neu aufzuzeichnen. Durch die Zeitverzögerung in der Videokontrolle können Fehler erst sehr spät erkannt werden.

Bei LinkedIn Austria werden Trainings in verschiedenen Sprachen aufgezeichnet. Dies führt zu einer weiteren Problemstellung. Die Content-Teams der verschiedenen Sprachdepartments haben historisch bedingt jeweils eigene Workflows für die Indexerstellung (siehe Glossar) sowie den Feedbackprozess entwickelt.

1.3 Zielsetzung

An dieser Stelle werden die Ziele und die zu erfüllenden Kriterien dieser Arbeit näher erläutert.

Das Ziel dieser Arbeit ist, die fehlerhaften Video-Files so schnell wie möglich zu erkennen und richtige Gegenmaßnahmen einzuleiten. Die Kontrolle der Files soll möglichst im Hintergrund stattfinden und den Trainer nicht bei den Aufzeichnungen stören.

Ein weiteres erklärtes Ziel ist, dass die Oberfläche und sämtliche Anleitungen in Englisch verfasst werden.

Darüber hinaus soll diese Arbeit auch den gesamten Workflow des Content-Teams beschleunigen und vereinfachen. Hierbei liegt ein besonderes Augenmerk auf der Bedienbarkeit. Die gesammelten Daten sollen grafisch dargestellt werden.

Leitideen der vorliegenden Arbeit betreffen insbesondere die Qualitätserhöhung, die Workflow-Beschleunigung und die Arbeitserleichterung.

Ein wesentliches Ziel dieser Arbeit ist auch, die eingegeben Daten zu verwalten und zu speichern.

1.4 LinkedIn-Qualitätskriterien eines Video-Trainings

Bei LinkedIn Austria werden zwei Dateitypen unterschieden – .mov-Files und .avi-Files. mov-Files werden mit iShowU unter Mac OSX aufgezeichnet, avi-Files mit Bandicam unter Windows.

Die aktuellen Qualitätsanforderungen von LinkedIn für .mov- und .avi-Files:

- Für die vorgegebene Auflösung von 16:9 werden folgende Auflösungen akzeptiert:
 - **1280 x 720** Pixel – dies ist die Standardauflösung.
 - **1360 x 768** Pixel
 - **1366 x 768** Pixel
 - **1600 x 900** Pixel
 - **1920 x 1080** Pixel

Die erstgenannte Auflösung hat sich als Standard etabliert. Die weiteren gelisteten Auflösungen können in Sonderfällen ebenso verwendet werden. Alle anderen Auflösungen werden nicht akzeptiert.

- Die **Frame Rate** (Bilder pro Sekunde) bei .avi-Files muss mindestens **25** betragen, bei .mov-Files mindestens **15** (bei iShowU ist die Frame Rate variabel). Die Frame Rate oder auch Bildfrequenz ist ein Maßstab für die Qualität einer Aufnahme. Das menschliche Gehirn nimmt ab etwa 14 bis 16 Bildern pro Sekunde (individuell verschieden) aufeinanderfolgende Bilder als bewegte (aber nicht unbedingt ruckelfreie) Szene wahr. Bei der Wahl der Frame Rate ist darauf zu achten, dass eine fließende Aufnahme entsteht. Die Wahl der Frame Rate hat einen unmittelbaren Einfluss auf den Speicherbedarf. Je höher die Frame Rate ist, desto mehr Speicher wird benötigt. Bei iShowU ist die Frame Rate variabel. Das bedeutet: Ist auf dem Bildschirm einiges an Bewegung vorhanden, wird die Frame Rate automatisch erhöht. Dabei darf sie allerdings nicht unter 15 liegen.
- Die **Sampling Rate** muss mindestens **48 kHz** betragen. Als Sampling Rate wird in der Signalverarbeitung die Häufigkeit bezeichnet, mit der ein Analogsignal, bei LinkedIn Austria die Stimme des Trainers, abgetastet wird. Bei einer Abtastrate von 48 kHz wird das analoge Signal 48.000 Mal pro Sekunde abgetastet. Für die Audionachbearbeitung ist es wichtig, genug „Information“ im

Audio-File zu haben. Die gewählten 48 kHz haben sich für die Audioanpassung als optimal herausgestellt. Höhere Abtastraten führen zu keiner merkbar besseren Audioqualität und verursachen unnötig größere Audio-Files. (Je größer die Sampling Rate, desto größer ist der benötigte Speicherplatz.)

- Es muss wenigstens **ein Audio-Track** mit mindestens **zwei Spuren** vorhanden sein. Dies ist nötig, weil neben der Sprachaufzeichnung im Studio auch Systemsound mitaufgezeichnet wird.
- Für das Audioformat wird **AAC** oder **PCM** gewählt. Beide Formate – die **Puls-Code-Modulation** und das **Advanced Audio Coding** – sind Audiodatenkompressionsverfahren. PCM und AAC haben sich in der Audiotechnik als Standard etabliert. Aus diesem Grund finden andere Kompressionsverfahren keine Berücksichtigung.
- Als Video-Codec bei .avi-Files wird **tscc** eingesetzt. Dieser Codec wurde speziell für die Bildschirmaufnahme entwickelt und wird bei LinkedIn als bevorzugter Codec verwendet.
- Die PCM-Bitrate muss mindestens **1400 Kbps** betragen, bei AAC **150 Kbps**. Die Bitrate bezeichnet das Verhältnis einer Datenmenge zu einem Zeitraum, typischerweise gemessen in Bit pro Sekunde. Für eine akzeptable Qualität darf die Bitrate bei den beiden Audioformaten die angegebenen Werte nicht unterschreiten.
- Der Audio-Track und das Video dürfen voneinander höchstens **200 Millisekunden** abweichen. Sonst wird die Aufnahme als asynchron gewertet. Anhand dieses Kriteriums wird sichergestellt, dass die Videos und Audiotracks auch synchron sind. Das gezeigte Bild muss synchron mit dem gesprochenen Text sein.

In der nachstehenden Tabelle werden die LinkedIn-Qualitätskriterien zusammengefasst. Dabei wird der Unterschied zwischen .avi-Files und .mov-Files nochmals verdeutlicht. Zusätzlich wird auch der Status der Signalleuchte im video-detective verdeutlicht.

Qualitätskriterium	.avi-File	.mov-File	Signalleuchte
Auflösungs-Verhältnis	16:9	16:9	grün
Standard-Auflösung	1280x720	1280x720	grün
Mögliche-Auflösung	1360x768	1360x768	gelb
Mögliche-Auflösung	1366x768		gelb rot
Mögliche-Auflösung	1600x900	1600x900	gelb
Mögliche-Auflösung	1920x1080	1920x1080	gelb
Alle anderen Auflösungen			rot
Frame Rate	min. 25 fps	min. 15 fps	grün
Frame Rate darunter			rot
Sampling Rate	min. 48 kHz	min. 48 kHz	grün
Sampling Rate darunter			rot
Audio-Track	min. 2 Spuren	min. 2 Spuren	grün
Audio-Track darunter			rot
Audioformat	PCM/AAC	PCM	grün
Anderere Audioformate			rot
Video-Codec	tsc	H.264	grün
Anderer Video-Codec			rot
Bitrate	min. 1400 Kbps/ min.150 Kbps	min. 1400 Kbps	grün
Bitrate darunter			rot
Video/Audio Synchronität	max. 200 ms	200 ms	grün
Video/Audio Synchronität größer			rot

Tabelle 1 Qualitätskriterien LinkedIn

1.5 Kapitelübersicht

Die Diplomarbeit besteht aus neun Kapiteln.

Nach der allgemeinen Einleitung des ersten Kapitels wird im **Kapitel 2** der Stand der Technik erläutert. Hierbei werden die verschiedenen Software-Alternativen aufgezeigt und ihre einzelnen Schwächen und Stärken erörtert.

Anschließend werden im **Kapitel 3** verschiedenen Programmiersprachen thematisiert. Mit Hilfe eines Kriterienkatalogs wird dabei die geeignete Sprache ermittelt.

Kapitel 4 befasst sich mit der grundlegenden Struktur und Vorgehensweise der Programmierung. Dazu wird auf Basis des Pflichtenheftes aus softwaretechnischer Sicht die Software programmiert.

Im **Kapitel 5** wird ein weiteres technisches Hilfsmittel erläutert, das sogenannte FFmpeg. Zusätzlich wird auch der Einsatz und die Relevanz im video-detective beschrieben.

Das **6. Kapitel** gibt einen kleinen Überblick über die Indexerstellung und dessen Implementierung im Projekt.

Kapitel 7 beschäftigt sich mit dem gesamten Anwendungsprozess des video-detective. Hierbei wird unter anderem auch die Wichtigkeit des Index näher erläutert, sowie die komplette Bedienoberfläche.

Im **Kapitel 8** wird die Fehlermeldung und Fehlerbehebung thematisiert. Dabei wird auf Basis der Qualitätskriterien die Fehlerbehebung erstellt.

Schließlich werden im **Kapitel 9** die Resultate der Diplomarbeit zusammengefasst. Zusätzlich wird ein Ausblick auf mögliche Weiterentwicklungen gegeben.

2 Stand der Technik

In diesem Kapitel wird Bezug genommen auf den aktuellen Stand der Technik und die möglichen Software-Alternativen erläutert.

Die Idee, eine spezielle Software zum Kontrollieren der Video-File-Einstellungen zu entwickeln, ist zwar naheliegend, allerdings verläuft eine Recherche nach einschlägigen Lösungen wenig ergiebig. Der video-detective soll direkt auf die firmeninternen Qualitätsstandards zugeschnitten sein. Solch ein Tool zu finden, ist sehr schwierig. Dennoch ergab die umfangreiche Recherche folgende Alternativen, die im nachfolgenden Abschnitt erläutert werden.

2.1 Alternative Software zur Kontrolle von Video-Settings

Die erste Alternative trägt den Namen **FLVCheck**. Mit ihr ist es möglich, fehlerhafte Flash-Video-Dateien zu identifizieren, die in Verbindung mit dem Flash Media Server Probleme machen könnten. Grund dafür sind z.B. fehlerhafte Metadaten, ungültige Timestamps, ungültiger Header oder defekte Audio-/Videodaten in den Videos. Mit FLVCheck kann also überprüft werden, ob mit Encodern von anderen Herstellern erstellte Flash-Video-Dateien (.flv) sich an die Dateispezifikationen halten.¹

Vorteile:

- Die Darstellung der einzelnen Auflösung ist sehr komfortable. Grundsätzlich besteht die Möglichkeit, nur Auflösungen mit einem Seitenverhältnis von 16:9 anzuzeigen.
- Die Unterscheidung zwischen einer festen Framerate und einer variablen Framerate ist einfach zu erkennen.
- Es können mehrere Audiospuren gleichzeitig überprüft werden.

Nachteile:

- Der Fokus dieser Alternative liegt hauptsächlich auf Flash-Videos. Es ist dementsprechend nicht möglich, die hausinternen Aufnahmen (.avi, .mov) zu kontrollieren.

¹ Vgl. [www.video-flash.de](http://www.video-flash.de/index/flvcheck-tool/) Alles rund um Web & Mobile: FLVCheck (siehe: <http://www.video-flash.de/index/flvcheck-tool/>), 2007.

- Grundsätzlich werden die fehlerhaften Dateien nur erkannt, eine Problemlösung wird dagegen nicht angeboten.
- Darüber hinaus ist es nicht möglich, die zu kontrollierenden Kriterien auf die gesamten firmeninternen Qualitätsstandards anzupassen.
- Eine Indexerstellung ist nicht möglich.

Die nächste Alternative trägt den Namen **VideoInspector**. Der kostenlose „VideoInspector“ informiert über alle Details einer Videodatei und insbesondere über die Codecs, die zum Abspielen erforderlich sind. So lassen sich Fehler leichter finden und Probleme bei der Wiedergabe gezielt beheben. Auch Angaben zu Dateigröße, Dauer, Auflösung, Framerate und Bitrate können mit der Software in Erfahrung gebracht werden. Das Tool unterstützt Formate wie AVI, Matroska (MKV), Quicktime, MPEG-1 und MPEG-2 (MPG).²

Vorteile:

- Die Überprüfung von .avi-Files ist möglich.
- Das Erkennen der unterschiedlichen Fehler erfolgt in der Regel sehr schnell.
- Das Audioformat wird auf einem Blick sichtbar, die Überprüfung mehrerer Audiospuren ist dabei ebenso möglich.
- Die Fehlererkennung findet im Hintergrund statt.

Nachteile:

- Es ist nicht möglich, eine automatische (zeitgesteuerte) Kontrolle durchzuführen.
- Es ist nicht möglich, die zu kontrollierenden Kriterien auf die firmeninternen Qualitätsstandards anzupassen (es werden immer alle Informationen dargestellt).
- Die Videos müssen alle einzeln darauf überprüft und beurteilt werden, ob ein etwaiger Fehler vorliegt (kein einfaches Ampelsystem).
- Die Indexerstellung ist nicht möglich.
- Die fehlerhaften Dateien werden zwar erkannt, allerdings wird keine Problemlösung angeboten.

² Vgl. VideoInspector Video Playing Trouble Annihilated VideoInspector (siehe: <https://www.kcsoftwares.com/?vtb>), 2017.

Eine weitere Alternative stellt die Software **GSpot** da. GSpot hilft bei der schnellen Fehlersuche, wenn Videos nicht korrekt abgespielt werden. Nicht jedes Video läuft auf jedem PC. Die Ursache liegt meist in unterschiedlichen Versionen der Codecs, mit denen der Film komprimiert wurde und die auf dem Windows-System installiert sind. Das kleine Gratis-Tool GSpot hilft bei der schnellen Analyse von Videodateien und listet alle für das korrekte Abspielen benötigten Codecs und DirectShow-Filter auf. Die Freeware kommt ohne Installation aus und ist nach dem Download sofort startklar.³

Vorteile:

- Die Software kann ohne Installation gestartet werden (schnelle Inbetriebnahme).
- Die Darstellung der unterschiedlichen Codecs erfolgt mit verschiedenen visuellen Effekten.
- Zusätzlich ist das Erkennen der Synchronität einer Aufnahme möglich.
- Die Oberfläche ist auch in Englisch verfügbar

Nachteile:

- Die Indexerstellung ist nicht möglich.
- Die zu kontrollierenden Kriterien lassen sich nicht an die firmeninternen Qualitätsstandards anpassen (es werden immer alle Informationen dargestellt).
- Die Videos müssen alle einzeln überprüft und beurteilt werden.
- Die Fehler werden zwar erkannt, allerdings werden keine Lösungsansätze angeboten.

³ Vgl. Chip.de GSpot Version 2.70a (siehe: http://www.chip.de/downloads/GSpot_13008930.html), 2017.

Die nachfolgende Tabelle fasst die Vorteile und die Nachteile der Alternativen Softwarevorschläge zusammen.

FLVCheck	Vorteile:	Nachteile
	<ul style="list-style-type: none"> • Darstellung der Auflösungen komfortable • Mehrere Audiospuren können gleichzeitig überprüft werden • Unterscheidung: feste/variable Framerate • Kostengünstig 	<ul style="list-style-type: none"> • Hauptsächlich Flash-Videos • Keine Problemlösung • Keine firmeninternen Qualitätsstandards • Keine Indexerstellung
Videolnspector	Vorteile:	Nachteile:
	<ul style="list-style-type: none"> • Keine Installation nötig • .avi-Files können überprüft werden • Fehlererkennung im Hintergrund • Einfache Bedienung • Updates im Preis inbegriffen 	<ul style="list-style-type: none"> • Keine firmeninternen Qualitätsstandards • Keine automatisierte Kontrolle • Nur Einzel-Kontrolle möglich • Keine Indexerstellung • Keine Problemlösung
GSpot	Vorteile:	Nachteile:
	<ul style="list-style-type: none"> • Englische Oberfläche verfügbar • Darstellung der Codecs, visuelle Effekte • Synchronität erkennbar • Keine Installation nötig 	<ul style="list-style-type: none"> • Keine Indexerstellung • Keine firmeninternen Qualitätsstandards • Nur Einzel-Kontrolle möglich • Keine Problemlösung

Tabelle 2 Vorteile und Nachteile Alternativer Software Lösungen

2.2 Eingeschlagener Weg zur Realisierung

Die möglichen Alternativen sind sehr einschlägig und für die Abdeckung aller firmeninternen Anforderungen nicht zu gebrauchen. Aufgrund dieser Analyse beziehungsweise Recherche wurde eine Software entwickelt, die alle firmeninternen Qualitätsstandards abdeckt und darüber hinaus einen neuen Workflow erschafft. Die Vorteile einer eigenen Software-Alternative liegen dabei klar auf der Hand:

- Die Software wird anhand der Anforderungen entwickelt (Steigerung der Effizienz).
- Die Software kann jederzeit angepasst und erweitert werden.
- Entstehung eines einheitlichen Workflows in Bezug auf die verschiedenen Sprachen
- Die Firma ist unabhängig von anderen Software-Herstellern (Kosten).
- Die Software-Tests erfolgen direkt mit dem schlussendlichen Benutzer. Dadurch ist in weiterer Folge ein nahtloser Austausch der neuen Alternative möglich.

Die Wahl einer geeigneten Programmiersprache ist für das Projekt und dessen Anforderungen entscheidend. Im nächsten Punkt wird genau auf diese Frage Bezug genommen.

3 Programmiersprache grundlegende Überlegungen

An dieser Stelle werden die Vor- und Nachteile diverser Programmiersprachen gegenübergestellt. Der Vergleich dient dazu, die richtige Wahl für die genannten Anforderungen zu treffen. Dabei spielen nicht nur Effizienz und Erweiterbarkeit eine wesentliche Rolle, sondern auch die Standardbibliothek und die Entwicklerumgebung. Folgende Entwicklungssprachen kamen für die Entwicklung der Software in Frage:

Java ist eine objektorientierte Programmiersprache, sie ist plattformunabhängig und einfach strukturiert. Die Programmiersprache Java dient innerhalb der Java-Technologie vor allem dem Formulieren von Programmen. Diese liegen zunächst als reiner, für Menschen verständlicher Text vor, als sogenannter Quellcode. Dieser Quellcode ist nicht direkt ausführbar. Erst der Java-Compiler als Teil des Entwicklungswerkzeugs übersetzt ihn in einen maschinenverständlichen Code, den sogenannten Java-Bytecode. Die Maschine, die diesen Bytecode ausführt, ist jedoch typischerweise virtuell – das heißt, der Code wird meist nicht direkt durch Hardware (etwa einen Mikroprozessor) ausgeführt, sondern durch entsprechende Software auf der Zielplattform. Zweck dieser Virtualisierung ist die Plattformunabhängigkeit: Das Programm soll ohne weitere Änderung auf jeder Rechnerarchitektur laufen können, wenn dort eine passende Laufzeitumgebung installiert ist.⁴

- Vorteile:
 - Weite Verbreitung (verschiedenste Computersysteme)
 - Umfangreiche Klassenbibliothek
 - Programmcode auf jeder Plattform lauffähig, sofern Java VM vorhanden
 - Applets (Applikationen über das Internet verfügbar)
 - Eclipse-Plattform (Werkzeugunterstützung)
 - Industriestandard⁵
- Nachteile:
 - Java-Programme benötigen eine Laufzeitumgebung.
 - Java-Applet-Laufzeitumgebungen sind häufig veraltet.
 - Für systemnahe Programme nicht geeignet⁵

⁴Vgl. wikipedia. Java (Programmiersprache) (siehe: [https://de.wikipedia.org/wiki/Java_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Java_(Programmiersprache))), 2017.

⁵Vgl. wikibooks Java Standard (siehe: https://de.wikibooks.org/wiki/Java_Standard:_Einleitung), 2016.

C++ kann als traditionelle prozedurale Programmiersprache mit einigen zusätzlichen Funktionen gesehen werden. Jedes gut geschriebene C++-Programm wird sowohl Elemente des objektorientierten Programmierens als auch solche des klassischen prozeduralen Programmierens enthalten. Im Gegensatz zu “modernen” Programmiersprachen wie Java und C# existieren in C++ einige Konzepte wie manuelle Speicherverwaltung und Zeigerarithmetik. Die Sprache C++ ist grundsätzlich und theoretisch plattformunabhängig.⁶

- Vorteile:
 - Großes Anwendungsgebiet
 - Viele zusätzliche Librarys
 - Compiler für fast jedes OS
 - Const-Correktness
 - Mehrfachvererbung
 - Templates
 - Nativ kompilierend
 - OOP und strukturiertes Programmieren möglich⁷
- Nachteile:
 - Die Sprache ist sehr veraltet.
 - Kleine Standardbibliothek
 - Teilweise komplizierte Syntax
 - Manuelle Speicherverwaltung nötig
 - Erstellung von GUIs (Graphical User Interfaces) aufwendig
 - Nicht standardisiertes name-mangling
 - Durch die Möglichkeit direkter Speicherzugriffe entsteht schnell ein Buffer Overflow.⁷

⁶ Vgl. ProgrammierBot C++ Programmieren lernen – Das Tutorial für Anfänger (siehe: <http://www.programmierenlernen24.de/c-programmieren-lernen/>), 2017.

⁷ Vgl. overblog C++ vs Java: Vor- und Nachteile (siehe: https://de.over-blog.com/C_vs_Java_Vor_und_Nachteile-1228321776-art185420.html/), 2011.

C# ist eine von der Firma Microsoft für das .NET-Framework entwickelte und von der European Computer Manufacturers Association (ECMA1) standardisierte Programmiersprache, die auf den Vorbildern C++ und Java aufbaut, aber auch etliche Weiterentwicklungen bietet. Die .NET-Plattform etabliert sich allmählich als Standard bei der Software-Entwicklung unter Windows und das Open-Source-Projekt Mono hat die Plattform erfolgreich auf andere Betriebssysteme (Android, Linux, MacOS-X, UNIX) portiert.⁸

- Vorteile:
 - Wird aktiv weiterentwickelt
 - Sehr große Standard-Library vorhanden, allgemein ist .NET ziemlich riesig.
 - Fertige Programme haben generell eine sehr geringe Dateigröße.
 - OOP (Einfachvererbung, Mehrfachvererbung nur mit Interfaces)
 - Echte generische Typen möglich bzw. vorhanden
 - ECMA-Standard
 - Plattformunabhängig, solange es .NET oder eine Implementierung davon (Mono) auf der Plattform gibt, außer plattformspezifische Funktionen werden benutzt
 - C#-Objekte erfordern weder Header- oder IDL-Dateien noch Typbibliotheken.
 - C# bietet eine einfache, sichere und intuitive Umgebung.
 - C-Code in DLL-Dateien kann über den C#-Code aufgerufen werden.
 - C# bietet Low-Level-Zugriff (bei dem der Einsatz von Zeigern möglich ist, wenn die Leistung eine besonders große Rolle spielt oder Zeiger für vorhandene DLLs benötigt werden).
 - C# wurde für die Entwicklung robuster, langlebiger Komponenten konzipiert, mit denen Situationen des wirklichen Lebens bewältigt werden können.⁹
- Nachteile:
 - Nicht die gleiche Hardwarenähe wie C++
 - .NET muss auf der Zielplattform vorhanden sein.

⁸ Vgl. Bernhard Baltes-Götz Einführung in das

Programmieren mit C# 4.0 (siehe: <https://www.uni-trier.de/fileadmin/urt/doku/csharp/v40/csharp4.pdf>), 2011.

⁹ Vgl. Eric Gunnerson C# Die neue Sprache für Microsofts .NET-Plattform (siehe: <http://openbook.rheinwerk-verlag.de/csharp/intro.htm>), 2002.

- Nicht komplett kompiliert, wird bei Programmstart noch mal übersetzt (ca. 5% Leistungsabfall gegenüber C++)¹⁰

3.1 Vor- und Nachteile der einzelnen Programmiersprachen

Die nachfolgende Tabelle fasst die Vorteile und die Nachteile der einzelnen Programmiersprachen zusammen.

Java	Vorteile:	Nachteile:
	<ul style="list-style-type: none"> • Weite Verbreitung • Umfangreiche Klassenbibliothek • Auf jeder Plattform lauffähig • Applets • Eclipse-Plattform • Industriestandard 	<ul style="list-style-type: none"> • benötigt Laufzeitumgebung • Applet-Laufzeitumgebungen häufig veraltet • Systemnahe Programme nicht geeignet
C++	Vorteile:	Nachteile:
	<ul style="list-style-type: none"> • Großes Anwendungsgebiet • Viele zusätzliche Librarys • Compiler für fast jedes OS • Const-Correktness • Mehrfachvererbung • Templates • Nativ kompilierend • Strukturiertes Programmieren 	<ul style="list-style-type: none"> • Sprache veraltet. • Kleine Standardbibliothek • Komplizierte Syntax • Manuelle Speicherverwaltung nötig • Erstellung von GUIs aufwendig • Nicht standardisiertes name-mangling • Buffer Overflow.
C#	Vorteile:	Nachteile:
	<ul style="list-style-type: none"> • Wird weiterentwickelt • Große Standard-Library • Programme geringe Dateigröße. • OOP • Echte generische Typen möglich • ECMA-Standard • Plattformunabhängig, • C#-Objekte erfordern weder Header- oder IDL-Dateien noch Typbibliotheken. • Bietet einfache/sichere Umgebung. • Low-Level-Zugriff • Robuster, langlebig 	<ul style="list-style-type: none"> • Nicht so Hardwarenah wie C++ • NET muss vorhanden sein • Nicht komplett kompiliert, wird noch mal übersetzt

Tabelle 3 Vorteile und Nachteile verschiedener Programmiersprachen

¹⁰ Vgl. Eric Gunnerson C# Die neue Sprache für Microsofts .NET-Plattform (siehe: <http://openbook.rheinwerk-verlag.de/csharp/intro.htm>), 2002.

3.2 Die Gewählte Programmiersprache

In diesem Abschnitt wird die gewählte Programmiersprache näher erläutert. Aufgrund dessen wurde eine Entscheidungsmatrix erstellt. Zuerst werden die Auswahlkriterien bestimmt. einer der wichtigsten Voraussetzung für die Auswahl der Programmiersprache ist dabei die Standard-Bibliothek. Genau aus diesem Grund, hat dieses Kriterium auch die höchste Gewichtung (siehe Tabelle: Entscheidungsmatrix Programmiersprache) Ein weiteres sehr wichtiges Kriterium ist die Möglichkeit DLL-Dateien einbinden zu können (Grundlegende Programmierung media_info). Die Entwicklerumgebung ist das Kriterium mit der drittgrößten Gewichtung. Zusätzlich sind noch Kriterien wie, einfache Erstellung der Benutzeroberfläche, Plattformunabhängigkeit und Weiterentwicklung der Programmiersprache relevant. Diese drei Punkte werden dabei jeweils mit gleicher Gewichtung betrachtet. In der nachstehenden Tabelle wird die unterschiedliche Wichtigkeit der einzelnen Kriterien verdeutlicht:

Kriterium	Gewichtung	Java		C++		C#	
		Punkte		Punkte		Punkte	
Standard-Bibliothek	30%	5	1,5	2	0,6	6	1,8
Entwicklerumgebung	15%	4	0,6	6	0,9	6	0,9
Erstellung Benutzeroberfläche	10%	3	0,3	3	0,3	4	0,4
Einbindung DLL-Dateien	25%	3	0,75	3	0,75	6	1,5
Plattformunabhängig	10%	6	0,6	6	0,6	4	0,4
Weiterentwicklung	10%	3	0,3	2	0,2	6	0,6
Summe	100%	24	4,05	22	3,35	32	5,6

Tabelle 4 Entscheidungsmatrix Programmiersprache

Gewichteten Punkte

Die Maximale Punktezahl die erreicht werden kann beträgt sechs Punkte. In der Tabelle wird deutlich, dass zum Beispiel Java (5 Punkte) eine größere Standard-Bibliothek besitzt als C++ (2 Punkte). Des Weiteren ist Java sowie C++ plattformunabhängig, deswegen wurden bei diesem Kriterium 6 Punkte vergeben. C# hingegen erreicht die Plattformunabhängigkeit nur mit .Net. In weiterer Folge bedeutet dies, je höher die Punktezahl desto eher trifft das jeweilige Kriterium zu. Zusätzlich wird noch die Gewichtung jedes Kriteriums ermittelt. Die Formel hierzu lautet: $\text{Gewichtung}(\%) / 100 \times \text{Punkte} = \text{Gewichteten Punkte}$. Die Gewichtung ist sinnvoll, wenn nicht alle Kriterien dieselbe Wichtigkeit oder Bedeutung haben. Aus

der Tabelle leitet sich die Programmiersprache C# als Alternative mit dem höchsten Gesamtergebnis ab (Gesamtpunkte 32 – Gewichteten Punkte 5,6). Aufgrund dessen stellte sich C# als geeignetste Programmiersprache für das Vorhaben heraus. Vor allem die Entwicklerumgebung (Visual Studio) und die große Bibliothek waren entscheidende Punkte.

Die einfache Erstellung der grafischen Benutzeroberfläche war ebenso ein sehr wichtiges Kriterium.

Des Weiteren ist es möglich, das fertiggestellte Projekt in eine webbasierende Form umzuwandeln (das wird in einer weiteren Ausbaustufe eine Rolle spielen, wird hier allerdings nicht weiter erläutert).

Ebenso sind die Einbindung und die Nutzung von vorgefertigten DLL-Dateien wie MediaInfo ein sehr positiver Aspekt. Mit der MediaInfo.dll können alle Informationen zu Video- und Audioformaten ausgelesen und angezeigt werden. Die Funktionsweise wird im Punkt: Grundlegende Programmierung media_info näher erläutert.

Ein letzter entscheidender Grund der ebenso für C# spricht ist die Unternehmensfusion, LinkedIn ist seit November 2016 Teil von Microsoft.

Im nachfolgenden Punkt wird die grundlegende Struktur und Vorgehensweise der Programmierung näher beschrieben.

4 Grundlegende Struktur und Vorgehensweise

An dieser Stelle wird die grundsätzliche Struktur der Kontrolle näher erläutert. Vor der Programmierung des video-detective waren Überlegungen nötig, was das Programm notwendigerweise erfüllen muss, was es erfüllen kann und wie es abgegrenzt ist.

4.1 Musskriterien

Das Programm hat nachfolgende Musskriterien zu erfüllen:

- Die Kontrolle darf den Trainer nicht stören.
- Es muss sichergestellt werden, dass pro Fehler und Zyklus nur eine Attention-Mail versendet wird.
- Das Attention-Dokument darf nur einmal pro Fehler im Verzeichnis erzeugt werden.
- Die Kontrolle mehrerer Projekte muss möglich sein.
- Es muss der Inhalt aller Unterordner kontrolliert werden.
- Der video-detective muss multiuserfähig sein.
- Die Kontrolle darf nicht zu viel Zeit in Anspruch nehmen.
- Die beiden Lämpchen (audio und video) dürfen nur bei bestimmten Ereignissen die Farbe ändern.
- Es muss immer genau das richtige Attention-Dokument generiert werden.
- Bei der automatisierten Kontrolle müssen die Videos richtig zugeordnet werden, auch wenn sich der Name des Videos ändert.
- Die Benutzeroberfläche muss in Englischer Sprache gehalten werden.
- Eine Auswertung der Daten muss möglich sein (grafische Darstellung).
- Es muss jederzeit eine Anpassung der Kriterien möglich sein.
- Die eingefügten Projekte müssen auch wieder gelöscht werden können.
- Der video-detective wird von verschiedenen Bereichen im Unternehmen genutzt, somit muss eine Funktionsabgrenzung vorhanden sein.
- Die gesammelten Informationen müssen zwischen den Benutzern austauschbar sein.

4.2 Wunschkriterien

Die Wunschkriterien sehen wie folgt aus:

- Die Benutzer können die Farbe der Kommentarzeile ändern.
- Die Benutzer können mehrere E-Mail-Adressen einfügen.
- Die E-Mail-Adresse kann automatisch vom angemeldeten Benutzer ausgelesen werden.
- Das Video kann mit einem Klick auf die Zelle angesehen werden.

4.3 Abgrenzungskriterien

Folgende Abgrenzungskriterien waren für die Erstellung relevant:

- Es werden nur Videodateien mit der Endung .mov, .avi oder .mp4 zur Kontrolle herangezogen.
- Die Kontrolle kann nur auf einem Windowssystem durchgeführt werden.

4.4 Notwendige Funktionen und Teilfunktionen

Anhand dieser Kriterien wurden folgende Funktionen und Eigenschaften dem video-detective hinzugefügt:

- Kontrolle im Hintergrund: Die Kontrolle findet automatisiert im Hintergrund des laufenden Systems statt. Die Aufnahme wird dabei in keinsten Weise beeinträchtigt. Zusätzlich ist es möglich die Kontrolle auch auf einem anderen System durchzuführen.
- Pro Fehler nur eine Fehlermeldung: Hierfür wurden verschiedene Arraylisten angelegt, befindet sich der Fehler bereits in der Arrayliste, wird keine weitere Fehlermeldung angezeigt. Die Fehlermeldung wird im Punkt: Fehlermeldung und Fehlerbehebung des video-detective genauer erläutert.
- Kontrolle mehrerer Projekte: Über die grafische Benutzeroberfläche können mehrere Projekte gleichzeitig eingefügt und überprüft werden. Das Einfügen wird über eine TextBox oder einem FolderBrowserDialog ermöglicht. Diese werden im Punkt: Der linke Teil des video-detective erörtert.
- Multiuserfähig: Der video-detective kann auf jedem beliebigen System installiert werden, die Anzahl der Benutzer spielt dabei keine Rolle.

- Geringer Zeitaufwand pro Kontrolle: Der video-detective speichert alle bereits kontrollierten Video-Files, somit muss nicht jedes File mehrmals kontrolliert werden. Dadurch wird die automatisierte Kontrolle immens beschleunigt. Ein Einblick in die Programmierung hierfür, wird im Punkt: Grundlegende Programmierung media_info gegeben.
- Grafische Darstellung der Fehler: Für die visuelle Darstellung der fehlerhaften Video-Files wurde ein einfaches Ampelsystem eingeführt. Dieses Ampelsystem wird im Kapitel: LinkedIn-Qualitätskriterien eines Video-Trainings verdeutlicht.
- Korrekte Fehlermeldung: Die Video-Files werden anhand der Filegröße in Bit und des Erstellungsdatums genau identifiziert. Somit ist eine exakte Unterscheidung der einzelnen Videos möglich, auch wenn sich der Name des Video-Files ändert.
- Funktionsabgrenzung: Der video-detective wurde hierfür in drei Teilbereiche gegliedert. Die drei Teilbereiche lauten: control, content und production. Die Funktionen des video-detective ändern sich anhand des ausgewählten Tabs. Die Erläuterung der einzelnen Bereiche findet im Punkt: Funktionen und Aufbau des video-detective statt.
- Auswertung der Daten: Zur Auswertung der Daten wurde eine automatisierte Indexerstellung konzipiert. Dadurch ist es jederzeit möglich einen Ist-Zustand des Projekts abzurufen. Die Indexerstellung wird im Punkt: Indexerstellung eines Video-Trainings verdeutlicht.
- Projekte entfernen: Die hinzugefügten Projekte können mittels der ArrayList jederzeit gelöscht werden. Somit werden nicht relevante Projekte einfach von der Kontrolle ausgeschlossen.
- Informationsaustausch: Die Informationen zum Beispiel Schnittanweisungen werden in einem DataGridView eingetragen. Diese Informationen können mittels der XML-Export- und Import-Funktion des video-detective ausgetauscht werden. Nähere Informationen hierfür werden im Punkt: Import und Export der XML-Datei im video-detective gegeben.

4.5 Hauptfunktionen und Teilfunktionen des video-detective (Skizze)

Anhand der notwendigen Funktionen wurde folgende Skizze entwickelt:

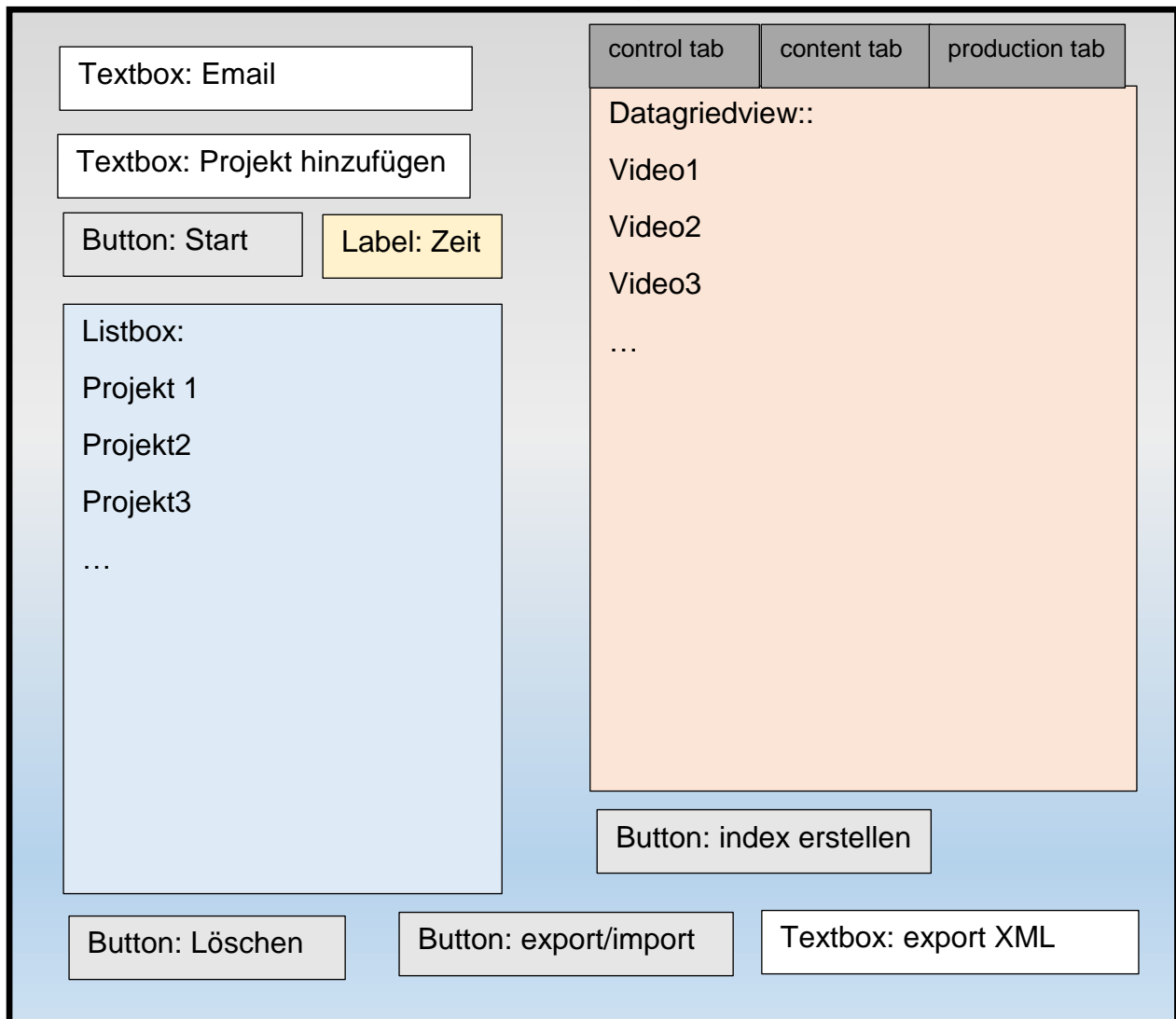


Abbildung 1: Grundlegende Funktionen und Teilfunktionen des video-detective

Auf der Abbildung 1: Grundlegende Funktionen und Teilfunktionen des video-detective werden alle wichtigen Funktionen des video-detective verdeutlicht.

- Die visuelle Darstellung der Daten erfolgt über das Datagridview.
- Zusätzlich ist eine Aufteilung in verschiedenen Bereiche vorhanden.
- Mit Hilfe des Buttons (export/import) wird der Datenaustausch ermöglicht.
- Die Indexerstellung wiederum erfolgt über den Button: index erstellen.
- Die automatisierte Kontrolle wird über den Start-Button gewährleistet, zusätzlich wird dabei die Zeitspanne bis zur nächsten Kontrolle angezeigt.

- Die Kontrolle mehrerer Projekte wird über die Auflistung mittels einer Listbox ermöglicht.
- Das Löschen der Projekte erfolgt über den Löschen-Button. Die Benachrichtigung bei einem Fehler wird automatisch, an die in der Textbox: Email eingegebene Email-Adresse versendet.

Diese Skizze dient als Grundlage für die Entwicklung des video-detective, anhand dessen wurde nachtendes use case diagram (Anwendungsfalldiagramm) konzipiert.

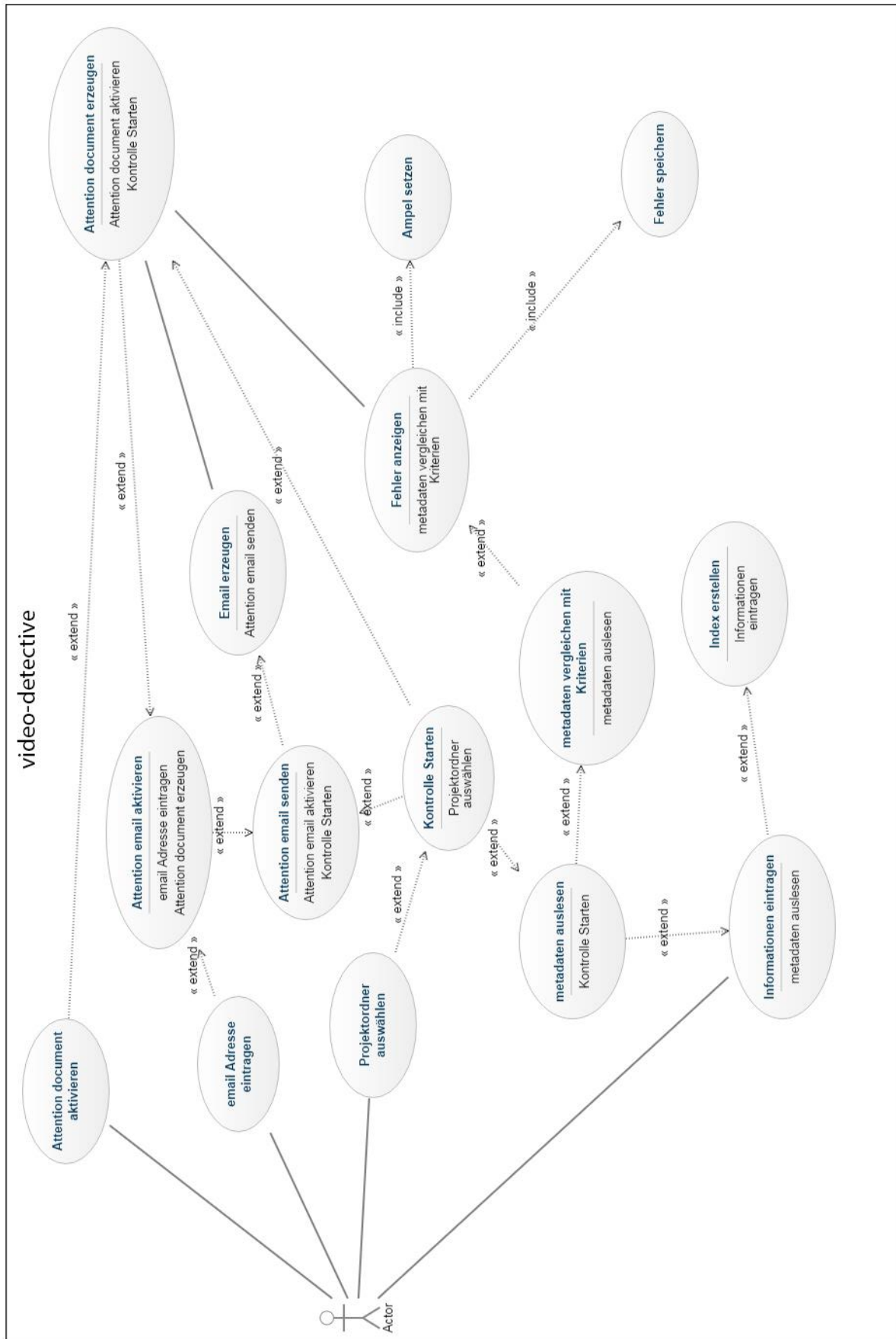


Abbildung 2: Use Case Diagram video-detective

Zu Beginn werden die Daten vom Projektordner initialisiert. Der video-detective überprüft den Ordner auf relevante Video-Files und speichert diese in ein Array. Die gespeicherten Video-Files durchlaufen alle einzeln die Kontrolle. Eine Kontrolle kann wie folgt aussehen:

Zuerst wird überprüft, ob es sich bei der Dateiendung um eine .avi-Datei oder eine .mp4-Datei handelt. Als zweite Instanz wird mittels eines AND-Operators die Framerate überprüft. (Bedingung: Die Framerate muss größer sein als 25 Bilder pro Sekunde.)

```
if (file.Extension == ".avi" && file.Video[0].FrameRate < 25000.0 || file.Extension == ".mp4" && file.Video[0].FrameRate < 25000.0)
```

Als Nächstes wird an dieser Stelle, mittels einer if-Abfrage, die Gültigkeit der Mail-Checkbox überprüft.

```
if (c_mail.Checked == true)
```

Wurde die Checkbox c_mail angehakt, wird ein Attention-Mail generiert:

```
mail("attention incorrect framerate!!!", "the file " + datei + " has the wrong framerate: " + file.Video[0].FrameRate + " \n \n Please look the email notes below, there is an instruction-file to solve the issue.", "attention_wrong_framerate.pdf");
```

Mit der nachstehenden Zeile wird die fehlerhafte Datei in die Arrayliste mail_framerate eingetragen. Somit ist es nicht möglich, den gleichen Fehler bei ein und demselben File noch mal zu erhalten.

```
mail_framerate.Add(datei);
```

Wurde die Checkbox c_textfile angehakt, werden die Attention-Dokumente direkt in den Projektordner kopiert. Der Code dafür sieht wie folgt aus:

```
if (c_textfile.Checked == true)
```

Die nachfolgende Integer-Variable (`check_number`) legt fest, welches Attention-Dokument kopiert bzw. verschickt wird:

```
check_number = 5;
```

Der Kopiervorgang des Attention-Dokuments erfolgt mit der folgenden Zeile:

```
copy_pdf("attention_wrong_framerate.pdf", datei);
```

Anhand der nachstehenden Zeile wird die fehlerhafte Datei in die Arrayliste `mail_framerate` eingetragen. Somit ist es nicht möglich, den gleichen Fehler bei ein und demselben File noch mal zu erhalten.

```
mail_framerate.Add(datei);
```

An dieser Stelle erfolgt nun ein Eintrag in die Arrayliste `video_ok_list`. Diese Aktion dient der Setzung der Kontrollleuchte (`video`):

```
video_ok_list.Add(datei);
```

Zusätzlich wird die zu kontrollierende Videodatei auch in die Arrayliste `ok_path` eingetragen. Dieser Eintrag dient der Setzung der Kontrollleuchte auf der linken Seite des `video-detective`:

```
ok_path.Add(datei.Substring(0, path_lenght));
```

Auf diese Art und Weise lassen sich alle Kriterien bzw. Qualitätsanforderungen abfragen. Die Qualitätsanforderungen werden an anderer Stelle (LinkedIn-Qualitätskriterien eines Video-Trainings) näher erläutert.

5 Grundlegende Programmierung media_info

Als Nächstes erfolgt der Aufruf der Methode media_info. Eigenschaft und Bedeutung dieser Methode wird in diesem Punkt erläutert.

In der Methode media_info werden die einzelnen Arraylisten überprüft. Diese Überprüfung ist notwendig, um festzustellen, ob das Video-File schon mal kontrolliert wurde oder nicht. Die Programmierung hierfür sieht wie folgt aus:

```
if (!mail_resolution.Contains(dateien2))
{
check_resolution(dateien2, aviFile);
}
if (!mail_length.Contains(dateien2))
{
check_length(dateien2, aviFile);
}
if (!mail_framerate.Contains(dateien2))
{
check_framerate(dateien2, aviFile);
}
if (!mail_samplingrate.Contains(dateien2))
{
check_samplingrate(dateien2, aviFile);
}
if (!mail_format.Contains(dateien2))
{
check_format(dateien2, aviFile);
}
if (!mail_audiobitrate.Contains(dateien2))
{
check_audiobitrate(dateien2, aviFile);
}
if (!mail_videocodec.Contains(dateien2))
{
check_videocodec(dateien2, aviFile);
}
```

Die Videos werden mittels einer If-Abfrage einzeln überprüft. Befindet sich eine Videodatei bereits in einer der Arraylisten bedeutet dies, dass der Fehler schon ausgegeben und eine Attention-Mail versandt wurde. Somit ist es nicht möglich, denselben „Fehler“ mehrmals zu erhalten.

5.1 MedialInfo grundlegende Funktionen

Beim Kontrollieren von Video-Files wird schnell die Grenze erreicht, denn das Betriebssystem an sich kann nur sehr wenige Metadaten eines Video-Files auslesen. Deswegen wird ein Tool benötigt, das es ermöglicht, die gesamten Metadaten auszulesen. Nach einer intensiven Recherche fiel die Wahl auf das Open-Source-Tool MedialInfo.

Was ist MedialInfo?

MedialInfo liefert technische und zusätzliche Tag-Informationen über Video- und Audiodateien.

Welche Informationen liefert MedialInfo?

- Grundsätzliches: Titel, Autor, Regisseur, Album, Titelnummer, Datum, Spieldauer ...
- Video: Codec, Bildseitenverhältnis, Bildfrequenz, Bitrate ...
- Audio: Codec, Sampling Rate, Kanäle, Sprache, Bitrate ...
- Text: Sprache der Untertitel
- Kapitel: Anzahl der Kapitel, Kapitelübersicht

Was kann das Programm?

- Liest viele Video- und Audioformate
- Verschiedene Darstellungsmöglichkeiten der Informationen (Text, Tabelle, Baumstruktur, HTML ...)
- Diese Darstellungen können selbst angepasst werden.
- Informationen können als Text, CSV, HTML ... ausgegeben werden.
- Grafische Benutzeroberfläche, Aufruf per Kommandozeile oder über die DLL
- Integration in MS-Windows (Drag&Drop, Kontextmenü)
- Internationalisierung: Jede Sprache wird unter jedem Betriebssystem unterstützt.¹¹

¹¹ Vgl. mediaarea.net, Version 0.7.91, Grafische Benutzeroberfläche, mit Installationsroutine, für Windows (siehe: <https://mediaarea.net/de/MedialInfo>), 2011.

Warum wurde-MediaInfo gewählt?

Der Grund für die Entscheidung für MediaInfo liegt klar auf der Hand. Bei MediaInfo besteht die Möglichkeit, nur die MediaInfo.dll zu verwenden. Das bedeutet, es ist möglich direkt auf die Funktion von MediaInfo über C# zuzugreifen. Das ist ein sehr großer Vorteil, denn nur die MediaInfo.dll-Datei wird ins Projekt eingebunden, der User des video-detective muss dieses Tool nicht extra installieren. Zusätzlich ist das Programm kostenlos.

5.2 Einbinden der MediaInfo-Funktionen in C#

- Zuerst muss die MediaInfo-Bibliothek heruntergeladen werden.
- Danach werden die beiden .dll-Files (MediaInfo.dll, MediaInfoNET.dll) in den debug-Ordner des C#-Projekts kopiert:

<C:\Users\deder\Documents\Visual Studio 2015\messenger\messenger\bin\Debug>

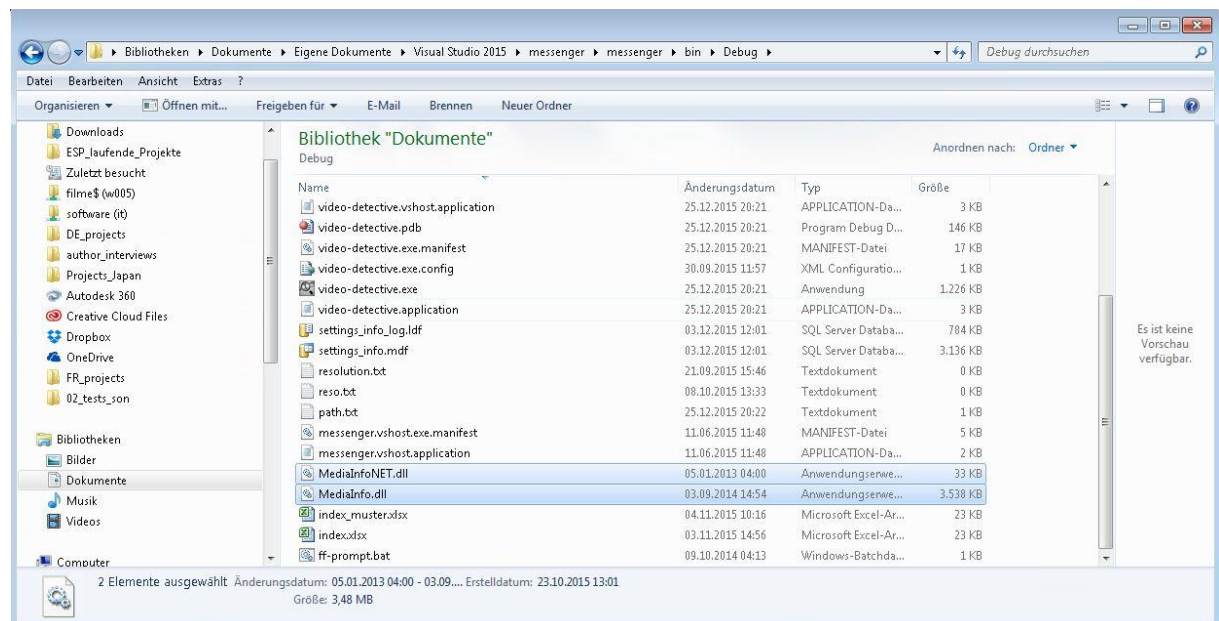


Abbildung 3: MediaInfo in das Projekt einbinden

- Nun müssen die beiden .dll-Files im C#-Projekt in Visual Studio eingebunden werden. Die MediaInfoNET.dll muss als Verweis hinzugefügt werden.

Das Hinzufügen eines Verweises funktioniert in Visual Studio wie folgt:

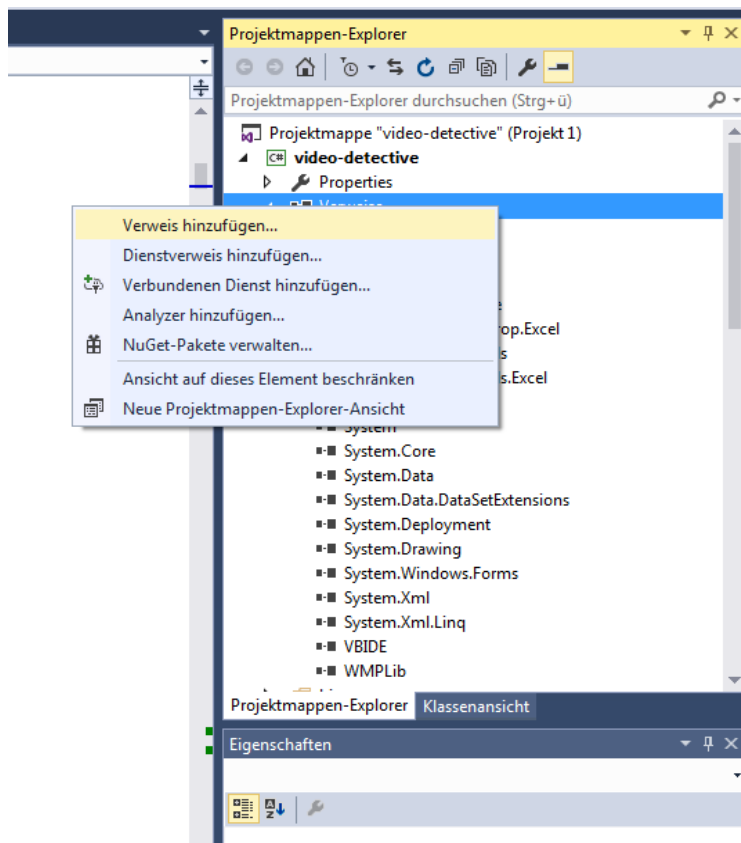


Abbildung 4: MediaInfo in Visual Studio einbinden

Im Projektmappen-Explorer wird zuerst der Unterpunkt „Verweis“ ausgewählt und mit der rechten Maustaste das Dialogfenster geöffnet. Ein Klick auf den Punkt „Verweis hinzufügen“ öffnet den Verweis-Manager:

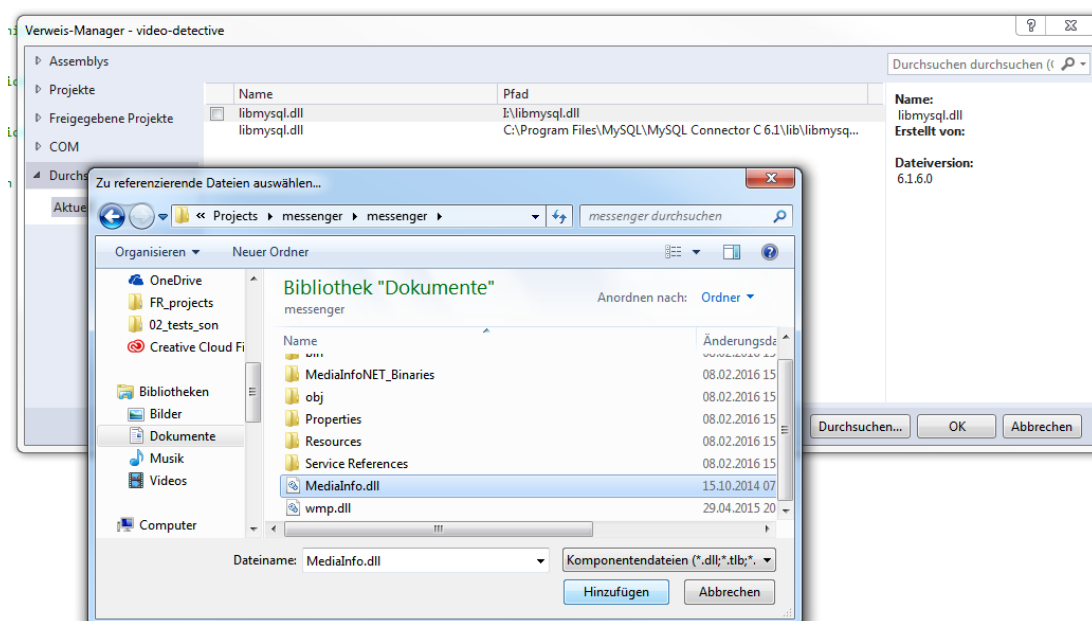


Abbildung 5: Verweismanager Fenster geöffnet

Wurde der Verweis erfolgreich hinzugefügt, erscheint er im Projektmappen-Explorer:

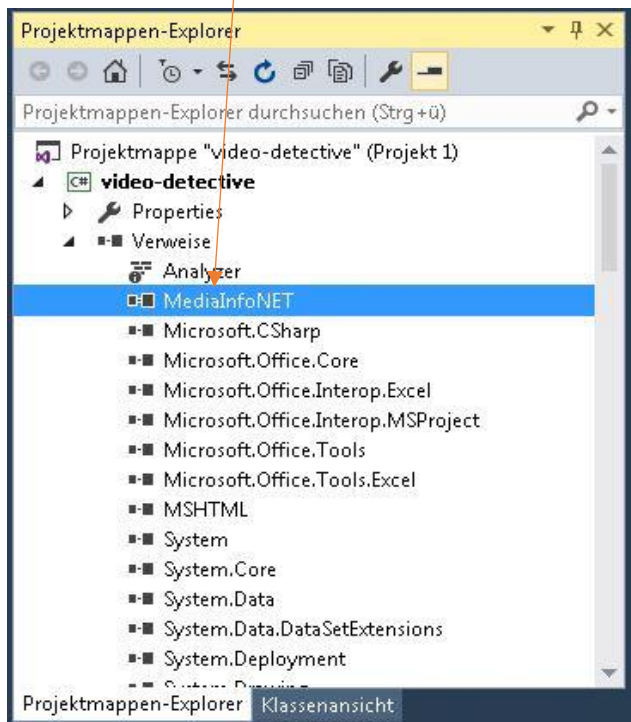


Abbildung 6: Visual Studio Projektmappen-Explorer Verweis MediaInfo

Danach wird die MediaInfo.dll einfach importiert (Rechtsklick auf das Projekt – Hinzufügen – Vorhandenes Element – MediaInfo.dll auswählen):

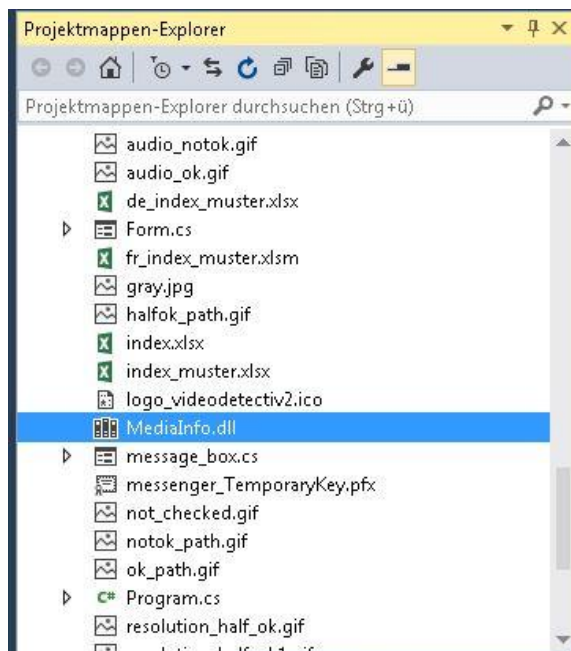


Abbildung 7: Visual Studio Projektmappen-Explorer MedInfo.dll

Damit MediaInfo nun im C#-Projekt genutzt werden kann, muss eine **Using-Anweisung** eingefügt werden:

```
using MediaInfoNET;
```

Nun ist es möglich, eine neue Instanz vom Typ MediaInfo anzulegen. Im nachstehenden Beispiel wurde eine Instanz vom Typ MediaInfo mit dem Namen „aviFile“ angelegt. In der Klammer steht ein Testordner, im wahren Code wird hier automatisch, mittels einer Schleife, der jeweilige vom Benutzer eingegebene Ordner eingetragen.

```
MediaFile aviFile = new MediaFile(@"C:\1234_test_video_files");
```

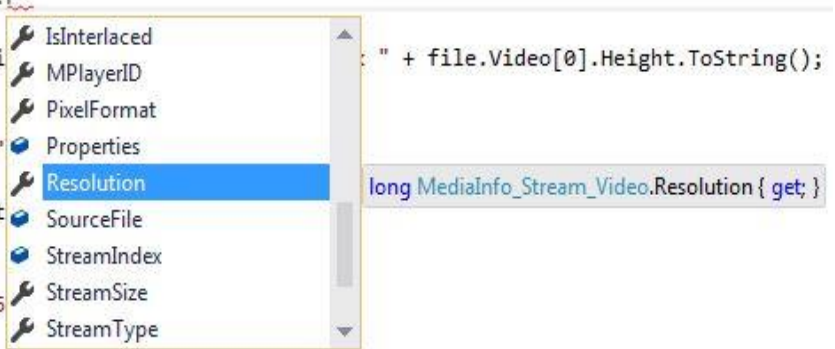
Mit der Instanz aviFile ist es möglich die Optionen von MediaInfo zu nutzen. Als Beispiel wird die Breite (in Pixel) eines Video-Files abgefragt. Die [0] nach video bedeutet, dass die erste Videospur kontrolliert werden soll (standardmäßig immer nur eine Videospur).

```
string resostring = aviFile.Video[0].Width.ToString();
```

Mit der Erweiterung .video ist es nun möglich, diverse Videoinformationen auszulesen. Einen kleinen Überblick wird in der nächsten Abbildung gegeben:

```
private void check_resolution(string datei, MediaFile file)
{
    try
    {
        resostring = file.Video[0].r...
        resolution_string = file.Vi... : " + file.Video[0].Height.ToString();
        if (resostring == "1280720"
        {
            resolution_1280.Add(dat...
        }
        else if (resostring == "136...
        {
            resolution_1360.Add(datei);

```



The screenshot shows a code editor with a dropdown menu open over the line `resostring = file.Video[0].r...`. The dropdown menu lists the following properties: IsInterlaced, MPlayerID, PixelFormat, Properties, Resolution (selected), SourceFile, StreamIndex, StreamSize, and StreamType. To the right of the dropdown, the type for the selected 'Resolution' property is displayed as `long MediaInfo_Stream_Video.Resolution { get; }`.

Abbildung 8: MediaInfo im Einsatz, Auflösung

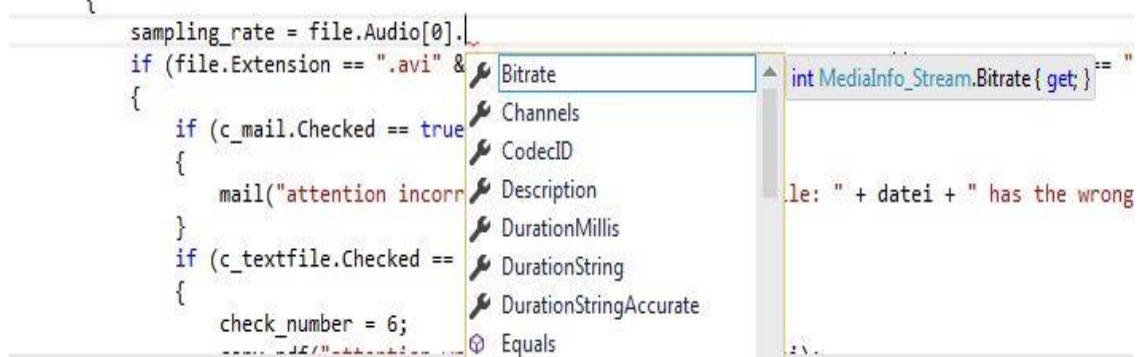
Die Erweiterung `.audio`, kann beispielsweise die Sampling Rate ganz einfach auslesen:

```
string sampling_rate = file.Audio[0].SamplingRate;
```

Die `[0]` am Ende von `.Audio` steht für die jeweilige Audiospur, aus der die Sampling Rate ausgelesen werden kann. Hierbei ist allerdings zu beachten, dass mehrere Audiospuren beziehungsweise Kanäle möglich sind. Diese Information ist für die schlussendliche Kontrolle von großer Bedeutung.

Bei der `.audio`-Erweiterung gibt es ebenso eine Menge an Informationen, die ausgelesen werden können. Einen kleiner Überblick hierzu, folgt auf der nachstenden Abbildung:

```
private void check_samplingrate(string datei, MediaFile file)
{
    try
    {
        sampling_rate = file.Audio[0].SamplingRate;
        if (file.Extension == ".avi" &
        {
            if (c_mail.Checked == true
            {
                mail("attention incorr
            }
            if (c_textfile.Checked ==
            {
                check_number = 6;
                ...
            }
        }
    }
}
```



The screenshot shows a code editor with a list of MediaInfo properties. The properties listed are: Bitrate, Channels, CodecID, Description, DurationMillis, DurationString, DurationStringAccurate, and Equals. A tooltip for the Bitrate property is visible, showing the type `int MediaInfo_Stream.Bitrate { get; }` and the value `le: " + datei + " has the wrong`.

Abbildung 9: MediaInfo im Einsatz, Bitrate

In diesem Punkt wurden die Eigenschaften und Vorteile von MediaInfo erörtert. Die Funktionen von MediaInfo sind ein wesentlicher Bestandteil des video-detective und kommen vielfach zum Einsatz.

6 FFmpeg grundlegende Funktionen

Damit die im Punkt: video audio split, beschriebene Split-Funktion eingesetzt werden kann, wird eine weitere Bibliothek benötigt: die FFmpeg-Bibliothek. Mit ihr ist es möglich, die einzelnen Audiokanäle vom Videostream zu trennen.

Was ist FFmpeg?

Das FFmpeg-Paket besteht aus einer Reihe von freien Computerprogrammen und Programmbibliotheken, welche digitales Video- und Audiomaterial aufnehmen, konvertieren, senden (streamen) und in verschiedene Containerformate verpacken können. Unter anderem enthält es eine Sammlung unterschiedlicher Audio-/Videoencoder und -Decoder. Sie befinden sich in der Libavcodec. FFmpeg ist ein Kommandozeilenprogramm, um von einem Video-, Audio- oder Bildformat zu einem anderen zu konvertieren. Es unterstützt auch das Aufnehmen und Kodieren von einer TV-Karte in Echtzeit. FFserver ist ein HTTP-Multimedia-Streaming-Server für Livestreaming über das Internet. Libavcodec enthält alle FFmpeg-Audio- und -Videoencoder. Diese Bibliothek kann von anderen Programmen benutzt werden, um dann die Filme darzustellen. Libavformat enthält die Container-Parser und -Ersteller für alle herkömmlichen Audio- und Video-Containerformate (z.B. avi, mov, mkv, ogg). Das FFmpeg-Paket ist kostenlos.¹²

6.1 FFmpeg – Bedeutung im video-detective

Im video-detective wird die FFmpeg-Bibliothek verwendet, um Video von Audio zu trennen. Diese Funktion befindet sich nur auf dem „production“-Tab, sie wird hauptsächlich von der Postproduktion verwendet.

Bei der Programmierung gilt dabei ein besonderes Augenmerk der Unterscheidung zwischen .avi- und .mov-Files. Diese Unterscheidung ist deswegen von Bedeutung, weil die verschiedenen Formate eine unterschiedliche Audiospurenbelegung aufweisen. Einen kleinen Einblick in die Programmierung von FFmpeg wird im nächsten Punkt gegeben.

¹² Vgl. ffmpeg.org, A complete, cross-platform solution to record, convert and stream audio and video (siehe: <https://ffmpeg.org/>), 2016.

6.2 FFmpeg-Programmierung im video-detective

Grundsätzlich werden im video-detective zwei .bat-Dateien erstellt. Diese Erstellung sieht folgendermaßen aus:

```
using (StreamWriter sw = System.IO.File.CreateText(avi_split))
{
    sw.WriteLine("for %a in (\\" + select_dir + "*.mov\\") do ffmpeg -i \\"%a\\" -
map_channel 0.1.0 \\" + select_dir + "split-" + folder_time + "-mov" +
"\\"%~na_voice_L_to_mono.wav\\"");
    sw.WriteLine("for %a in (\\" + select_dir + "*.mov\\") do ffmpeg -i \\"%a\\" -
map_channel 0.1.2 -map_channel 0.1.3 \\" + select_dir + "split-" + folder_time +
"-mov" + "\\"%~na_systemsound_stereo.wav\\"");
    sw.WriteLine("exit");
}
```

In diese .bat-Datei werden zwei Zeilen geschrieben. Diese zwei Zeilen werden benötigt, um die beiden Audiospuren vom Video zu trennen und in eine .wav-Datei zu konvertieren. Nachdem die .bat-Datei erzeugt wurde, muss sie nur noch ausgeführt werden. Das Ausführen der .bat-Datei sieht in der Programmierung wie folgt aus:

```
Process proc = null;
string batDir = string.Format(avi_split2);
proc = new Process();
proc.StartInfo.WorkingDirectory = batDir;
proc.StartInfo.FileName = "extract_audio_mov.bat";
proc.StartInfo.CreateNoWindow = true;
proc.StartInfo.WindowStyle = ProcessWindowStyle.Hidden;
proc.Start();
```

Die .bat-Datei wird mittels der Eingabeaufforderung im Hintergrund ausgeführt. Der Benutzer sieht dabei nur die Statusleiste im video-detective. Um das Ausführen der .bat-Datei zu ermöglichen, ist es notwendig, die ffmpeg.exe zu importieren. Diese ausführbare Datei befindet sich im Hauptverzeichnis des video-detective-Projekts:

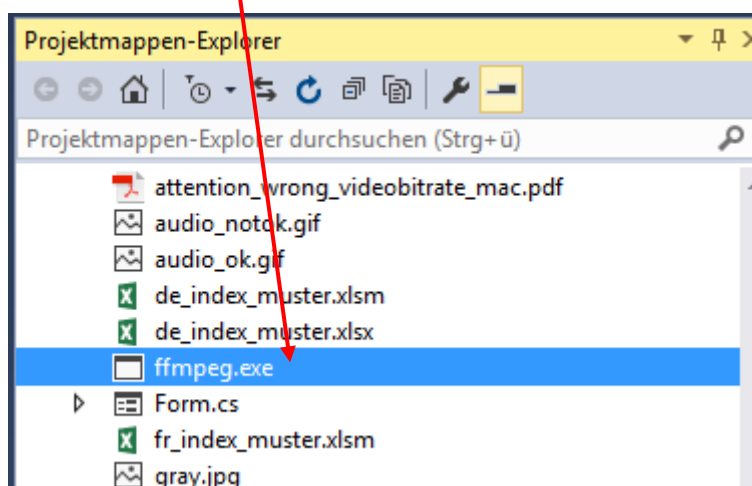


Abbildung 10: FFmpeg im Projekt einbinden

Schlussendlich entstehen zwei Audio-Tracks, einer mit der Namensweiterung „_voice_L_to_mono.wav“ und einer mit der Erweiterung „_systemsound_stereo.wav“. Diese werden im zuvor automatisiert erstellten Ordner „split“ gespeichert.

Die einzelnen Video-Files sehen im Explorer vor der Trennung wie folgt aus:

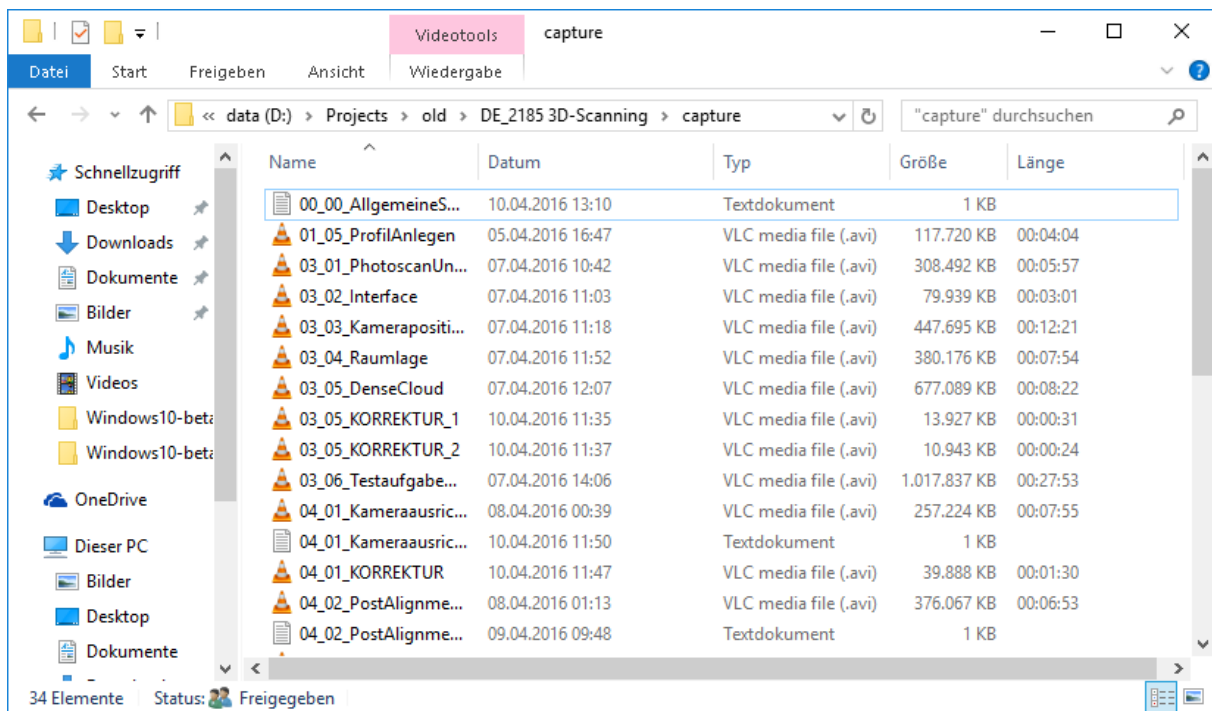


Abbildung 11: Ffmpeg im Einsatz, vor dem Split

Nach erfolgreicher Trennung wurde ein Ordner mit dem Namen „split“ erstellt, die getrennten .wav-Dateien werden in der nachfolgenden Abbildung verdeutlicht:

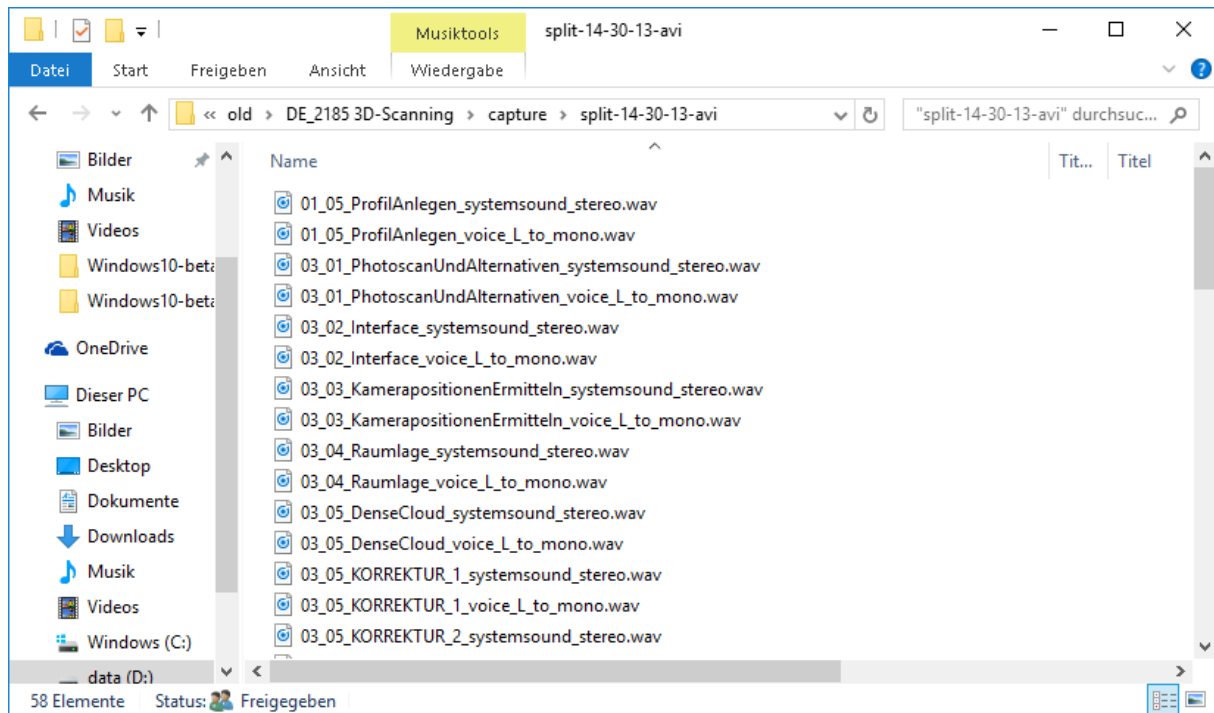


Abbildung 12: FFmpeg im Einsatz, nach dem Split

Nach erfolgreicher Trennung von Video und Audio können die Files nun von der Post Produktion verwendet werden (der Post Produktions-Workflow ist nicht Teil dieser Diplomarbeit).

In diesem Punkt wurde die mögliche Split-Funktion des video-detective mit Hilfe von FFmpeg näher erläutert. Der nächste Abschnitt thematisiert die Indexerstellung und dessen Implementierung.

7 Die automatisierte Indexerstellung im video-detective

An dieser Stelle näher auf die Programmierung der Indexliste eingegangen. Wie in Punkt Index beschrieben, ist die Indexliste ein wesentlicher Bestandteil für den Workflow des Content-Teams (deswegen befindet sich diese Option auch nur im „content“-Tab).

Die Besonderheit bei der Erstellung dieser Indexliste ist, dass alle Daten vollkommen automatisch in eine Excel-Mappe exportiert werden. Der User kann dabei auswählen, welche Indexliste (deutscher Index, französischer Index, englischer Index) er erstellen möchte. Diese Unterscheidung spielt beim Export der Daten eine große Rolle, denn abhängig von der Sprache werden die Daten unterschiedlich dargestellt.

Damit das Erstellen der Excel-Mappe möglich ist, müssen zuerst die Office-Bibliotheken in das C#-Projekt geladen werden:

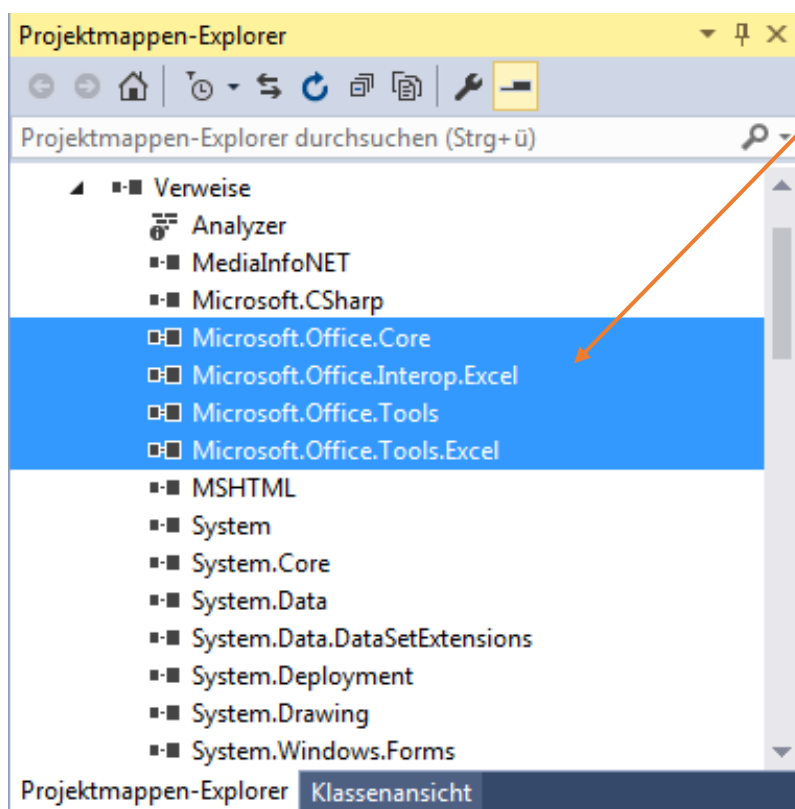


Abbildung 13: Office Verweise im Projekt hinzufügen

Nach der Implementierung ist es nun möglich, Excel in C# zu verwenden. Zunächst wird die Excel-Applikation gestartet:

```
Excel.Application excel = new Excel.Application();
```

Danach wird die Vorlage der deutschen Excel-Liste geladen (die String-Variable „source“ steht für den Pfad der Vorlage):

```
Excel.Workbook workbook = excel.Workbooks.Open(source, n, n, n, n, n, n, n, n, n, n, n, n, n, n, n);
```

Das Erstellen eines neuen Worksheets sieht im Code-Bereich wie folgt aus:

```
Excel.Worksheet worksheet = new Excel.Worksheet();
```

Festlegung des aktiven Worksheets (in unserem Fall das zuvor erzeugte Worksheet):

```
worksheet = (Excel.Worksheet)excel.ActiveWorkbook.ActiveSheet;
```

Das Anzeigen erfolgt mit folgender Zeile:

```
excel.Visible = true;
```

Nun werden die einzelnen Zellen mit Daten gefüllt. Im nachstehenden Beispiel wird die Zelle mit der Zeilennummer 9 und der Spaltennummer 5 (alphabetisch E) mit der Variable Zeit befüllt:

```
worksheet.Cells[9, 5] = index_time;
```

Nach Abschluss der automatisierten Eintragung der einzelnen Daten wird die Excel-Liste schlussendlich gespeichert:

```
workbook.SaveAs(currentDir + @"\Resources\de_index.xlsx");
```

Mit diesen grundsätzlichen Befehlen werden nun die einzelnen Excel-Listen in den verschiedenen Sprachen erzeugt. Die automatisierte Erstellung und die verschiedenen Berechnungen, die der video-detective hier noch vornimmt, sind zu komplex und würden an dieser Stelle den Rahmen der Dokumentation sprengen.

Die nächsten beiden Punkte beschäftigen sich mit dem kompletten Anwendungsprozess des video-detective. Dabei wird genau Bezug auf die einzelnen Funktionen genommen.

8 Funktionen und Aufbau des video-detective

In diesem Punkt erfolgt zuerst ein Überblick über den video-detective. Die einzelnen Buttons und Eingabefelder werden erklärt.

Die gesamte Startoberfläche des video-detective sieht folgendermaßen aus:

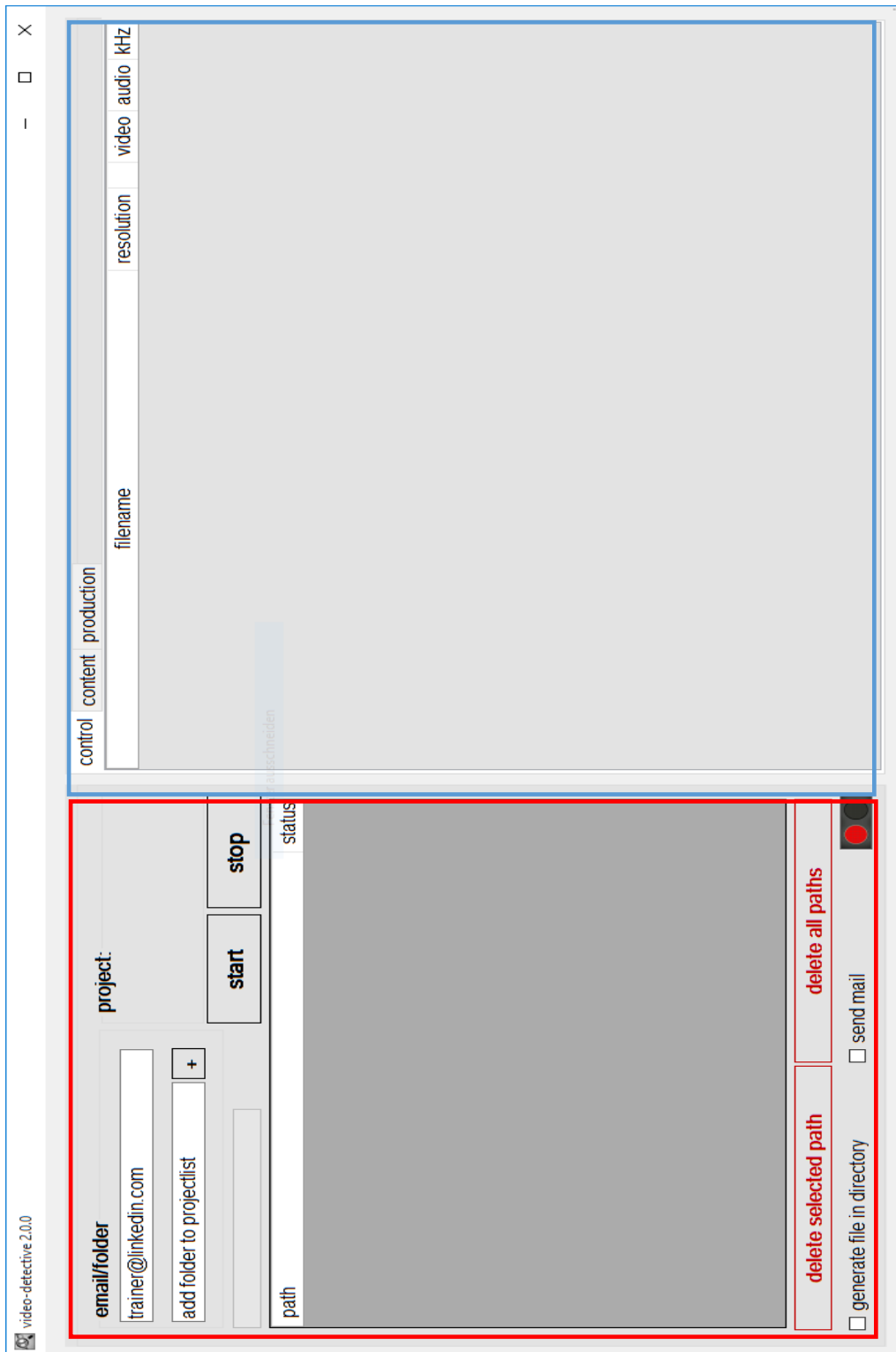


Abbildung 14: video-detective Startoberfläche

Die Oberfläche des video-detective (Abbildung 13: video-detective Startoberfläche) teilt sich in zwei Bereiche auf:

- Der **linke Teil** ist für die gesamte Eingabe verantwortlich. Hier werden diverse Einstellungen (email, add folder, send mail, ...) vorgenommen, bevor der Klick auf den „Start“-Button erfolgt. Nach Eingabe eines Projekts wird der Pfad im datagridview: dgv_path angezeigt. Zusätzlich können die einzelnen Pfade auch gelöscht werden. Nach erfolgreicher Kontrolle beginnt der Timer zu laufen. Am unteren rechten Rand wird eine „Ampel“ dargestellt. Diese springt auf Grün, sobald der Timer gestartet wurde. Das Beenden der kontinuierlichen Kontrolle erfolgt durch das Betätigen des „Stop“-Buttons.
- Der **rechte Teil** wird hauptsächlich für die Darstellung verwendet. Hier erfolgt eine Auflistung der einzelnen Video-Files. Die Anzeige gliedert sich in drei Bereiche: „control“, „content“, „production“. Je nachdem, welche Darstellung der Benutzer benötigt, kann er zwischen den einzelnen Tabs hin- und herspringen. Zusätzlich lassen sich die dargestellten Video-Files in eine Excel-Tabelle (Buttons: „export de – index“, „export fr – index“, „export excel index“) exportieren.

8.1 Der linke Teil des video-detective

In diesem Punkt wird die linke Seite des video-detective näher beschrieben.

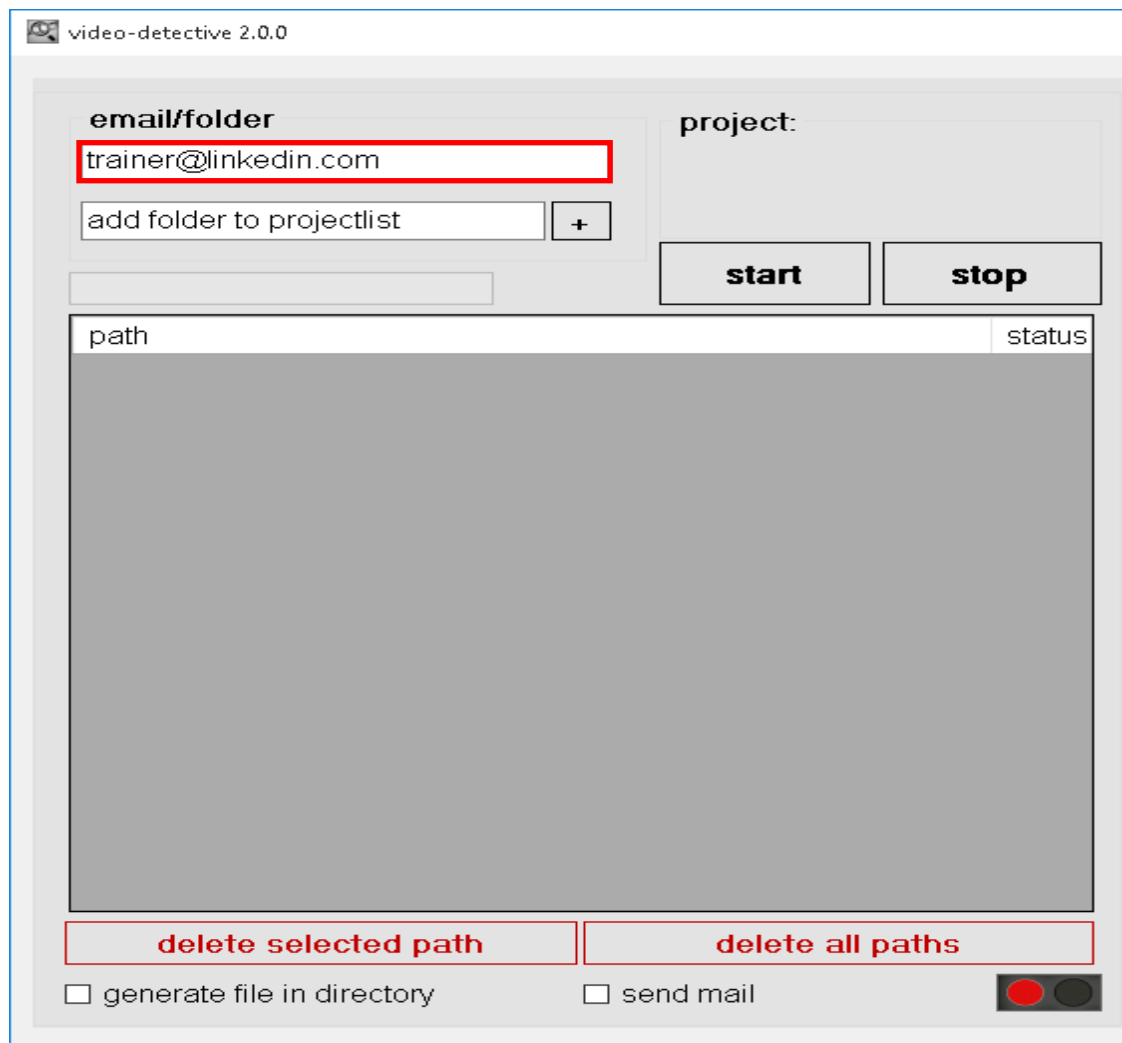


Abbildung 15: video-detective Startoberfläche linke Seite

Wie anhand der Abbildung (Abbildung 14: video-detective Startoberfläche linke Seite) erkennbar, erfolgt im Textfeld **txt_mail** die Eingabe der E-Mail-Adresse. Sollte bei der Kontrolle ein Fehler auftreten, wird an die angegebene Adresse eine „Attention-Mail“ geschickt. Zusätzlich wird im Anhang auch noch eine Anleitung zum Beheben des Fehlers mitgeschickt. Die E-Mail-Adresse wird standardmäßig aus dem Benutzernamen generiert, zum Beispiel Benutzernamen: deder + String: @linkedin.com. Die Adresse kann natürlich jederzeit geändert werden.

Die E-Mails werden mittels eines Gmail-Kontos übermittelt. Der video-detective stellt dabei eine Verbindung zu den Gmail-Servern her und versendet die generierten E-Mails. Das Versenden der Mails wird in einem anderen Punkt näher erläutert.

In der nachfolgenden Abbildung sieht man, wie beispielsweise ein Pfad hinzugefügt wird. Dabei öffnet sich ein „folderBrowser-Dialog“:

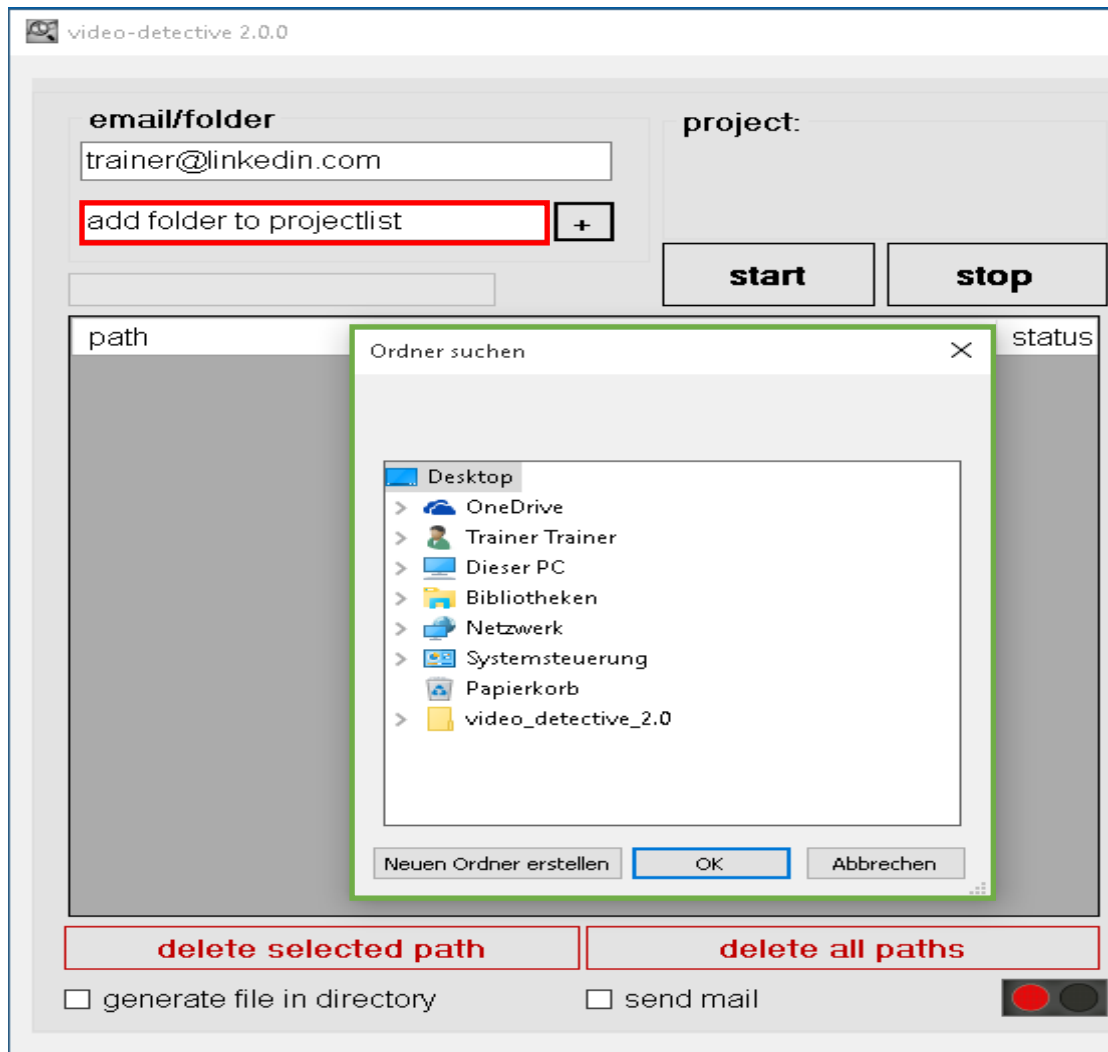


Abbildung 16: video-detective Startoberfläche Ordner hinzufügen

In dem Textfeld **txt_pfad** (Abbildung: video-detective Startoberfläche hinzufügen) wird der Pfad bzw. das Projekt hinzugefügt, das später kontrolliert werden soll. Die Eingabe erfolgt entweder mit Copy und Paste (Bestätigung mit „Enter“-Key) oder über das Klicken auf die „+“-Schaltfläche. Wird auf den „+“-Button geklickt, öffnet sich ein „**folderBrowser-Dialog**“. Hier wird das Projekt ausgewählt. Der ausgewählte Pfad wird im Textfeld: txt_pfad angezeigt.

Nicht außer Acht gelassen werden darf, dass es nicht möglich ist, den gleichen Pfad noch mal einzufügen. Sollte der User dies versuchen, erhält er eine Fehlermeldung.

Außerdem ist es auch nicht möglich, einen ungültigen Pfad in das Textfeld zu kopieren. Nach Einfügen des Pfads bzw. Projekts wird überprüft, ob dieser Pfad auch vorhanden

ist. Sollte der Pfad nicht vorhanden oder ungültig sein, so erscheint ebenfalls eine Fehlermeldung.

Nach erfolgreichem Hinzufügen eines Pfads beziehungsweise eines Projekts, verändert sich die Oberfläche wieder, diese Veränderung wird in der nachfolgenden Abbildung deutlich:

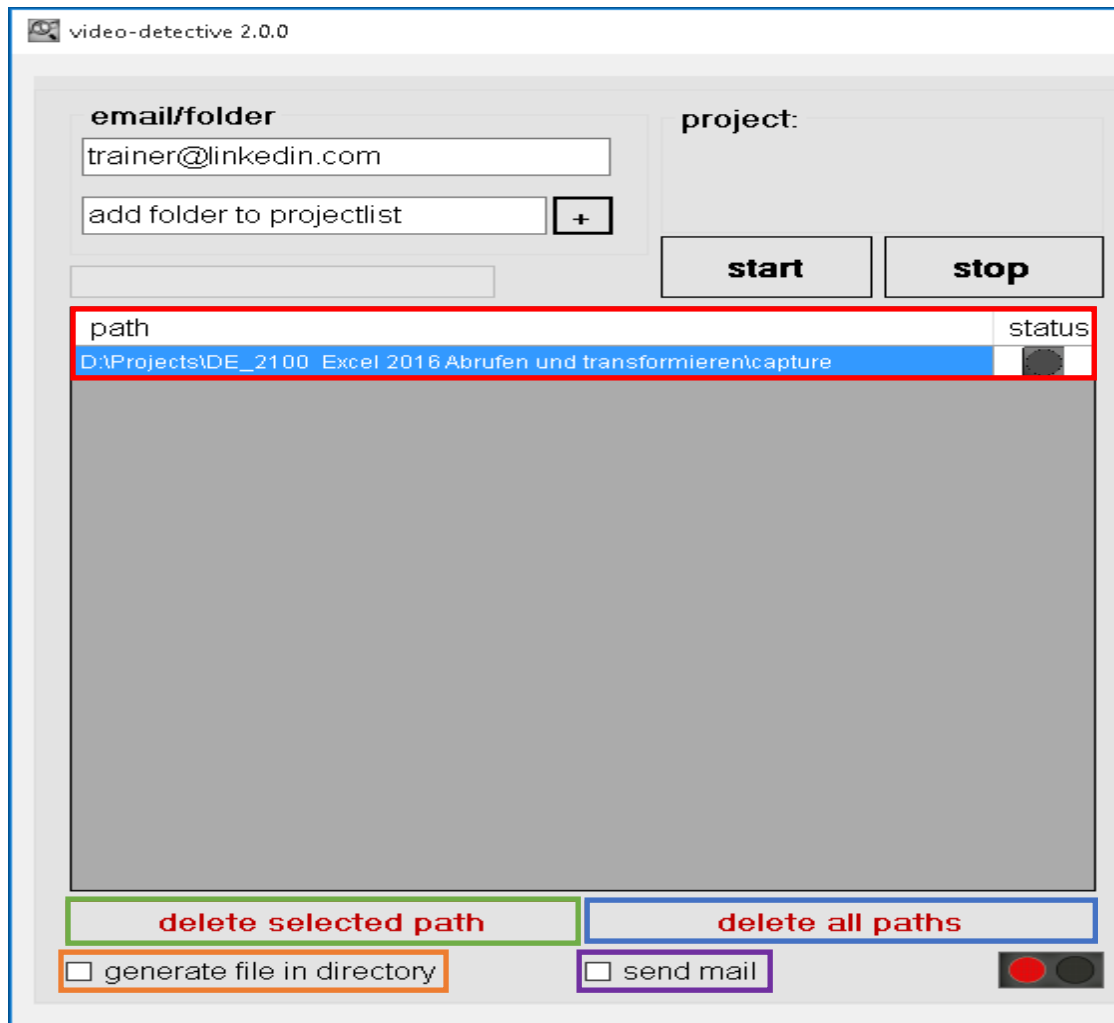


Abbildung 17: video-detective Startoberfläche Ordner wurde hinzugefügt

Der Pfad des Projekts erscheint im datagridview: **dgview_path**. Rechts daneben wird die **Status**-Spalte (Abbildung: video-detective Startoberfläche Ordner wurde hinzugefügt) abgebildet. Hat die Ampel die graue Farbe, bedeutet dies, dass das Projekt noch nicht kontrolliert wurde. Ist die Ampel rot, befindet sich in diesem Projekt mindestens ein fehlerhaftes Video-File. Eine grüne Ampel signalisiert, dass alles in Ordnung ist.

Mit dem Button „**delete selected path**“ wird der ausgewählte Pfad gelöscht. Damit das Löschen erfolgreich ist, muss natürlich ein Pfad ausgewählt sein, andernfalls erfolgt

eine Fehlermeldung. Nach dem Löschen eines Pfads bzw. Projekts geht auch der Status verloren. Dies bedeutet, der Pfad muss wieder neu „kontrolliert“ werden.

Mit dem Klick auf den Button „**delete all paths**“ werden alle Pfade bzw. Projekte gelöscht, inklusive der Informationen, die durch die Kontrolle ermittelt wurden.

Am unteren Ende des video-detective sind zwei Checkboxen ersichtlich. Durch das Aktivieren dieser Checkboxen werden Zusatzoptionen freigeschaltet:

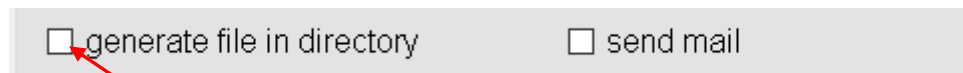


Abbildung 18: Attention Dokument Auswahl

Mit der Checkbox: „**generate file in directory**“ (Abbildung 17: Attention Dokument Auswahl) ist es möglich, beim Auftreten eines fehlerhaften Files die Anleitung gleich direkt in den jeweiligen Pfad zu kopieren. Voraussetzung ist, dass Schreibrechte in den jeweiligen Projektpfaden vorhanden sind. Diese Anleitungen wurden für alle möglichen Fehlerszenarien erstellt. Die Checkbox ist beim Starten des video-detective standardmäßig deaktiviert.



Abbildung 19: Attention Email Auswahl

Mit der Checkbox: „**send mail**“ (Abbildung 18: Attention Email Auswahl) ist es möglich, beim Auftreten eines fehlerhaften Files eine Attention-Mail zu verschicken. In dieser Mail steht genau, welches Video-File einen Fehler aufweist und wie sich der jeweilige Fehler beheben lässt. Zusätzlich ist es möglich, direkt auf den Link des fehlerhaften Files zu klicken und der Standard-Video-Player wird gestartet. Die Checkbox ist beim Starten des video-detective standardmäßig deaktiviert.

Auf der nächsten Seite wird der direkte Klick auf ein Projekt simuliert.

Wird direkt auf den **Pfad** im datagridview geklickt, so wird die Kontrolle gestartet, allerdings nur für diesen einen Pfad. Die Oberfläche ändert sich der Abbildung (Abbildung 19: video-detective ein Pfad aktiviert) entsprechend:

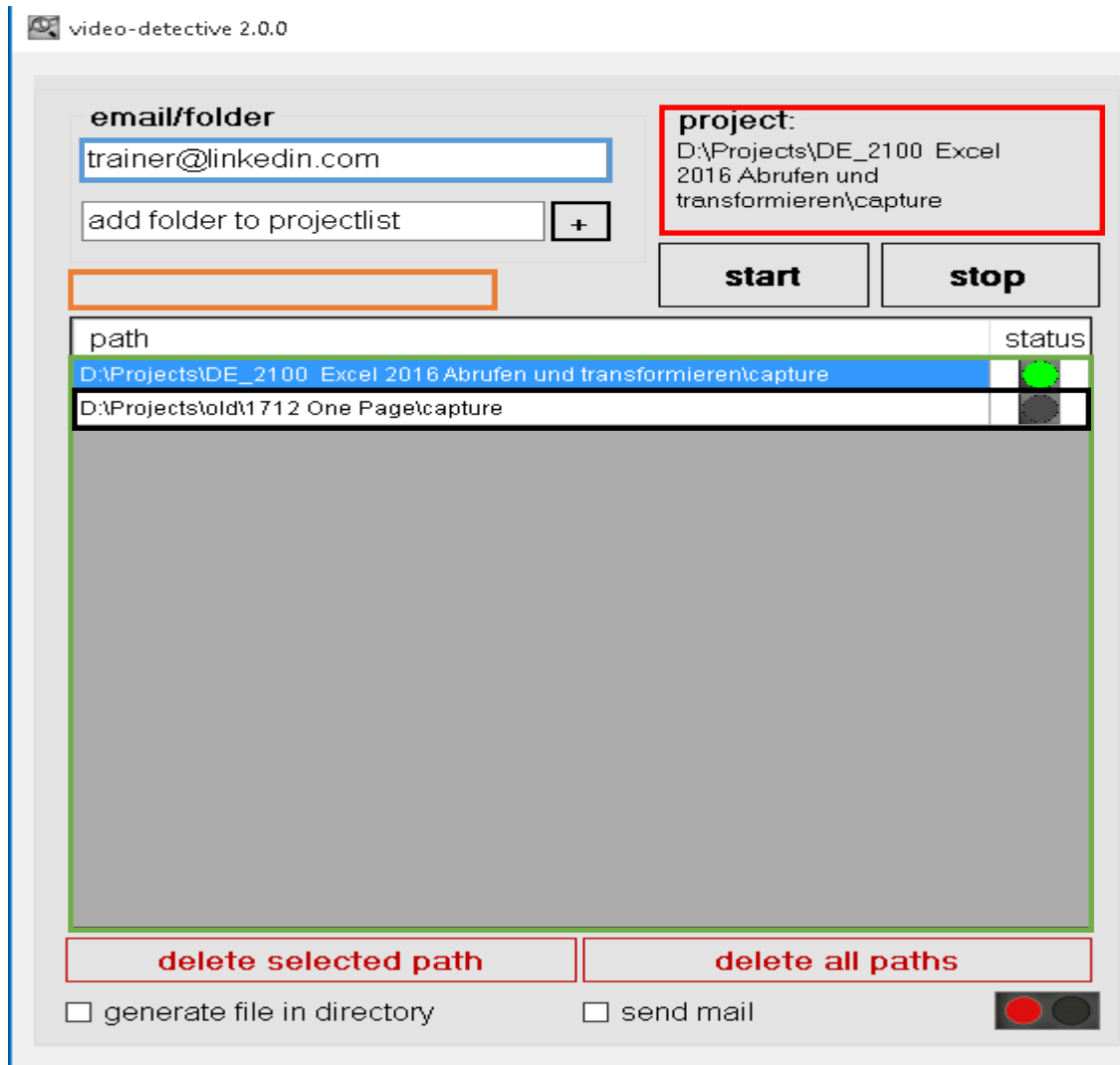


Abbildung 20: video-detective ein Pfad aktiviert

Nach dem direkten Klick auf den **Pfad** wird das Textfeld `txt_mail` deaktiviert. Das Projekt wird nun auf Fehler überprüft, nach erfolgreicher Kontrolle erscheint die Anzeige des Status in der Status-Spalte. Zusätzlich erfolgt eine Darstellung des ausgewählten Pfads im Label: `lbl_project`. Die `ProgressBar` bleibt bei der Einzelkontrolle allerdings grau. Im datagridview `dgv_path` befinden sich jetzt zwei Projekte. Das erste Projekt beinhaltet anscheinend kein fehlerhaftes Video-File, deswegen hat die Status-Ampel eine grüne Farbe. Das zweite Projekt wurde noch nicht überprüft, deswegen ist die Ampel grau. Die beiden Optionen „generate file in directory“ und „send mail“ sind in diesem Beispiel deaktiviert.

Wird nun der „start“-Button betätigt, so wird eine komplette Kontrolle durchgeführt. Dies bedeutet, alle im datagridview befindlichen Projekte werden überprüft. Die Oberfläche des video-detective ändert sich dabei wie folgt (Abbildung 20: video-detective gesamte Kontrolle aktiviert):

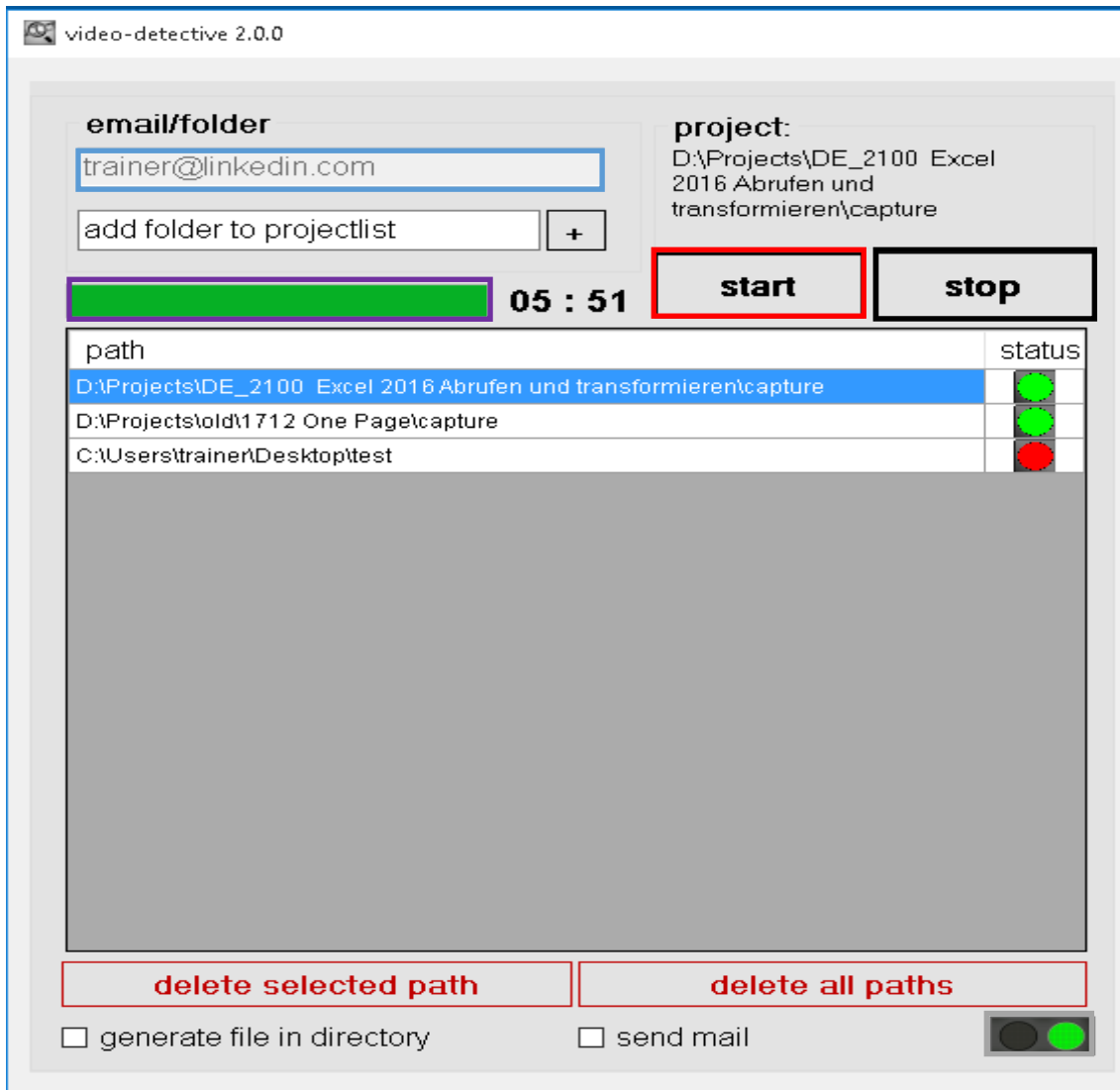


Abbildung 21: video-detective gesamte Kontrolle aktiviert

Nach dem Klick auf die „start“-Schaltfläche wird das Textfeld `txt_mail` deaktiviert. Die gesamten Projekte im `dgv_path` werden nun überprüft. Nach der Überprüfung wird der Timer eingeschaltet und die Zeit im Label `lbl_time` angezeigt. Die `ProgressBar` zeigt dabei den aktuellen Status der Überprüfung an.

Ist die `ProgressBar` auf 100%, bedeutet dies, dass die Kontrolle beendet ist und nach x Minuten wieder beginnt. Im datagridview `dgv_path` befinden sich jetzt drei Projekte. Das erste Projekt beinhaltet kein fehlerhaftes Video-File, deswegen hat die Status-Ampel eine grüne Farbe. Das zweite Projekt beinhaltet ebenfalls kein fehlerhaftes

Video-File. Das dritte Projekt allerdings enthält anscheinend mindestens ein fehlerhaftes Video-File und genau deswegen springt die Ampel auf Rot. Zusätzlich wird noch anhand der **PictureBox** am rechten unteren Ende signalisiert, dass die gesamte Kontrolle gestartet wurde:

- **Grün** bedeutet gestartet.
- **Rot** bedeutet nicht gestartet.

Mit Klick auf eines der Projekte, wird das ausgewählte Projekt im Label `lbl_project` angezeigt. Mit dem **stop**-Button lässt sich der Timer wieder stoppen. Nach dem Stopp wird das Label `lbl_time` „geleert“. Die `ProgressBar` wird dabei ebenfalls auf 0% gesetzt. Die beiden Optionen „generate file in directory“ und „send mail“ sind in diesem Beispiel deaktiviert.

Wird nun nochmals der **start**-Button betätigt dann erfolgt eine erneute Kontrolle, allerdings wurden die bereits gemeldeten Fehler gespeichert, dies bedeutet, es findet keine zusätzliche Fehlermeldung, zum bereits kontrollierten File statt. Somit werden häufige E-Mails desselben Fehlers vermieden.

Die linke Seite des video-detective wurde somit erläutert, im Anschluss wird der rechte Teil genauer unter die Lupe genommen.

8.2 Der rechte Teil des video-detective (deaktivierter Zustand) Control-Tab

In diesem Punkt wird die rechte Seite des video-detective erklärt (deaktivierter Zustand):

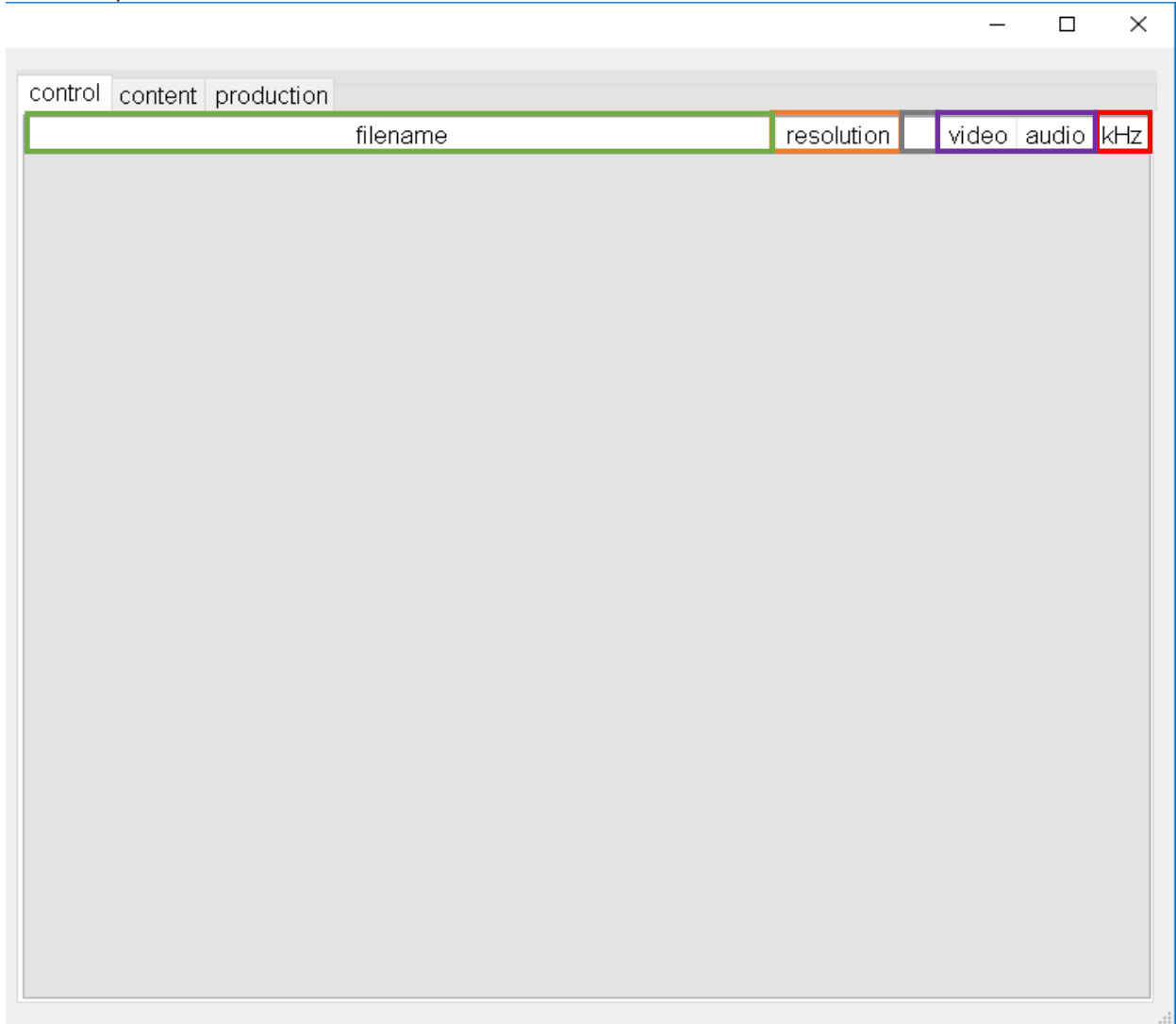


Abbildung 22: video-detective Startoberfläche rechte Seite

Hier wird der rechte Teil des video-detective dargestellt, sofern kein Projekt angeklickt wurde. Der rechte Teil gliedert sich in drei Bereiche: „control“, „content“, „production“. Wir befinden uns nun im „control“-Tab. In diesem Bereich werden die für das Studiomanagement relevanten Informationen angezeigt.

In der **ersten Spalte** wird der Name des jeweiligen Video-Files angezeigt. Direkt daneben steht die **Auflösung** des geprüften Videos. Hierbei ist zu beachten, dass es verschiedene akzeptable Auflösungen gibt. Die Kennzeichnung erfolgt anhand eines Ampelsystems in der nächsten Spalte.

Als Nächstes wird die Spalten „video“ und „audio“ dargestellt. Diese beiden Spalten signalisieren die Korrektheit der Audio- und Videoeinstellungen. Die Kennzeichnung erfolgt auch hier anhand eines Ampelsystems. Wie die Ampel auf die einzelnen Kriterien reagiert, wird in diesem Punkt: LinkedIn-Qualitätskriterien eines Video deutlich.

Die letzte Spalte des datagridview dgv_file ist die „kHz“-Spalte. In dieser Spalte wird die Sampling Rate angezeigt. Diese spezifische Information spielt für den internen Workflow eine große Rolle. Allerdings wird in dieser Arbeit nicht näher darauf eingegangen.

Content-Tab

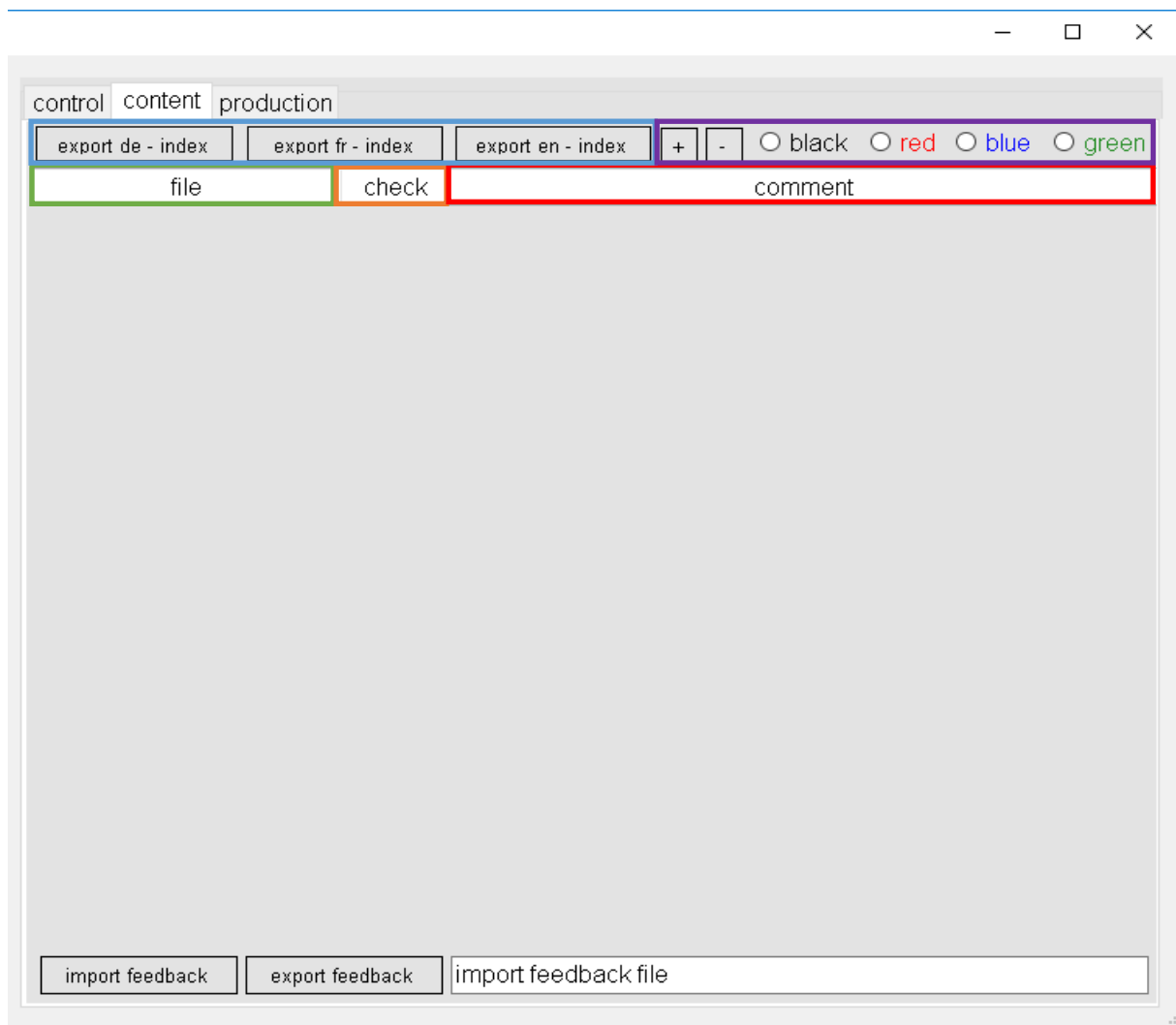


Abbildung 23: video-detective Startoberfläche Content Tab

In dieser Abbildung befinden wir uns nun im „content“-Bereich. Es sind die einzelnen für das Content-Team wichtigen Spalten sichtbar.

In der **ersten Spalte (filename)** steht der Name des Video-Files. Durch das direkte Klicken auf einen Videonamen wird der in Windows eingestellte Standard-Video-Player gestartet. Somit ist es möglich, das Video direkt zu kontrollieren.

Die **zweite Spalte (check)** wird automatisch mit einer Checkbox befüllt. Diese dient dem Content-Team zur Kontrolle, welches Video bereits überprüft wurde. Diese Information wird bei einem Neustart des video-detective in einem XML-File gespeichert. Dies bedeutet, sobald dasselbe Projekt noch mal überprüft wird, erscheinen die zuvor eingegebenen Informationen. Somit ist es möglich direkt an gleicher Stelle weiterzumachen.

Die letzte **Spalte (comment)** wird vom Content-Team verwendet, um diverse Informationen für ein Video-File zu speichern. Hier werden zum Beispiel die Videoteile eingetragen, die der Trainer noch mal aufzeichnen muss. Diese Informationen sind für die Indexerstellung relevant. Die Indexerstellung wird später detailliert erklärt. Zusätzlich ist es möglich, eine oder mehrere Kommentarzeilen **hinzuzufügen (oder zu entfernen)**. Die Farbe der Kommentarzeile kann mittels der **radioButtons (r_red, r_black, r_blue, r_green)** geändert werden.

Die drei Buttons „**export de – index**“, „**export fr – index**“ und „**export en – index**“ werden benötigt, um den Index anhand der eingegebenen Informationen zu erstellen.

8.3 Indexerstellung eines Video-Trainings

In diesem Punkt wird die Indexerstellung erklärt. Im gesamten Arbeitslauf zur Erstellung eines fertigen Trainings spielt die sogenannte Index-Excel-Liste eine entscheidende Rolle. Ein Teil dieser Liste wird vom Content-Team gemeinsam mit dem Trainer erstellt. Der Index ist sozusagen ein Drehbuch.

Befindet sich der Trainer im Aufnahmestudio, wird die Indexliste als Feedback-Liste zwischen Trainer und Content-Team verwendet. Sollten zum Beispiel irgendwelche Sätze neu einzusprechen sein, wird dies in der Indexliste notiert und der Trainer kann sie zur gegebenen Zeit neu einsprechen.

Der video-detective bietet als Zusatzoption die Erstellung genau dieser Indexlisten an. Damit wird der Workflow für das Content-Team erheblich beschleunigt. Es werden automatisch sämtliche Videonamen in die Indexliste übertragen. Zusätzlich werden auch die Dauer der Video-Files und die Kommentare eingetragen.

Die drei Buttons für den index-excel export sehen folgendermaßen aus:



Abbildung 24: video-detective content tab index Optionen

Grundsätzlich ist es möglich, drei verschiedene Indexlisten zu erstellen, die Excel-Listen werden mit den zuvor eingefügten Informationen befüllt. Wie eine solche Indexliste aussieht, wird anhand des nachfolgenden Beispiels deutlich:

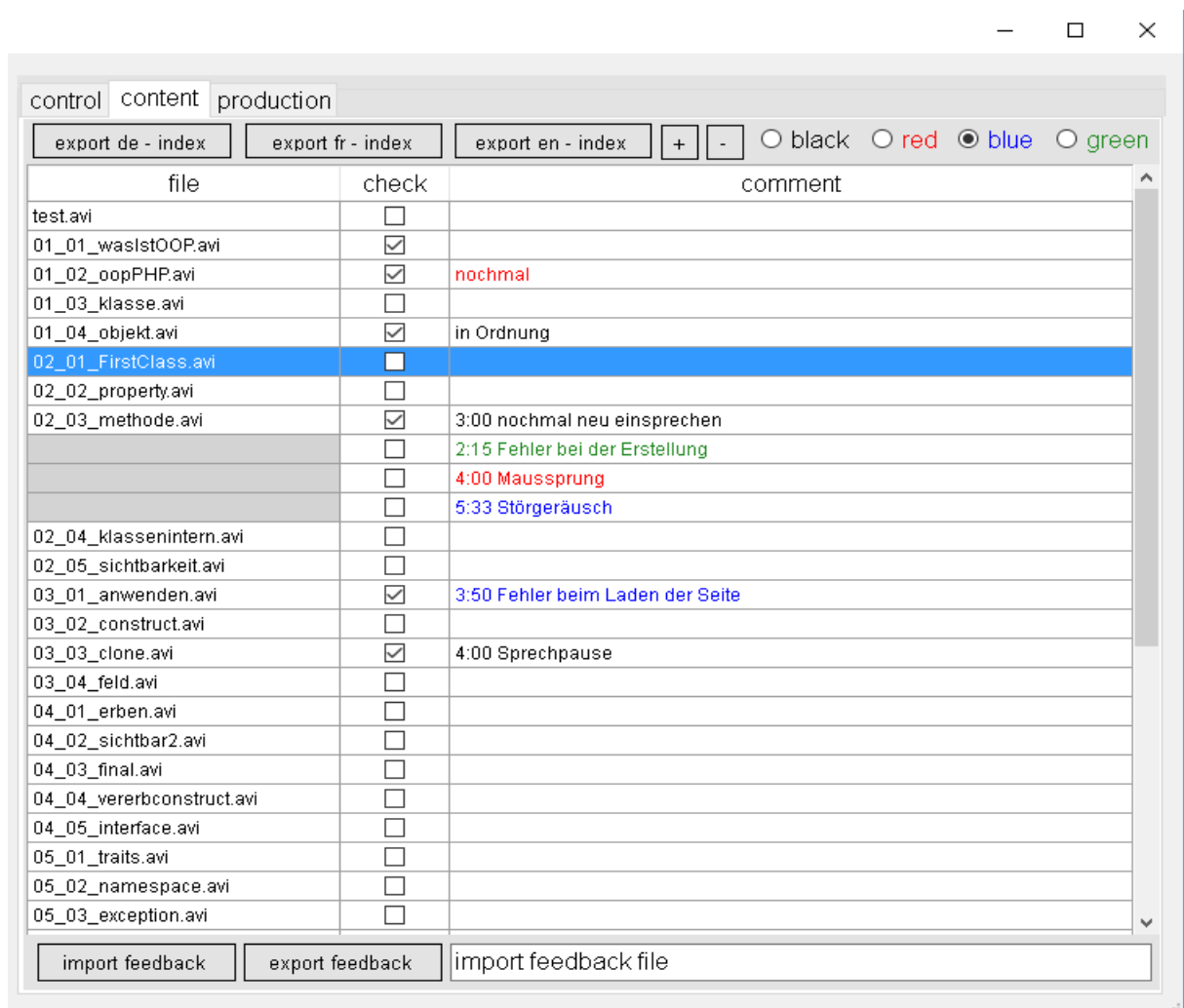


Abbildung 25: video-detective Content Tab, nach der Kontrolle

In der Abbildung (Abbildung 24: video-detective Content Tab, nach der Kontrolle) wird deutlich, dass diverse Informationen zu den einzelnen Videos eingetragen wurden. Mit Klick auf den Button „**export de – index**“, wird ein deutscher Index erstellt. Die automatisiert erstellte Index-Liste wird in der nächsten Abbildung (Abbildung 25: Exportierter Index) verdeutlicht:

Abbildung 26: Exportierter Index

Die Informationen aus dem datagridview wurden in die Indexliste übertragen. Wurde in der Spalte „checked“ das Video-File angehakt, dann wird die jeweilige Zeile im Index grau hinterlegt. Zusätzlich ist es möglich, die Kommentare bzw. Anmerkungen, anhand von unterschiedlichen Farben zu selektieren. Diese Farbeinstellung wird ebenfalls in die Indexerstellung übernommen. Die Indexerstellung ist ab Office 2010 möglich.

8.4 Import und Export der XML-Datei im video-detective

An dieser Stelle wird der Import und Export der XML-Feedback-Datei näher erläutert. Diese Option befindet sich am unteren Ende des content-Tabs (Abbildung 26: video-detective Feedback-Export-Funktion):



Abbildung 27: video-detective Feedback-Export-Funktion

Die Idee dahinter ist, dass die Videos von externen Mitarbeitern geprüft und die Informationen zu den jeweiligen Video-Files eingetragen werden. Damit nun das Content-Team dasselbe Feedback gleich direkt im video-detective bearbeiten kann, muss es zuvor in irgendeiner Form importiert werden. Und genau diese Anforderung wird mit dem XML-Export/Import realisiert.

Damit ein Feedback exportiert werden kann, muss natürlich zuerst eine Kontrolle der Files stattgefunden haben und zudem sollten diverse Informationen eingetragen sein.

Mit Klick auf den „export feedback“-Button, öffnet sich ein FolderBrowser-Dialog. Auf der Abbildung (Abbildung 27: video-detective Export-Funktion im Einsatz) ist dieser Dialog sichtbar. In dem Fenster „Ordner suchen“ wird der Speicherort der Feedback-Datei ausgewählt. Wird die Auswahl mit einem Klick auf den „ok“-Button bestätigt wird die XML-Datei im gewünschten Speicherort erstellt, sofern Schreibrechte vorhanden sind. Die erstellte XML-Datei wird in der Abbildung 28: video-detective exportiertes XML-File veranschaulicht gemacht.

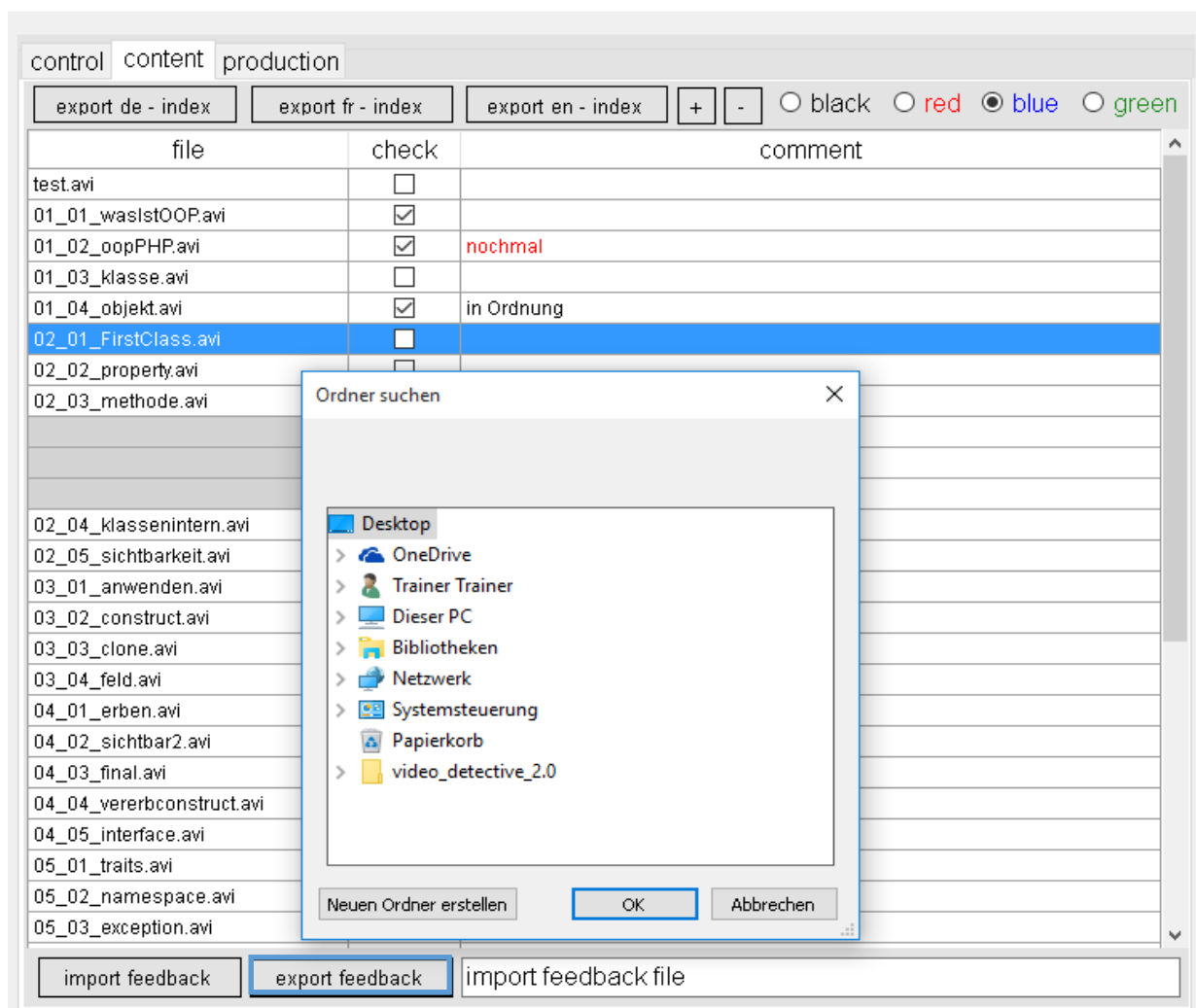


Abbildung 28: video-detective Export-Funktion im Einsatz

Der Name der XML-Datei setzt sich wie folgt zusammen:

„Projektordnername“ + „Projektordnergröße in Byte“ + „Text: content“

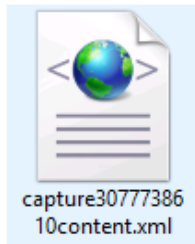


Abbildung 29: video-detective exportiertes XML-File

Die Besonderheit beim Import der XML-Dateien liegt nun darin, dass der Name der exportierten XML-Datei geändert werden kann. Der video-detective ordnet die importierte Datei immer dem richtigen Projekt zu. Der Import der XML-Datei wird auf der Abbildung 29: video-detective Import XML-File näher erläutert.

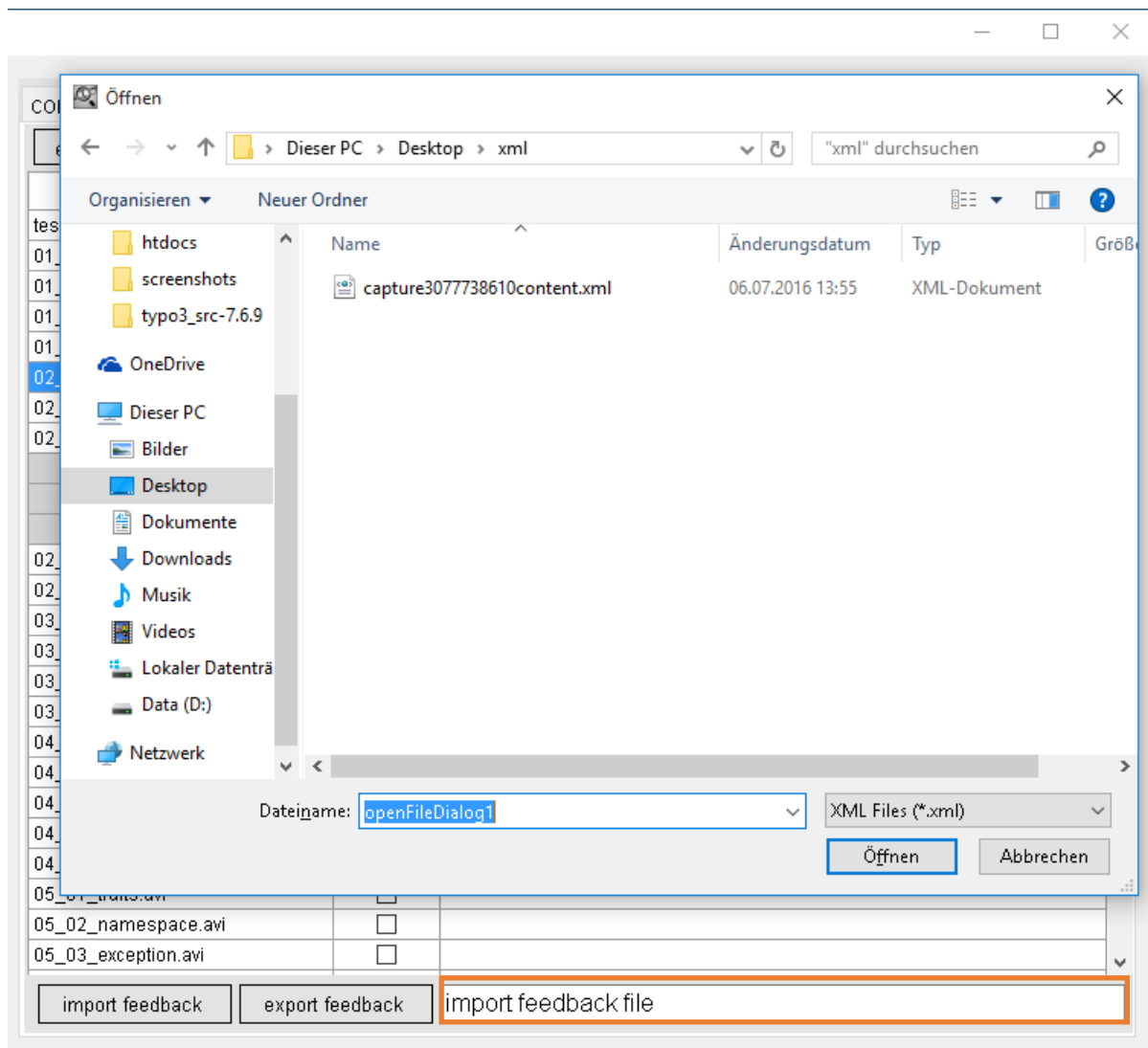


Abbildung 30: video-detective Import XML-File

Auf der Abbildung wird das Import-Fenster dargestellt. Es öffnet sich mit direktem Klick

auf die **TextBox**. Im openFileDialog wählt der User die gewünschte XML-Datei aus und nach Klick auf „import feedback“ wird die XML-Datei im datagridview angezeigt. Der User kann nun direkt das extern erstellte Feedback bearbeiten und dem Trainer weiterleiten. Der Trainer bearbeitet das Feedback und leitet es zum Content-Manager weiter

8.5 Production-Tab

An dieser Stelle wird der Produktionsbereich erläutert. Dieser Bereich beinhaltet alle notwendigen Informationen für das Postproduktion-Team. Die Abbildung (Abbildung 30: video detective rechts Production Tab) zeigt den Content Bereich ohne ausgewähltes Projekt.

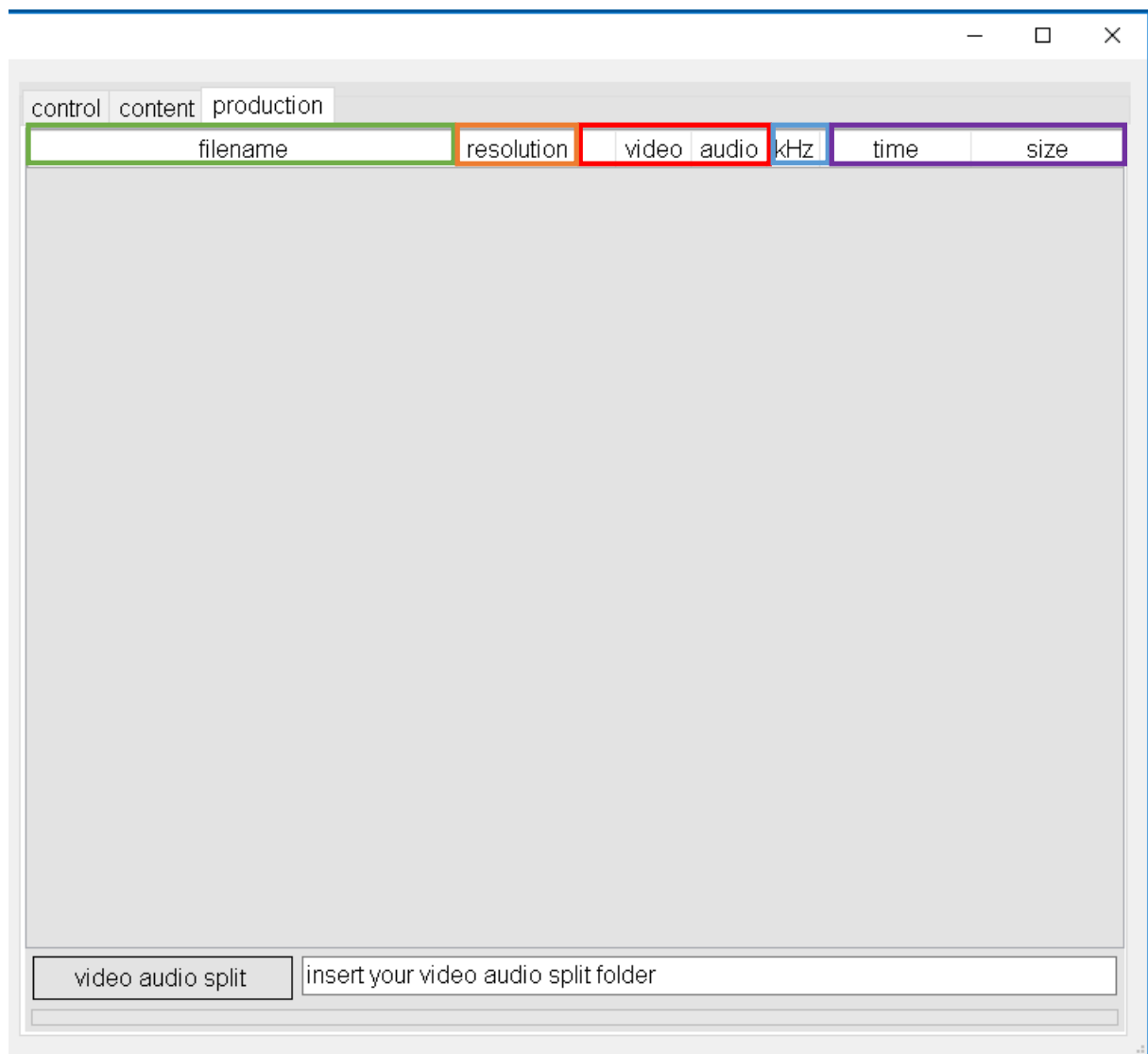


Abbildung 31: video detective rechts Production Tab

In der **ersten Spalte** steht wieder der jeweilige File-Name. Direkt daneben in **Spalte zwei** befindet sich die Auflösung und in **Spalte drei bis fünf** wird das bekannte Ampelsystem angezeigt. Die „kHz“-**Spalte** dient zur Anzeige der Sampling Rate und in der **Spalte „time“** wird die Dauer des jeweiligen Video-Files in Sekunden angezeigt. Die letzte **Spalte „size“** beinhaltet die Größe der Video-Files in Byte.

Die Besonderheit am Produktionsbereich ist allerdings die „video-audio-split“-Funktion. Im nächsten Punkt wird diese Funktion explizit erläutert.

8.6 video audio split Funktion im video-detective

Die Split-Funktion wurde in Zusammenarbeit mit der Postproduktion entwickelt und dient im weitesten Sinne der Trennung von Tonspur und Video.

Die video-audio-split-Funktion befindet sich nur im Produktionsbereich und sieht (Abbildung 31: video detective rechts Video Audio Split Funktion) folgendermaßen aus:



Abbildung 32: video detective rechts Video Audio Split Funktion

Bei der Nachbearbeitung der einzelnen Dateien ist es notwendig, die Tonspur vom Video zu trennen, und genau dies passiert hier. Die Trennung erfolgt mittels der ffmpeg-Bibliothek. Über das Textfeld **txt_video_audio** wird der Projektpfad eingefügt (Abbildung 32: video detective rechts video audio split Funktion im Einsatz):

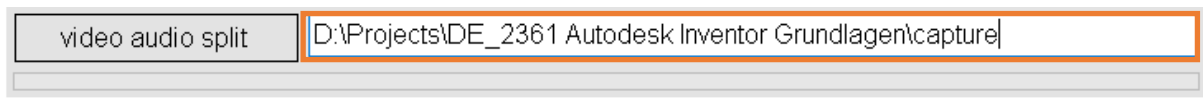


Abbildung 33: video detective rechts video audio split Funktion im Einsatz

Mit Klick auf den Button **btn_video_audio**, verändert sich die Oberfläche der Abbildung (Abbildung 33: video detective rechts video audio split Funktion Status Anzeige) entsprechend:

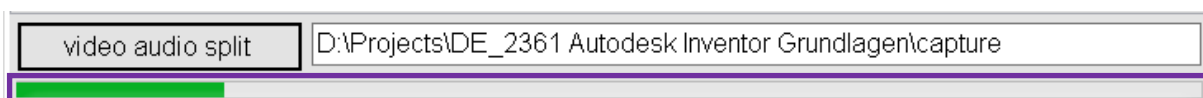


Abbildung 34: video detective rechts video audio split Funktion Status Anzeige

Nun findet die Trennung von Video und Audio statt. Die Anzeige des Fortschritts erfolgt in der Progressbar `prb_audio_video`. Der Effekt wird sichtbar, sobald der User zu dem eingefügten Pfad navigiert. Im Projekt wurde ein neuer Ordner hinzugefügt.

Der hinzugefügte Ordner sieht im Windows-Explorer beispielsweise wie folgt aus (Abbildung 34: video detective rechts video audio split Funktion, generierter Ordner):

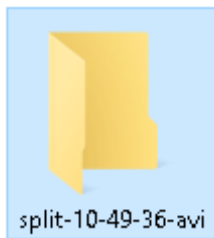


Abbildung 35: video detective rechts video audio split Funktion, generierter Ordner

In diesem Ordner befinden sich die einzelnen extrahierten Audiospuren. Diese Audiospuren werden in weiterer Folge für die Bearbeitung verwendet. Wird ein weiteres Mal auf den „video audio split“-Button geklickt, dann wird ein nächstes Verzeichnis mit anderen Namen erstellt. Die Überlegung dahinter, es ist auch möglich mehrere Splits des gleichen Projekts zu erstellen. Somit ist es möglich den Inhalt der Projekte zu ändern und trotzdem wird das Verzeichnis nicht gelöscht.

Die dahinterstehende Programmierung wurde im Punkt FFmpeg-Programmierung im video-detective näher erläutert.

Im nächsten Punkt werden nochmals die Qualitätsanforderungen der Trainings erläutert, welche Kriterien dabei für den video-detective relevant sind und wie er darauf reagiert wird ebenfalls verdeutlicht.

9 Fehlermeldung und Fehlerbehebung des video-detective

Bevor die Darstellung der Attention-E-Mails und Attention-Dokumente erläutert wird, werden nochmals die Qualitätskriterien von LinkedIn Austria aufgelistet.

Es wird, wie oben bereits beschrieben, zwischen .mov-Files, die mit iShowU unter Mac OSX aufgezeichnet werden, und .avi-Files, die mit Bandicam unter Windows aufgezeichnet werden, unterschieden.

Die aktuellen Qualitätsanforderungen von LinkedIn für .mov- und .avi-Files:

- Die Auflösung muss 16:9 sein, das bedeutet, folgende Auflösungen werden akzeptiert:
 - **1280 x 720** Pixel, dies ist die Standardauflösung, der detective meldet grün.
 - **1360 x 768** Pixel, dies ist nicht die Standardauflösung, wird aber unter Umständen verwendet, der detective meldet gelb.
 - **1366 x 768** Pixel, dies ist nicht die Standardauflösung, wird aber unter Umständen verwendet, der detective meldet gelb.
 - **1600 x 900** Pixel, dies ist nicht die Standardauflösung, wird aber unter Umständen verwendet, der detective meldet gelb.
 - **1920 x 1080** Pixel, dies ist nicht die Standardauflösung, wird aber unter Umständen verwendet, der detective meldet gelb.

Bei allen anderen Auflösungen meldet der detective rot.

- Die Frame Rate (Bilder pro Sekunde) muss bei .avi-Files mindestens **25** betragen, bei .mov-Files mindestens **15** (bei iShowU ist die Frame Rate variabel).
- Die Sampling Rate muss mindestens **48 kHz** betragen.
- Es muss wenigstens **ein Audio-Track** vorhanden sein mit mindestens **zwei Spuren**.
- Das Audioformat muss **AAC** oder **PCM** sein.
- Der Video-Codec bei .avi-Files ist **tsc**.
- Die PCM-Bitrate muss mindestens **1400 Kbps** betragen, bei AAC **150 Kbps**.

- Der Audio-Track und das Video dürfen maximal nur 200 Millisekunden Unterschied aufweisen. Sonst wird die Aufnahme asynchron.

Trifft eines der oben genannten Kriterien nicht zu, zeigt der video-detective ein **rotes** Signallämpchen in den jeweiligen Spalten. Sobald eines der kontrollierten Video-Files den Kriterien nicht entspricht, leuchtet das Signallämpchen auf der linken Seite ebenfalls **rot**. Diese Richtlinien unterliegen laufenden Veränderungen und Verbesserungen, natürlich wird der video-detective auch dementsprechend angepasst. Die Programmierung dahinter wurde im Punkt Grundlegende Struktur und Vorgehensweise erläutert.

Zusätzlich wird eine Attention-E-Mail verschickt und/oder ein Attention-Dokument im Projektordner erzeugt. Diese Attention-Funktion wird zur Fehler-Visualisierung genutzt. Dadurch ist es möglich zeitnahe Gegenmaßnahmen einzuleiten. Die Attention-Dokumente wurden auf die Firmeninterne Aufzeichnungssoftware hin erstellt. Das Attention-Dokument wird der Attention-Email als Anhang angehängt. Wie diese Attention-E-Mail nun im Detail aussieht, wird im nächsten Punkt deutlich.

9.1 Beispiel für ein Attention-E-Mail (Fehlermeldung)

Tritt bei der Kontrolle ein fehlerhaftes File auf und das Signallämpchen leuchtet rot, wird auf Wunsch eine sogenannte Attention-E-Mail versendet. Diese Attention-Mail beinhaltet eine genaue Beschreibung des Fehlers und eine Anleitung zum Beheben des Fehlers. Die E-Mails werden mittels eines Gmail-Accounts versendet. Eine Attention-Mail sieht beispielsweise wie folgt aus:

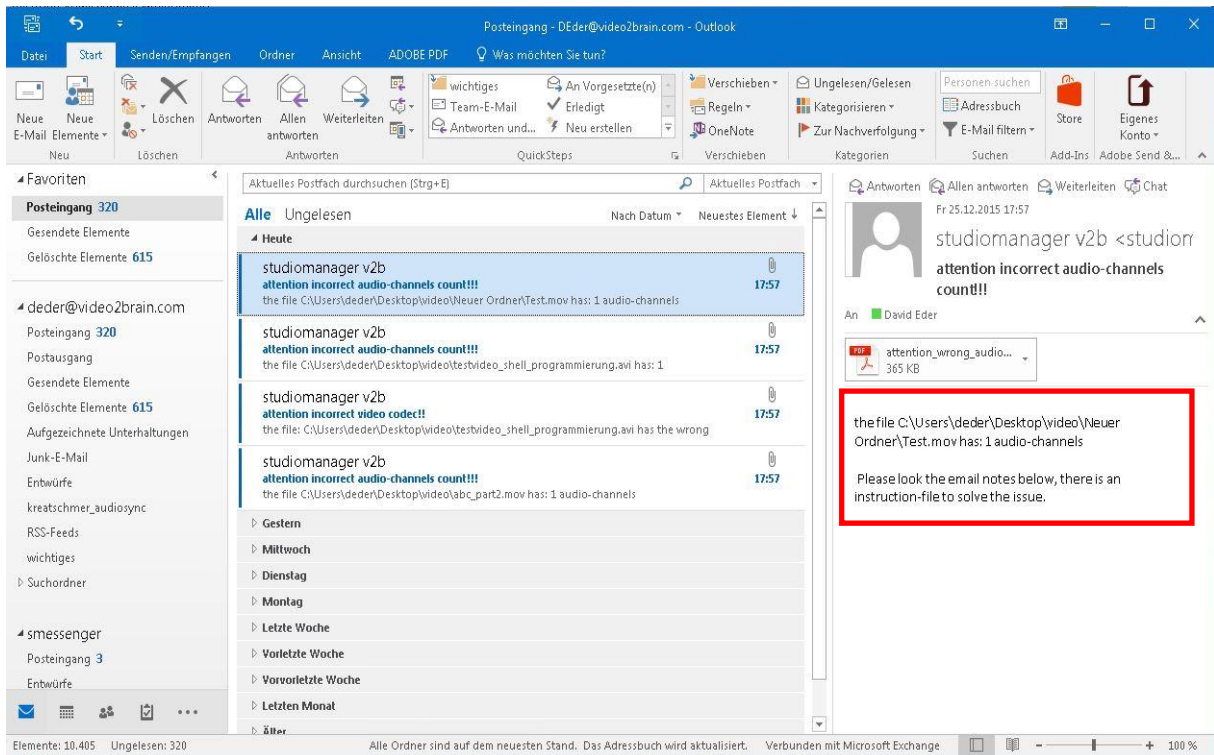


Abbildung 36: Outlook Attention Email Darstellung

Es wird für jeden Fehler eine eigens darauf zugeschnittene Attention-E-Mail versendet. Im Beispiel hier sieht man, dass die Anzahl der Audiospuren fehlerhaft ist. Im Anhang befindet sich auch gleich die Anleitung (das Attention-Dokument), um den Fehler zu beheben. Eine Erläuterung des Attention-Dokuments erfolgt im nächsten Abschnitt.

Das jeweilige Attention-Dokument eines auftretenden Fehlers wird im nächsten Punkt näher betrachtet.

9.2 Beispiel für ein Attention-Dokument (Fehlermeldung)

Das Attention-Dokument des oberen Beispiels sieht wie folgt aus:

Attention!!!
Audio Settings Issue

There is something wrong with your recording settings.
Please follow these instructions:

- Please make sure that the **audio section** below has these settings:



- Record a short Testvideo and see if you get this message again. If you do not get this message again you can continue recording.

Abbildung 37: Attention Dokument Darstellung

Genauso wie bei den Attention-E-Mails wird für jeden einzelnen Fehler ein Attention-Dokument generiert. Hierbei wird unterschieden, ob es sich um ein .mov- oder ein .avi-File handelt. Die Unterscheidung ist wichtig, damit auch die richtige Anleitung als Anhang verschickt wird (bei Bandicam sehen die Einstellungen anders aus als bei iShowU). Zusätzlich können diese Attention-Mails auch direkt in den Projektordner gelegt werden, sofern Schreibrechte vorliegen.

10 Zusammenfassung und Ausblick

Zusammenfassend wird an dieser Stelle über die erreichten Ziele resümiert und ein Ausblick zu dieser Arbeit gegeben.

10.1 Ergebnisse

Die erreichten Ziele:

- Das Erkennen der fehlerhaften Aufnahmen wurde erheblich verbessert.
- Der gesamte Workflow des Videoproofings und der Indexerstellung wurde vereinfacht/beschleunigt.
- Die Kontrolle findet komplett im Hintergrund statt und behindert den Trainer in keinsten Weise bei der Aufzeichnung.
- Die Oberfläche und sämtliche Anleitungen wurden in Englisch verfasst und überprüft.
- Die gesammelten Daten werden über die automatisierte Indexerstellung grafisch dargestellt.
- Der video-detective wird im Content-Team sowie im Postproduktion-Team verwendet.
- Es wurde zu jedem auftretenden Fehler eine Anleitung entwickelt.

Der video-detective wird im Unternehmen eingesetzt und ist Bestandteil der Videokontrolle.

10.2 Ausblick

Wie sieht nun die Zukunft des video-detective aus?

Das Tool wurde mit sehr großer Begeisterung im Unternehmen angenommen. Zugleich wurden einige Erweiterungsvorschläge eingereicht, die hier aufgezählt werden sollen:

- Das Postproduktion-Team verwendet noch andere kleine Tools, diese werden in Zukunft zur Gänze vom video-detective abgedeckt werden.
- Die Kontrolle wird noch um ein paar Kriterien erweitert (Pegelmessung, Lautstärkenanpassung, ...).
- Die Indexerstellung wird noch weiter optimiert.
- Zusammenarbeit mit den amerikanischen Kollegen, einheitliche Lösung für den Informationsfluss zwischen Trainer und Content-Team

Das gesamte Tool wird mit ASP.Net online virtualisiert werden, somit kann es auf jedem System verwendet werden.

Anlage A

Die vorliegenden Screenshots zeigen die gesamte Benutzeroberfläche während eines Anwendungsprozesses.

Gesamte Oberfläche des video-detective

Die gesamte Oberfläche des video-detective sieht im control-Tab wie folgt aus:

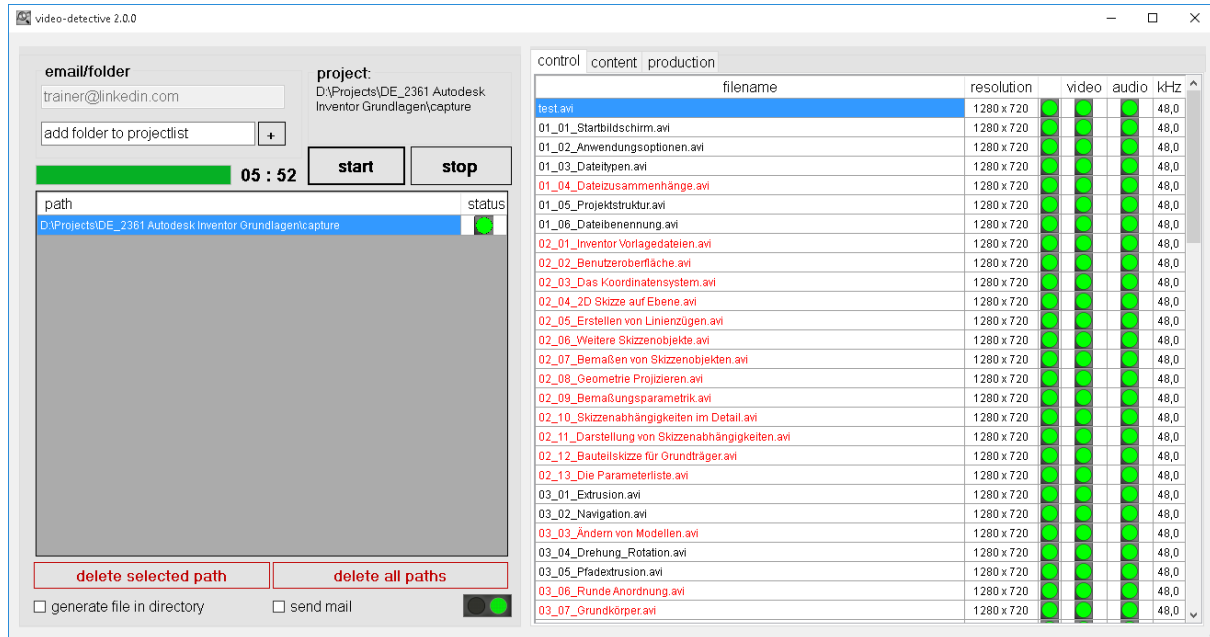


Abbildung 38: video detective gesamte Oberfläche im gestarteten Modus, Control Tab

Im content-Tab sieht die gesamte Oberfläche folgendermaßen aus:

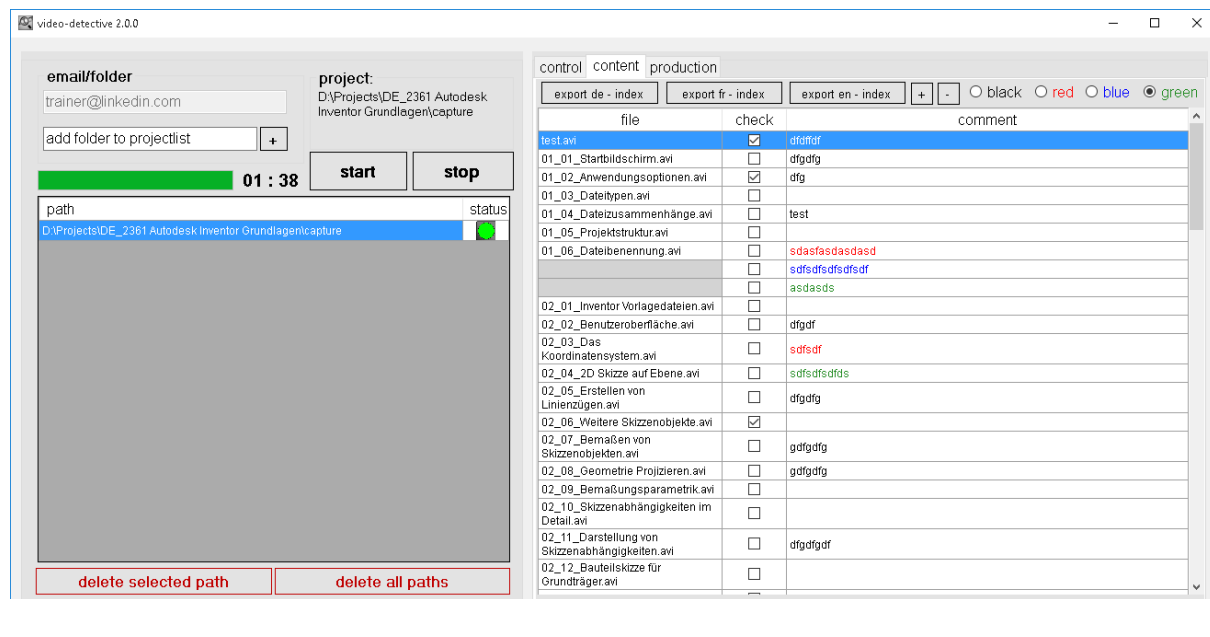


Abbildung 39: video detective gesamte Oberfläche im gestarteten Modus, Content Tab

Die gesamte Oberfläche des video-detective sieht im production-Tab wie folgt aus:

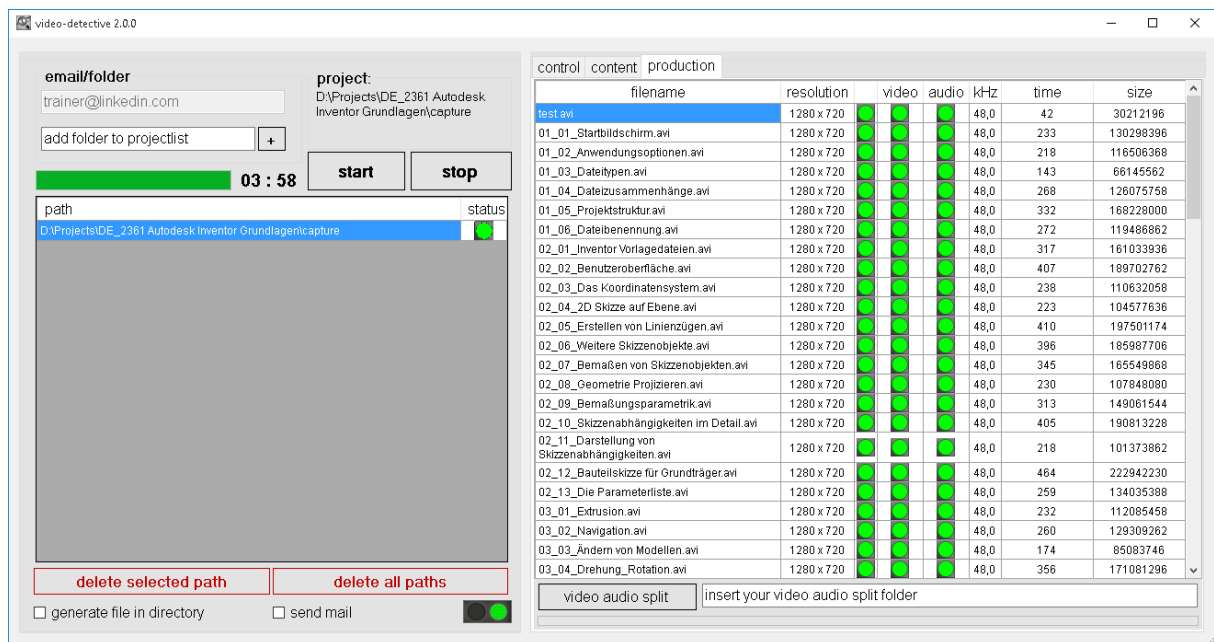


Abbildung 40: video detective gesamte Oberfläche im gestarteten Modus, Production Tab

Anlage B

In dieser Anlage wird nun der gesamte Ablauf einer Kontrolle nochmals verdeutlicht.

- E-Mail-Adresse eingeben
 - Die Standardadresse lautet Benutzer + „@linkedin.com“
 - Versenden der Mails erfolgt über Gmail-Account
- Projekt/e hinzufügen
 - Einfügen über Browserdialog oder Copy and Paste
 - Gültigkeit wird überprüft
- Eventuell Zusatzoptionen wählen
 - E-Mail versenden
 - Anleitung direkt im Projektordner erzeugen
- Bereich auswählen
 - Control
 - Content
 - Production
- Starten
 - Die komplette Kontrolle starten, Klick auf start-button
 - Einzelkontrolle mit Auflistung, direkter Klick auf Pfad
- Kontrolle läuft
 - Resultat wird angezeigt
- Ergebnis begutachten
 - Gegebenenfalls Maßnahmen einleiten
 - Kommentare einfügen
 - Index-Excel-Liste exportieren
 - XML-Datei exportieren

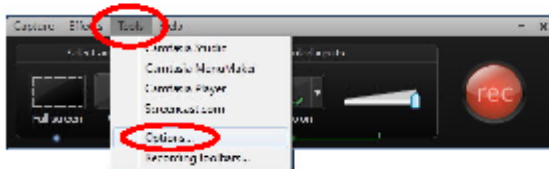
Anlage C

An dieser Stelle wird die Windows-Anleitung gezeigt. Dieses Attention Dokument wird generiert, sobald der Codec-falsch eingestellt wurde.

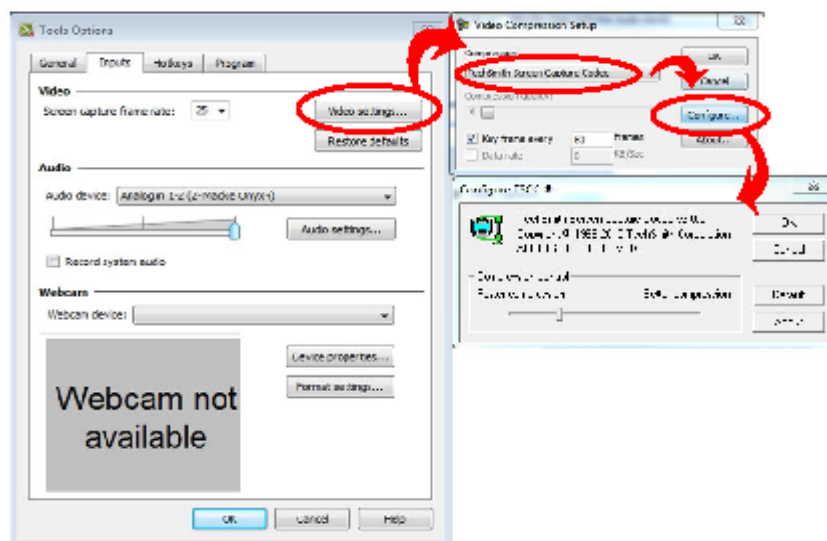
Attention!!!

There is something wrong with your recording settings.
Please check the following settings:

- In Camtasia Recorder Go to **Tools > Options**



- Go to the **Inputs** Tab and click on **Video settings**. In the next dialog box under **Compressor** choose **TechSmith Screen Capture Codec** and click **Configure**. Under **Compression control** push the slider to the position shown below. Afterwards click **OK** in all Dialog Boxes.



- Record a short Testvideo and see if you get this message again. If you do not get this message again you can continue recording.

Anlage D

An dieser Stelle wird die Mac-Anleitung verdeutlicht. Dieses Attention Dokument wird generiert, sobald eine Asynchronität während der Aufzeichnung auftritt.

Attention!!!

Sync Issue

There is something wrong with your recording settings.
Please follow these instructions:

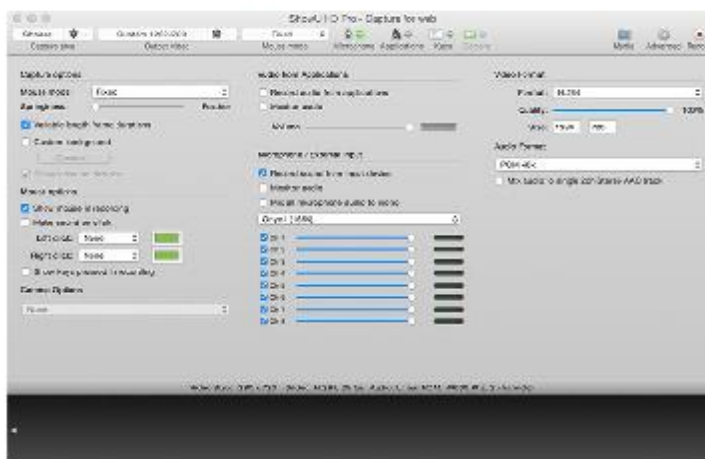
- Please power off the Mackie Audio Mixer by pressing the Power Switch in the back of the device.



- Afterwards restart the Mac
- Log on with the „trainermac“ account as usual.
- Once logged in please Power on the Mackie Audio Mixer in the back of the device.



- Start „iShowu HD“ and record a short Testvideo.
Make sure the settings in „iShowu HD“ look like this:



- If you do not get this message again you can continue recording.

Literaturverzeichnis

- [Rheinwerk Computing2016] Java ist auch eine Insel von Christian Ullenboom. URL: <<http://openbook.rheinwerk-verlag.de/javainsel/>>, verfügbar am 07.09.2016
- [Rheinwerk Computing2016] C# 6 mit Visual Studio 2015 von Andreas Kühnel. URL: <https://www.rheinwerk-verlag.de/professionell-entwickeln-mit-c-6-und-visual-studio-2015_3994/>, verfügbar am 14.08.2016
- [Carl Hanser Verlag GmbH & Co. KG2015] Der C++-Programmierer: C++ lernen – professionell anwenden – Lösungen nutzen von Ulrich Breyman, URL: <<https://fachbuch.hanser-ebooks.de/ebook/bid-1899877-der-c-programmierer-c-lernen-professionell-anwenden-loesungen-nutzen.html>> verfügbar am 07.04.2015
- [mediaarea2016] Jérôme Martinez, MediaInfo, URL: <<https://mediaarea.net/de/MediaInfo>>, verfügbar am 28.09.20016
- [FFmpeg2016] FFmpeg; FFmpeg multimedia framework, URL: <<https://ffmpeg.org/about.html>>, verfügbar am 19.09.20016
- [Microsoft2016] Microsoft C# Lernprogramme, URL: <[https://msdn.microsoft.com/de-de/library/aa288436\(v=vs.71\).aspx](https://msdn.microsoft.com/de-de/library/aa288436(v=vs.71).aspx)>, verfügbar am 17.10.2016
- [Hauser2009] Mathias Hauser: Csharpme.de – Das etwas andere C# URL: <<http://www.csharpme.de/>>>, verfügbar am 23.11.20016
- [Weber2007] Christian Weber; URL:<[http://www.lgis.informatik.uni-kl.de/archiv/wwwdvs.informatik.uni-](http://www.lgis.informatik.uni-kl.de/archiv/wwwdvs.informatik.uni-kl.de/)

- I.de/courses/seminar/WS0203/ausarbeitung1.pdf>, verfügbar am 20.10.20016
- [Techsmith2016] Camtasia Videoeditor und Screen-Recorder. URL: < <https://www.techsmith.de/camtasia.html> >, verfügbar am 18.11.2016
- [shinywhitebox2016] ISHOWU HD PRO Realtime screen capture for your Mac.URL: < <https://shinywhitebox.com/ishowu-hd-pro>>, verfügbar am 01.11.2016
- [VideoInspector2017] Video Playing Trouble Annihilated VideoInspector URL: < <https://www.kcsoftwares.com/?vtb> >, verfügbar am 24.01.2017
- [video-flash2007] . www.video-flash.de Alles rund um Web & Mobile: FLVCheck. URL: < <http://www.video-flash.de/index/flvcheck-tool>>, verfügbar am 24.01.2017
- [Chip.de2017] GSpot, GSpot Version 2.70a URL: <http://www.chip.de/downloads/GSpot_13008930.htm >, verfügbar am 24.01.2017
- [overblog2011] C++ vs Java: Vor- und Nachteile. URL: < https://de.overblog.com/C_vs_Java_Vor_und_Nachteile-1228321776-art185420.html>, verfügbar am 24.01.2017

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Graz, am

.....
David Eder