1. Idee und Ziel

Ich war schon immer ein Tierfreund und als ich wiederholt Artikel im Stader Tageblatt gelesen habe, dass Wölfe, die Weidetiere wie Schafe oder Rinder reißen, zum Abschuss freigegeben werden, entstand die Idee zu meinem Projekt. Das Programmieren einer KI, die Bilder anhand der Körperumrisse eines Wolfes von einem anderen Tier unterscheiden kann, ist das erste Zwischenziel. Durch das Verbinden mit einer Wildkamera wird anschließend ein autonomes Meldesystem entstehen, welches Wölfe registrieren kann und an einen Verantwortlichen melden wird. Somit können auch Wölfe davor geschützt werden, dass sie, in einem für sie natürlichen Lebensraum, abgeschossen werden.

2. Zwischenziele

Zwischenziel 1 zum Regionalwettbewerb 2024

- Ein neuronales Netz programmieren, welches dafür geeignet ist Wölfe zu erkennen
- Trainingsbilder aus dem Internet herunterladen (Wölfe – Hunde und Katzen)
- Neuronales Netz trainieren

Zwischenziel 2

- Kamerasystem bauen
- Neuronales Netz mit dem Kamerasystem verbinden

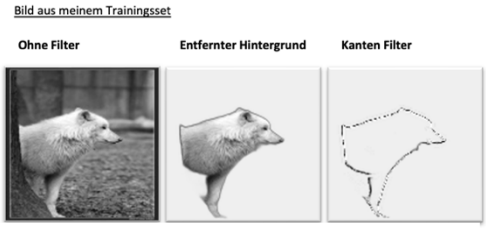
3. Vorgehensweise

3.1 Wie bin ich vorgegangen?

- Mit der Python-Library **TensorFlow** ein **neuronales Netz** erstellt
- **Trainingsbilder** heruntergeladen (Wölfe vs. Hunde u. Katzen)
- **Hintergrund** meiner Testbilder **entfernt**
- **Kanten** in den Bildern **hervorgehoben**
- Nach jedem Schritt immer ein **Diagramm** der **Trainingsgenauigkeit** in Abhängigkeit der **Menge** der **Trainingsbilder** erstellt

3.2 Welche Module habe ich verwendet?

- OpenCV (Open Computer Vision Library) von Intel und Willow Garage (2024)
- Pillow (Python Image Library) von Secret Labs AB (2024)
- rembg (remove Background) vom Nutzer danielgatis auf GitHub (2024)
- TensorFlow, vom Google Brain Team (2024)



(Abb. 2 aus der Langfassung)

5. Ergebnisse und Zielerreichung

Die **beste** von mir erzielte **Genauigkeit** von **80% - 85%** ist für erste Testdurchläufe bereits **zufriedenstellend**, aber motivieren mich noch **weiterzuarbeiten**!

5.2 Zielerreichung

Mein Ziel war es ein **neuronales Netz** mit Python zu programmieren, welches automatisch Wölfe erkennen kann. Die Tabelle 1 zeigt die Ergebnisse der Testdurchläufe. Ich habe bereits eine **große Menge** an **Trainings-** bzw. **Testbildern** heruntergeladen und ein **neuronales Netz** mit TensorFlow **programmiert** und dieses mit den Trainingsbildern **trainiert**. Weiterführend habe ich schon sehr **viele Computerexperimente** durchgeführt und konnte die **einzelnen Tests** durch Auswertungen miteinander **vergleichen** und den **Aufbau** des **neuronalen Netzes** immer weiter **verbessern**.

6. Weiterentwicklung / Ausblick

- Neuronales Netz weiter **verbessern**
- **Hardware** bauen: Kamerasystem zur Erkennung von Wölfen
- Programm mit selbst **aufgenommenen Bildern** trainieren
- eine Funktion in den Code einbauen, dass ein **Verantwortlicher benachrichtigt** wird, wenn ein Wolf erkannt wird

(Abb. 3)

(Abb. 1 aus der Langfassung)

4. Testdurchläufe

Insgesamt 11 Versuche (3.1.1 bis 3.1.11 der Langfassung). Hier sehen Sie ein Beispiel für einen Testdurchlauf:

(1) Versuchsbeschreibung Versuch 9, Modell 3

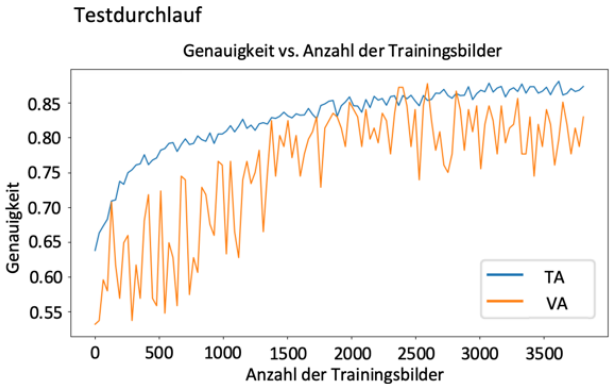
Da der dritte Versuch bis jetzt die besten Ergebnisse erzielte, die Genauigkeit jedoch noch nicht am Limit war, habe ich das dritte Modell nochmal mit doppelt so vielen Trainingsbildern trainiert.

(2) Ergebnis

Die **Genauigkeit** des Modells ist erstaunlich **gut**. Am Ende schwankt sie zwischen **75%** und **85%** (Graph: Diagramm 10 aus der Langfassung).

(3) Deutung

Da die **Genauigkeit** in den **letzten** 500-1000 **Trainingsschritten nicht** weiter **gestiegen** ist, glaube ich, dass auch **mehr Trainingsbilder** die Genauigkeit des Modells **nicht** weiter **verbessern** können.



TA: Trainings Accuracy
VA: Validation Accuracy (Mit Bildern, die das Modell nicht kennt)
(Diagramm 10 aus der Langfassung)

5.1 Ergebnisse der Testdurchläufe

Meine Kriterien zum Vergleichen der einzelnen Netze sind ebenfalls gut, da die bloße Genauigkeit nicht ausreicht, um neuronale Netze zu vergleichen, da die Menge der Trainingsdaten auch eine große Rolle spielt. Wenn man eine hohe Genauigkeit erreichen möchte, muss man auf zwei Dinge gleichzeitig achten: das Netzwerk und die Trainingsdaten

Testdurchlauf	Trainingsgenauigkeit	Testgenauigkeit
TensorFlow Test	100%	95% - 98%
Versuch Nr. 1: Model 1 mit Hintergrund	95%	75% - 85%
Versuch Nr. 2: Model 1 ohne Hintergrund	95%	60% - 85%
Versuch Nr. 3: Model 1, SW-Bilder u. verstärke Kanten	90%	50% - 60%
Versuch Nr. 4: Model 2, SW-Bilder u. verstärke Kanten	90%	60% - 70%
Versuch Nr. 5: Model 3, SW-Bilder u. verstärke Kanten	85%	75% - 85%
Versuch Nr. 6: Model 4, SW-Bilder u. verstärke Kanten	85%	60% - 75%
Versuch Nr. 8: Model 6, SW-Bilder u. verstärke Kanten	85%	60% - 80%
Versuch Nr. 9: Model 3, SW-Bilder u. verstärke Kanten	87%	80%
Versuch Nr. 10: Model 7, SW-Bilder u. verstärke Kanten	80%	57% - 75%
Versuch Nr. 11: Model 8, SW-Bilder u. verstärke Kanten	85%	60& - 73%

(Tabelle 1: Ergebnisse der Testdurchläufe der Modelle)

Literaturverzeichnis
-danielgatis (Entwickler) (12.2023 aktualisiert): rembg https://github.com/danielgatis/rembg/releases (heruntergeladen am 2.1.2024)
-Grit Klempow (Autor) (25.01.2023): Lauter Ruf nach Wolfsabschüssen Stader Tageblatt
-Google Brain Team (Entwickler) (4.9.2023 aktualisiert): TensorFlow https://de.wikipedia.org/wiki/TensorFlow (heruntergeladen am 2.1.2024)
-harishvutukuri (Entwickler) (2020): Dogs vs wolves https://www.kaggle.com/datasets/harishvutukuri/dogs-vs-wolves (heruntergeladen am 04.01.2024)
-Imagenet (Entwickler) (keine Angabe): https://images.ev/dataset/white-wolf-image-classification-dataset (heruntergeladen am 04.01.2024)
- Institut für Ökologie und Evolution, Universität Bern (keine Angabe): Abb. 1 aus der Langfassung https://infoplattform-grossraubtiere.ch/bei-ihrer-raeckkehr-achten-woelfe-auf-das-beutenangebot/
-Intel, Willow Garage (Entwickler) (18.12.2023 aktualisiert): OpenCV https://de.wikipedia.org/wiki/OpenCV (heruntergeladen am 2.1.2024)
- Reidt, Lutz: Abb. 3 https://www.deutschlandfunkkultur.de/wolf-woelfe-gefahr-weidetiere-nutztiere-bayern-schweiz-100.html
-Secret Labs AB (Entwickler) (17.10.2023 aktualisiert): Python Imaging Library https://en.wikipedia.org/wiki/Python_Imaging_Library (heruntergeladen am 2.1.2024)
-shuantheshoep (Entwickler) (2020): Microsoft Cats vs Dogs https://www.kaggle.com/datasets/shuantheshoep/microsoft-catsvsdogs-dataset (heruntergeladen am 04.01.2024)
-Zöth, Paul (Entwickler) (2024): mein GitHub-Projekt: https://github.com/Paul193247/Jugendforscht-Wolfserkennung
- Zöth, Paul (Ersteller) (2024): Abb. 2 aus der Langfassung
- Zöth, Paul (Ersteller) (2024): Tabelle 1
- Zöth, Paul (Ersteller) (2024): Tabelle 2
- Zöth, Paul (Ersteller) (2024): Diagramm 10

Danksagung
An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich bei meinem Projekt unterstützt und motiviert haben. Zuerst gebührt mein Dank Herrn Privatdozent Dr. Carmesin, der mein Projekt betreut und mir bei der Organisation des Projektes und dem Setzen der Zwischenziele geholfen hat. Ein besonderer Dank gilt Frau von Bagen, die mir mit viel Geduld, Interesse und Hilfsbereitschaft zur Seite stand. Zu guter Letzt möchte ich mich bei meiner Familie bedanken, die stets ein offenes Ohr für mich hatte.