

Titel der großen Studienarbeit

Projektarbeit

des Studienganges **Angewandte Informatik / Betriebliches Informationsmanagement**
an der Dualen Hochschule Baden-Württemberg Mannheim

von

Maximilian Ludwig, Kevin Wrona, Fabian Brandmüller

22. Januar 2020

Bearbeitungszeitraum
Betreuer der DHBW

23.09.2019 - 20.04.2019
Eckhard Kruse

Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema: "Titel der großen Studienarbeit" selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort

Datum

Unterschrift

Abstract

Inhaltsverzeichnis

I. Einleitung	I
I.1. DevOps	I
II. Theorie	I
III. Methode	II
III.1. Vorgehen im ersten Teil	III
III.2. Vorgehen im zweiten Teil	III
IV. Ergebnis	III
IV.1. test1	IV
IV.1.1. test2	IV
IV.1.1.1. test3	IV
V. Diskussion	IV

Abbildungsverzeichnis

I.1. DevOps Lifecycle	
Quelle: https://nareshit.com/wp-content/uploads/2018/09/DevOps-LifeCycle-Features-Naresh-IT.png	II

Abkürzungen

- VCS Version Control System
- CI/CD Continuous Integration/Continuous Deployment
- API Application Programming Interface
- DSL Domain Specific Language

KPI Key Performance Indicator

CRI Container Runtime Interface

JCasC Jenkins Configuration as Code

PV Persistent volume

PVC Persistent volume claim

NFS Network File System

iSCSI internal Small Computer System Interface

I. Einleitung

”Several researchers have stated that facial expression recognition appears to play one of the most important roles in human communication”¹ Dieses Zitat von Katherine B. Leeland gibt einen Einblick in die Relevanz der Emotionserkennung für den Menschen. Dabei ist diese Relevanz nicht erst in jüngerer Zeit entstanden. Bereits Darwin stellte die Frage, ob von den Gesichtsausdrücken einer Person nicht auch der emotionale Zustand abgeleitet werden kann.² Nun ist diese Studienarbeit kein wissenschaftliches Werk über die psychologischen Emotionen, die hinter verschiedenen Gesichtsausdrücken stehen, jedoch thematisiert diese Arbeit dieses Themengebiet indirekt. Es geht hier darum eine künstliche Intelligenz zu entwickeln, die erkennen kann in wie weit ein Pokerface einer Person anhand dessen Gesichtsausdruck vorliegt. Diese Aufgabe scheint nun erst ein Mal nicht viel mit dem oben bereits genannten Themengebiet zu haben, dem ist jedoch nicht so. Ein Pokerface wird im Allgemeinen als ein emotionsloser Gesichtsausdruck definiert. Dies impliziert, dass eine Person mit Pokerface keine Emotion zu erkennen gibt. Daher könnte bei einer nicht messbaren Emotion ein Pokerface vorliegen. Dies mittels KI zu testen und einen sogenannten ”Pokerface Detektor” zu entwickeln ist daher Ziel dieser Arbeit. Mit diesem Pokerface Detektor sind verschiedenste Einsatzmöglichkeiten denkbar, im Folgenden einige mögliche Szenarien:

- Polizeiverhöre

Es ist denkbar, dass ein Detektor wie der, der in dieser Arbeit entwickelt werden wird bei Polizeiverhören eingesetzt wird. Für die Beamten kann es nicht immer direkt ersichtlich sein, ob der verhörte mit einem Pokerface lügt, oder doch die Wahrheit sagt. Da die Möglichkeiten dies zu prüfen ebenfalls nicht sehr zahlreich sind, wäre es eine Vereinfachung für die Beamten wenn eine einfache Webcam zusammen mit einem Computer reichen würde um die Lügner zu entlarven.

- Gerichtsverhandlungen

Das zweite Einsatzgebiet ist ähnlich zu dem ersten. Bei Gerichtsverhandlungen gelten die gleichen Voraussetzungen wie bei einem Verhör der Polizei. Zwar müssen die hier vorgeladenen eine eidstattliche Erklärung abgeben nur die Wahrheit zu sagen, jedoch

¹Vgl. Leeland, *Face Recognition: New Research*, S. 1.

²Vgl. Leeland, *Face Recognition: New Research*, S. 2.

ist zu bezweifeln ob dies auch jeder so handhabt. Nun soll nicht der Eindruck entstehen dass das hier gebaute Werkzeug ein Lügendetektor ist. Es ist ebenfalls nicht möglich, dass von einem Pokerface immer auf eine Lüge geschlossen werden kann. Jedoch ist ein Pokerface ein Zeichen dafür, dass sich diese Person ihren emotionalen Zustand nicht anmerken lassen möchte. Und dies wiederum deutet eher daraufhin dass die Person nicht die Wahrheit sagt oder nur teilweise. Abgesehen von Einsatzgebieten die zur Entlarvung von Lügen führen kann auch ein klassischeres Szenario verwendet werden:

- Pokerspiel

Es ist anzunehmen, dass der erste Begriff der mit dem Wort Pokerface in Verbindung gebracht wird, das Pokerspiel selber ist. Und auch in diesem kann ein Pokerface Detektor nützlich sein. So kann ein Mitspieler zum Beispiel mittels einer Kamera das Gesicht des Gegenübers scannen und analysieren ob ein Pokerface vorliegt oder nicht, und dementsprechend agieren. Nachdem verschiedene Anwendungsszenarien beleuchtet wurden wird im Folgenden die konkrete Forschungsfrage beleuchtet.

I.1. Aufgabenstellung

In diesem Abschnitt soll nun noch die konkrete Forschungsfrage behandelt werden. Das Projekt selber wird an der DHBW in Mannheim durchgeführt und betreut von Prof. Dr. Erckhard Kruse.

Wie eingangs erwähnt soll mittels künstlicher Intelligenz ein Pokerface erkannt werden. Dafür wird wiederum eine Bilderkennungssoftware notwendig, die ebenfalls angefertigt werden soll. Dieses Softwareprodukt soll Bilder die mit einer Webcam aufgenommen werden nach ihren Emotionen analysieren. Je nachdem welche Emotionen gezeigt werden wird dann von der Software ein Rückschluss gezogen auf ein eventuell vorhandenes Pokerface. Ein konkretes Einsatzgebiet nach Abschluss der Entwicklung ist nicht vorgesehen, da es sich um ein Forschungsprojekt handelt. jedoch sind wie bereits beschrieben einige verschiedene Einsatzmöglichkeiten denkbar, an die das Werkzeug leicht angepasst werden kann.

Gliederung einleitung

- Einstieg + Relevanz des Themas
- use cases Relevanz der Use Cases

Gliederung

- unterschied es gibt face recognition emotion recognition
- use cases dafür

DevOps LifeCycle and Features

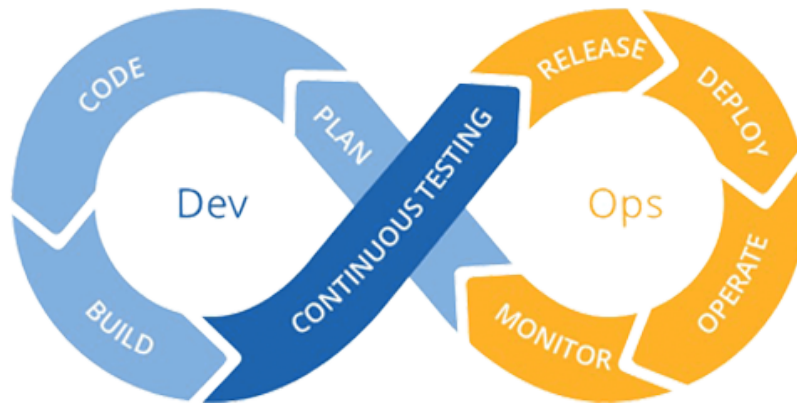


Abbildung I.1.: DevOps Lifecycle

Quelle: <https://nareshit.com/wp-content/uploads/2018/09/DevOps-LifeCycle-Features-Naresh-IT.png>

- unser use case wie er reinpasst
- emotion rec = absplaltung der face regocnition
-

³.

I.2. DevOps

II. Theorie

Seite 3 ganz viele Emotionsdefis

³Ansible2.

III. Methode

Diese Arbeit besteht aus zwei größeren Teilen. Zu nächst einmal den ersten, der die Entwicklung eines Tools beinhaltet, welches eine funktionierende Continuous Integration/Deployment Pipeline automatisiert aufsetzt. Der zweite Teil hingegen befasst sich basierend auf Aufgabe eins damit, dieses Tool automatisiert auf einem Kubernetes Cluster zu deployen, indem die dafür benötigten Applikationen eingerichtet und hochgefahren werden. In diesem zweiten Teil der Arbeit soll es dabei weniger um das eigentliche Aufsetzen und Konfigurieren von Kubernetes gehen, sondern um das automatisierte Deployment des Tools aus Aufgabe 1. Beide Arbeitsteile sollen methodisch mit der MoSCoW Priorisierung bearbeitet werden. Diese Art der Priorisierung teilt die zu bearbeitenden Anforderungen in vier Kategorien ein:¹

- Must - Core Anforderungen die unbedingt umgesetzt werden müssen
- Should - Anforderungen die ebenfalls umgesetzt werden müssen, jedoch um Nachhinein noch durch Change Request verändert werden können.
- Could - Anforderungen die nach den Must und Should Anforderungen umgesetzt werden sollen, sofern noch Ressourcen und Zeit vorhanden sind um diese zu bearbeiten
- Won't - Anforderungen die nicht in diesem Projekt bzw. Release erfolgen, jedoch in einer zukünftigen Version bearbeitet werden sollen.

III.1. Vorgehen im ersten Teil

Die erste Aufgabe, die automatisierte Erstellung einer CI/CD Pipeline, soll nun die folgenden funktionalen, so wie nicht-funktionalen MoSCoW Anforderungen erfüllen:

- Must
 - Kommunikation zwischen den Applikationen ist gewährleistet
 - Indikation von Pipeline und Analysestatus
 - Die CI/CD Pipeline wird automatisch eingerichtet

¹Projektmanagement.

- Should
 - Dynamisches, benutzerfreundliches Tool zur Einrichtung der Umgebung
- Could
 - Jenkins Pipelineskript soll mit konfigurierbar sein
- Won't
 - Projektspezifische Testauswahl
 - Projektspezifisches Deployment der getesteten Applikation

Dabei sollen die einzelnen Anforderungen entsprechend ihrer Priorität abgearbeitet werden. So kann am Ende der Erfolg der Arbeit deutlich besser eingeordnet werden

III.2. Vorgehen im zweiten Teil

Bezüglich des zweiten Teils der Arbeit, dem automatisierten Deployment der Applikation, sollen folgende Anforderungen mit Ihrer Priorisierung aufgestellt werden:

IV. Ergebnis

IV.1. test1

IV.1.1. test2

IV.1.1.1. test3

Gitea Setup **V. Diskussion**

Das nunmehr letzte Kapitel soll sich mit der kurzen Zusammenfassung der Ergebnisse des letztens Teils und deren Bewertung widmen. Des Weiteren sollen die angewandten Methoden reflektiert werden, offene Fragen beantwortet und auch weitere Punkte

aufgezeigt werden die verbessert oder noch implementiert werden können. Dazu soll zunächst die Ergebnisse kurz zusammengefasst werden.

Literatur

Literaturquellen

- [1] Katherine B. Leeland. *Face Recongnition: New Research*. 1. Aufl. Nova Science Publishers INC, 2008. ISBN: 978-16045646625.