

Wolfsabschreckung

Future Internet / Digitales Kulturerbe
Anwendungspraktikum & Seminar

Projekt des Studiengangs I-16
Fachbereich Automatisierung und Informatik
Hochschule Harz

Autoren: Alexander Luft (23945), Dirk Neumann (24160),
Fabian Theuerkauf (23947)

Dozent: Prof. Dr. O. Drögehorn

Veranstaltung/Semester: Future Internet/ Digitales Kulturerbe
Anwendungspraktikum
Sommersemester 2019

I Inhaltsverzeichnis

1	Einleitung	7
2	Problemstellung	7
3	Aktueller Stand	7
3.1	Politische Situation Deutschland	7
3.2	Beispielhafter Einfluss des Wolfes	8
3.3	Herdenschutz.....	9
3.3.1	Weideschutzzäune	9
3.3.2	Herdenschutzhunde.....	10
3.3.3	Halsbänder für Wölfe	10
4	Projektziele	11
4.1	Soll-Kriterien	11
4.2	Kann-Kriterien	11
4.3	Projektabgrenzung	12
5	Projektplanung	12
5.1	Gantt Diagramm.....	14
5.2	Netzplan	15
6	Entwicklungswerzeuge	16
7	Architekturentwurf.....	17
7.1	Logische Architektur	17
7.2	Technische Architektur.....	18
8	Logischer Ablauf im Steuerungsmoduls.....	19
9	Implementierung	20
9.1	Erstkonfiguration	20
9.1.1	Raspberry Pi Konfiguration	20
9.1.1.1	Betriebssystem.....	20
9.1.1.2	Einrichten von SSH	20
9.1.1.2.1	SSH aktivieren mit Maus und Tastatur.....	20
9.1.1.2.2	SSH aktivieren ohne Maus und Tastatur	22
9.1.1.2	Einrichtung von C#	22
9.1.1.3	Prototypen bauen	24
9.1.1.3.1	Verkabelung Druckknopf	24
9.1.1.3.2	Verkabelung LED-Band	25
9.1.1.3.3	Verkabelung Kamera.....	27
9.1.1.3.4	Gehäuse	28
9.1.1.4	Einrichtung des Clouddienstes	29

9.2	Implementierung der Softwaremodule.....	30
9.2.1	Schnittstelle zum Clouddienst	30
9.2.2	Tags ausgeben.....	31
9.2.3	Konfigurationsdatei	32
9.2.4	Soundausgabe– und Lichtabstrahlung-Thread	33
10	Modultests.....	34
10.1	Testdurchführung	34
10.1.1	Erstellen der Testdaten	34
10.1.2	Klassifizierung der Testbilder.....	37
10.1.3	Abspeichern der Ergebnisse	38
10.2	Testauswertung	38
10.2.1	Aufbereitung der Ergebnisdaten	38
10.2.2	Analyse der Ergebnisdaten	39
10.2.3	Handlungsempfehlung	43
10.3	Fehlerbetrachtung/ Kritische Einschätzung der Tests.....	43
11	Systemtest	45
11.1	Testumgebung	45
11.2	Erster Systemtest.....	47
11.3	Zweiter Systemtest.....	48
12	Diskussion der Ergebnisse	49
13	Fazit.....	49
14	Ausblick	50
15	Quellen.....	51
15.1	Literaturquellen.....	51
15.2	Bildquellen	53
16	Anhang	55
16.1	Glossar	55
16.2	Verwendete Programme.....	56
16.3	Projektdateien	56

II Abbildungsverzeichnis

Abbildung 1: Darstellung des Projektplanes in Form eines Gantt Diagrammes	14
Abbildung 2: Zweiter Teil des Gantt Diagramms.....	14
Abbildung 3: Netzplan inklusive Legende	15
Abbildung 4: Logische Architektur des Prototypen.....	17
Abbildung 5: Technische Architektur des Prototypen	18
Abbildung 6: Ablaufdiagramm des Steuerungsmoduls.....	19
Abbildung 7: Hauptmenü der Raspberry Pi Konfiguration.....	21
Abbildung 8: Interface Optionen des Raspberry Pi.....	21
Abbildung 9: Konfigurationsfenster zum aktivieren des SSH-Servers auf den Raspberry Pi	21
Abbildung 10: HelloWorld.cs Datei mit C#-Code	22
Abbildung 11: Beispielcode für ein C#-Fensterprogramm	23
Abbildung 12: Druckknopf mit Kabelverbindung zum Raspberry Pi	24
Abbildung 13: Raspberry Pi mit Anschlüssen ans das Lichtband	25
Abbildung 14: Konfigurationsmenü des Raspberry Pi.....	26
Abbildung 15: Kamera Flachbandkabel in CSI-Anschluss am Raspberry Pi.....	27
Abbildung 16: Raspberry Pi Gehäuse samt anbauten.....	28
Abbildung 17: Website des Azure-Vision-Service mit anzuwählenden Menüpunkt.....	29
Abbildung 18: Webansicht des abonnierten Service	29
Abbildung 19: Inhalt einer .ini Konfigurationsdatei für Python-Skripte.....	32
Abbildung 20: Download der 64bit-Version von „XnViewMP“ für Windows als .zip-Datei.....	35
Abbildung 21: Laden der Originalbilder für die Stapelverarbeitung in "XnViewMP"	35
Abbildung 22: Auswählen der einzelnen Bearbeitungsschritte	36
Abbildung 23: Festlegen von Speicherort und Name der bearbeiteten Bilder und optionales Speichern der Routine.....	36
Abbildung 24: Aufbereitete Ausgangsdaten	40
Abbildung 25: Farbmethode Daten gemittelt über Kontraststufen.....	41
Abbildung 26: ANOVA Farbmethode Daten gemittelt über Kontraststufen.....	42
Abbildung 27: Position des Prototypen, gehalten von einem der Projektgruppenmitglieder - mit Handykamera aufgenommen.....	45
Abbildung 28: Aussichtsplattform des Wolfgeheges - mit Handykamera aufgenommen.....	46
Abbildung 29: Wölfe im ersten Systemtest - mit Handykamera aufgenommen	47
Abbildung 30: Wölfe im zweiten Systemtest	48

III Tabellenverzeichnis

Tabelle 1: Projektplan.....	12
Tabelle 2: Vergleich der Programmiersprachen.....	16
Tabelle 3: Benennungen der Testbilder	37
Tabelle 4: Übersicht der Stempel aus MSA-CV, die mit Wölfen in Verbindung stehen.....	39
Tabelle 5: Kennzahlen im Kastendiagramm	41
Tabelle 6: Fehlerlösungsansätze	48

IV Abkürzungsverzeichnis

Abkürzung	Bedeutung
HSH	Herdenschutzhunde
JSON	JavaScript Object Notation
VSCE	Visual Studio 2019 Community Edition
MSA-CV	Microsoft Azure Computer Vision
PG	Programm
PLA	Polylactide
SPI	Serial Peripheral Interface
WB	Wolfsbild
WSS	Weidenschutzzäune

V Versionsgeschichte

Autor	Version	Datum	Änderungen
Fabian Theuerkauf	0.1	21.05.2019	Dokument erstellt
Alexander Luft	0.2	25.05.2019	Projektplanung, Architekturentwurf
Projektgruppe	0.3	27.05.2019	Dokument überarbeitet
Fabian Theuerkauf	0.4	27.05.2019	Formatierung angepasst, Kapitel 2 & 4 eingefügt
Fabian Theuerkauf	0.5	28.05.2019	Kapitel 4 überarbeitet, Tests eingefügt, Kapitel 3 eingefügt
Dirk Neumann	0.6	02.06.2019	Erweiterung des Testkapitels
Fabian Theuerkauf, Dirk Neumann	0.7	03.06.2019	Version 0.6 Review
Fabian Theuerkauf	0.8	06.06.2019	Kapitel Entwicklungswerkzeuge, Architekturentwurf, Projektplanung überarbeitet
Projektgruppe	0.9	18.07.2019	Kapitel 11,12,13,14 eingefügt
Fabian Theuerkauf	0.10	28.07.2019	Abbildung Netzplan und Technische Architektur überarbeitet, automatische Nummerierung im Fließtext eingefügt
Fabian Theuerkauf	0.11	5.08.2019	Kapitel 8 eingefügt
Alexander Luft	0.12	15.08.2019	Kapitel 3 & 3.1 eingefügt
Alexander Luft	0.13	17.08.2019	Kapitel 3.2 eingefügt
Alexander Luft	0.14	18.08.2019	Kapitel 9.1.1.2 bis 9.1.2 eingefügt
Dirk Neumann	0.15	21.08.2019	Testkapitel inhaltlich fertiggestellt
Dirk Neumann	0.16	26.08.2019	Testkapitel stilistisch überarbeitet

Alexander Luft	0.17	26.08.2019	Kapitel 9.1.3 bis 9.1.4 eingefügt
Alexander Luft	0.18	27.08.2019	Kapitel 9.2 eingefügt
Projektgruppe	0.19	29.08.2019	Kapitel 9 überarbeitet
Projektgruppe	0.20	30.08.2019	Quellen überarbeitet, Bildreferenzen vervollständigt

1 Einleitung

Das Projekt wird im Zuge des Anwendungspraktikums der Lehrveranstaltungen Future Internet/Internet of Things und Digitales Kulturerbe durchgeführt. Die Studierenden sollen strukturierte Entwicklungsarbeit, mit dem Fokus auf Projektmanagement, anhand einer selbst gewählten Aufgabenstellung, sowie die Grundlagen des Arbeitens mit Clouddienstanbietern erlernen und vertiefen.

Seitdem Wölfe wieder in Deutschland heimisch werden, entstehen vermehrt Konflikte zwischen dem Artenschutz des Wolfes und der Nutzviehhaltung. Durch das Reißen von Nutztieren entstehen wirtschaftliche Schäden, welche nicht nur den Nutztierhalter, sondern auch Bund und Länder finanziell betreffen [2,3,4]. Dazu wurden in der Vergangenheit bereits einige Schutzmaßnahmen etabliert, darunter am stärksten Herdenschutzzäune und Herdenschutzhunde, die sich jedoch nicht als wirtschaftlich erweisen [1].

2 Problemstellung

Der Wolf spielt eine wichtige Rolle bei der Entwicklung und den Erhalt verschiedener Ökosysteme. Er wird aber aufgrund wirtschaftlicher Aspekte dennoch bejagt bzw. geschossen. Deswegen soll ein System geschaffen werden, welches sowohl die Weidetiere des Menschen schützt und dementsprechend die wirtschaftlichen Schäden die der Wolf verursacht reduziert, als auch die Wölfe.

3 Aktueller Stand

Sobald eine Tierart sich in einem Gebiet wieder ansiedelt, führt dies dazu, dass das Umfeld auf diese Tierart reagiert. Diese Reaktionen können dabei positive oder negative Auswirkungen haben auf den Lebensraum haben. Eine positive Auswirkung wäre es, wenn die neuangesiedelte Tierart den natürlichen Kreislauf des Lebens beeinflusst, wie z.B. das Bejagen von kranken Tieren. Eine negative Auswirkung wäre z.B. das Auftreten der Tiere als invasive Art, d.h. sie verdrängen die einheimischen Tier- oder Pflanzenarten. Im Folgenden wird betrachtet wie die Politik in Deutschland auf den Wolf reagiert, wie der Wolf seinen neuen Lebensraum beeinflusst und wie der Herdenschutz als gesellschaftliche Reaktion auf den Wolf in Deutschland geregelt ist.

3.1 Politische Situation Deutschland

In Deutschland ist jedes Bundesland für das Management eines oder mehrerer Wolfsrudel selbst verantwortlich, darunter fällt auch die Entscheidung über Bejagung. So ist der Wolf in Sachsen zwar im Jagtrecht aufgenommen, hat dort allerdings keine Jagdzeiten [25], somit darf der Wolf in Sachsen dennoch nicht gejagt werden. Das allgemeine Jagdverbot des Wolfes ist im 5. Kapitel des Bundesnaturschutzgesetzes, kurz BNatSchG [23], festgeschrieben, allerdings gibt es auch dort Ausnahmen. So wurde festgelegt, dass Wölfe die sich gegenüber Menschen auffällig verhalten oder fortwährend hinreichend geschützte Nutztiere erbeuten vom Jagdverbot ausgenommen werden. Damit ist gesichert, dass der Wolf nicht grundlos bejagt werden kann. Dennoch kam es vermehrt in der ersten Hälfte des Jahres 2019 zu einer vermehrten politischen Diskussion um das Thema Wolf.

Diese Auseinandersetzung führte dazu, dass sich das Bundesministerium für Umwelt, Naturschutz und nuklearer Sicherheit sowie das Bundesministerium für Ernährung und Landwirtschaft um die aktuelle rechtliche Lage stritten und am 20. Mai 2019 einen Gesetzesentwurf vorlegten, der den aktuellen Schutzstatus des Wolfes geringfügig änderte. Einer im Entwurf vorkommender Punkt ist dabei, dass Wölfe eines Rudels zum Abschuss frei gegeben werden können, auch wenn ein Schaden, zum Beispiel die Tötung eines Schafes, nicht direkt diesem Wolf des Rudels zugesprochen werden kann [23]. Allerdings wurden keine weiteren Maßnahmen erdacht, wie das Weidevieh vor dem Wolf geschützt werden kann.

Auch findet sich das Thema Wolf im Wahlkampf der verschiedenen Politischen Parteien wieder, so werden oft Obergrenzen, für die Anzahl der Wölfe im Land und die Aufnahme im landeslokalen Jagdrecht gefordert. Zum anderen wird eine vermehrte Förderung für die Schutzmaßnahmen des Weideviehs, sowie höhere Entschädigungszahlungen angestrebt.

Im gesellschaftlichen Diskurs wird der Wolf als etwas Negatives für die regionale Umgebung dargestellt, dabei wird nicht beachtet, wie positiv der Wolf auf ein Ökosystem einwirken kann.

3.2 Beispielhafter Einfluss des Wolfes

Die Wolfspopulation in Deutschland lag im Jahr 2017/2018 bei 75 Rudel, dabei leben in einem Rudel im Durchschnitt drei bis elf Tiere. Dies ergibt eine Gesamtpopulation von ca. 829 Wölfen. Diese Zahl ist allerdings mit Vorsicht zu betrachten, da alleine die Annahme der Rudelgrößen auf Durchschnittswerte beruhen. Die meisten Wölfe halten sich in Bayern, Brandenburg, Mecklenburg-Vorpommern, Sachsen, Sachsen-Anhalt und Thüringen auf [8].

Diese Ausbreitung des Raubtieres, führt zu regionalen Problemen mit dem Tier. Denn der Wolf ist ein anpassungsfähiger Beutegreifer und seine Nahrungszusammensetzung hängt stark von den lokalen Bedingungen, sowie von der aktuellen Jahreszeit ab. Die natürlichste Beute des Wolfes sind Wildtiere wie Rehe, Hasen, aber auch andere kleinere Raubtiere. Das führt dazu, dass der Wolf durch seine Anwesenheit indirekt auf das ihm umgebene Ökosystem eingreift.

So wurden 1995 wieder Wölfe in den Yellowstone Nationalpark angesiedelt, dies führte dazu, dass die Population der Biber in den Jahren wieder gestiegen ist, da die Wölfe, die in den Jahren zuvor stark angestiegenen Population, der Elche wieder verringerten. Da die Anzahl der Elche abnahm, konnten sich die Bäume in direkter Nachbarschaft der Flüsse und Bäche wieder erholen, welche die Biber für die Überwinterung brauchen. Dies führte wiederum dazu, dass die Biber mehr Dämme bauten. Diese Dämme hatten Auswirkungen auf die Strömungen der Flüsse, was wiederum dazu führte, dass das Wasser sich abkühlte und dadurch die Fischbestände sich erholten. Die sich erholenden Fischbestände, führten dann wieder dazu, dass die Bären bessere Nahrungsbedingungen vorfanden, was sich auch wieder positiv auf die Population auswirkte. Dieser Kaskadeneffekt ist auch noch in anderen Bereichen des Yellowstone Nationalparks zu finden, wie eine wachsende Vogel, Insekten oder Pflanzen Population [9].

Dieser Einfluss auf das Ökosystem des Yellowstone Nationalparks ist allerdings nicht direkt auf Deutschland übertragbar, da alleine die Fläche des Yellowstones ca. 9000 km² groß ist, dies entspricht ungefähr die Hälfte der Fläche von Sachsen (18450 km²) und solch ein flächendeckendes Areal in Deutschland nicht gegeben ist. Denn die Fläche aller Naturschutzgebiete in Deutschland beträgt gerade einmal 2145km² [10].

Von diesen Naturschutzgebieten ist der Harz mit seinen 247 km² die größte Region, diese Fläche ist aber gerade mal so groß, wie das Revier eines Rudels, das zwischen 100 und 350 km² liegen kann. Die Reviergröße hängt vor allem von der verfügbaren Nahrung ab [11]. Dies führt dazu, dass Wölfe in Deutschland sich immer wieder in besiedelten oder beweideten Gebieten wiederfinden.

Dort finden sie dann Weidetiere als Nahrungsquelle vor, die durch die meist vorherrschende Einzäunung keine Möglichkeit haben, vor dem Wolf zu flüchten und es somit für den Wolf ein leichtes ist diese Beute zu jagen. Dies führt zu einem Konflikt zwischen Menschen und Wolf, welcher in den vergangenen Jahren nicht gegeben war. Dadurch war der effektive Herdenschutz vor Wildtieren rückläufig.

So wurden hauptsächlich Elektrozäune zur Weideabgrenzung genutzt und Herdenschutzhunde nicht mehr angeschafft. Der Wolf aber ist ein adaptives Raubtier und diese herkömmlichen Schutzmaßnahmen sind nicht ausreichend genug um diesen abzuwehren. Daher wird häufiger der Abschuss des Wolfes gefordert, doch dies ist keine geeignete Lösung, da der Wolf, wie am Beispiel des Yellowstone Nationalparks erkennen kann, einen positiven Einfluss auf das Ökosystem haben kann [6].

3.3 Herdenschutz

Bisher gab es mehrere Ansätze, die den Herdenschutz garantieren sollen ohne dabei den Wolf in seiner Existenz zu bedrohen. Von diesen haben sich bisher Weideschutzzäune in Kombination mit den Herdenschutzhunden am effizientesten herausgestellt, wobei auch diese jeweils Probleme mit sich führen [1]. Im Folgenden werden sowohl diese beiden Maßnahmen, als auch der Lösungsansatz mit Wolfhalsbänder hinsichtlich ihrer Umsetzung und Probleme näher beschrieben.

3.3.1 Weideschutzzäune

Die Weideschutzzäune, kurz WSS, sind in Deutschland Industriestandard, wenn es darum geht Viehweiden mit einem gewissen Grundschatz auszustatten. Die WSS gibt es in unterschiedlichen Formen, so z.B. mit oder ohne Stromversorgung. Unabhängig davon ob der Zaun unter Strom steht oder nicht, bildet der WSS für den Wolf eine psychische und keine physische Barriere, da dieser durchaus in der Lage wäre den Zaun zu überspringen. Um dem Wolf das Herüberspringen des WSS zusätzlich zu erschweren, können Flatterbänder auf dem Zaun montiert werden, was allerdings einen zusätzlichen Arbeitsaufwand und Kosten für die Viehwirte bedeutet. Wenn der Viehwirt sich für einen WSS mit Stromfluss entscheidet, muss beachtet werden, dass der Zaun ordnungsgemäß aufgestellt ist und regelmäßig von Bewuchs befreit wird, da dieser sonst die stromführenden Elemente des WSS erden kann und somit die abschreckende Wirkung durch die kurzen, aber ungefährlichen Stromschläge verliert. Der Aufwand die Zäune zu installieren und zu warten ist insgesamt also sehr aufwändig, was sich natürlich je nach Herdengröße unterschiedlich stark auswirkt [1,6].

3.3.2 Herdenschutzhunde

Der Einsatz von Herdenschutzhunden, kurz HSH, ist nicht nur eine der traditionellen Formen des Herdenschutzes, sondern auch eine der effizientesten. Durch das Ausrotten der Wölfe geriet diese Tradition allerdings in Vergessenheit, weshalb die Wiedereinführung die Viehhalter vor Herausforderungen stellt. So kostet ein HSH den Besitzer jährlich zwischen 800 und 1000€. Des Weiteren muss der Viehhalter in der Lage sein, sich dem Hund gegenüber durchzusetzen und ihn zu sozialisieren, was gerade für ungeübte Hundehalter durchaus zu Problemen führen kann. Dies ist bei HSH besonders wichtig, da sie aufgrund ihrer Aufgabe einen stark ausgeprägten eigenen Willen besitzen und, im Gegensatz zu anderen Hunden, nicht auf den Menschen fixiert sind, sondern auf ihre Herde. Da diese Hunde oft vom Halter unabhängig agieren, muss in der Ausbildung des Hundes sichergestellt sein, dass er Menschen gegenüber nicht aggressiv auftritt. Diese Punkte stellen aber nur die grundsätzlichen Aspekte der Sozialisierung dar, denn auch das Gewöhnen der Junghunde an ihre jeweilige Herde, ist sehr zeitaufwändig. Da die HSH ihre, zur Arbeit notwendigen, Verhaltensmuster, wie z.B. das Abschrecken von Beutegreifern, von älteren Artgenossen erlernen, müssen je Herde immer mindestens zwei HSH im Einsatz sein. Da Hunde Rudeltiere sind, ist dies für eine artgerechte Haltung ohnehin notwendig [1].

3.3.3 Halsbänder für Wölfe

Eine, im Vergleich zu HSH und WSS, moderne Lösungsstrategie zur Abwehr von Wölfen sind Halsbänder für Wölfe mit unterschiedlichen Funktionen. Ursprünglich wurden diese Halsbänder für die Beobachtung der Wanderwege der Wölfe eingesetzt, allerdings wurde mittlerweile auch der Nutzen der Halsbänder für den Herdenschutz erkannt. So gibt es z.B. ein Halsband-System mit GPS Sender, welches dem Schäfer eine SMS sendet, falls sich der Wolf der Herde nähert. Ein anderes Beispiel ist ein Halsband, welches ebenfalls mit GPS arbeitet und dem Wolf einen Elektroschock gibt, wenn er der Herde zu nahe kommt. Unabhängig von der konkreten Funktion, teilen alle Halsbänder die gleichen Probleme. Es ist schwer, den scheuen und vorsichtigen Wölfe, die entsprechenden Halsbänder anzulegen und diese gegebenenfalls zu reparieren, falls sie beschädigt werden. Auch ist diese verhältnismäßig neue Technologie noch nicht erprobt genug, um eine Aussage auf die Wirksamkeit der Halsbänder treffen zu können [1,5]. Eines der größten Probleme, ist allerdings das jedem Wolf ein Halsband angelegt werden muss. Vor allem bei sich reproduzierenden Rudel führt dies zu einem sehr hohen finanziellen und organisatorischen Aufwand.

4 Projektziele

Alle der bisher vorgestellten Technologien zum Herdenschutz haben jeweils ihre funktionalen Nachteile. Ferner aber, sind alle der genannten Maßnahmen unwirtschaftlich, wenn man die Kosten für deren Anschaffung und die laufenden Kosten mit den Schäden vergleicht, die entstehen, wenn eine Herde nicht geschützt werden kann und ein Wolf ein oder mehrere Tiere reißt [2,3,4].

Dem entgegen ist das Ziel des Projektes „Wolfsabschreckung“ ein System zu konzipieren und realisieren, welches Wölfe zuverlässig und zu jeder Zeit erkennt und abschreckt und dabei keine großen Kosten erzeugt. Im Zielzustand müssen Wölfe bei einer möglichst großen Vielfalt von unterschiedlichen Szenarien, z.B. verschiedene Witterungsbedingungen und Bildqualitäten, als solche erkannt werden. Diese Szenarien und Bildqualitäten werden im Kapitel „10. Tests“ näher erläutert. Des Weiteren muss das System in der Lage sein, Wölfe von wolfsähnlichen Hunden zu unterscheiden. Im Kontext der Veranstaltung „Future Internet / Digitales Kulturerbe Anwendungspraktikum“ wird aber ein verringelter Umfang, für die den soeben definierten Zielzustand angenommen, welcher im Folgenden im Detail erklärt wird.

4.1 Soll-Kriterien

Die grundlegende Funktionalität, die Erkennung von Wölfen mithilfe eines Minicomputer, muss in jedem Fall gewährleistet sein. Dazu soll ein Hardwareprototyp realisiert werden, der im Falle der Erkennung eines Wolfs die entsprechenden Abwehrmaßnahmen einleitet. Die Erkennung des Wolfes soll, als Projektanforderung, über einen Dienst eines ausgewählten Clouddienstanbieters erfolgen. Dieser Dienst führt dann, entsprechend der Implementierung des Anbieters, eine Bildklassifizierung durch und sendet die Ergebnisse, also die erkannten Objekte mit dem dazugehörigen Konfidenzintervall, zurück an das Hardwaremodul. Aufgrund der erkannten Objekte muss die Software des Prototyps dann entscheiden, ob eine Abwehrmaßnahme eingeleitet werden soll. Die Abwehrmaßnahmen bestehen dabei zum einen aus einer visuellen Abschreckung über LEDs in unterschiedlichen Farben und zum anderen aus einer akustischen Abschreckung über Lautsprecher.

4.2 Kann-Kriterien

Bei einer Erweiterung des Prototyps muss beachtet werden, dass der Funktionsumfang so erweitert wird, dass die Abschreckung von Wölfen, sichergestellt bleibt. Dazu sollte das erweiterte System zusätzlich Nachtaufnahmen über eine Infrarotkamera möglich machen. So können ganztägig Aufnahmen erstellt und so Wölfe rund um die Uhr abgeschreckt werden. Die Daten könnten bevor sie an den Clouddienst geschickt werden optimiert werden. Die Wolfserkennung findet weiterhin durch den Clouddienst statt. Hier kann eine speziell trainierte künstliche Intelligenz für bessere Ergebnisse sorgen als die Standardvariante. Für den Fall, dass ein Wolf erkannt wird, kann die Wolfsabwehr vielfältiger gestaltet werden. Durch die Variation von verschiedenen Farben und Tönen kann verhindert werden, dass sich ein Wolf an ein Muster bei der Abwehr gewöhnt.

Die Station selbst kann mit einem Stativ und Gehäuse gegen Witterungsverhältnisse verstärkt werden und zusätzlich mit Diebstahlschutz oder eindeutiger Zuordnung zu einem Besitzer versehen werden.

4.3 Projektabgrenzung

Wie bereits erläutert, wird ein reduzierter Umfang des ursprünglichen Zielzustandes angenommen. Das heißt, die Wolfserkennung des Prototyps soll nur unter optimalen Bedingung, also ohne Rücksicht auf schlechte Witterungsverhältnisse wie z.B. Nebel oder starken Regen, ein zuverlässiges Ergebnis liefern. Dementsprechend wird nur eine Standard RGB-Kamera und keine für Nachtaufnahmen benötigte Infrarotkamera verwendet. Auch die bereits erwähnte Unterscheidung zwischen Wölfen und wolfsähnlichen Hunden wird, wenn überhaupt, nur rudimentär versucht umzusetzen, da die für das Projekt geplante Zeit nicht ausreicht, eine dafür geeignete Künstliche Intelligenz zu trainieren und zu testen.

Im Gegensatz zu bestehenden Projekten, soll der zu bauende Prototyp den Wolf nicht direkt schaden, wie es z.B. das Elektroschock-Halsband tut [5], sondern ihn nur abschrecken. Des Weiteren sind die gewählten Abwehrmechanismen nicht wolfsgebunden, das heißt dem Wolf muss keine Hardware angelegt werden, wie es z.B. bei einem Sensorhalsband der Fall ist.

5 Projektplanung

Um einen ersten Überblick über die einzelnen Aufgaben zu erhalten, sowie deren Strukturierung zu erleichtern, wurde ein Projektplan erstellt. In diesem werden, in zeitlicher Abhängigkeit, genauer in Intervallen von 7 Tagen, die zu erledigen Aufgaben dargestellt. Der erarbeitete Projektplan ist in Tabelle 1 dargestellt.

Tabelle 1: Projektplan

	Was muss der Stand zu Tag X sein? (Fragen müssen beantwortet sein)
25.03.	-Thema ist entschieden -Was ist dabei: Cloud, Digitales Kulturerbe, IoT?
28.03.	-Wo besteht ein Bedarf unseres Projektes? -Welche Marktlücke wollen wir decken? -Was bedeutet „fertig“? (Funktionale Abgrenzung)
31.03.	-Projektprozess ist definiert (Team- und Projektstruktur) -Projektzeitplan (Gant, Netzplan) -Wie kann verifiziert und validiert werden? -Abhängigkeiten, Schutzziele, Risiken sind definiert -grobe Architektur ist fertig definiert
Mi 03.04. nach dem Treffen	-Die Literatur die verwendet werden soll ist recherchiert, auf Wissenschaftlichkeit geprüft (Das wird nicht dokumentiert) und als Rechercheliteratur festgelegt -Literaturthemen (wichtigstes zuerst): 1. Fakten zum aktuellen Stand (Zäune, Hütehunde, wirtschaftliche Schäden durch den Wolf)

	<p>2. Wolf: Lebensraum, Verhalten</p> <p>3. Anforderungen an Hardwarekomponenten -> technische Produktblätter</p>
So 07.04.	-Exposé (Präsentation) ist fertig
Di 09.04	Neues Recherchethema: Cloud: Verschieden Clouddienste, Anbindung an Pi, Videoverarbeitung
14.04.	<p>-Der erste Teil der ausgewählten Literatur wurde zusammengefasst und so aufbereitet, dass er den anderen Teammitgliedern vorgestellt werden kann und Ergebnisse diskutiert werden können (Wolf, Hardware).</p> <p>-Eine Liste mit nötigen Hardwarekomponenten und den Eigenschaften, die diese haben müssen, wurde festlegt.</p>
15.04+16.04	Vorstellung der Rechercheergebnisse im Anwendungspraktikum Future Internet/Digitalen Kulturerbe Auswertung der Abstracts im Seminar
21.04.	<p>-Sollte die Recherche der Cloudanwendungen kein eindeutiges, stark überwiegendes Ergebnis geliefert haben, werden nun einige kleinere Tests gestartet, um zu prüfen wie gut sich mit den einzelnen Clouddiensten arbeiten lässt.</p> <p>Gut: zeiteffizient, wenig neues Wissen anzueignen</p> <p>-Hardwareliste ist fertig und eingereicht</p>
28.04.	<p>-Cloudrecherchen sind abgeschlossen. Die zu verwendende Technologie steht fest. Durch kleinere Test ist der Umgang mit dieser bereits erprobt.</p> <p>-Die Hardware wurde bestellt.</p>
05.05.	Bilddaten können mit Hilfe eines Laptops in die Cloud gesendet werden.
12.05.	<p>Getestet: Auswertung der Bilddaten durch einen ausgewählten Dienst eines ausgewählten Cloudanbieters und Aufzeigen etwaiger Grenzfälle/Qualitätsgrenzen mit dem Ziel: Reproduzierbare Ergebnisse.</p> <p>Die Kamera für den Minicomputer ist angeschlossen und konfiguriert.</p>
19.05.	<p>Die nötige Hardware für Abschreckmaßnahmen ist am Minicomputer installiert und getestet.</p> <p>Der Dienst in der Cloud entscheidet, ob abgewehrt werden soll und sendet die Daten an den Laptop zurück.</p>
26.05.	Der Minicomputer sendet die (gegebenenfalls veränderten) Bilddaten in die Cloud, welche die entsprechende Antwort an den Minicomputer zurückschickt.
02.06.	Modulimplementierung ist abgeschlossen
09.06.	Erste Tests des Prototypens wurden durchgeführt
16.06.	<p>Tests sind ausgewertet Analyse der geleisteten Arbeit</p> <p>Dokumentation und Präsentation sind fertig</p> <p>Präsentation + Abgabe</p>

5.1 Gantt Diagramm

Das Gantt Diagramm stellt die Aufgaben die im Projektplan ermittelt wurden dar. Dies geschieht durch einzelne Balken, die sowohl Startdatum, Dauer und Enddatum repräsentieren. Dadurch kann der gesamte Projektzeitraum in einer horizontalen Zeitachse dargestellt werden. Die zu erledigenden Aufgaben werden, wie in den Abbildungen 1 und 2 zu sehen, in den Balken selbst abgebildet, die Meilensteine des Projektes werden als schwarze Dreiecke dargestellt.

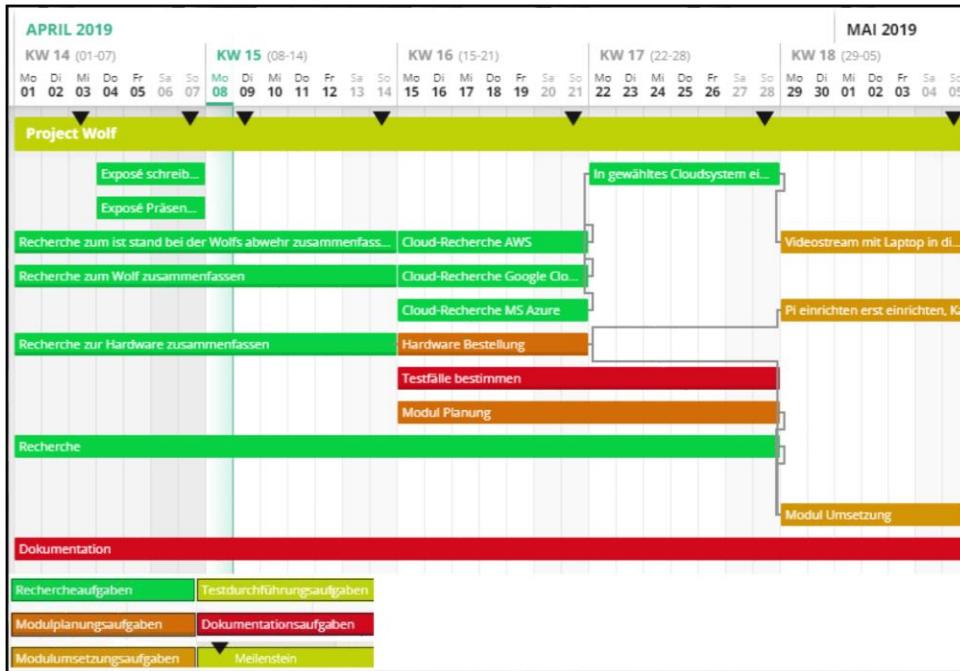


Abbildung 1: Darstellung des Projektplanes in Form eines Gantt Diagrammes

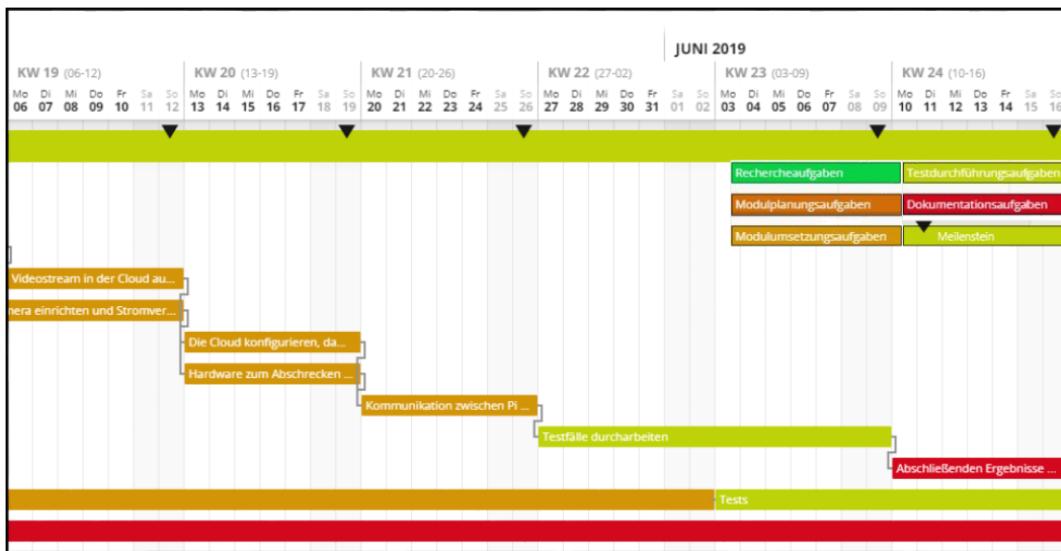


Abbildung 2: Zweiter Teil des Gantt Diagramms

5.2 Netzplan

Der Netzplan stellt, wie in Abbildung 3 zu erkennen, den Ablauf des Projektes in Form von Abhängigkeiten und Reihenfolgen der Aufgaben grafisch dar. Der Projektzeitraum ist durch die relative kurze Zeitspanne, von nur 77 Tagen, streng bemessen. Das Zusammensetzen der einzelnen Hardwarekomponenten stellt dabei mit seinen 2 Tagen Puffer, über mehrere Aufgaben hinweg, den engsten Zeitraum dar. Der Aufgabenbereich der Clouddrecherche und der Anwendung und Implementierung der Clouddienste, hat mit vier Tagen Puffer etwas mehr Zeit.

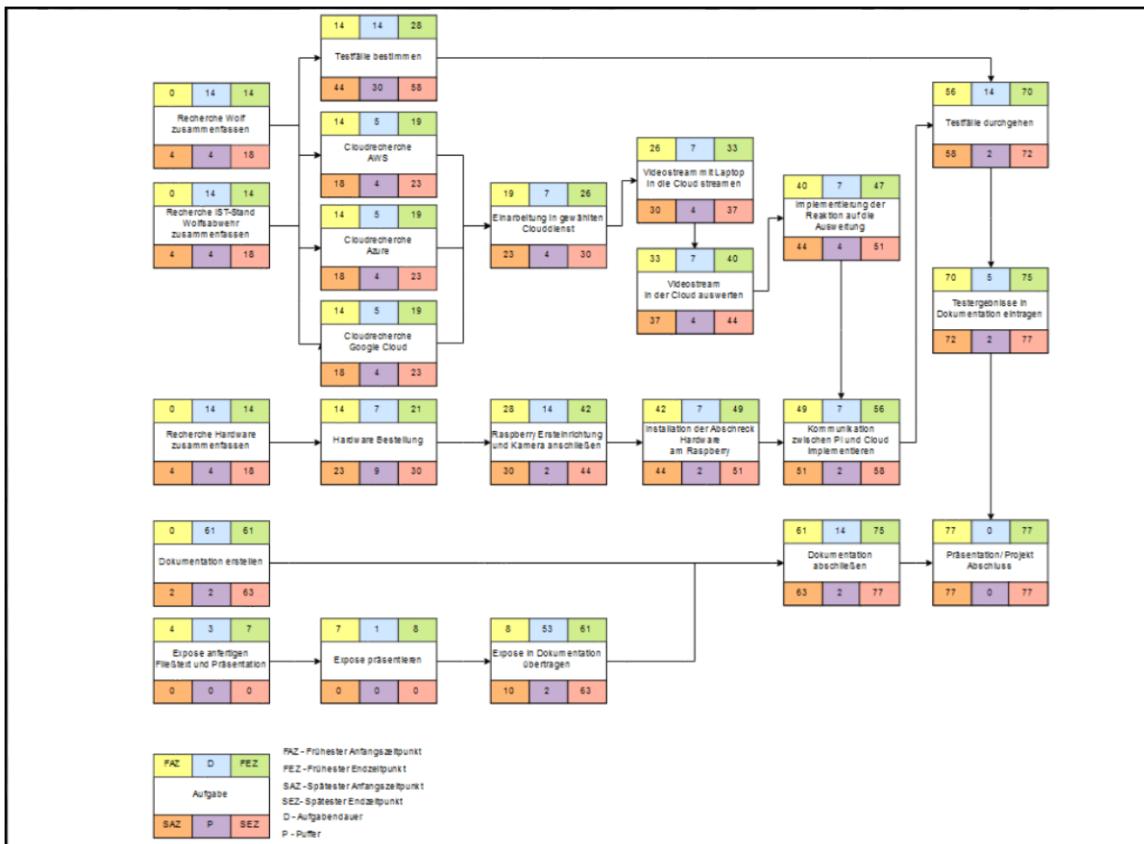


Abbildung 3: Netzplan inklusive Legende

6 Entwicklungswerzeuge

Die, für die Fortschrittsübersicht wichtigen, Zeitpläne wurden mit Hilfe von Agantty erstellt. Der daraus resultierende Netzplan, sowie die UML-Diagramme mit Hilfe des Internetdiensts draw.io.

Um die benötigte Software zu implementieren, muss eine geeignete Programmiersprache als Werkzeug gewählt werden. Als erste Auswahl wurden vier verschiedene objektorientierte Sprachen gewählt, von denen eine ausgesucht werden muss. Jede dieser Programmiersprachen besitzt Vor- und Nachteile, welche in der folgenden Tabelle dargestellt werden.

Tabelle 2: Vergleich der Programmiersprachen

Programmiersprache	Vorteile	Nachteile
Java	-plattformübergreifend -leicht zu lernen -leicht lesbar	-bei kleinen Projekten Overhead Code -benötigt Ressourcen für die virtuelle Maschine
Python	-leicht lesbar -leicht zu lernen -Vielzahl von mächtigen Bibliotheken	-langsam -typunsicher
C#	-leicht lesbar -leicht zu lernen -Produkt von Microsoft	-höherer Ressourcenverbrauch im Vergleich zu C++ -nicht plattformübergreifend
C++	-plattformübergreifend -effizient (bei guter Programmierung) -hardwarenah -Produkt von Microsoft	-eigenständige Speicherverwaltung

Wie Tabelle 2 zeigt, bieten C# und C++ den Vorteil, ebenso wie der genutzte Clouddienstanbieter, ein Microsoftprodukt zu sein. Dementsprechend wäre das Projekt, in Sicht auf die genutzten Technologien homogen, was zur Folge hat, dass keine Probleme mit zu implementierenden Schnittstellen auftreten. Des Weiteren liegt der Gedanke nah, dass der Microsoft Azure Support, bei Verwendung von anderen Microsoft Produkten, wie C# und C++, umfangreicher ist, als der bei nicht von Microsoft entwickelten Programmiersprachen, weshalb die nähere Auswahl auf diese beiden Sprachen fällt.

Aufgrund der fehlenden Erfahrung bei der Arbeit mit C++ und den bereits vorhandenen Vorkenntnissen im Umgang mit C#, wird für die Implementierung die Sprache C# genutzt.

Zur Programmierung mit C# wird die Entwicklungsumgebung Visual Studio 2019 Community Edition, kurz VSCE, genutzt, da diese ebenfalls ein Produkt von Microsoft ist und speziell für die Entwicklung von .Net Sprachen geschaffen wurden ist. Für VSCE sprechen hauptsächlich die bereits oben genannten Vorteile der Homogenität und des Supports der Sprache durch Microsoft. Zusätzlich dazu besitzen auch hier die Teammitglieder bereits Vorkenntnisse im Umgang mit der IDE und deren zusätzlichen Werkzeugen.

7 Architekturentwurf

Im Folgenden wird die Architektur, des zu konstruierenden Prototypen beschrieben. Dazu wird erst eine Unterteilung in verschiedene Aufgaben vorgenommen, welche dann im nächsten Schritt umgesetzt werden sollen.

7.1 Logische Architektur

Um den Wolf abschrecken zu können benötigt der Prototyp ein Hardware-Modul, welches in der Lage ist, Bilder oder Videos aufzunehmen. Da, im Rahmen des Projektes, nur der Wolf abgeschreckt werden soll, wird, wie in Abbildung 4 zu sehen ist, ein Modul benötigte, welches Objekte verschiedener Art zuverlässig voneinander unterscheiden kann. Wenn dann ein Wolf erkannt wird, benötigt der Prototyp ein oder mehrere Module, welche den Wolf, durch verschiedene Abwehrmaßnahmen verjagen können, ohne diesen jedoch direkt zu gefährden.

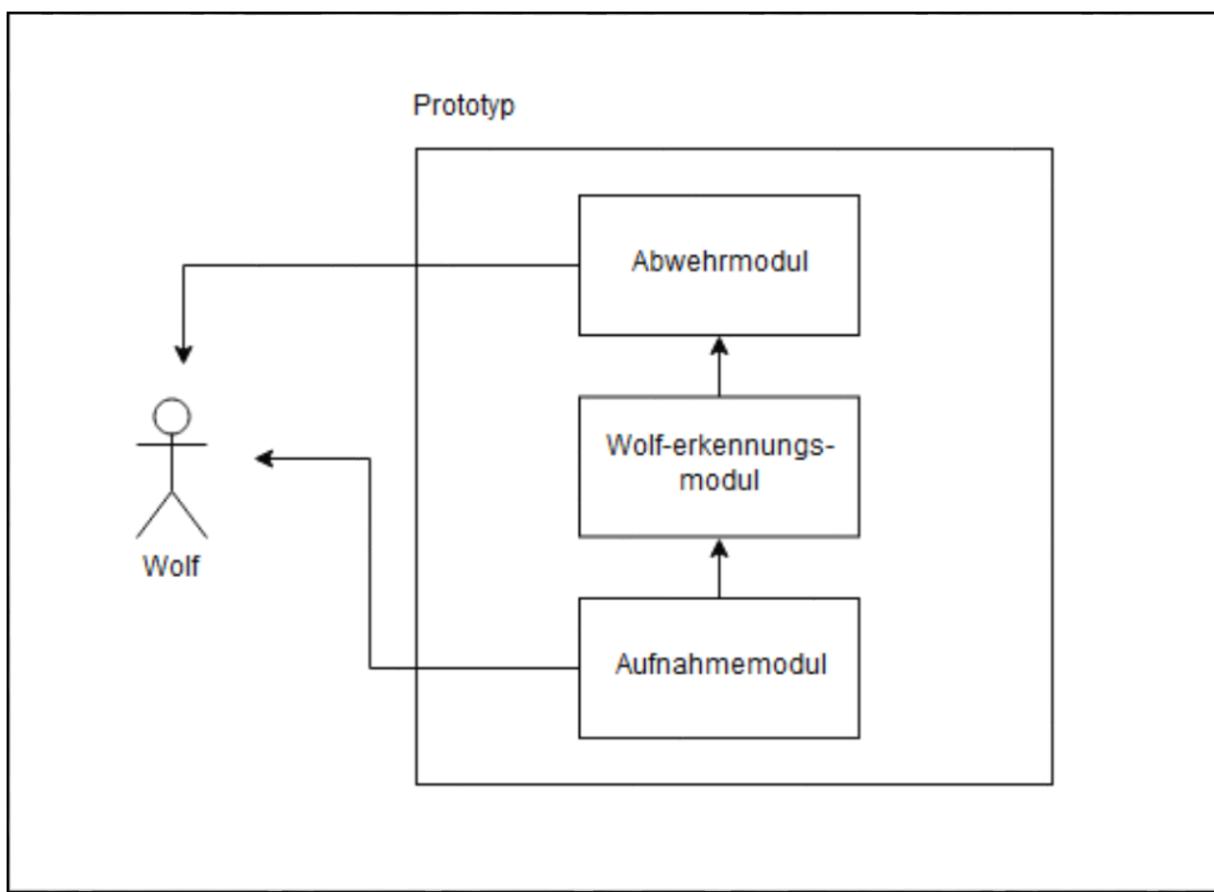


Abbildung 4: Logische Architektur des Prototypen

7.2 Technische Architektur

Die Erfassung des näheren Umfeldes des Minicomputers wird durch ein Kameramodul geleistet. Dieses nimmt in konfigurierbaren Zeitintervallen Bilder auf und speichert diese lokal auf dem Minicomputer. Der dabei abgedeckte Winkel, sowie die Kamerareichweite und die Fähigkeit zu Nachtaufnahmen, hängen von dem ausgewählten Kameramodul ab. Auf dem Minicomputer werden mehrere Aufgaben von einem Programm koordiniert. Zum einen wird in einem konfigurierbaren Zeitintervall ein Bild aus dem Datenstrom der Kamera auf die SD-Karte des Minicomputers gespeichert. Zum anderen wird dieses Bild von einem anderen Unterprogramm wieder geladen und an MSA-CV gesendet. Dort wird es dann ausgewertet und die dabei entstehenden Bildklassifizierungsinformationen zurück an den Minicomputer gesendet. Wie in Abbildung 5 zu sehen ist, wird anschließend anhand dieser Informationen entschieden, ob Abwehrmaßnahmen ausgeführt werden sollen oder nicht.

Als Abwehrmaßnahmen wird eine Kombination von visuellen und akustischen Reizen verwendet. Das Zusammenspiel aus den Lichtimpulsen und den lauten und unnatürlichen Geräuschen sollen den Wolf abschrecken. Aufgrund der Anpassungsfähigkeit, bzw. der beim Wolf ausgeprägten Lernfähigkeit, werden die Lichteffekte und die akustischen Signale variiert. Auch diese Unterprogramme werden durch das bereits genannte Koordinationsprogramm gesteuert.

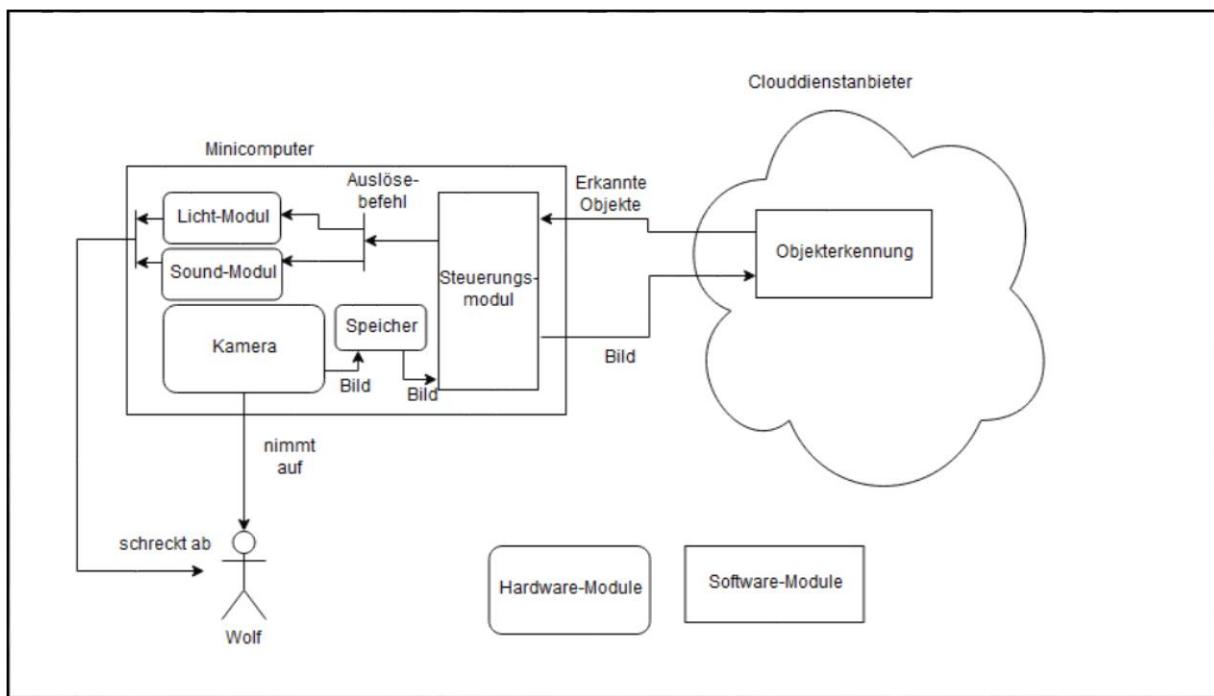


Abbildung 5: Technische Architektur des Prototypen

8 Logischer Ablauf im Steuerungsmoduls

Um einen detaillierteren Überblick über die Implementierung der notwendigen Aufgaben, des Steuermoduls, zu erhalten, sowie dessen Strukturierung zu erleichtern, wurde ein Ablaufdiagramm, in Abbildung 6 zu sehen, erstellt. In diesem wird die Verteilung der einzelnen Softwareaspekte, wie z.B. das Erkennen von Wölfen oder deren Abschreckung auf mehrere Threads dargestellt.

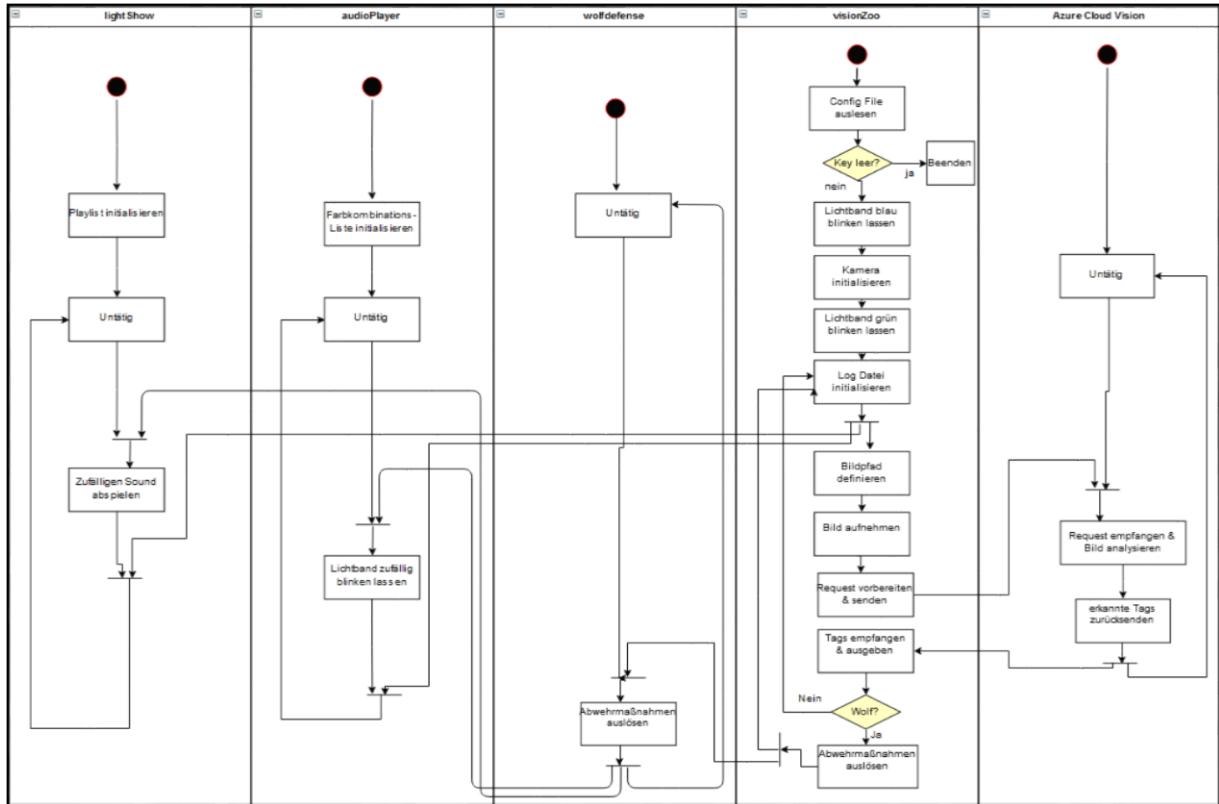


Abbildung 6: Ablaufdiagramm des Steuerungsmoduls

9 Implementierung

Damit der Prototyp funktionsfähig ist, müssen einige Vorbereitungen am Raspberry Pi selbst getroffen werden, wie z.B. das vornehmen der Erstkonfigurationen. Außerdem muss die benötigte Hardware verkabelt, angebaut und eingerichtet werden. Des Weiteren muss eine Verbindung zum Clouddienst eingerichtet werden. Nach Vollendung dieser Vorbereitungen muss die Software entwickelt werden, um die einzelnen Hardwaremodule anzusteuern und so für die Wolfsabwehr einsetzen zu können.

9.1 Erstkonfiguration

Im folgendem wird erklärt, wie der Raspberry Pi konfiguriert werden muss, damit dem Prototyp die Software- und Hardwareseitigen Voraussetzungen, zum Abschrecken des Wolfes, zur Verfügung stehen.

9.1.1 Raspberry Pi Konfiguration

Bevor der Raspberry Pi für seine mögliche Verwendung genutzt werden kann, müssen einige Softwarekomponenten eingerichtet und konfiguriert werden, unter anderen das Betriebssystem, die Verbindung über SSH, sowie die Einrichtung von C# und von Python.

9.1.1.1 Betriebssystem

Das Betriebssystem des Raspberry Pi benötigt eine 8, besser wäre 16, Gigabyte große Micro-SD Speicherkarte. Für den Prototypen wurde Raspbian, ein auf Debian basierenden Betriebssystemen, welches für den Raspberry Pi angepasst wurde, installiert. Zur Installation wird eine .Iso-Datei [12] des Betriebssystems benötigt. Diese muss auf die Speicherkarte kopiert werden, dafür wurde das Programm Etcher [PG3] genutzt. Nach Beendigung dieses Vorgangs, kann die SD Karte in den Raspberry Pi eingesteckt werden. Nun ist der PI in der Lage, mit diesem Betriebssystem zu starten.

9.1.1.2 Einrichten von SSH

In den neueren Versionen von Raspbian ist das SSH bereits installiert, aber noch nicht aktiviert. Damit wir aber über Programme wie PuTTY [PG2] oder unter Linux/Mac OS über das Terminal auf den Raspberry Pi über einen externen Rechner im selben Netzwerk wie der Raspberry Pi zugreifen können, müssen wir dieses aktivieren.

9.1.1.2.1 SSH aktivieren mit Maus und Tastatur

Wenn an dem Raspberry Pi neben einem Bildschirm auch eine Maus und Tastatur angeschlossen ist, dann kann man das SSH einfach über die Konfiguration des Raspberry Pi aktivieren. Um die Konfiguration aufzurufen reicht es aus im Terminal folgendem Code einzugeben.

```
pi@raspberrypi ~ $ sudo raspi-config
```

Mit bestätigen der Eingabe öffnet sich das, in Abbildung 7 zu sehende, Konfigurationsmenü des Pi. Dort muss dann der Menüpunkt 5 ausgewählt werden, damit die SSH Einstellungen angezeigt werden, die in Abbildung 8 zu sehen sind.

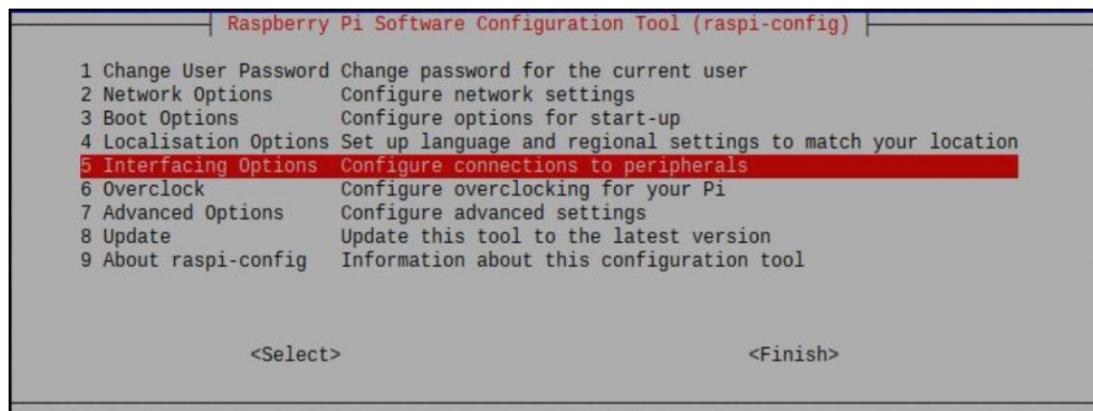


Abbildung 7: Hauptmenü der Raspberry Pi Konfiguration

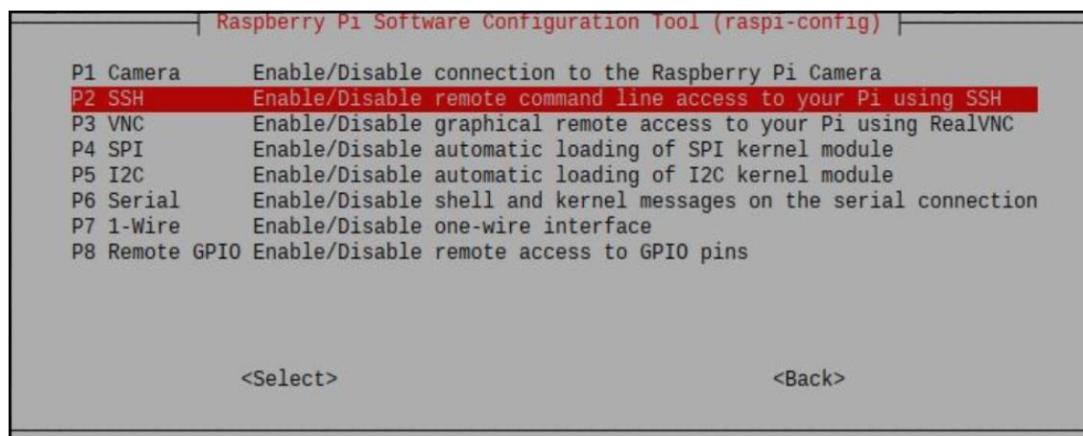


Abbildung 8: Interface Optionen des Raspberry Pi

In dem in Abbildung 8 geöffneten Menü, wählt man dann Punkt zwei auf um auf die SSH Einstellungen zuzugreifen, in dem darauf sich öffnenden Fenster, in Abbildung 9 zu sehen, muss dann die gestellte Frage mit „Ja“ beantwortet werden.

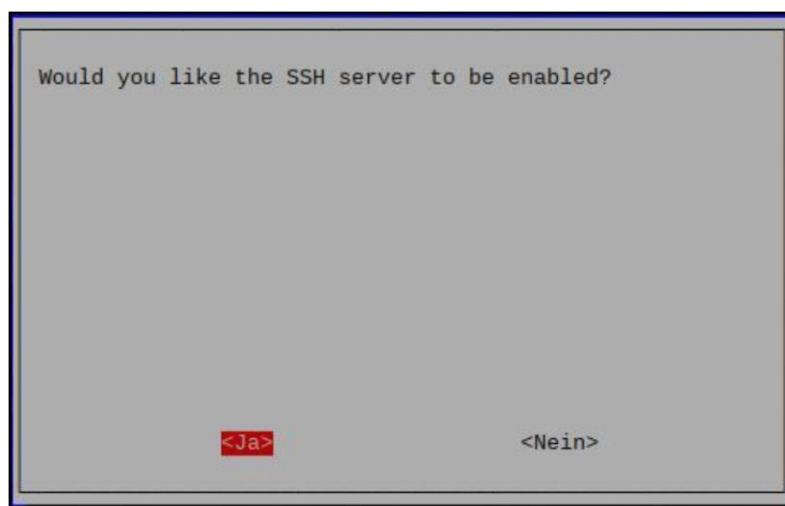


Abbildung 9: Konfigurationsfenster zum aktivieren des SSH-Servers auf den Raspberry Pi

9.1.1.2.2 SSH aktivieren ohne Maus und Tastatur

Falls der Zugriff auf den Raspberry Pi direkt über SSH erfolgen soll und es keine Möglichkeit gibt den SSH Zugriff, wie im vorherigen Kapitel beschrieben, zu aktivieren, dann muss eine Datei auf die Bootpartition der SD-Karte erstellt werden, auf der das Betriebssystem ist. Die zu erstellende Datei kann unter Linux/Mac OS wie gewohnt über die grafische Oberfläche erstellt werden, zu beachten ist, dass der Dateiname „ssh“ lauten muss und keine Dateiendung haben darf, die Datei benötigt außerdem keinen Inhalt.

Unter Windows muss die Datei mit Hilfe der Windows Power Shell erstellt werden. Dafür muss folgender Code in die Shell eingegeben werden.

```
New-Item -Name "ssh" -ItemType File -Path "(Pfad der SD-Karte)"
```

9.1.2 Einrichtung von C#

Obwohl der Raspberry Pi mit seinem Linux Betriebssystem nicht für das .NET Framework von Microsoft ausgelegt ist, ist es dennoch möglich durch das Mono Projekt C# Programme zu nutzen und zu entwickeln.

Um C# auf dem Raspberry Pi nutzen zu können, muss Mono installiert werden, dafür muss folgender Befehl im Terminal eingegeben werden.

```
sudo apt-get install mono-complete
```

Nach der Installation kann in einem beliebigen Texteditor dann C#-Code geschrieben werden, wichtig dabei ist, dass die erstellte Datei, wie in Abbildung 10 zu sehen ist, die Endung .cs haben muss. Dieser Code muss dann mit nachfolgendem Terminalbefehl kompiliert werden.

```
mcs Dateiname.cs
```



```
HelloWorld.cs
Datei Bearbeiten Suchen Optionen Hilfe
using System;

public class HelloWorld
{
    static public void Main()
    {
        Console.WriteLine("Hello World!");
    }
}
```

Abbildung 10: HelloWorld.cs Datei mit C#-Code

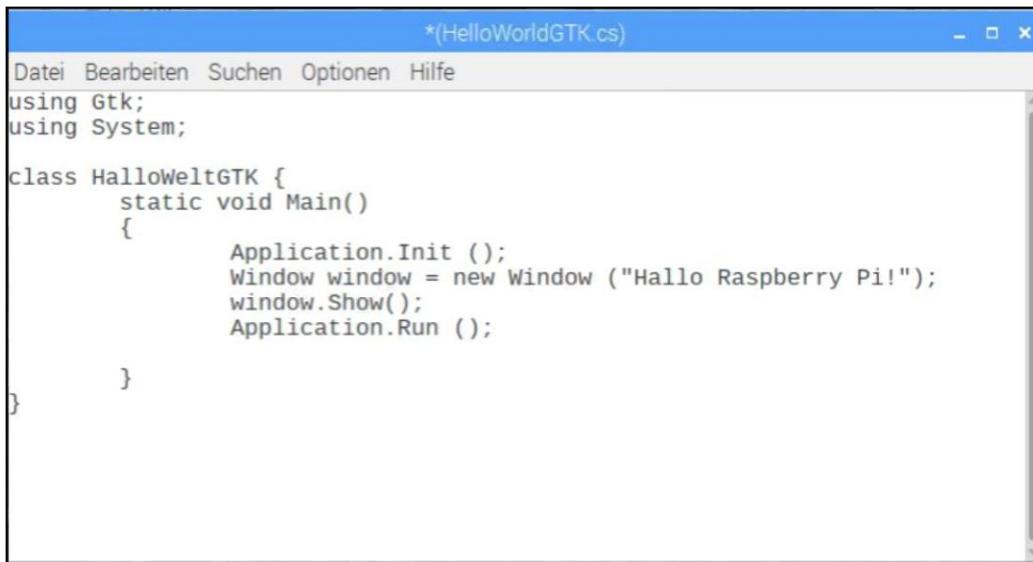
Die nun kompilierte Datei wird als .exe Datei gespeichert, da der Raspberry Pi dieses vom Betriebssystem aus nicht ausführen kann, muss folgender Befehl ausgeführt werden.

```
mono Dateiname.exe
```

Durch den mono Befehl wird eine virtuelle Umgebung auf dem Raspberry Pi geschaffen, auf der die .exe Datei laufen kann. Es ist auch möglich Grafik bzw. C#-Fensterprogramme auf den Raspberry Pi laufen zu lassen. Dafür muss die Grafikbibliothek GTK installiert werden, dies geht mit folgendem Befehl.

```
sudo apt-get install gtk-sharp2
```

Nach der Installation kann nun wieder mit einem Texteditor C#-Code geschrieben werden. Bei einem Fensterprogramm muss beachtet werden, dass wir Gtk im Programmcode mit dem Schlüsselwort „using“ einbinden. In der Main Methode wird ein Window Objekt erstellt, dieses wird mit der Methode „show()“ angezeigt, dem Konstruktor wird ein Text übergeben, dieser wird der Titel des erstellten Fensters. Ein Beispiel Programm ist in der Abbildung 11 zu sehen.



The screenshot shows a Windows Notepad window with the title bar labeled *(HelloWorldGTK.cs). The menu bar includes Datei, Bearbeiten, Suchen, Optionen, and Hilfe. The main content area contains the following C# code:

```
using Gtk;
using System;

class HalloWeltGTK {
    static void Main()
    {
        Application.Init ();
        Window window = new Window ("Hallo Raspberry Pi!");
        window.Show();
        Application.Run ();
    }
}
```

Abbildung 11: Beispielcode für ein C#-Fensterprogramm

Im nächsten Schritt muss die Datei wieder kompiliert werden, dabei ist darauf zu achten, dass eine Zusätzliche Option im Terminalbefehl mit angegeben wird, diese lautet wie folgt.

```
mcs dateiname.cs -pkg:gtk-sharp-2.0
```

Sobald die Datei kompiliert wurde, kann sie wieder über das mono Kommando ausgeführt werden.

9.1.3 Prototypen bauen

Da der Prototyp neben dem Raspberry Pi noch aus einigen anderen Hardware Komponenten besteht und diese mitunter mit dem Raspberry verbunden werden mussten, wird in diesem Kapitel genau auf diese Verkabelungen aber auch die softwareseitige Konfiguration eingegangen.

9.1.3.1 Verkabelung Druckknopf

Der Druckknopf hat zwei Anschlüsse, an jeweils einen Anschluss wird ein Kabel gelötet, die Farbe des Kabels spielt dabei keine Rolle, beim Prototypen wurde sich jeweils für ein rotes und ein schwarzes Kabel entschieden. Die Kabel werden dann jeweils, wie in Abbildung 12 zu sehen ist, an den Pin 5 und Pin 6 des Raspberry Pi angeschlossen.

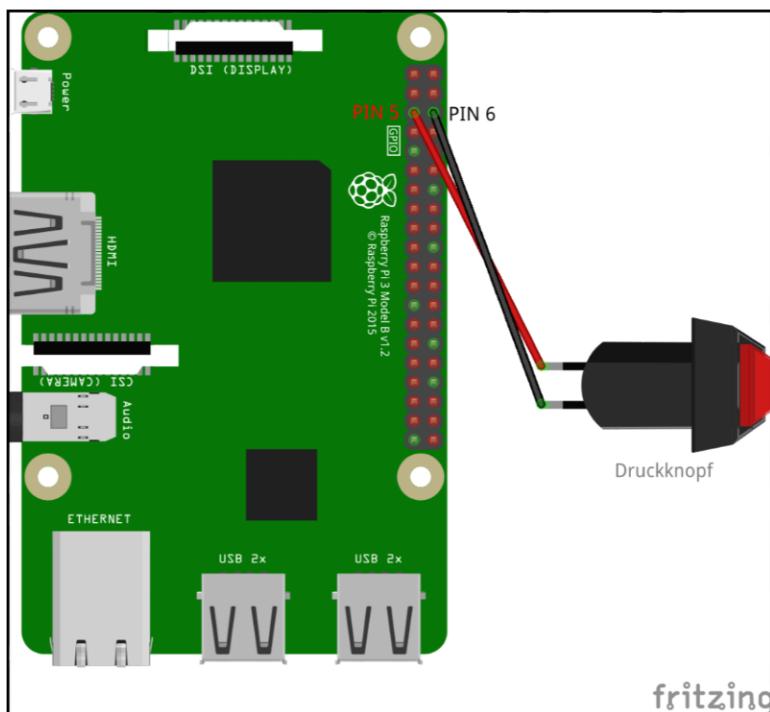


Abbildung 12: Druckknopf mit Kabelverbindung zum Raspberry Pi

Damit später die Pins über den Knopf angesteuert werden können, müssen einige zusätzliche Pakete installiert werden. Zuerst werden ein paar grundlegende Pakete über folgenden Befehle installiert.

```
sudo apt-get install python-dev
sudo apt-get install python3-dev
sudo apt-get install gcc
sudo apt-get install python-pip
```

Als nächstes muss RPI.GPIO mit dem nächsten Befehl heruntergeladen, dieses Paket ermöglicht die gewünschte Ansteuerung der Pins über Python.

```
wget
https://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.5.11.tar.gz
```

Im nächsten Schritt wird das heruntergeladen Paket entpackt und installiert, dafür muss folgender Befehl eingegeben werden.

```
sudo tar -zxvf RPi.GPIO-0.5.11.tar.gz  
-----  
cd RPi.GPIO-0.5.11  
-----  
sudo python setup.py install  
oder  
sudo python3 setup.py install
```

Wenn alle diese Schritte erfolgreich waren, kann RPI.GPIO in jedes Python Programm importiert werden und die PINs des Raspberry Pi können angesteuert und abgefragt werden.

9.1.3.2 Verkabelung LED-Band

Die Verkabelung des LED-Bands mit dem Raspberry Pi wird anhand folgender Abbildung 13 verdeutlicht. Es ist wichtig zu beachten, dass die Farben exemplarisch für diesen Prototyp genutzte Lichtband gelten, bei einem Nachbau, sollte darauf geachtet werden, dass die Bezeichnungen der Kabel am Lichtband, mit denen in der Abbildung 13 übereinstimmen, sollten die Farben der Kabel dabei gleich sein, ist dies umso besser.

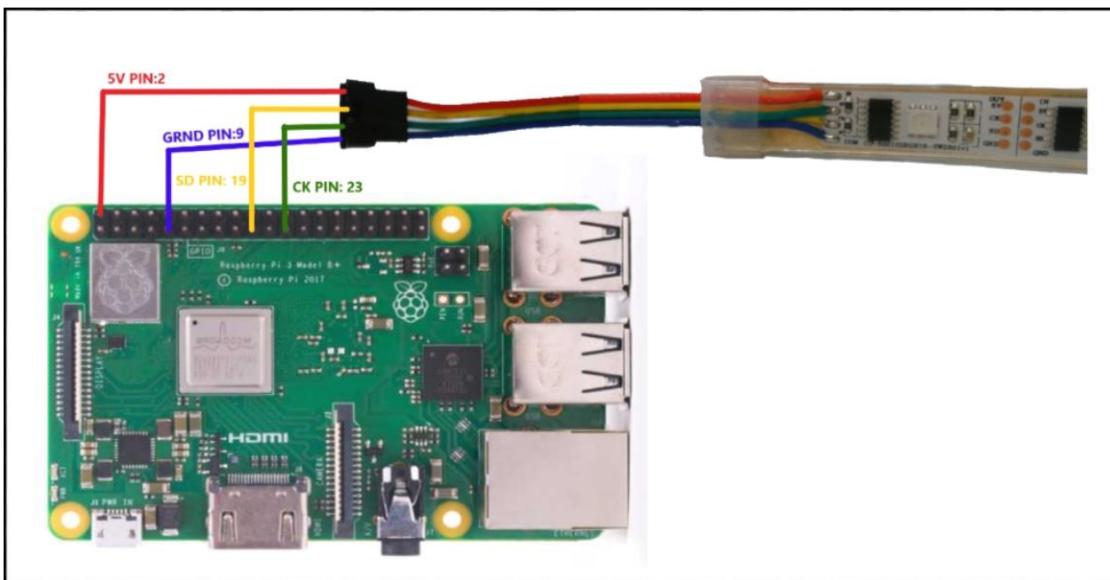


Abbildung 13: Raspberry Pi mit Anschlüssen ans das Lichtband

Um das Lichtband über Python Programme ansteuern zu können, muss das RPI.GPIO Paket installiert werden. Zusätzlich wird von Adafruit die Bibliothek „adafruit-ws2801“ installiert, dafür muss folgender Befehl im Terminal eingegeben werden.

```
sudo apt-get update  
sudo apt-get install python-pip -y  
sudo pip install adafruit-ws2801
```

Da diese Bibliothek den SPI Bus benötigt, muss dieser in der Raspberry Pi Konfiguration noch aktiviert werden. Dafür wird die Konfiguration durch folgendem Befehl geöffnet. Und im Menü wird der Punkt 5, wie in Abbildung 14 zu sehen ist, ausgewählt. Dort kann im Nachfolgendem Fenster unter Menüpunkt 4 der SPI Bus aktiviert werden.

```
sudo raspi-config
```

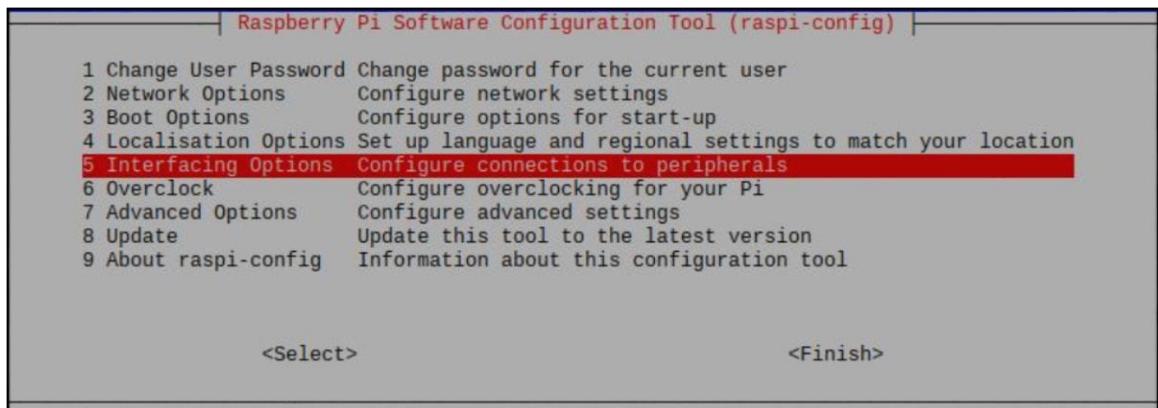


Abbildung 14: Konfigurationsmenü des Raspberry Pi

Ein Beispielcode mit dem Dateinamen „ws2801_example.py“ kann im GitHub heruntergeladen werden [13].

9.1.3.3 Verkabelung Kamera

Die Kamera wird über das, an ihr befindlichen, Flachbandkabel mit dem Raspberry Pi verbunden. Der Anschluss erfolgt über die 15-polige serielle MIPI-Schnittstelle, konkreter über das CSI, dem Camera Serial Interface. Dieser Anschluss befindet sich zwischen der Klinken- und HDMI-Buchse, was in folgender Abbildung 15 zu sehen ist. An der CSI-Steckverbindung muss dafür das obere Teil etwas nach oben gezogen werden und dann das Flachbandkabel mit der farbigen Markierung in Richtung Klinkenstecker-Anschluss gesteckt werden. Dann muss der Verschluss wieder nach unten gedrückt werden und der Kontakt ist hergestellt.

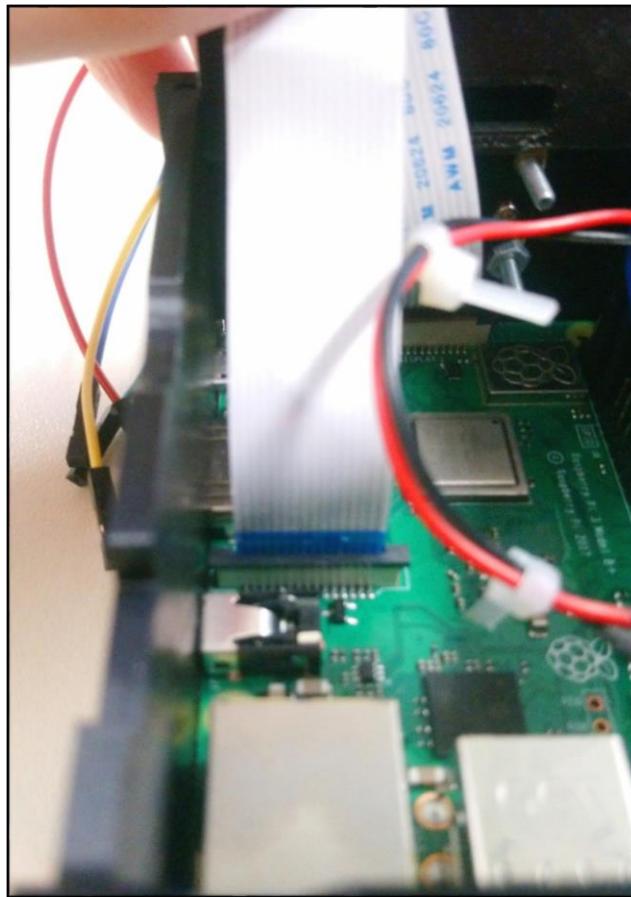


Abbildung 15: Kamera Flachbandkabel in CSI-Anschluss am Raspberry Pi

Um die Funktionsfähigkeit der Kamera zu testen, kann man im Terminal des Raspberry Pi einfach den folgenden Befehl eingeben, dann wird ein Bild aufgenommen und auf dem Pi gespeichert.

```
raspistill -o test.jpg
```

Möchte man ein Video aufnehmen muss der nachfolgende Befehl eingegeben werden. Der Wert **-w** gibt die Breite, **-h** die Höhe und **-t** die Dauer des Videos an.

```
raspivid -o video.h264 -w 1280 -h 720 -t 60000
```

In Python kann die Kamera direkt über die PiCamera Bibliothek angesteuert werden. Der nachfolgende Code kann in einem Python Datei eingefügt werden. Sobald der Code ausgeführt wird, wird sich ein Fenster öffnen, das den Bildausschnitt der Kamera zeigt.

```
from picamera import PiCamera
from time import sleep

camera = PiCamera()

camera.start_preview()
sleep(5)
camera.stop_preview()
```

9.1.3.4 Gehäuse

Das Gehäuse des Prototyps wurde mit Hilfe eines 3D-Drucker hergestellt. Dabei wurde PLA Filament mit einer Stärke von 1.75mm genutzt. Die Grundstruktur des Gehäuses entstammte einer Vorlage von Thingiverse [14] und wurde für den Prototyp angepasst. Die Mesh-Datei ist im GitHub hinterlegt [15]. In Abbildung 16 ist das Gehäuse samt Kamera und Druckknopf dargestellt.

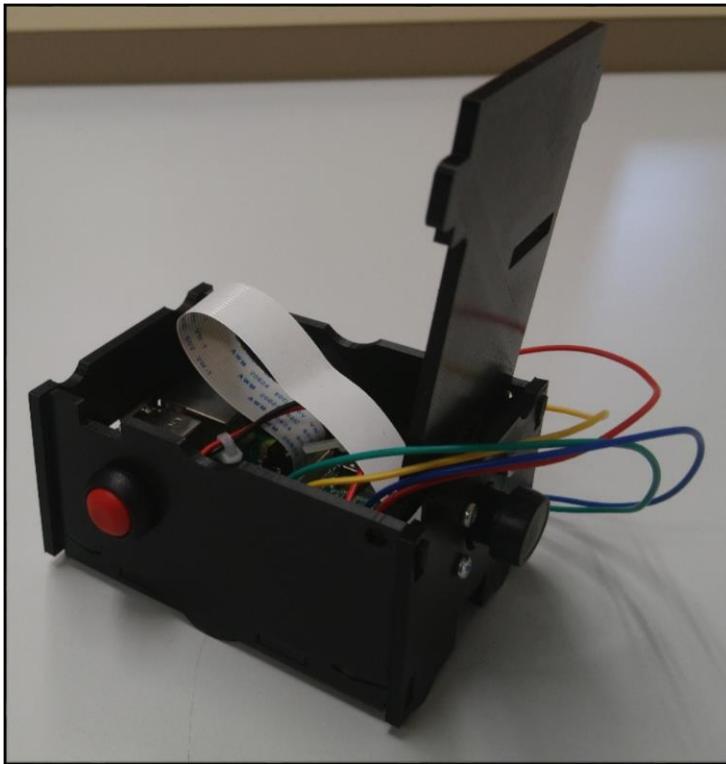


Abbildung 16: Raspberry Pi Gehäuse samt anbauten

9.1.4 Einrichtung des Clouddienstes

Um den Clouddienst einzurichten muss ein Microsoft Konto auf der Microsoft Azure Webseite erstellt werden [16]. Das genutzte Python-Skript benötigt zur Verbindung zum Clouddienst einen Schlüssel und eine Linkadresse damit eine Anfrage an den gewünschten Service gestellt werden kann. Dafür muss der Service abonniert werden, dies kann auf der Website des gewünschten Service, wie in Abbildung 17 zu sehen ist, getestet werden [17].

The screenshot shows the Microsoft Azure Cognitive Services Catalog page for 'Maschinelles Sehen'. At the top, there's a navigation bar with links like 'Übersicht', 'Lösungen', 'Produkte', 'Dokumentation', 'Preise', 'Schulungen', 'Marketplace', 'Partner', 'Support', 'Blog', 'Mehr', and 'Kostenloses Konto'. Below the navigation, a title 'Cognitive Services-Verzeichnis' and a subtitle 'Verwalten Sie Cognitive Services in der Azure-Cloud, oder testen Sie den Dienst mit temporärem Zugriff' are displayed. A sub-navigation bar includes 'Cognitive Services erkunden:', 'Cognitive Services-Homepage', 'Cognitive Services testen', 'Preise', and 'Dokumentation'. The main content area features tabs for 'Bildanalyse', 'Spracheingabe', 'Sprache', 'Entscheidung', and 'Suchen', with 'Bildanalyse' being the active tab. A descriptive text explains that users can experience how machine vision enables apps and services to recognize and analyze images. Below this, two service cards are shown: 'Maschinelles Sehen' (with a camera icon) and 'Gesichtserkennung' (with a face icon). Each card has a 'Kostenlos testen' button, which is highlighted with a red box in the screenshot. Other buttons include 'Weitere Informationen', 'Containerunterstützung', and various demo links.

Abbildung 17: Website des Azure-Vision-Service mit anzuwählenden Menüpunkt

Wenn man auf der Website den Menüpunkt „Kostenlos testen“ auswählt, kann man zwischen drei Abonnementtypen wählen, unabhängig von der Auswahl, folgt man den Anweisungen der Webseite.

Wenn man den Service abonniert hat, sieht man auf der Website, wie in Abbildung 18 zu sehen, zwei Schlüssel und die Linkadressen zum Service.

The screenshot shows the Microsoft Azure portal dashboard for the 'Maschinelles Sehen' service. At the top, there's a navigation bar with links like 'Übersicht', 'Lösungen', 'Produkte', 'Dokumentation', 'Preise', 'Schulungen', 'Marketplace', 'Partner', 'Support', 'Blog', 'Mehr', and 'erika_wolf89@outlook.de'. The user's email address is also displayed. Below the navigation, a message says 'Maschinelles Sehen wurde erfolgreich zu Ihrem Abonnement hinzugefügt.' In the main content area, there's a 'Status' section for 'Maschinelles Sehen'. It shows that the API-Key is currently active and will expire in 7 days. It also lists several URLs for different API versions and two specific keys: 'Schlüssel 1: 47eb7231eae5449d8a489776b13a51ab' and 'Schlüssel 2: 3d107ed7c6b94f2ab3c8e0f1e4136f62'. To the right of the status information, there's a 'Hallo!' greeting and the user's email address '@outlook.de (Abmelden)'.

Abbildung 18: Webansicht des abonnierten Service

9.2 Implementierung der Softwaremodule

Dieser Abschnitt erklärt, wie der Programmcode implementiert wurde, damit der Prototyp die einzelnen Hardware Module und den Clouddienst ansteuern kann, um die Abwehrmaßnahmen gegen den Wolf auslösen zu können.

9.2.1 Schnittstelle zum Clouddienst

Damit das Python-Skript, welches für die Verbindung zum Clouddienst notwendig ist, funktioniert, müssen vorher die Python-Module matplotlib und pillow installiert werden. Dafür muss im Terminal folgende Kommandos eingegeben werden.

```
pip3 install Pillow  
pip3 install Matplotlib
```

Im Skript muss als erstes der Schlüssel und die Service-URL, die in Kapitel 9.1.4 über das Abonnieren des Service erhalten wurden, in eine Variable eingebunden werden. Zusätzlich muss an den String der URL noch das Wort „analyze“ angefügt werden.

```
subscription_key=  
"1778cd1943264088b68ada56ef6da798"  
assert subscription_key  
  
vision_base_url=  
https://westcentralus.api.cognitive.microsoft.com/vision/v2.0/  
  
analyze_url = vision_base_url + "analyze"
```

Als nächstes müssen eine Header und eine Parameter Variable festgelegt werden, welche dafür sorgen, dass sowohl der Schlüssel an den Clouddienst übermittelt werden kann, als auch der Clouddienst die Informationen besitzt, welche Parameter bei der Antwort vorhanden sein müssen.

```
headers={'Ocp-Apim-Subscription-Key':subscription_key,'Content-Type':'application/octet-stream'}  
  
params={'visualFeatures':'Categories,Description,Color'}
```

Damit nun eine Anfrage zum Clouddienst gestellt werden kann, muss vorher noch die Bibliothek „requests“ importiert werden, diese HTTP Bibliothek für Python ermöglicht es HTTP/1.1 Anfragen zu versenden. Hat man die Bibliothek in sein Skript importiert, kann mit der „post“ Methode eine Anfrage an den Clouddienst gestellt werden, dabei wird neben der „params“ und der „header“ Variable auch ein Bild in Form einer „data“ Variable übergeben. Wenn man diese Methode an eine Variable übergibt, kann der Programmablauf mit der

Methode „raise_for_status()“ solange gestoppt werden, bis eine Antwort vom Clouddienst empfangen wurde.

```
// an den Skript Anfang
import requests

image_path = "/home/pi/image.jpg"
image_data = open(image_path, "rb").read()

response =
requests.post(analyze_url, headers=headers,
params=params, data=image_data)

response.raise_for_status()
```

9.2.2 Tags ausgeben

Sobald das Programm eine Antwort vom Clouddienst erhält, kann die Antwort mit der Methode .json() in einen JSON-String umgewandelt und an eine Variable übergeben werden. Durch die JSON-Notation kann durch Keys auf einzelne Elemente im JSON-String zugegriffen werden, welche dann wieder an eine Variable übergeben werden. Diese neue Variable, mit der Bezeichnung „tags“ ist dann ein Array und man kann über die Indizes auf die einzelnen Elemente in der Variable zugreifen. Im nachfolgendem Programmcode, ist das Verfahren abgebildet.

```
# The 'analysis' object contains various fields
# that describe the image. The most
# relevant caption for the image is obtained from
# the 'description' property.

analysis = response.json()
tags = analysis["description"]["tags"]
# Ausgabe aller Elemente der tags Variable
print(tags)
# Ausgabe des ersten Elements der tags Variable
print(tags[1])
```

9.2.3 Konfigurationsdatei

Wichtige Variablen wie Nutzernamen, Passwörter, Schlüssel oder einfach Konstanten, sollten in einer Konfigurationsdatei gespeichert werden. Für das Python-Skript des Prototypen wurde die Bibliothek „configparser“ genutzt. Da diese Bibliothek bereits in Python enthalten ist, muss nichts extra installiert werden, die Bibliothek muss lediglich in den Programmcode importiert werden. Im nachfolgendem Codeabschnitt wird gezeigt, wie man die Konfigurationsdatei in das Programm lädt und eine Variable aus der Datei übergibt und in Abbildung 19 sieht man, wie die Konfigurationsdatei aufgebaut sein muss.

```
import configparser

# Load the configuration file
config = configparser.ConfigParser()
config.sections()
config.read('config.ini')

subscription_key=config.get('azure','subscription_key')
```



```
[config - Editor]
Datei  Bearbeiten  Format  Ansicht  ?
[azure]
subscription_key= 70dc7e1915b0466f84f03b2b60d78dab
vision_base_url= https://westcentralus.api.cognitive.microsoft.com/vision/v2.0/

[image]
image_patch= /home/pi/image.jpg
```

Abbildung 19: Inhalt einer .ini Konfigurationsdatei für Python-Skripte

9.2.4 Soundausgabe- und Lichtabstrahlung-Thread

Die Wiedergabe von Geräuschen zur Wolfsabschreckung muss parallel mit der Bilderkennung stattfinden, daher wurde dieser Programmabschnitt als Thread konzipiert und durch Semaphore gesteuert. Einmal gestartet, wartet es, bis es ausgeführt wird und läuft dann solange, bis es wieder gestoppt wird. Da Python keine Möglichkeit liefert Musikdateien abzuspielen, wurde pygame installiert [18]. Eine Bibliothek die es ermöglicht Wiedergabelisten zu erstellen und diese abzuspielen. Der nachfolgende Programmcode zeigt, wie man mit pygame Musik abspielen kann.

```
import pygame

pygame.mixer.init()
playlist = list()
playlist.append("artillery.wav")
playlist.append("bear.wav")

pygame.mixer.music.load(random.choice(self.
playlist))
pygame.mixer.music.set_volume(1.0)
pygame.mixer.music.play()
```

Ziel der Wolfsabschreckung ist es, dass die Geräusche immer zufällig abgespielt werden. Dies erfordert allerdings, dass eine Zufallsfunktion für die Wiedergabeliste vorhanden ist, diese ist in pygame allerdings nicht vorhanden. Sodass die Bibliothek random [19] in den Programmcode importiert wurde um eine Zufallswiedergabe von Musikdateien zu ermöglichen. Dafür muss, wie im nachfolgendem Programmcode zu sehen, die Wiedergabeliste dem „random“ Objekt durch die Methode .choice() übergeben werden.

```
import random

random.seed()

#playlist stammt aus dem oberen Programmcode
#von pygame
pygame.mixer.music.load(random.choice(self.
playlist))
```

Auf GitHub [20] ist der komplette Programmcode mit dem Namen „audioPlayer.py“ zu finden.

Wie bei der Soundausgabe, muss auch die Lichtabstrahlung am Lichtband parallel mit der Bilderkennung stattfinden. Daher ist auch dieser Programmteil ein Thread und wird durch Semaphore gesteuert. Die Ansteuerung des Lichtbandes funktioniert über die GPIO-Pins des Raspberry Pi, daher wird die Bibliothek Adafruit_WS2801 [21] genutzt, die diese Ansteuerung ermöglicht. Das Lichtband kann dadurch in verschiedenen Farben blinken, in welchen Farben das Lichtband leuchtet muss im Programmcode definiert werden.

Auf GitHub [22] ist der komplette Programmcode mit dem Namen „lightShow.py“ zu finden.

10 Modultests

Im Rahmen des Anwendungspraktikums werden zunächst die verschiedenen Lichtverhältnisse untersucht. Dabei wird zum einen untersucht wie die Bildauswertung bei Tag funktioniert, wie zuverlässig sie bei Dämmerung ist und wie aussagekräftig bei Nacht. Zusätzlich soll untersucht werden inwieweit sich eine hohe Luftfeuchtigkeit, das heißt Nebel, auf die Bildqualität auswirkt. Dies kann überprüft werden indem die Bilder [WB1-24] in ihren Kontrastwerten angepasst werden. Gesetzt den Fall, dass die Tests eindeutige Ergebnisse liefern, kann darüber nachgedacht werden, die Ergebnisse zu nutzen um die Bilder zu optimieren bevor sie an den Clouddienst geschickt werden. So können auch in einer Situation in der das Bild eigentlich optimal scheint die Ergebnisse der Bildklassifizierung noch weiter verbessert werden. Die beim Test verwendeten Bilder sollen nach Möglichkeit den Wolf aus verschiedenen Winkeln zeigen und dabei auch Hindernisse wie zum Beispiel Bäume, Sträucher, Zäune, usw. beinhalten. So kann eine Vielzahl von verschiedenen Aufnahmeszenarien abgedeckt werden. Außerdem sollen zusätzlich Bilder von Hunden überprüft werden, um zu evaluieren wie gut die Bildklassifizierung zwischen Hund und Wolf unterscheiden kann.

Im Rahmen der Veranstaltung soll allerdings nur die von Microsoft Azure zur Verfügung gestellte Bildklassifizierung auf ihre Qualität in Bezug auf die Erkennung von Wölfen überprüft werden. Wie ein solcher Clouddienst generell verbessert werden kann ist nicht Gegenstand dieses Tests.

Jedes der Bilder die für den Test erstellt werden wird an MSA-CV gesendet und dort klassifiziert. Die Ergebnisse werden von MSA-CV zurückgeschickt und auf dem Testrechner zusammen mit den Bildern abgespeichert. Die Klassifizierungsdaten können dann hinsichtlich ihrer Signifikanz auf die Ergebnisse der Bildklassifizierung überprüft werden. Liefern die Tests eindeutige Ergebnisse, können Handlungsempfehlungen ausgesprochen werden.

10.1 Testdurchführung

Der Test besteht aus mehreren Schritten. Zunächst muss ein genaueres Ziel des Tests definiert werden. Dies ist bereits im vorherigen Abschnitt geschehen. Im nächsten Schritt muss überlegt werden, wie dieses Testziel am besten erreicht werden kann. Die Entscheidung fiel in diesem Fall auf das Erstellen und Auswertenlassen von Beispieldaten und die Analyse der Ergebnisse. Die dafür zu nutzende Hardware und Software sowie alle anderen Voraussetzungen wurden bereits im Pflichtenheft definiert. Entsprechend ist es der nächste Schritt diese Maßgaben zu nutzen und den Test durchzuführen. Dazu werden im Folgenden die einzelnen Schritte der Testdurchführung genauer erklärt. Dazu gehört zunächst das Erstellen der Testdaten, anschließend die Auswertung selbiger und zu Schluss das Abspeichern der Ergebnisse.

10.1.1 Erstellen der Testdaten

Für den Test wurden 24 verschiedene Bilder rausgesucht. Diese zeigen Wölfe und Hunde aus verschiedenen Perspektiven, oder auch von Hindernissen verdeckt. Ein Teil der Bilder wurde mit Wild- oder Nachtkameras aufgenommen und sind bereits entfärbt.

Diese 24 Bilder werden mit dem Bildbearbeitungsprogramm „XnViewMP“ [PG1] bearbeitet. Für diesen Test wurde wie in Abbildung 20 zu erkennen die 64bit Variante für Windows gewählt.

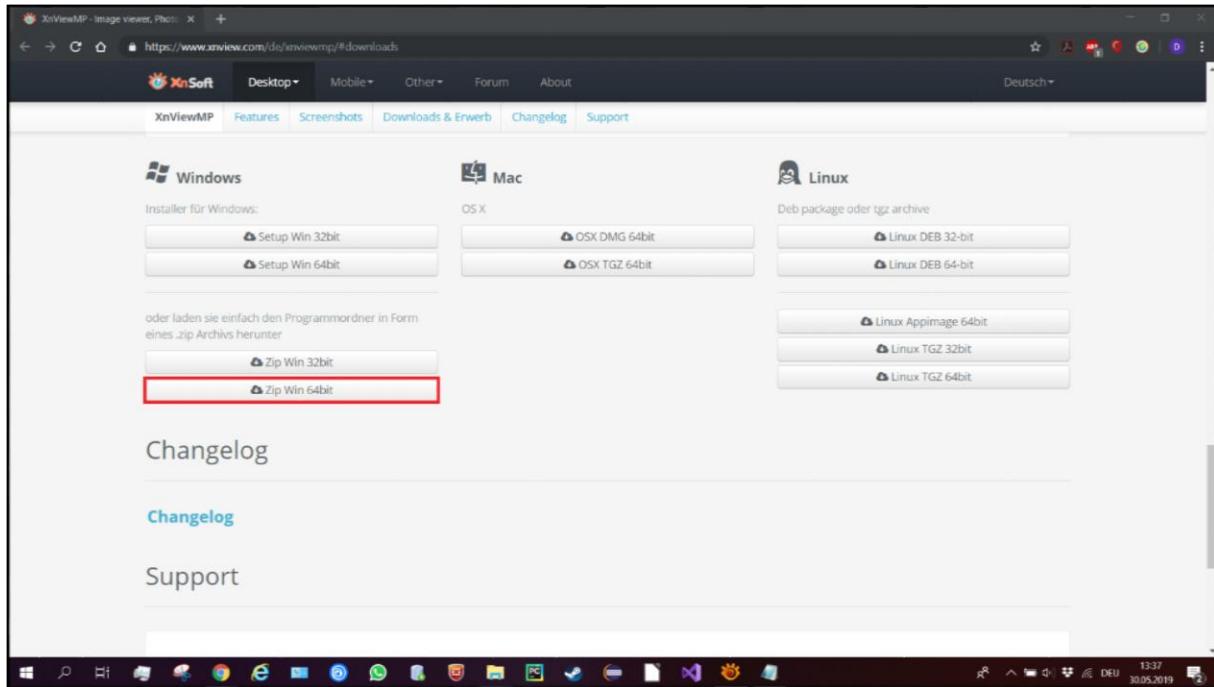


Abbildung 20: Download der 64bit-Version von „XnViewMP“ für Windows als .zip-Datei

Nachdem der Download abgeschlossen ist, kann die .zip-Datei entpackt werden und das Programm über die „xnviewmp.exe“ gestartet werden. Eine Installation ist nicht notwendig. XnViewMP bietet die Möglichkeit der Stapelbearbeitung, das heißt, eine vorher definierte Bildbearbeitungsroutine wird auf alle 24 Bilder angewendet. Dazu müssen diese wie in Abbildung 21 nur in dem Programm aus dem Dateibaum herausgesucht und selektiert werden. Anschließend können die selektierten Bilder der Stapelverarbeitung hinzugefügt werden.

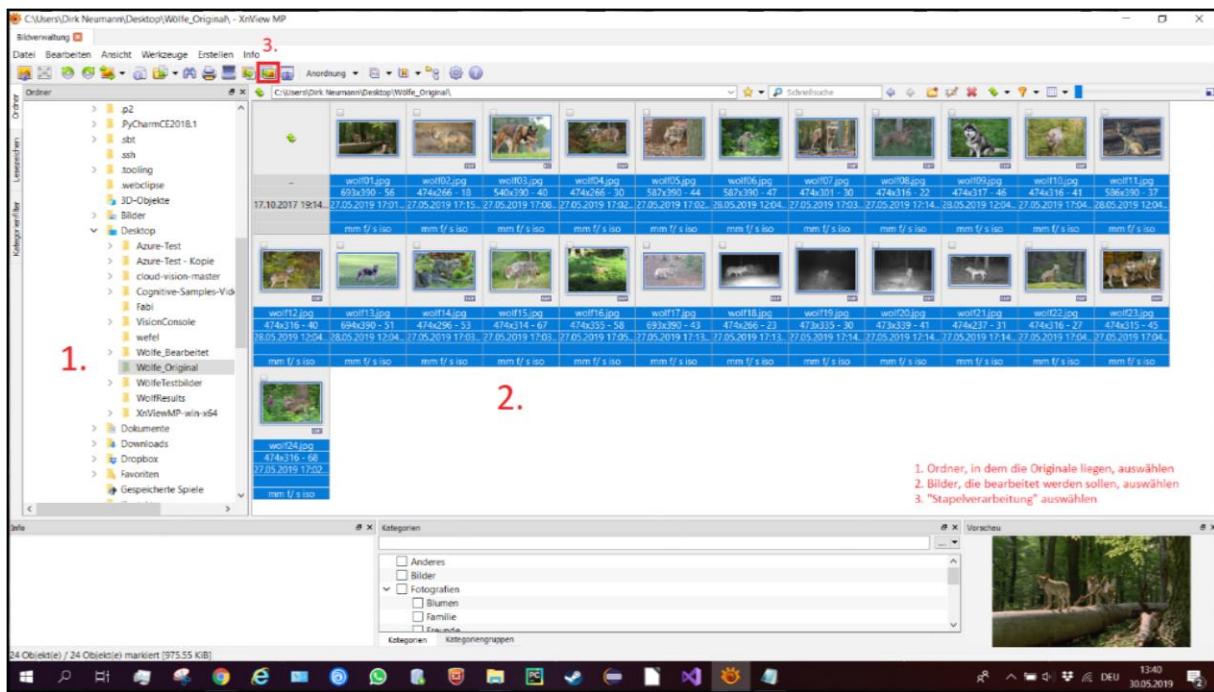


Abbildung 21: Laden der Originalbilder für die Stapelverarbeitung in "XnViewMP"

Für den Test wurden 21 verschiedene Routinen erstellt, welche sich zum einen in drei Methoden der Farbbearbeitung untergliedern lassen und auch in 7 Abstufungen von Fabian Theuerkauf, Dirk Neumann, Alexander Luft

Kontrastbearbeitungen. Diese drei Methoden und sieben Kontrastabstufungen werden jeweils miteinander kombiniert. Das Erstellen der Routinen ist in den Abbildungen 22 und 23 genauer zu sehen.

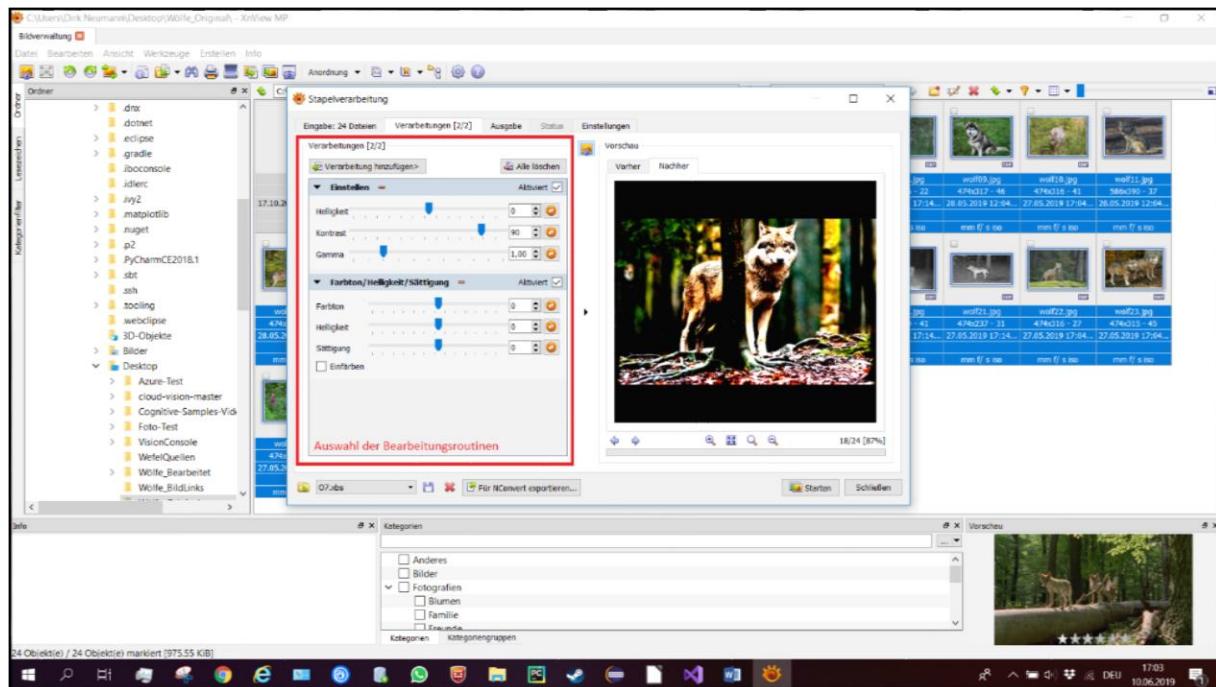


Abbildung 22: Auswählen der einzelnen Bearbeitungsschritte

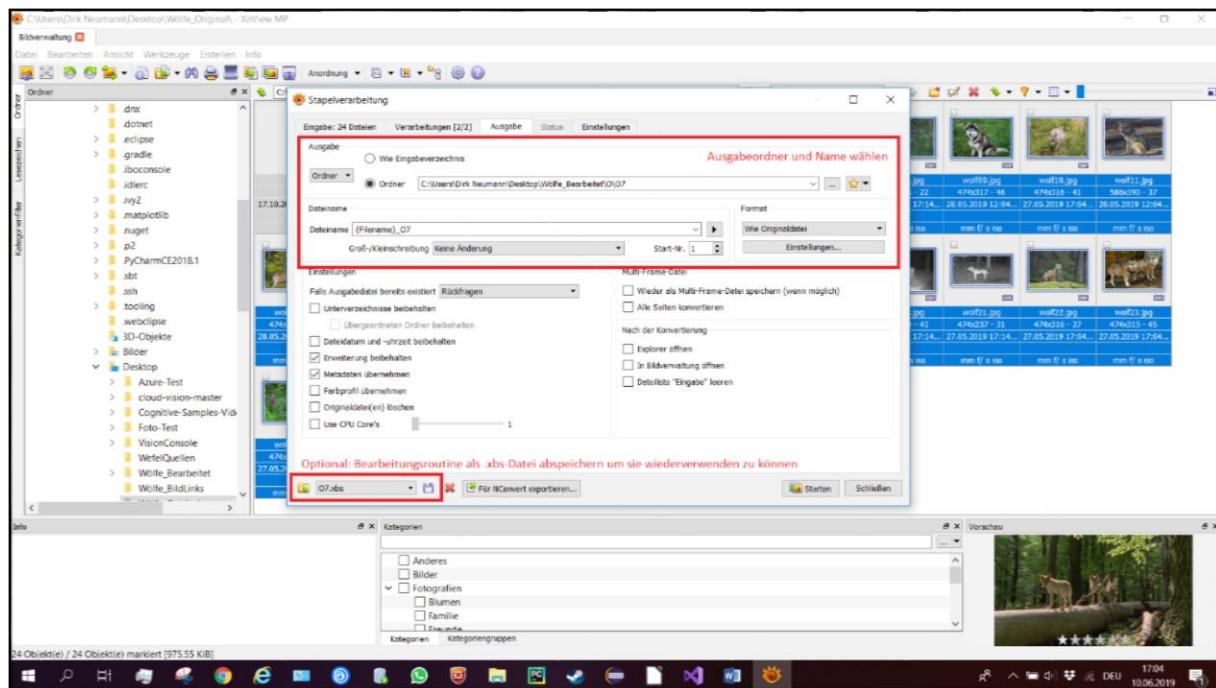


Abbildung 23: Festlegen von Speicherort und Name der bearbeiteten Bilder und optionales Speichern der Routine

Diese 21 verschiedenen Bildbearbeitungsroutinen wurden nach einer Kombination aus Buchstaben und Zahlen benannt, die in Tabelle 3 erklärt wird. Anschließend wurden die 21 Routinen abgespeichert und auf jedes der 24 Bilder angewendet.

Tabelle 3: Benennungen der Testbilder

O	Das Bild wurde in Bezug auf die Farbsättigung und die Helligkeit nicht bearbeitet.
D	Das Bild wurde verdunkelt. Auf einer Skala von -127 (komplett schwarz) bis +127 (komplett weiß) lag das Original bei 0. Nach der Bearbeitung liegt die Kopie auf der Skala bei -50.
SW	Das Bild wurde entfärbt. Die Helligkeit ist gleich geblieben.
1	Der Kontrastwert liegt bei -90 (Skala von -127 bis +127)
2	Der Kontrastwert liegt bei -60 (Skala von -127 bis +127)
3	Der Kontrastwert liegt bei -30 (Skala von -127 bis +127)
4	Der Kontrastwert liegt bei 0. Entspricht dem Originalbild (Skala von -127 bis +127)
5	Der Kontrastwert liegt bei +30 (Skala von -127 bis +127)
6	Der Kontrastwert liegt bei +60 (Skala von -127 bis +127)
7	Der Kontrastwert liegt bei +90 (Skala von -127 bis +127)

10.1.2 Klassifizierung der Testbilder

Zur Klassifizierung der Bilder wurde im Rahmen der Lehrveranstaltung ein 7-Tage kostenloses Testkonto erstellt, mit dem maximal 20 Bilder pro Minute und 5000 Bilder insgesamt von MSA-CV ausgewertet werden können. Die Qualität der Analyse ist in dieser Version nicht reduziert, womit sich die Testergebnisse nicht von denen mit einem bezahlten Konto unterscheiden.

Um die Testdaten an MSA-CV zu schicken wurde ein Testprogramm [PG4] geschrieben, um das Hochladen der einzelnen Bilder zu automatisieren und eine Vorschau der Ergebnisse für den Tester sichtbar zu machen. Dieses beinhaltet alle Bibliotheken die notwendig sind um es in einer Windowsumgebung nutzen zu können. Um das Testprogramm in vollem Umfang nutzen zu können ist eine installierte MS-Excel Version notwendig. Das Programm schreibt die für die Wolfserkennung relevanten Klassifizierungsergebnisse in eine Excel Datei, um mit diesen weiterarbeiten zu können.

Nach Programstart wird der Tester gebeten seine Testdaten zu selektieren in dem er den Ordner wählt, in dem diese liegen. Alle JPG-Dateien die in diesem Ordner oder seinen Unterordnern liegen werden nun von dem Testprogramm registriert. Anschließend öffnet sich das Testfenster und der Tester kann den Test starten, sofern er unter „Settings“ einen gültigen API-Schlüssel und Endpunkt für MSA-CV angibt. Ein solcher kann mit einem kostenlosen Nutzerkonto generiert werden. Pro zu analysierendem Bild dauert ein Testdurchgang ca. drei Sekunden. Diese Einstellung wurde getroffen, um auch mit dem umfangärmsten Konto bei MSA-CV testen zu können. Andernfalls würden die Anfragen nicht beantwortet werden, wenn sie 20 je Minute übersteigen. Der Testfortschritt ist im Testfenster erkennbar, um abschätzen zu können, wie lang der Test noch dauern wird. Während die Testdaten ausgewertet werden, ist keine weitere Handlung des Nutzers notwendig. Der Test kann also unbeaufsichtigt weiterlaufen.

10.1.3 Abspeichern der Ergebnisse

Im Verlauf des Tests werden die Bilder nach und nach an MSA-CV geschickt und die Ergebnisse empfangen. Von jedem Bild wird eine Kopie erzeugt und auf diese werden die Bildklassifizierungsergebnisse geschrieben. Der Nutzer sieht bereits während der Test läuft was von der Bildklassifizierungssoftware erkannt wurde. Zur erleichterten Unterscheidung werden alle Klassifizierungsdaten, die auf die Erkennung eines Wolfes schließen lassen rot eingefärbt um sich von den restlichen, gelbeingefärbten, Ergebnissen zu unterscheiden. Das Ergebnisbild wird dann in einen neuen Unterordner gespeichert der in dem vom Nutzer als Startordner selektierten Ordner liegt. Dieser wird „AzureResults“ genannt. Zu beachten ist, dass falls dieser bereits existiert, die Bilder einfach in diesen abgelegt werden. Der Dateiname bleibt erhalten, entspricht also dem Dateienamen des Originals. Sind in diesem Ordner bereits Bilder mit dem Namen der Ergebnisse vorhanden werden diese überschrieben. Ist, wie unter „10.1.2 Klassifizierung der Testbilder“ beschrieben, eine Excel Version vorhanden, wird zusätzlich zu den Ergebnisbildern eine Excel Datei angelegt. Hierfür ist eine weitere Einschränkung der Originaldaten notwendig. Deren Dateinamen müssen dem Muster „wolf{Zahl}_{legaler Ausdruck aus Tabelle 1}\“ entsprechen, um die Ergebnisse des Tests richtig abspeichern zu können. Aus dem Teil „wolf{Zahl}“ wird ermittelt, um das wievielte Testbild es sich handelt und der zweite Teil „{legaler Ausdruck aus Tabelle 1}\“, welche der 21 Bildbearbeitungsroutinen auf das ursprüngliche Original angewandt wurde um dieses Testbild zu erzeugen. Sind diese Benennungsvorschriften eingehalten, wird die Tabelle in Excel während des Tests richtig gefüllt. Eine weitere Einschränkung gibt es jedoch noch. Es ist darauf zu achten, dass die Tabelle nicht vorher geöffnet war uns falls doch, das im Taskmanager unter Windows kein Task von MS-Excel mehr aktiv ist.

Falls dies doch der Fall ist, kann die Tabelle bei Teststart nur im Lesemodus geöffnet werden und das Programm stürzt ab, sobald versucht wird das Dokument am Ende zu speichern.

10.2 Testauswertung

Die im Verlauf der Tests entstandenen Ergebnisdaten werden nun ausgewertet. Da die Ergebnisse je Testbild jedoch sehr unterschiedlich sein können, müssen diese Daten zunächst aufbereitet werden, sodass homogene Datenreihen entstehen mit denen weitergearbeitet werden kann. Ist die geschehen wird untersucht, ob die unterschiedlichen Bildbearbeitungsroutinen einen signifikanten Teil zur Qualität der Bildklassifizierung in MSA-CV beitragen. Die Ergebnisse aus dieser Untersuchung können dann genutzt werden, um Handlungsempfehlungen auszusprechen wie Bilddaten am besten aufzubereiten sind, um sie anschließend auswerten zu lassen und möglichst genaue Bildklassifizierungsinformationen vom Clouddienst MSA-CV zu erhalten. So kann die Effizienz des Systems gesteigert oder gegebenenfalls auch um neue Funktionalität erweitert werden. Dies wäre zum Beispiel der Fall, sollte sich herausstellen, dass die Originaldaten von Nachtaufnahmen mit Wölfen sehr ungenaue Ergebnisse, also eine geringe Erkennungssicherheit für Wölfe, liefern, aber eine der Bearbeitungsroutinen die Ergebnisse derart aufwertet, dass Nachtaufnahmen dennoch sehr zuverlässig ausgewertet werden können.

10.2.1 Aufbereitung der Ergebnisdaten

Nachdem alle Einschränkungen berücksichtigt wurden und der Test abgeschlossen ist, befindet sich im vom Tester zu Beginn gewählten Ordner zum einen ein Unterordner mit allen Testbildern und den darauf gedruckten Klassifizierungsdaten und zum anderen eine Excel Tabelle, welche für jedes Bild an der entsprechenden Stelle in der Tabelle einen Eintrag beinhaltet.

Nach ersten Tests fiel auf, dass es mehrere Stempel gibt, welche in Verbindung zu Wölfen stehen und entweder „canis“ oder „wolf“ als Subzeichenkette beinhalten. Diese wurden

in anschließenden Tests notiert und beim Eintragen in die Excel-Tabelle abgekürzt, um die Einträge je Tabellenzelle möglichst kurz zu halten und so die Tabelle nicht zu unleserlich werden zu lassen. Zusätzlich wird zu dem abgekürzten Stempel auch die Sicherheit abgespeichert, mit der dieser Stempel vom Bildklassifizierungsalgorithmus erkannt wurde. Der Wert für die Sicherheit bei der Erkennung wird dabei in Prozent angegeben und auf zwei Nachkommastellen gerundet. Wurden mehrere Stempel erkannt, werden die Einträge für diese durch „|“ getrennt, wurde kein Stempel in Verbindung zu einem Wolf gefunden, bleibt die Zelle leer. Tabelle 4 gibt einen Überblick darüber, welche Stempel erkannt wurden, was sie laut Systematik der Biologie bedeuten und wie sie abgekürzt wurden.

Tabelle 4: Übersicht der Stempel aus MSA-CV, die mit Wölfen in Verbindung stehen

Stempelname	Systematik der Biologie aus [7,8]	Abkürzung
canis	Lateinische Bezeichnung der Gattung von wolfs- und schakalartigen Tieren (canis lupus ≡ Wolf)	C
wolf	Das Tier „Wolf“	W
canis lupus tundrarum	(Alaska-)Tundrawolf	CLT
redwolf	Rotwolf – Diese Spezies ist etwas kleiner, kann für Schafe aber genauso gefährlich werden wie richtige Wölfe. Ehemals ausgestorben, erfolgreich wieder in North Carolina ausgewildert.	RW
wolfdog	Wolfshund – Mischling aus Wolf und Hund	WD

Die Einträge in der Excel Tabelle können also sehr unterschiedlich aussehen. Um eine wissenschaftliche Methodik aus die Daten anwenden zu können, müssen sie also so aufbereitet werden um mit ihnen weiter arbeiten zu können. Zunächst werden dafür alle Einträge bei denen mehrere Stempel erkannt wurden auf denjenigen Stempel reduziert, dessen Sicherheit am höchsten war und dann von dem Präfix „{Abkürzung}“ befreit. Danach steht also nur noch eine Zahl in der entsprechenden Zelle. Die leeren Zellen werden mit „0,00“ ersetzt um zu zeigen, dass hier kein Wolf erkannt wurde. Diese Form der Datenaufbereitung stützt sich auf folgende Annahme:

Unabhängig davon welcher der in Tabelle 4 gelisteten Stempel erkannt wird, dieser symbolisiert ein Raubtier, in dessen Beuteschema ein Schaf auf einer Weide fallen würde. Besagtes Tier stellt entsprechend eine potenzielle Bedrohung für die Schafe dar und muss abgewehrt werden. Die Art des Wolfes spielt keine Rolle und selbst wenn es sich um einen Mischling aus Hund und Wolf handelt, so ist dieser mit hoher Wahrscheinlichkeit nicht domestiziert, da Paarungen zwischen Wolf und Hund hauptsächlich mit Wildhunden oder streunenden Hunden zustande kommen.

Nach genannter Datenaufbereitung liegen die Daten so vor, dass mit Ihnen weitergearbeitet werden kann, um zu bestimmen welche Bildbearbeitungsroutinen welchen Einfluss auf die korrekte Bildklassifizierung haben.

10.2.2 Analyse der Ergebnisdaten

Um die erfassten Daten auszuwerten, wurde die Datenanalysemethode der Varianzanalyse, kurz ANOVA vom englischen analysis of variance, gewählt. „Die Idee der einfaktoriellen Varianzanalyse besteht darin, den Einfluss des Faktors A auf die Zielvariable Y an Hand der durch A bedingten Variation zu beurteilen, und zwar im Rahmen eines Vergleichs mit der durch den Versuchsfehler bedingten Variation“ [7]. In diesem Fall ist der Faktor A die Methode die auf die Bilder angewendet wurde, also die zuvor erstellte Bildbearbeitungsroutine.

Die Zielvariable Y ist die Sicherheit mit der ein Wolf erkannt wurde. Die durch den Versuchsfehler bedingte Variation ist in diesem Fall die Variation unter den verschiedenen Originalbildern, also die Variation zwischen zum Beispiel einem Bild, auf welchem der Wolf von vorne zu sehen ist, und einem Bild, auf welchem der Wolf von der Seite zu sehen ist. Die Nullhypothese H0 dieser Datenanalysemethode ist es, dass die Mittelwerte der verschiedenen Methoden gleich sind. Diese muss verworfen werden, wenn der p-Wert der bei der ANOVA berechnet wird unter einem selbst festgelegten Konfidenzintervall α liegt. Wird beispielsweise $\alpha = 0.1$ gewählt so bedeutet dies, dass es uns genügt, wenn wir uns zu $0.9 \leq 90\%$ sicher sind H0 zu verwerfen zu können. Eine 100-prozentige Sicherheit können wir uns nicht geben lassen, also wählen wir für diesen Test 95% Sicherheit und damit $\alpha=0.05$.

Insgesamt gilt: Werden die Daten so kumuliert, dass überprüft werden kann, welchen Einfluss eine Methode auf die Ergebnisse hat, so kann mit der ANOVA überprüft werden, ob diese Methode signifikante Auswirkungen auf die Ergebnisse hat. Ist $p < 0.05$, dann hat diese Methode signifikante Auswirkungen auf die Ergebnisse.

Im Folgenden wird der Ablauf bei der ANOVA für eine Untersuchung erläutert. Dies geschieht am Beispiel der Farbmethoden, also der Verdunklung und der Entfärbung. Um dies zu untersuchen müssen wir die Ausgangsdaten zunächst richtig kumulieren. Abbildung 24 zeigt alle Ausgangsdatensätze nachdem sie wie in Kapitel 10.2.1 beschrieben aufbereitet wurden. Aus diesen werden die nötigen Datensätze erzeugt, welche dann für die ANOVA benötigt werden. In diesem speziellen Fall ist die Methode A die Farbveränderung, also Original, verdunkelt und entfärbt, und der Versuchsfehler weiterhin die Unterschiede zwischen den Motiven der verschiedenen Bilder. Dabei ist das Problem, dass wir je Farbmethode noch einmal sieben Abstufungen, nämlich die sieben Kontraststufen, haben. Um diese jeweils 7 Datensätze zu kumulieren verwenden wir das arithmetische Mittel. So wird aus den sieben Einträgen von W1:O1, also Wolf 1 im Original mit Kontrast -90, bis W1:O7, also Wolf 1 im Original mit Kontrast +90 ein Wert. Dieser ist in Abbildung 25 in Zelle O:wolf1 niedergelegt. Das Gleiche passiert nun auch für die anderen jeweils sieben Datensätze die dasselbe Motiv und dieselbe Farbmethode haben.

	O1	O2	O3	O4	O5	O6	O7	D1	D2	D3	D4	D5	D6	D7	SW1	SW2	SW3	SW4	SW5	SW6	SW7
W1	51,57	85,04	75,26	63,02	72,36	64,08	27,46	41,98	78,76	70,50	71,79	42,96	0,00	0,00	0,00	0,00	63,21	42,46	35,35	27,83	
W2	78,61	81,69	88,49	87,53	89,13	86,00	39,47	66,20	80,95	84,29	88,08	24,47	66,38	56,74	51,25	73,18	80,44	86,08	86,40	76,74	73,28
W3	36,58	64,91	67,28	65,97	69,51	65,67	0,00	0,00	61,89	66,07	72,69	63,80	60,24	0,00	57,74	60,33	70,52	75,65	77,30	66,88	0,00
W4	42,87	86,91	82,98	59,89	65,32	52,28	61,12	45,55	75,07	83,62	28,66	53,59	0,00	0,00	0,00	90,02	94,91	92,60	90,39	87,92	73,29
W5	95,67	97,95	98,22	93,76	88,19	88,37	75,79	95,00	96,63	93,48	94,34	85,36	76,32	80,90	55,41	86,65	90,20	91,92	94,09	96,19	86,62
W6	95,66	94,71	92,31	89,81	85,01	79,98	54,16	93,35	89,63	89,91	70,32	69,55	80,47	63,73	92,72	95,29	95,49	95,04	83,47	51,44	0,00
W7	93,89	86,61	89,45	85,12	90,98	66,70	0,00	81,29	95,13	84,89	78,23	60,40	0,00	0,00	84,36	94,12	88,97	86,12	88,90	88,20	35,42
W8	0,00	51,67	54,20	78,75	73,09	0,00	0,00	0,00	56,60	53,69	59,14	0,00	0,00	0,00	0,00	80,13	89,70	83,20	69,43	0,00	0,00
W9	42,60	52,92	32,78	57,16	55,04	0,00	0,00	38,06	0,00	0,00	0,00	0,00	0,00	0,00	35,74	31,26	27,90	53,80	50,95	40,86	35,46
W10	51,80	75,07	78,13	76,20	61,49	41,34	0,00	39,72	63,18	77,06	65,80	50,05	38,74	0,00	36,78	68,18	74,27	77,83	78,06	0,00	57,02
W11	0,00	50,82	72,62	77,96	80,99	57,07	0,00	0,00	0,00	56,49	45,78	0,00	0,00	0,00	0,00	34,55	62,51	59,22	76,17	39,85	
W12	96,75	98,67	99,57	98,87	99,15	99,23	92,43	97,04	98,09	98,76	98,72	98,14	97,69	96,62	93,74	97,36	98,61	98,96	98,10	98,82	97,74
W13	0,00	29,95	0,00	39,43	37,67	38,54	25,23	0,00	0,00	0,00	0,00	0,00	0,00	0,00	41,98	43,17	31,65	33,21	43,26	0,00	0,00
W14	67,34	74,18	72,62	76,55	79,17	72,99	60,79	85,28	76,98	78,03	79,11	66,36	0,00	0,00	62,92	81,42	68,74	61,86	64,74	82,11	73,58
W15	89,79	95,32	98,68	96,44	96,97	93,74	26,39	92,22	94,00	97,10	88,00	84,60	44,53	0,00	0,00	42,63	91,89	96,81	90,80	61,20	27,48
W16	0,00	70,19	92,88	90,68	91,35	79,01	0,00	0,00	71,98	85,82	91,95	93,90	32,54	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
W17	0,00	63,59	76,36	89,47	76,33	42,78	0,00	0,00	54,78	68,03	76,04	67,08	31,19	51,63	0,00	0,00	0,00	49,59	0,00	0,00	0,00
W18	0,00	39,22	65,04	63,32	0,00	0,00	0,00	0,00	39,08	45,30	37,44	0,00	0,00	0,00	0,00	39,55	67,64	61,55	28,06	0,00	0,00
W19	0,00	0,00	0,00	0,00	0,00	0,00	30,11	0,00	27,84	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	30,11	
W20	0,00	0,00	26,64	34,73	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	28,61	0,00	0,00	0,00
W21	0,00	0,00	30,35	46,64	51,77	0,00	0,00	0,00	0,00	26,24	34,99	0,00	0,00	0,00	0,00	0,00	30,35	46,64	51,77	0,00	0,00
W22	30,95	0,00	0,00	0,00	0,00	0,00	0,00	0,00	27,60	32,65	0,00	0,00	0,00	0,00	0,00	0,00	0,00	32,72	27,64	68,57	
W23	91,16	98,08	95,49	89,31	86,71	93,30	80,24	86,43	92,40	97,43	90,22	87,00	64,43	0,00	71,91	78,44	79,84	78,83	80,46	93,28	98,31
W24	67,59	86,32	85,51	72,06	64,48	46,06	64,52	73,68	86,34	78,57	61,10	44,81	0,00	0,00	0,00	83,30	77,49	83,80	81,19	86,28	80,35

Abbildung 24: Aufbereitete Ausgangsdaten

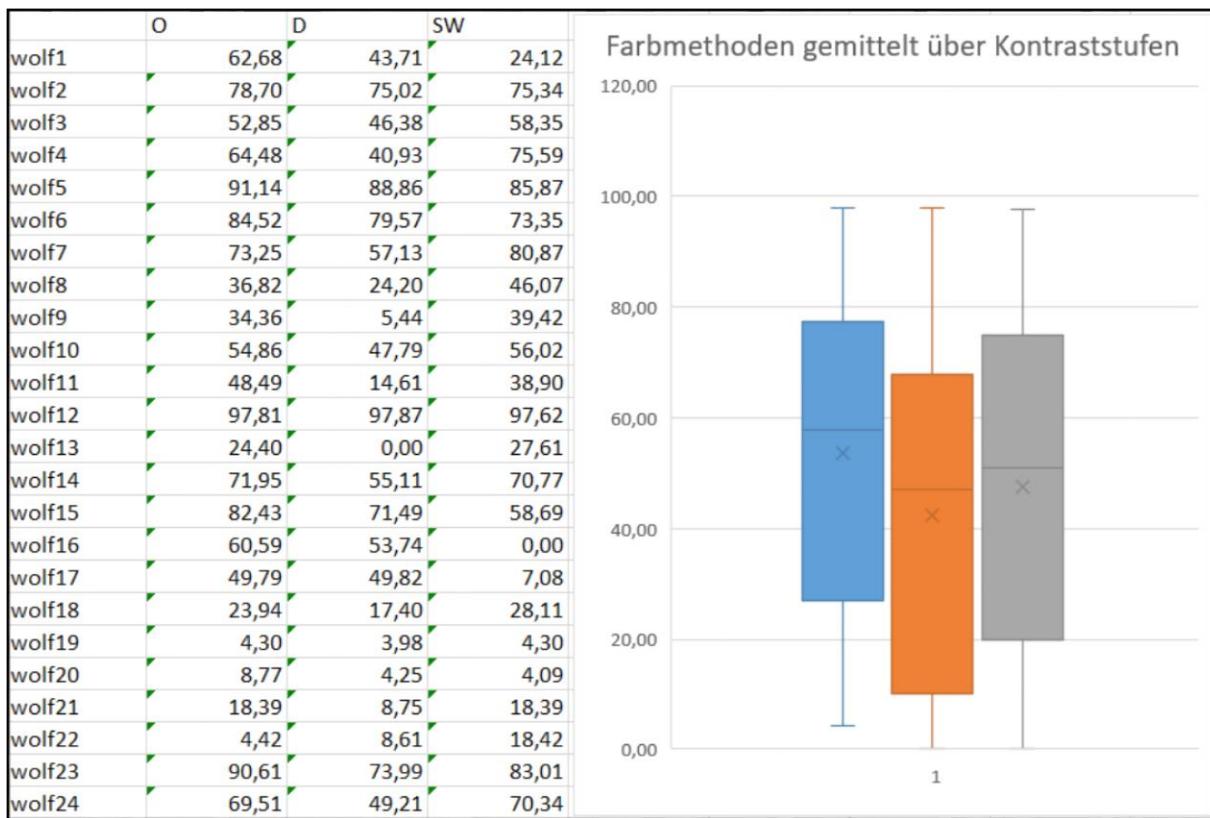


Abbildung 25: Farbmethode daten gemittelt über Kontraststufen

Nachdem alle nötigen Datensätze erzeugt wurden, werden diese in einem Diagramm dargestellt. Ein Kastendiagramm wie in Abbildung 25 zu sehen, ist dafür am besten geeignet, denn dieses stellt die gesamte Datenreihe dar, mit allen für einen Überblick notwendigen Kennzahlen. Diese sind in Tabelle 5 genauer beschrieben.

Tabelle 5: Kennzahlen im Kastendiagramm

Name	Grafische Darstellung	Bedeutung
Antennen/ Whiskers	Dünne senkrechte Linien am oberen und unteren Ende der Box, begrenzt durch horizontale Markierungen oder Diagrammende	Intervall in dem alle Werte liegen. Die obere und untere Grenze entsprechen Maximum und Minimum der Daten.
Box	Farbiges Rechteck	Bereich in dem 50% aller Werte liegen. Wird von erstem und drittem Quantil begrenzt.
Median	Durchgezogene Linie innerhalb der Box	Median der Daten. 50% der Daten liegen oberhalb der Linie, 50% unterhalb. \triangleq zweitem Quantil
Arithmetisches Mittel	X innerhalb der Box	Alle Werte aufsummiert geteilt durch Anzahl der Werte.

Wie in der Kastengrafik zu sehen, sind die Intervalle der Farbmethoden sehr ähnlich. Die Maxima sind alle nahe der 100%. Bei Betrachtung der Daten in Abbildung 26, ist zu erkennen, dass die Maxima bei 97.81%, 97.87% und 97.62% liegen sich also nur um Hundertstel unterscheiden. Die Minima liegen bei 4.3%, 0% und 0% was ebenfalls relativ nah beieinander ist, bedenkt man, dass die Werte von fast 0% bis fast 100% gestreut sind. Die Boxen überschneiden sich mehr als sie sich unterscheiden, was ebenfalls sehr ähnliche Ergebnisse bedeutet. Die arithmetischen Mittel, um die sich die Nullhypothese dreht, liegen augenscheinlich allerdings nicht sehr dicht beieinander, was aber auch hier wieder in Relation auf die gesamte Streuung betrachtet werden muss.

Anova: Einfaktorielle Varianzanalyse			Methode: Farbmethoden O D SW			
ZUSAMMENFASSUNG						
Gruppen	Anzahl	Summe	Mittelwert	Varianz		
O	24	1289,06	53,71083333	821,7937056		
D	24	1017,861429	42,41089286	881,7552149		
SW	24	1142,331429	47,59714286	902,6832092		

ANOVA						
Streuungsursache	Quadratsummen (SS)	Freiheitsgrade (df)	Mittlere Quadratsumme (MS)	Prüfgröße (F)	P-Wert	kritischer F-Wert
Unterschiede zwischen den Gruppen	1535,704441	2	767,8522203	0,883864731	0,417808335	3,129643983
Innerhalb der Gruppen	59943,33898	69	868,7440432			
Gesamt	61479,04342	71				

Abbildung 26: ANOVA Farbmethode daten gemittelt über Kontraststufen

Werden die Datenreihen mit der von MS-Excel zur Verfügung gestellten ANOVA ausgewertet, wird die Ausgabe generiert, wie sie in Abbildung 26 zu sehen ist. Interessant für die Bewertung der Nullhypothese sind hier nur der p-Wert in der unteren Tabelle und die in der oberen Tabelle dargestellten Mittelwerte, also arithmetische Mittel, und die Varianzen pro Gruppe.

Hier ist wie zuvor schon aus dem Kastendiagramm zu erkennen, dass die Mittelwerte zwar nicht sehr nahe beieinanderliegen, aber die sehr großen Varianzen diesen Unterschied relativieren. Der p-Wert zeigt abschließend mit einem Wert von 0.41, dass die Nullhypothese angenommen werden kann, da gilt $p \geq \alpha$. Das bedeutet, die Mittelwerte sind, vor gegebenem statistischen Hintergrund, gleich und die Variation der Werte zwischen den Farbmethode ist kleiner als die Variation der Werte innerhalb einer Methode. Es konnte also gezeigt werden, dass die Veränderung der Farbe, egal ob es sich dabei um die Verdunklung oder die Entfärbung handelt, keinen signifikanten Einfluss auf die Qualität der Wolfserkennung hat.

So wie im vorangegangenen Teil dieses Kapitels können auch die anderen Datensätze ausgewertet werden. Dafür wurden zusätzliche ANOVA durchgeführt, um weitere Erkenntnisse aus den Testdaten zu sammeln.

Nachdem festgestellt wurde, dass die Farbmethode keinen Einfluss auf die Wolfserkennung haben, wurde beleuchtet wie groß der Einfluss des Kontrastes innerhalb einer Farbmethode ist. Zu untersuchen ist dabei, ob die Kontrastveränderung unter den Bildern einer Gruppe, zum Beispiel innerhalb aller Schwarz-Weiß-Bilder, signifikante Unterschiede bewirkt. Die ANOVA dafür sind in der Exceldatei „DataAnalysis.xlsx“ in dem Datenblatt „ANOVA Kontrast“ zu finden. Wie sich aus den dort zu findenden Auswertungstabellen, ähnlich denen in Abbildung 26, erkennen lässt, hat der Kontrast auf alle drei Farbmethode einen signifikanten Einfluss, da der p-Wert der jeweiligen ANOVA immer kleiner als 0,05 ist. Der größte Mittelwert der Wolfserkennung aus den Ergebnissen liegt jedoch bei zwei von drei Farbmethode bei 100% des Kontrastes, das heißt die Verstärkung oder Verminderung des Kontrastes hat bei diesen einen negativen Einfluss auf die Wolfserkennungssicherheit.

Die Farboriginalbilder hatten dabei einen p-Wert von ca. 1,02E-4 und die Schwarz-Weiß-Bilder einen p-Wert von ca. 1,018. Nur bei den abgedunkelten Bildern, welche Dämmerungsverhältnisse simulieren sollen, ist ein positiver Effekt gemessen worden, wenn der Kontrast in dem Bild leicht verringert wird. Bei diesen Aufnahmen lag der höchste Mittelwert der Wolfserkennung bei 70% des Originalkontrastes, der p-Wert bei ca. 1,57E-5

Als drittes wurde untersucht ob die unterschiedlichen Wolfsbilder einen großen Einfluss auf die Wolfserkennungsergebnisse haben. Dazu wurde zu jedem Wolfsbild das arithmetische Mittel aus den 21 Bearbeitungen eines Bildes gebildet und dazu die ANOVA erstellt. Mit einem p-Wert von ca. 8,74E-66 ist sicher, dass die unterschiedlichen Wolfsbilder einen sehr hohen Einfluss auf die Ergebnisse aus der Bilderkennung haben.

10.2.3 Handlungsempfehlung

Die durchgeführten Tests haben gezeigt, dass das Verändern von aufgenommenen Bildern insgesamt kaum einen signifikanten Einfluss auf die Wolfserkennung hat. Ist ein Effekt auf die Wolfserkennung zu verzeichnen, so ist die Standardeinstellung zumeist am besten geeignet. Die einzige Bildbearbeitung, die einen positiven Effekt auf die Erkennung von Wölfen hatte, war das Kontrastreduzieren bei den verdunkelten, also Dämmerungssimulierenden, Aufnahmen. Hier könnte in weiteren Tests ermittelt werden, ab welchem Schwellenwert bei der Verdunklung im Bild das Kontrastverringern einen positiven Effekt auf die Wolferkennungssicherheit hat. Aufgenommene Bilder, die dunkler als dieser Schwellenwert sind, können dann bearbeitet werden bevor sie an die Bildklassifizierung des Clouddienstanbieters gesendet und verarbeitet werden.

10.3 Fehlerbetrachtung/ Kritische Einschätzung der Tests

In den vorherigen Kapiteln wurden die Daten erfasst, abgespeichert und ausgewertet. In diesem wird nun der Testablauf kritisch betrachtet und die Aussagekraft der einzelnen Elemente überprüft. Währenddessen wird vor allem auf die Vereinfachungen und Annahmen geachtet, die im Rahmen des Testes getroffen wurden und ob die Tests unter diesen Bedingungen weiterhin aussagekräftig sind.

Die bei der Testdatenerstellung getroffenen Annahmen, dass eine leichte Verdunklung eine Dämmerungssituation und eine Entfärbung eine Nachtsituation (Schwarz-Weiß-Nachtkamera) simulieren kann, ist zunächst in Ordnung. Hier ist kritisch zu betrachten, dass auch Nachtaufnahmen unter den Originalbildern waren, für welche eine Verdunklung und Entfärbung keinen Einfluss haben. Die als Nachtaufnahme bearbeiteten Tagaufnahmen sind dem gegenüber nicht aussagekräftige Nachtbilder. Sie sind zwar entfärbt, aber zum Beispiel Hintergründe wären auf echten Nachtaufnahmen aufgrund ihrer zu hohen Entfernung von einer Nachtsichtlichtquelle nicht ausreichend beleuchtet. Ein weiterer Kritikpunkt sind die verschiedenen Auflösungen/Bildgrößen der Bilder. Durch diese Qualitätsunterschiede können Abweichungen in der Wolfserkennung entstehen, die in der Testdatenauswertung nicht berücksichtigt wurden.

Die Datenanalyse kann nicht kritisch betrachtet werden, da diese extern vom Clouddienstanbieter durchgeführt wurde. Die Bilder wurden MSA-CV zur Verfügung gestellt und dort ausgewertet. Das Einordnen der Ergebnisdaten und die Übertragung in eine Excel Tabelle wurde durch ein Programm ausgeführt. Die entstandene Ergebnistabelle wurde stichprobenartig überprüft. Bei der darauffolgenden Datenaufbereitung, wurde das Maximum aus den Bildklassifizierungsdaten mittels eines weiteren Programmes herausgefiltert. Dass dieses Vorgehen, das Reduzieren auf das Maximum, so annehmbar ist, wurde in Kapitel 10.2.1 erläutert und als Annahme festgelegt.

Diese Annahme könnte auch im Rahmen weiterer Tests so übernommen werden. Kritisch zu betrachten ist, ob diese Annahme in Verbindung mit einer anderen Bildklassifizierung oder einer abweichenden Version von MSA-CV bestehen bleibt. Unterschiedliche Möglichkeiten der Klassifizierung eines Bildes, könnten zu Erkennungslabeln führen, die bei der Wolfsabschreckung nicht betrachtet werden müssen (z.B. Wolfsfledermaus, englisch „wolf bat“).

Die Auswertung der Daten mit Hilfe der ANOVA entspricht deren Ziel, sodass hier die richtige Methode gewählt wurde. Ein Kritikpunkt ist, dass Daten nicht in jedem Fall per Mittelwertbildung (arithmetisches Mittel) kumuliert werden können, bevor sie in eine solche Betrachtung einfließen. Zwei unterschiedliche Wolfbilder können nicht zwangsweise dazu verwendet werden die Wolfserkennungswerte über die Kontraststufen zu mitteln. Dieser Fall ist jedoch bei der Analyse des Einflusses der Farbmethoden aufgetreten. Aufgrund der Tatsache, dass die Wolfsmotive, also die verschiedenen Wolfsbilder einen signifikanten Einfluss auf die Wolfserkennungswerte hatten (p-Wert bei ca. 8,74E-66) beeinflusst dieser die Betrachtung der Farbmethode.

Dieser Abschnitt des Tests war dementsprechend nicht ausreichend wissenschaftlich konzipiert. Unter folgender Annahme sind die Ergebnisse wissenschaftlich verwendbar: Die großen Unterschiede in den Aufnahmen spiegeln ein hohes Maß an Varianz wieder, welches so auch in der freien Natur vorhanden wäre. Dazu gehören Lichtbedingungen, Hintergrundmotiv, Hindernisse im Bild, Geschwindigkeit des Wolfes beim Durchqueren des Kamerablickfeldes und Witterungsverhältnisse. Um diese Faktoren in einer Testmatrix abzubilden sind sie für den Rahmen dieses Projektes zu umfangreich. Die reduzierte Betrachtungsweise ist, wie sie auch hier angesetzt wurde, die verschiedenen Faktoren als Wiederholungsfehler zu definieren. Im Kontext der Wolfsabschreckung bedeutet dies, dass die Unterschiede so betrachtet werden, als wären sie durch einen Messfehler entstanden, der auftritt, wenn man denselben Sachverhalt mehrmals misst. Dieser Messfehler bildet die Realität mit den vielen verschiedenen standort- und zeitabhängigen Faktoren ab. Mit einer größeren Auswahl an Originalbildern, wäre es möglich diesen Messfehler zu relativieren. Die untersuchte Menge von 24 Bildern ist dafür nicht ausreichend.

Zusammenfassend ist festzustellen, dass der Modultest der Bilderkennung aussagekräftige Ergebnisse in Bezug auf die Kontrastabhängigkeit der Wolfserkennung liefert hat. Der Sachverhalt, ob Tag-, Dämmerungs- oder Nachtaufnahmen signifikante Auswirkungen auf die Wolfserkennung haben, konnte allerdings nicht beantwortet werden. Für weitere detaillierte Resultate sollten der Test noch einmal in erweitertem Umfang wiederholt werden. Um die Testmethodik zu erweitern, kann eine selbsttrainierte künstliche Intelligenz eingesetzt werden. Des Weiteren könnten Stationen mit Hilfe aussagekräftiger Ergebnisse kalibriert werden, um standortabhängige Faktoren wie Lichtverhältnisse oder Hindernisse, bei der Wolfserkennung ausschließen zu können.

11 Systemtest

Nach dem vorangegangenen Modultest, soll jetzt das System als Ganzes getestet werden. Dabei soll zum einen die Qualität der Bilder im Zusammenhang mit den entsprechenden Analyseergebnissen untersucht werden, sowie die Abhängigkeit des Gesamtsystems von der gegebenen Bandbreite vor Ort. Aus entsprechenden Zeitgründen, sowie dem fehlenden Diebstahlschutz ist ein Test an einer frei zugänglichen Schafsweide nicht möglich. Um den Test aber dennoch so realitätsnah wie möglich durchführen zu können, wird das System mit, in Gefangenschaft lebenden, Wölfen untersucht. Dafür standen die Wölfe des Tierparks Thale zur Verfügung.

11.1 Testumgebung

Der Test wurde, in Absprache mit der Tierparkleitung, am Zaun des Wolfsgeheges des Tierparks in Thale durchgeführt. Aufgenommen wurden die Testdaten, des zweiten Systemtests, in ca. 1,50 m Höhe durch den Zaun des Geheges, wie in Abbildung 27 zu sehen ist.

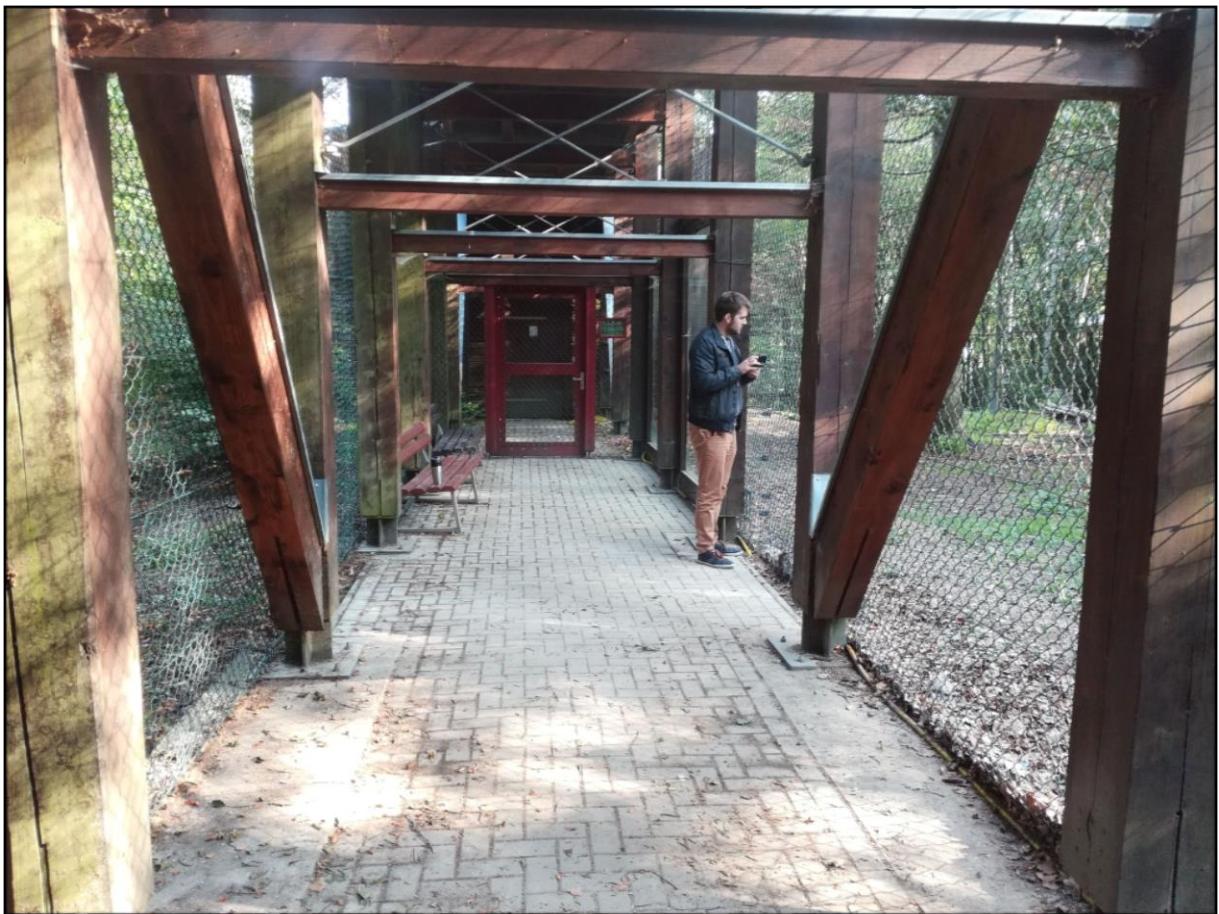


Abbildung 27: Position des Prototypen, gehalten von einem der Projektgruppenmitglieder - mit Handykamera aufgenommen

Um die benötigte Internetverbindung zu gewährleisten, befand sich ein weiteres Teammitglied auf einer erhöhten Plattform, wie sie in Abbildung 28 zu sehen ist, welche sonst als Aussichtsplattform für Besucher dient. Der Prototyp befand sich im Test physisch unter der Plattform.



Abbildung 28: Aussichtsplattform des Wolfgeheges - mit Handykamera aufgenommen

11.2 Erster Systemtest

Der erste Systemtest fand am 19.06.2019 am späten Nachmittag statt.

Die Testdaten wurden von der, im Abbildung 28 gezeigten, Holzplattform aufgenommen. Die Wölfe lagen entsprechend der heißen Temperaturen gut sichtbar im Schatten der Bäume und konnten problemlos fotografiert werden, wie in Abbildung 29 zu sehen ist.



Abbildung 29: Wölfe im ersten Systemtest - mit Handykamera aufgenommen

Obwohl die Bedingungen entsprechend gut waren, konnte der Prototyp die Wölfe nicht erkennen. Die anschließende Fehlerauswertung ergab folgende, potenzielle Fehlerquellen:

- Schlechte Qualität der Mobilfunkabdeckung vor Ort
- Fehler bei der Netzwerkkonfiguration des Prototyps
- Ablauen des Microsoft Azure Accounts

Für die Fehlerbehebung vor Ort fehlte, aufgrund der Öffnungszeiten des Tierparks, anschließend aber die Zeit, sodass der erste Systemtest negativ, aber mit potenziell lösbareren Problemen, beendet wurde.

11.3 Zweiter Systemtest

In Vorbereitung auf den zweiten Feldtest, welcher am 18.07.2019 stattfand, wurden die bereits bekannten Fehler versucht zu beheben, wie in Tabelle 6 zu sehen ist.

Tabelle 6: Fehlerlösungsansätze

Fehler	Lösung
Schlechte Qualität der Mobilfunkabdeckung vor Ort	Ein alternatives Skript für den Prototypen erstellen, welches die Auflösung der Bilder und damit die Dateigröße reduziert, dementsprechend aber schlechtere Bildqualität liefert.
Fehler bei der Netzwerkkonfiguration des Prototyps	Netzwerkeinstellungen des Prototypen so einstellen, dass dieser automatisch die benötigten Konfigurationen durchführt, wie z.B. eine dynamische IPv6 Adresse oder dem DNS Namen.
Ablauen des Microsoft Azure Accounts	Anlegen eines neuen Microsoft Azure Accounts.

Bei dem zweiten Systemtest befand sich der Prototyp nicht auf der Plattform, sondern wie bereits beschrieben unter dieser. Des Weiteren erhielt das Projektteam vom Tierpark Thale weitere Unterstützung. Ein Tierpfleger begleitete den Systemtest und fütterte die Wölfe währende der Aufnahmen, sodass diese näher an den Prototypen gelockt werden konnten, wie in Abbildung 30 zu sehen ist.



Abbildung 30: Wölfe im zweiten Systemtest

Im Gegensatz zum ersten Systemtest konnten diesmal positive Ergebnisse erzielt werden, dementsprechend kann der Fehler, welcher den ersten Systemtest hat scheitern lassen, nun weiter eingegrenzt werden: Die Netzwerkkonfiguration des Prototypen war falsch eingestellt, sodass dieser keine Internetverbindung aufbauen konnte.

12 Diskussion der Ergebnisse

Im Rahmen der Veranstaltungen ist es gelungen, einen funktionsfähigen Prototypen zu bauen. Dieser weist allerdings noch Mängel auf, die in dem bestehenden Kontext nicht mehr gelöst werden können. Die Notwendigkeit einer stabilen und wenn möglich auch qualitativ guten Internetverbindung ist die größte Schwäche des Prototyps. In ländlichen Regionen ist eine solche womöglich nicht immer gewährleistet, sodass die Wolfserkennung nicht funktioniert. Auch bei der Hardware treten noch Mängel auf. Die genutzte Kamera bietet für einen ersten Prototyp eine ausreichende Qualität, sollte aber bei der Weiterentwicklung durch ein leistungsfähigeres Modell ausgetauscht werden, welches in der Lage ist, Bilder mit besserer Qualität aufzunehmen, sodass die Wolfserkennung zusätzlich verbessert wird.

Zusätzlich zu den eben beschriebenen Hardware- und Softwaremängeln, treten auch bei dem Testaufbau Probleme auf. So fand ein Test der Wolfserkennung mit in Gefangenschaft lebenden Wölfen statt, welche regelmäßig gefüttert werden und entsprechend an den Kontakt mit Menschen, auch mit größeren und lauter Menschenmengen, schon gewöhnt sind und dementsprechend ein Verhalten zeigen, welches für freilebende Wölfe unnatürlich ist. Des Weiteren wurde auf Wunsch der Tierparkleitung Thale auf ein Auslösen der Abwehrmechanismen verzichtet, sodass nicht getestet werden konnte, ob die Wölfe überhaupt auf diese mit Flucht reagieren. Auch ist nicht getestet, ob die Abwehrmechanismen des Prototyps genug variiert werden, um einen effektiven und langfristigen Schutz vor Wölfen zu leisten.

13 Fazit

Im Kontext dieses Projektes wurde die Thematik Rückkehr des Wolfes nach Deutschland genauer betrachtet. Der wachsende Konflikt innerhalb dieser Thematik zwischen dem Umweltschutz auf der einen Seite und der wirtschaftlichen Situation der Viehbesitzer auf der anderen Seite wurde dabei genauer untersucht, um einen möglichen Lösungsansatz zu finden und nachzuverfolgen. Bei diesem Ansatz handelte es sich um die Abschreckung der Wölfe, um auf der einen Seite die Tiere der Viehbesitzer zu schützen, auf der anderen Seite aber keine existenzielle Bedrohung der Wölfe in Aussicht zu stellen. Die Wolfsabschreckung hatte damit das Ziel den Konflikt nachhaltig zu lösen. Der Begriff Nachhaltigkeit ist dabei unterteilt in ökonomisch nachhaltig für die Viehbesitzer, ökologisch nachhaltig für die Spezies Wolf und deren positive Auswirkungen auf die Biodiversität in seinem Revier und sozial nachhaltig im Bezug darauf, Bedenken und daraus resultierende Debatten und Konfliktpotenziale in Politik und Gesellschaft beizulegen. Um diesen neuen Ansatz der Wolfsabschreckung weiter nachzuverfolgen, wurde der bisherige Stand des Herdenschutzes und des Wolfsverhaltens genauer untersucht, wobei ein Kompromiss zwischen Effektivität der Abschreckung und Tierschutz gefunden werden sollte. Das Ergebnis dieser Nachforschungen war die Festlegung auf eine audiovisuelle Abschreckung welche orts- und nicht wolfgebunden sein sollte. Im nächsten Schritt wurden die entsprechenden Anforderungen aufgestellt und zu den bestehenden architektonischen Anforderungen hinzugefügt. Anschließend begann die konzeptionelle Phase, in der die Systemarchitektur entworfen wurde und Entscheidungen zur zu verwendenden Technologien getroffen wurde. Dazu gehörten der Clouddienstanbieter, Entwicklungsumgebungen, Programmiersprachen, Hardware und Hilfssoftware.

Nach Beginn der Implementierungsphase wurden einige dieser Entscheidungen wieder verworfen und noch weitere Festlegungen getroffen von Aspekten die im Vorhinein falsch entschieden oder nicht bedacht worden waren. Eine dieser Entscheidungen ist dabei die Wahl der zu nutzende Programmiersprache. Ursprünglich wurde sich für C# als

Entwicklungsgerüst entschieden. Nach nicht lösbarer Kompatibilitätsproblemen musste diese Wahl allerdings revidiert werden, sodass für die Implementierung der nötigen Softwaremodule die Programmiersprache Python ausgewählt wurde, da sie die native Sprache des Raspberry Pis ist und für alle benötigten Hardwaremodule kompatible Bibliotheken bietet. Ab Mitte der Implementierungsphase wurde zudem der Modultest begonnen und nach ersten Versuchen erweitert.

Die entstandenen Daten wurden ausgewertet und ein Fazit zum Bilderkennung mit MSA-CV erstellt. Nach erfolgreicher Gesamtimplementierung wurden Systemtests begonnen, wobei auch hier nach einem ersten Durchlauf Verbesserungen am System durchgeführt wurden, welche die Ergebnisse im zweiten Versuchsdurchlauf positiv beeinflusst haben.

Nach Ablauf der Projektlaufzeit ist damit ein funktionsfähiger Prototyp unter Berücksichtigung der zu Beginn aufgestellten Anforderungen konzipiert, realisiert und getestet. Die Tests haben dabei ergeben, dass der Prototyp grundlegend funktioniert, für einen professionellen Einsatz allerdings noch nicht geeignet ist. Die Systemtests waren nicht geeignet um die Marktfähigkeit des Produktes im realen Einsatzgebiet zu überprüfen und weitere Randbedingungen die bei längerem Einsatz zu beachten sind wurden im Rahmen des Projektes vernachlässigt. Als grundlegende Erkundung dieses Gebiets ist das Projekt dennoch erfolgreich gewesen. Es konnte gezeigt werden, dass es möglich ist mit Hilfe eines Minicomputers und einer einfachen Kamera Wölfe über einen Bildklassifizierungsalgorithmus erkennen zu lassen. Ob und wie diese Erkenntnis in Zukunft genutzt werden wird ist vorerst nicht ersichtlich, weitere Möglichkeiten was und wie verbessert werden kann wird im Ausblick dargestellt.

14 Ausblick

Nach erfolgreicher Konzeption und Realisierung eines Prototyps kann diese Grundlagenerkundung genutzt werden um ein potenziell marktreifes Produkt weiter zu entwickeln. Dabei sind vordergründig die Hardwarekomponenten und der Bildklassifizierungsalgorithmus als Punkte mit dem größten Ausbaupotenzial zu nennen. Im Bereich der Hardware ist eine hochauflösende Kamera mit großem Aufnahmewinkel, einer feste oder unabhängige Stromversorgung, sowie Witterungs- und Diebstahlschutz zu nennen. Im Bereich der Bildklassifizierung könnten bessere Ergebnisse mit einer lokalen, einer drittanbieterunabhängigen und/ oder einer selbst auf einen bestimmten Bereich trainierten künstlichen Intelligenz erzielt werden.

Werden diese Erweiterungen berücksichtigt oder ähnliche Verbesserungen eingeführt, die die Funktionalität des Gesamtsystems stabilisieren oder erweitern, kann ein solches Produkt die anfangs gesetzte Motivation als neues Projektziel erreichen und so einen Beitrag zur nachhaltigen Entwicklung der Pflanzen- und Tierwelt leisten. Zudem skaliert ein solches Projekt auch global, denn jedes System für sich ist unabhängig. Kumulierte man die Daten einzelner Systeme ließe sich zudem ein Beitrag zum Wolfsmonitoring und -management leiten.

Darüber hinaus ist die verwendete Technologie nicht nur auf Wölfe anwendbar. So wie Schafe vor Wölfen geschützt werden können, könnten auch Hühner für Füchse oder Mardern geschützt werden. Genauso kann mit dieser Technologie nicht nur die Populationsentwicklung von Wölfen überwacht werden, sondern auch die anderer gefährdeter oder vom Aussterben bedrohter Tierarten. Damit sind tiererkennende Überwachungs- und Abschreckungssysteme umfangreich anwendbar und bergen ein großes Potenzial in Bezug auf wirtschaftliche als auch nachhaltige Projekte.

15 Quellen

15.1 Literaturquellen

- [1] Deutschland. Sachsen / Landesamt für Umwelt, Landwirtschaft und Geologie [Herausgeber/in], *Erprobung und Bewertung von Schutzmaßnahmen für Nutztiere vor dem Wolf, insbesondere der Einsatz von Herdenschutzhunden und Elektronenzäunen*. Dresden: Sachsen / Landesamt für Umwelt, Landwirtschaft und Geologie, 2014 [Online]. Verfügbar: <https://www.publikationen.sachsen.de/bdb/artikel/22053/documents/30077>. [Zugriff: 14.04.2019].
- [2] Deutschland. Dokumentations- und Beratungsstelle des Bundes zum Thema Wolf [Herausgeber/in], *Wolfsverursachte Schäden, Präventions- und Ausgleichszahlungen in Deutschland 2015: Zusammenstellung der wolfsverursachten Schäden, Präventions- und Ausgleichszahlungen in Deutschland 2015 nach den Angaben der Bundesländer*. Görlitz: Dokumentations- und Beratungsstelle des Bundes zum Thema Wolf; 2016. [Online]. Verfügbar: https://dbb-wolf.de/mehr/literatur-download/berichte-zu-praevention-und-nutzterschaeden?file=files/publisher/downloads/Sch%C3%A4den_Pr%C3%A4vention_Kompensation_DE2015_Endversion.pdf. [Zugriff: 14.04.2019].
- [3] Deutschland. Dokumentations- und Beratungsstelle des Bundes zum Thema Wolf [Herausgeber/in], *Wolfsverursachte Schäden, Präventions- und Ausgleichszahlungen in Deutschland 2016*. Görlitz: Dokumentations- und Beratungsstelle des Bundes zum Thema Wolf; 2017. [Online]. Verfügbar: https://www.dbb-wolf.de/mehr/literatur-download/berichte-zu-praevention-und-nutzterschaeden?file=files/publisher/Management/Herdenschutz/Herdenschutzbericht%202016/Sch%C3%A4den_Pr%C3%A4vention_Kompensation_DE2016_final.pdf. [Zugriff: 14.04.2019].
- [4] Deutschland. Dokumentations- und Beratungsstelle des Bundes zum Thema Wolf [Herausgeber/in], *Wolfsverursachte Schäden, Präventions- und Ausgleichszahlungen in Deutschland 2017*. Görlitz: Dokumentations- und Beratungsstelle des Bundes zum Thema Wolf; 2019. [Online]. Verfügbar: https://www.dbb-wolf.de/mehr/literatur-download/berichte-zu-praevention-und-nutzterschaeden?file=files/publisher/Management/Herdenschutz/Herdenschutzbericht%202017/Sch%C3%A4den_Pr%C3%A4vention_Kompensation_DE2017_final.pdf. [Zugriff: 14.04.2019].
- [5] Deutschland. Gregor Louisoder Umweltstiftung [Herausgeber/in], Materialsammlung zum Herdenschutz für Schafhalter und Naturschützer in Bayern ; 2019. [Online]. Verfügbar: https://www.umweltstiftung.com/fileadmin/user_upload/Leitfaden_Herdenschutz_fina1_3.2..pdf [Zugriff: 13.05.2019].
- [6] Radinger, Elli H. (2017): Die Weisheit Der Wölfe, Wie sie denken, planen, für einander sorgen –Erstaunliches über das Tier, das dem Menschen am ähnlichsten ist, 6.Auflage, Ludwig Verlag ISBN: 978-3-453-28093-9
- [7] W. Timischl. *Angewandte Statistik*. Vienna: Springer Vienna, 2013, pp. 362-364
- [8] <https://dbb-wolf.de/Wolfsvorkommen/territorien/karte-der-territorien> [Zugriff: 17.08.2019]
- [9] <https://www.yellowstonepark.com/things-to-do/wolf-reintroduction-changes-ecosystem> [Zugriff: 17.08.2019]
- [10] <https://www.bfn.de/themen/gebietsschutz-grossschutzgebiete/nationalparke.html> [Zugriff: 17.08.2019]
- [11] https://dbb-wolf.de/Wolf_Steckbrief/portrait [Zugriff: 17.08.2019]

- [12] <https://www.raspberrypi.org/downloads/raspbian/> [Zugriff: 18.08.2019]
- [13] https://github.com/NeumannDirk/Wolfsabschreckung/tree/master/Wolfsabschreckung-Python-Skripte/ws2801_example [Zugriff: 29.08.2019]
- [14] <https://www.thingiverse.com/thing:1368797> [Zugriff: 18.08.2019]
- [15] <https://github.com/NeumannDirk/Wolfsabschreckung/tree/master/3D-Druck-Geh%C3%A4use> [Zugriff: 29.08.2019]
- [16] <https://azure.microsoft.com/de-de/> [Zugriff: 26.08.2019]
- [17] <https://azure.microsoft.com/de-de/services/cognitive-services/directory/vision/> [Zugriff: 26.08.2019]
- [18] <https://www.pygame.org/news> [Zugriff: 26.08.2019]
- [19] <https://docs.python.org/3/library/random.html> [Zugriff: 26.08.2019]
- [20] <https://github.com/NeumannDirk/Wolfsabschreckung/tree/master/Wolfsabschreckung-Python-Skripte> [Zugriff: 29.08.2019]
- [21] https://github.com/adafruit/Adafruit_Python_WS2801/ [Zugriff: 26.08.2019]
- [22] <https://github.com/NeumannDirk/Wolfsabschreckung/tree/master/Wolfsabschreckung-Python-Skripte> [Zugriff: 29.08.2019]
- [23] https://www.bmu.de/fileadmin/Daten_BMU/Download_PDF/Glaeserne_Gesetze/19_Lp/bnatschg_2_aenderung/Entwurf/bnatschg_2_aenderung_gesetzentwurf_bf.pdf [Zugriff: 26.08.2019]
- [24] http://www.gesetze-im-internet.de/bnatschg_2009/ [Zugriff: 26.08.2019]
- [25] https://ljk-sachsen.de/wp-content/uploads/2017/05/SaechsJagdVO_27082012.pdf [Zugriff: 26.08.2019]

15.2 Bildquellen

[WB1]

<https://images.ecosia.org/1SgdlOP4r4trCziiTyogcgvFE=/0x390/smart/http%3A%2F%2Ffootage.framepool.com%2Fshotimg%2Fqf%2F616256141-wolf-ungarn-stamm-pflanze-raubtier.jpg> [29.07.2019]

[WB2]

https://media.wendland-net.de/files/l/m/lmaf66yhdv8/images_zVHoe8PYqEERbgy7_L.jpg

[29.07.2019]

[WB3]

<https://images.ecosia.org/DBJqPHrVWHGI0q6BHfBe44BKbeQ=/0x390/smart/https%3A%2F%2Fimg.fotocommunity.com%2Fder-wolf-im-hund-f99125b8-75d3-4c8b-b963-81cdeb903451.jpg%3Fheight%3D1080> [29.07.2019]

[WB4]

<https://www.zdf.de/assets/planet-e-modul-lebensraum-wald-bild-wolf-100~1280x720?cb=1536065608466> [29.07.2019]

[WB5]

https://images.ecosia.org/slepg2uXWuqMJzlLOJcNdYZgBd0=/0x390/smart/http%3A%2F%2Fstatic.geo.de%2Fbilder%2F18%2F35%2F63061%2Farticle_image_big%2Fwolf-im-wald-hinter-einem-baumstamm-c-10966689.jpg [29.07.2019]

[WB6]

<https://www.ksta.de/image/3447696/max/1920/1080/3f613b6f5f92ea2f6c712e3afae39bf8/UI/img-0850-jpg.jpg> [29.07.2019]

[WB7]

https://www.augsburger-allgemeine.de/img/wirtschaft/crop54383091/806956850-cv16_9-w940/Wer-im-Wald-auf-einen-Wolf-trifft-sollte-ihm-Raum-lassen.jpg [29.07.2019]

[WB8]

<https://tse2.mm.bing.net/th?id=OIP.Wsm8KflgaFtE8EI-dLVeUAHaE8&pid=Api> [29.07.2019]

[WB9]

https://gnadenhofsondershausen.files.wordpress.com/2011/03/dsc_1023.jpg [29.07.2019]

[WB10]

https://www.bayerwald-wanderer.de/wp-content/uploads/2016/04/IMG_6838.jpg [29.07.2019]

[WB11]

https://www.jagdverband.de/sites/default/files/Rolfes%20Wolf%20WR%2011%201237_0.JPG [29.07.2019]

[WB12]

https://www.running-magazin.com/wp-content/uploads/0125_GettyImages-175228841.jpg [29.07.2019]

[WB13]

https://www.ndr.de/nachrichten/niedersachsen/lueneburg_heide_unterelbe/wolf2534_v-contentx1.jpg [29.07.2019]

[WB14]

<https://images5.alphacoders.com/346/thumb-1920-346942.jpg> [29.07.2019]

[WB15]

https://tse3.mm.bing.net/th?id=OIP.vUuQgjbHNaje_PJy1jtZ_wHaFR&pid=Api [29.07.2019]

[WB16]

https://c1.staticflickr.com/9/8011/7544511368_be0b1e8030.jpg [29.07.2019]

[WB17]

<https://images.ecosia.org/AZ0c9yDO3dfwHwi5rvWymDec5DE=/0x390/smart/https%3A%2F%2Fimage.jimcdn.com%2Fapp%2Fcms%2Fimage%2Ftransf%2Fdimension%3D698x10000%3Aformat%3Djpg%2Fpath%2Fs6b475586b582e4aa%2Fimage%2Fie218567b6e030ff2%2Fversion%2F1485354657%2Fimage.jpg> [29.07.2019]

[WB18]

https://www.ndr.de/nachrichten/schleswig-holstein/screenshot453380_v-contentx1.jpg [29.07.2019]

[WB19]

https://www.rnz.de/cms_media/module_img/234/117485_1_org_die-gesamtzahl-der-woelfe-in-deutschland-wird-von-experten-auf-etwa-350-geschaetzt-foto-wolfsbetreuer-de.jpg [29.07.2019]

[WB20]

https://www.jagdverband.de/sites/default/files/styles/large/public/IMG_0045.JPG?itok=b-Qf6jZU?t=29.07.2019

[WB21]

https://www.lvz.de/var/storage/images/lvz/region/oschatz/erstmals-tappt-wolf-auf-oschatzer-flur-in-fotofalle/712843073-1-ger-DE/Erstmals-tappt-Wolf-auf-Oschatzer-Flur-in-Fotofalle_big_teaser_article.jpg [29.07.2019]

[WB22]

https://naturdetektive.bfn.de/fileadmin/_processed_/5/0/csm_Wolf_bayrischer_Wald_gemeinfrei_70c1745cb8.jpg [29.07.2019]

[WB23]

<https://2.bp.blogspot.com/YMqZ2S16lLU/TsUetY77J7I/AAAAAAAACIo/I6jKR8R5aVI/s1600/Bay+Wald+252826%2529.jpg> [29.07.2019]

[WB24]

<https://c8.alamy.com/compde/hggxjx/einsamer-grauer-wolf-grey-wolf-canis-lupus-klaffen-im-wald-im-sommer-hggxjx.jpg> [29.07.2019]

16 Anhang

16.1 Glossar

Wort	Bedeutung
Adafruit	Open-Source-Hardware Unternehmen
Adaptiv	anpassungsfähig
Agantty	Browserwerkzeug zum Erstellen und Verwalten von Gantt Diagrammen https://app.agantty.com/
Array	Datenstruktur in der Informatik, für viele Daten des gleichen Datentyps
Azure	Eine von der Microsoft Corporation gehostete Clouddienstleistungsplattform.
Bildklassifizierung	Analyse eines Bildes, bei der im Bild enthaltende Teilelemente einer Klasse von Objekten zugeordnet werden. Diese Zuordnung basiert zumeist auf Daten mit der das Analysesystem vorher trainiert wurde.
Bus-System	Ist ein System zur Datenübertragung zwischen Teilnehmern über einen gemeinsamen Übertragungsweg
C#	Programmiersprache
Draw.io	Browserwerkzeug zum Erstellen von Grafiken zu Modellierungszwecken https://www.draw.io/
JavaScript Object Notation	ist ein kompaktes Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen
IDE	Integrierte Entwicklungsumgebung, stellen hilfreiche Werkzeuge bereit, die dem Softwareentwickler häufig wiederkehrende Aufgaben abnehmen
Invasive Arten	Gebietsfremde Arten, welche negative Auswirkungen auf andere Arten oder Biotope haben.
Kaskadeneffekt	eine Verkettung von Ereignissen oder Prozessen, wobei alle Ereignisse auf die vorhergehenden aufbauen
Künstliche Intelligenz	Teilgebiet der Informatik, beschäftigt sich u.a. mit dem maschinellen Lernen
Maschinelles Lernen	Ein künstliches System lernt aus Beispielen und kann diese nach Beendigung der Lernphase verallgemeinern.
Master-Slave-Prinzip	Ist eine hierarchische Verwaltung des Zugriffs auf eine gemeinsame Ressource in einem gemeinsamen Datenkanals
Polylactide	sind synthetische Polymere, die zu den Polyester gehören. Aus ihnen wird Kunststoff gefertigt, der aus regenerativen Quellen gewonnen wird (wie beispielsweise Maisstärke).
Python	Programmiersprache
Semaphore	Koordiniert asynchrone Abläufe in Threads, funktioniert wie ein Regler, der Freigibt oder Reserviert
Serial Peripheral Interface	Ein Bus-System das digitale Schaltungen nach dem Master-Slave-Prinzip miteinander Verbindet
Thread	Abarbeitungsstrang innerhalb eines Prozesses

16.2 Verwendete Programme

- [PG1] XnViewMP: <https://www.xnview.com/de/xnviewmp/>
[Zugriff: 28.07.2019]
- [PG2] PuTTY <https://www.putty.org/>
[Zugriff: 30.08.2019]
- [PG3] Etcher <https://www.balena.io/etcher/>
[Zugriff: 30.08.2019]
- [PG4] Testprogramm Modultest <https://github.com/NeumannDirk/Wolfsabschreckung>
[Zugriff: 30.08.2019]

16.3 Projektdateien

Link: <https://github.com/NeumannDirk/Wolfsabschreckung>