Bildanalyse Software

Betreuer: Matthias Bachmann

Studierende: Roger Bollmann

Datum: September 2015

Inhaltsverzeichnis

[Aufgabenstellung 3](#_Toc415132699)

[Thema 3](#_Toc415132700)

[Mitwirkende Personen 3](#_Toc415132701)

[Ausgangslage 3](#_Toc415132702)

[Ziel der Arbeit 3](#_Toc415132703)

[Aufgabenstellung 3](#_Toc415132704)

[Erwartete Resultate 3](#_Toc415132705)

[Recherche 4](#_Toc415132706)

[Markanalyse 4](#_Toc415132707)

[ExperVision 4](#_Toc415132708)

[Maestro Recognition Server 4](#_Toc415132709)

[MaxxVault LLC 5](#_Toc415132710)

[Methoden zum Abfangen von Bilder 5](#_Toc415132711)

[Übertragungsmethoden 5](#_Toc415132712)

[Textanalyse Software 5](#_Toc415132713)

[Anforderungen 5](#_Toc415132714)

[Konzept 5](#_Toc415132715)

[Proof of Concept „PoC“ 5](#_Toc415132716)

[Sender 6](#_Toc415132717)

[Empfänger 6](#_Toc415132718)

[Translator 6](#_Toc415132719)

[Testing 6](#_Toc415132720)

[Unit Test 6](#_Toc415132721)

[User Akzeptanz Tests 6](#_Toc415132722)

[Fazit 6](#_Toc415132723)

# Aufgabenstellung

## Thema

Abfangen von Bilder von einem Webserver und umwandeln in Text zur Analyse.

## Mitwirkende Personen

Studierender Roger Bollmann

Betreuungsperson Matthias Bachmann

## Ausgangslage

Als Reaktion auf den Diebstahl von zahlreichen Kundendaten, sowie des anschliessenden Verkaufs von illegalen CDs, hat die Eidgenössische Finanzmarktaufsicht (FINMA) alle Schweizer Finanzinstitute aufgefordert, den Zugriff auf Kundendaten verstärkt zu überwachen. Das Unternehmen für das ich arbeite, integriert nun eine Überwachungssoftware, die den HTTP-Traffic nach Kundendaten absucht. Die Software ermöglicht jedoch lediglich eine systematische Analyse von Text. Bilder, die ebenfalls Kundendaten enthalten könnten, werden ignoriert. Grund dafür ist, dass auf dem Markt keine wirklich effizienten Bildanalyse-Softwares angeboten werden.

## Ziel der Arbeit

Das Ziel der Arbeit ist ein Programm zu entwickeln, welches die versendeten Bilder von einem Webserver in Text umwandelt, um eine Analyse durch die Überwachungssoftware zu ermöglichen. Das Programm sollte aus zwei Teilen bestehen, einem Sender und einem Empfänger. Der Sender wird zuständig sein für das Versenden der Bilder, wobei der Empfänger die Bilder empfangen und in Text umwandeln soll.

## Aufgabenstellung

1. Recherche
   1. Analyse von verschiedenen Methoden zum Abfangen von Bilder auf einem Webserver
   2. Analyse von verschiedenen Übertragungsmethoden
2. Anforderungen ermitteln und dokumentieren
3. Erstellen eines Konzepts
   1. Vergleichen von verschiedenen Textanalytik-Applikationen
4. Erstellen eines Proof of Concepts
   1. Implementierung eines Senders, welcher die Bilder abfängt und an den Empfänger weiterleitet
   2. Implementierung eines Empfängers, welcher die Bilder empfängt und in Text umwandelt
5. 5.Fazit

## Erwartete Resultate

1. Durchführen einer Recherche
2. Definition einer Anforderungsanalyse
3. Ausarbeitung eines Konzept
4. Durchführen eines Proof of Concept „PoC“
5. Ausführung von Tests
6. Persönliches Fazit

# Recherche

## Markanalyse

### ExperVision

<http://www.expervision.com/ocr-software>

Negative

* + - 5200$ kostet die Lizenz für OpenRTK (SDK) Developer Tool, damit man selber entwickeln kann
    - Muss alles manuell eingestellt werden, Directory welche gescannt werden sollen. Vor allem für Dokumentenablagen und Emails zu empfehlen
    - Keine out of the Box Automatisierung, man muss sehr viel selber Programmieren, damit man das gewünscht Resultat erreicht.
    - Ein reines OCR Tool
    - Es können nur Bilder umgewandlet werden, jedoch kann nicht nachvollzogen werden, wer dieses Bild gesehen oder angeschaut hat
    - Keine Lösung zum anzeigen welcher User was gesehen hat
    - Keine Lösung für veschickte Bilder von Webserver nur Client oder Netzlauftwerk Scanning.

Positiv

* + - Unterstützt sehr viele Sprachen
    - Sehr viel Erfahrung (Gründung 1987)
    - Verfügt über viele Funktionen (Client/Server Scanning, Cloud-Service)
    - Unterstütz auch Mobile Bilder Umwandlung
    - Verfügt über ein API

### Maestro Recognition Server

<https://www.cvisiontech.com/products/general/maestro-recognition-server.html?lang=eng>

Negativ

* + - Keine Lösung zum anzeigen welcher User was gesehen hat
    - Keine Lösung für veschickte Bilder von Webserver nur Client oder Netzlauftwerk Scanning.

Positiv

* + - Ist sehr schnell auch bei grossen Datenmengen
    - Sobald ein Bild in einem Verzeichnis auftaucht, wird es automatisiert umgewandelt
    - Vorallem zum Umwandlen von PDF’s in suchbare Texte.

### MaxxVault LLC

<http://www.primerecognition.com/maxxvault.htm>

Architektur: <http://www.primerecognition.com/prime_network.htm>

Negativ

* + - Keine Lösung zum anzeigen welcher User was gesehen hat
    - Keine Lösung für veschickte Bilder von Webserver nur Client oder Netzlauftwerk Scanning.
    - Lizenz startet bei 4500$ für Job Server und zusätzlich noch pro Verifikations Server

Positiv

* + - Speichert die Daten auf einem sicheren und einfach zugreifbaren System.
    - Um möglichst viele Fehler auszuschliessen werden die Bilder durch mehrere ORC Engines umgewandlet.
    - Die Software hat ein Failover, falls es ein Fehler gibt bei einer OCR Engine, wird es ignoriert und für das nächste Bild vorbereitet.
    - Unterstütz 11 Sprachen
    - Kann durch Hinzufügen von Verfikations Server performanter Bilder in Text umwandeln (Cluster System)

Momentan sieht es so aus, dass es sehr gute Lösungen im Bereich ORC (Optical Character Recognition) gibt, jedoch keine welche meiner Anforderung entspricht. Bei allen Produkten gibt es einen Viewer, welche alle hinzugefügten Netzlaufwerke scannen und automatisiert prozessieren kann. Die hat zur Folge, dass bei einem neu hinzugefügten Netzlaufwerk, die Konfiguration angepasst werden muss. Bei grösseren Unternehmen möchte man viele Änderungen in der Produktion jedoch vermeiden.

## Methoden zum Abfangen von Bilder

## Übertragungsmethoden

## Textanalyse Software

# Anforderungen

# Konzept

# Proof of Concept „PoC“

## Sender

## Empfänger

## Translator

# Testing

## Unit Test

## User Akzeptanz Tests

# Fazit