

BACHELORTHESIS
Philipp Jovannes Grages

Bewegungserkennung in Fußgängerbereichen mit einer Wärmebildkamera unter Wahrung der Privatsphäre

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Philipp Jovannes Grages

Bewegungserkennung in Fußgängerbereichen mit
einer Wärmebildkamera unter Wahrung der
Privatsphäre

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stephan Pareigis
Zweitgutachter: Prof. Dr. Tim Tiedemann

Eingereicht am: 16. Juni 2022

Philipp Johannes Grages

Thema der Arbeit

Bewegungserkennung in Fußgängerbereichen mit einer Wärmebildkamera unter Wahrung der Privatsphäre

Stichworte

Bewegungserkennung, Wärmebildkamera, Objekterkennung, Maschinelles Lernen, Künstliche Intelligenz, RetinaNet

Kurzzusammenfassung

Im Rahmen dieser Bachelorarbeit wurde die Erkennung von sich bewegenden Objekten in Fußgängerbereichen mithilfe einer Wärmebildkamera untersucht. Es wurde erforscht, Fußgänger, Fahrradfahrer und Hunde mit einer Boxen-Objekterkennung zu erfassen. Dazu wurde ein RetinaNet benutzt welches auf dem COCO Datensatz vortrainiert ist. Das Netzwerk wurde mit den Bildern einer Wärmebildkamera nachtrainiert, um Objekte in diesen zu erkennen. Die Objekte auf den Wärmebildern konnten mit einer durchschnittlichen Präzision von 28% erkannt werden. Der Vorteil von Wärmebildern gegenüber sichtbaren Bildern ist, dass Personen nicht einfach identifiziert und so in ihrer Privatsphäre besser geschützt werden können.

Philipp Johannes Grages

Title of Thesis

Motion detection in pedestrian areas by a thermal imaging camera while maintaining privacy

Keywords

motion detection, thermal imaging, object detection, machine learning, artificial intelligence, RetinaNet

Abstract

As part of this bachelor thesis, the detection of moving objects in pedestrian areas was investigated using a thermal imaging camera. The detection of pedestrians, cyclists and dogs with a boxed object detection was explored. For this purpose, a RetinaNet pre-trained on the COCO dataset was used. The network was retrained with the images from a thermal imaging camera to detect objects in them. The objects on the thermal images could be detected with an average precision of 28%. The advantage of thermal images over visible images is that people cannot be easily identified and thus their privacy can be better protected.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	ix
1 Einleitung	1
1.1 Problemstellung	1
1.2 Eingrenzung des Themengebiets	2
1.3 Struktur der Arbeit	2
2 Grundlagen	4
2.1 Stand der Forschung	4
2.2 RetinaNet	6
2.3 Labeling	7
3 Entwurf	8
3.1 Anforderungen	9
3.2 Hardware	10
3.2.1 Gehäuse	10
3.2.2 RaspberryPi 4	10
3.2.3 DHT 22	10
3.2.4 Flir A35	11
3.2.5 Basler	11
3.2.6 Akku	11
3.2.7 Ethernet Switch	11
3.2.8 DC Step up	11
3.3 Software	12
3.3.1 Aufnahme der Bilder	12
3.3.2 Objekterkennung	13

4 Durchführung	14
4.1 Daten	14
4.1.1 Temperatureinfluss auf die Daten	14
4.1.2 Probleme	15
4.2 Training	17
4.2.1 Klassifizieren der Bilder	17
4.2.2 Trainieren des Netzwerkes	18
5 Auswertung der Netzwerk-Metriken	19
6 Ergebnisse	25
6.1 Ergebnisse des Netzwerkes	25
6.2 Privatsphäre	30
7 Schlusswort	32
Literaturverzeichnis	34
A Anhang	38
Selbstständigkeitserklärung	40

Abbildungsverzeichnis

2.1	Wärmebild von der gleichen Szenerie unter unterschiedlichen Bedingungen [14]	5
2.2	RetinaNet Architektur[23]	7
2.3	LabelImg Programm [27]	7
3.1	Kameragehäuse	8
3.2	Kontextdiagramm der Hardware zur Aufnahme der Bilder	9
4.1	Personen bei 5, 8°C Außentemperatur	15
4.2	Personen bei 12, 8°C Außentemperatur	15
4.3	Wärmebild einer Gruppe	16
4.4	Bild im sichtbaren Spektrum einer Gruppe	16
4.5	Bild eines Fahrradfahrers	16
4.6	Wärmebild von einem Hund mit kurzen Fell	17
4.7	Wärmebild von einem Hund mit langem Fell	17
5.1	mAP des Netzes über den Trainingsverlauf	20
5.2	mAP des Netzes für kleine Objekte	21
5.3	mAP des Netzes für mittelgroße Objekte	21
5.4	50% Überlappung	22
5.5	75% Überlappung	22
5.6	Lokalisierung Fehlerrate über den Trainingsverlauf. Hellblau Evaluierungsdaten, Dunkelblau Trainingsdaten	23
5.7	Klassifikation Fehlerrate über den Trainingsverlauf. Hellblau Evaluierungsdaten, Dunkelblau Trainingsdaten	24
6.1	Links erkannte Personen nach 25000 Schritten, rechts Referenzbild	25

6.2	Referenzbild zu 6.1	26
6.3	Wärmebild ohne Annotationen	26
6.4	Personen weiter entfernt; siehe Tabelle A.1	27
6.5	Fahrradfahrer	28
6.6	Gruppe von Personen und ein Hund; siehe Tabelle A.2	29
6.7	Hund mit seinem Besitzer; siehe Tabelle A.3	29
6.8	RetinaNet50 auf dem COCO Datensatz[23]	30
6.9	Wärmebild einer Gruppe von Personen	31

Tabellenverzeichnis

A.1	Tabelle zu Abbildung 6.4	38
A.2	Tabelle zu Abbildung 6.6	39
A.3	Tabelle zu Abbildung 6.7	39

1 Einleitung

In den heutigen Städten wird es immer wichtiger, die Infrastruktur zu optimieren. So ist eines der Ziele einer digitalen Stadt die Verbesserung des öffentlichen Personenverkehrs [9, S. 15 ff]. Kameras, die für diesen Zweck aufgestellt werden, erfassen jedoch personenbezogene Daten, welche zu starken Sicherheitsrisiken führen können und unter Umständen Persönlichkeitsrechte verletzen. [20]. Um die Risiken bei der Erhebung von personenbezogenen Daten so gering wie möglich zu halten, wird in dieser Arbeit anstelle einer Kamera im sichtbaren Wellenlängenbereich eine Wärmebildkamera benutzt.

1.1 Problemstellung

Wenn über sichtbares Licht gesprochen wird, ist das Licht gemeint, welches für das menschliche Auge sichtbar ist. Dieses sichtbare Spektrum hat den Wellenlängenbereich von $400 - 700nm$ [29, S. 1094]. In dieser Arbeit wird eine Basler a2A1920-51gvPro verwendet, die in diesem sichtbaren Wellenlängenbereich arbeitet. Im weiteren Verlauf wird die Basler als Kamera bezeichnet. Im Gegensatz dazu arbeitet eine Wärmebildkamera, hier eine Flir A35, in einem Wellenlängenbereich von $7,5 - 13\mu m$ [16]. Die Wärmebildkamera bildet die Temperatur als Schwarz-Weiß-Bild ab. So erscheinen wärmere Objekte heller. Bei einem Bild von einer Wärmebildkamera sieht man nicht die Farben der Kleidung. Man erkennt nur, wie viel Wärme die Kleidung der Person abstrahlt. Dieses ist von Vorteil im Hinblick auf die Privatsphäre, da es die Identifikation einer Person erschwert. Um Personen zu erkennen, muss es einen Kontrast zwischen ihnen und dem Hintergrund geben. So erscheinen Personen heller, wenn sie wärmer als die Umgebung sind und dunkler, wenn umgekehrt die Umgebung eine höhere Temperatur aufweist, als die Personen. Je näher die Temperatur der Umgebung und die der zu identifizierenden Objekte aneinander sind, desto geringer ist der Kontrast zueinander und umso schwieriger ist es, die Objekte korrekt zu identifizieren. Wärmebildkameras sind erheblich teurer als Kameras im sichtbaren Bereich. So kostet eine günstige Wärmebildkamera $77,20\text{€}$,

hat jedoch nur eine Auflösung von 24×32 Pixel [10]. Eine Kamera im sichtbaren Bereich kostet dagegen nur 27,73 € und hat dabei eine Auflösung von 640×480 Pixel [11]. Die Kamera im sichtbaren Bereich hat eine bessere Auflösung um den Faktor 400. Möchte man eine Wärmebildkamera mit einer höheren Auflösung haben, so kostet sie erheblich mehr.

1.2 Eingrenzung des Themengebiets

In dieser Arbeit soll untersucht werden, wie gut Teilnehmer in Fußgängerbereichen mit einer Wärmebildkamera erkannt werden können, welchen Einfluss unterschiedliche Temperaturen auf die Erkennbarkeit von Teilnehmern haben und ob sie eine Alternative zu den üblichen Kameras darstellen. Da es datenschutzrechtlich schwierig ist, eine Kamera fest aufzustellen, werden die Bilder von einer in der Hand getragenen Plattform aus gemacht. Um eine Kamera fest aufzustellen, müsste erst geprüft werden, ob die schutzwürdigen Interessen der betroffenen Personen überwiegen [12, BDSG §4 Abs. (1)]. Auch müssten dann entsprechende Hinweise aufgestellt werden, die die betroffenen Personen frühestmöglich informieren [12, BDSG §4 Abs. (2)].

Die Bilder werden nicht über alle Jahreszeiten aufgenommen. So wurden weder Bilder unter 0°C noch über 25°C gemacht. Zu den zu klassifizierenden Objekten gehören nur Personen, Hunde und Fahrradfahrer. Es wird nicht versucht, Autos oder andere motorisierte Fahrzeuge zu erkennen, da diese für gewöhnlich nicht in Fußgängerbereichen fahrend anzutreffen sind. Weiterhin wird nicht versucht, Rollerfahrer, Skateboardfahrer und Rollstuhlfahrer zu erkennen, obwohl diese oft in diesen Bereichen vorzufinden sind.

Ziel ist es, einen ersten Eindruck über die generelle Machbarkeit von der Erkennung mit einer Wärmebildkamera zu bekommen und herauszufinden, ob diese Bilder möglicherweise die Privatsphäre besser schützen können als die anderer Kameras.

1.3 Struktur der Arbeit

Die Arbeit wird sich zuerst damit befassen, wie der derzeitige Forschungsstand zur Objekterkennung mit einer Wärmebildkamera ist. So wird kurz erklärt, welches bereits trainierte Netz für die Objekterkennung benutzt wird. Anschließend werden im Entwurf die

Anforderungen abgesteckt: das Ziel, das versucht wird zu erreichen und der Lösungsansatz, der dazu verwendet wird. Auch wird auf die dazu benutzte Hardware und Software eingegangen. Die Durchführung des Experimentes wird als nächstes erörtert und erste Eindrücke aus den daraus resultierenden Daten. Dann wird beschrieben, wie das Training des Netzwerkes mit den Daten durchgeführt wurde. Abschließend werden die Ergebnisse ausgewertet, die Datenerfassung im Hinblick auf die Privatsphäre der betroffenen Personen bewertet und ein Fazit gezogen.

2 Grundlagen

In dieser Arbeit soll der Schwerpunkt auf der Objekterkennung liegen, die unter dem Aspekt der Privatsphäre möglichst keine Rückschlüsse auf die einzelnen Individuen zulässt. Dafür schauen wir uns den Forschungsstand für Objekterkennung mit einer Wärmebildkamera an. Für viele Anwendungen ist die Identifikation der Objekte auch bei schlechten Sichtverhältnissen und nicht der Schutz der Persönlichkeitsrechte der treibende Faktor.

2.1 Stand der Forschung

Ivašić-Kos hat nachts Wärmebildkameras zur Überwachung von geschützten Arealen benutzt. So können Wärmebildkameras auch bei schlechter oder keiner vorhandenen Beleuchtung weiterhin Personen erkennen, da sie die Wärmeabstrahlung direkt für die Bilder benutzen[21, S. 20]. Für die Erkennung der zu überwachenden Objekte wurde hierfür das YOLO-Objekterkennungs-Netzwerk benutzt, welches mit Bildern im sichtbaren Wellenlängenspektrum trainiert wurde. Dafür wird das vortrainierte Netzwerk jetzt mit den Bildern einer Wärmebildkamera nachtrainiert. Hier ist zu berücksichtigen, dass die Bilder einer Wärmebildkamera signifikant anders sind, als die einer Kamera im sichtbaren Bereich. So hat eine Wärmebildkamera nur einen Wert pro Pixel, während eine Kamera im sichtbaren Bereich drei Werte, rot, grün und blau, pro Pixel hat. Trotzdem wird davon ausgegangen, dass sich die einzelnen Schichten eines RGB-Bildes den Werten eines Wärmekamerabildes genug ähneln, um das Netz für dieses zu verwenden[21, S. 21]. Wenn das Netzwerk nicht vorher auf Wärmebildkameras trainiert wurde, aber mit Wärmebildern benutzt wird, erzielt es nur eine durchschnittliche Erkennungsrate von 23%. Dieses ist sehr viel schlechter als die erzielte durchschnittliche Erkennungsrate von 90% auf dem Standard COCO Datensatz. Nachdem das Netzwerk mit den Wärmebildern nachtrainiert wurde, erzielte es eine durchschnittliche Erkennungsrate von 97%[21, S. 23].



Abbildung 2.1: Wärmebild von der gleichen Szenerie unter unterschiedlichen Bedingungen [14]

Die meistbenutzten Datensätze zum Trainieren von Objekterkennung sind ImageNet, PASCAL VOC und COCO. Sie benutzen alle Bilder im sichtbaren Spektrum [30]. So müssen für Wärmebildkameras selbst erstellte Datensätze benutzt werden, da es noch keine vergleichbar großen Datensätze zu den vorher genannten drei gibt. Je nach Umwelteinflüssen können aufgenommene Bilder einer Wärmebildkamera sehr unterschiedlich ausfallen. So zeigt Abbildung 2.1, wie unterschiedlich Bilder von Personen in der gleichen Szenerie ausfallen können, wenn sie zu unterschiedlichen Tageszeiten und Wetterbedingungen aufgenommen werden. Mithilfe eines Zwei-Stufen-Algorithmus zum Erkennen von Personen können auch schwierig zu differenzierende Individuen auf Wärmebildern bestimmt werden. In der ersten Stufe können mit einer Kanten- und Gradienteninformation-Vorverarbeitung mögliche Personen hervorgehoben, der Hintergrund unterdrückt und ein Ausschnitt um die Person generiert werden. Die produzierten Ausschnitte enthalten Personen, Teile von Personen und andere nicht personenbezogene Objekte, wie zum Beispiel Fahrzeuge, Tiere, et cetera. In der zweiten Stufe sollen die aus der ersten Stufe generierten, Vordergrund-Objekte nach den Personen gefiltert und die Personen entsprechend im Bild markiert werden [14]. Dies geschieht mit der AdaBoost Technik. Diese verwendet mehrere unabhängige Klassifizierer, welche nur etwas besser als der Zufall sein müssen. Jeder dieser Klassifizierer arbeiten unabhängig voneinander. Am Ende werden sie zu einem Gesamtergebnis zusammengetragen [14].

Sie benutzen einen eigens erstellten Datensatz aus Wärmebildern, aufgenommen von einem festen Standort bei unterschiedlichen Wetterbedingungen. Insgesamt sind dabei 284

Bilder mit einer Auflösung von jeweils 360×240 Pixel aufgenommen worden. Der Zwei-Stufen-Ansatz konnte in einigen Bildern, ohne falsche Positive, alle Personen korrekt identifizieren. Allerdings gab es bei manchen Bildern Probleme, wo sowohl zwei Personen als eine Person und umgekehrt eine Person als zwei Personen klassifiziert wurden. Auch wurden teilweise Autos als Personen klassifiziert. Insgesamt kommt der Ansatz auf eine Genauigkeit von 95%, ohne die falschen Positive mit einzubeziehen[14].

Die aktuellen Forschungsergebnisse vermitteln, was schon mit einer Wärmebildkamera möglich ist, und wo die Objekterkennung bei Wärmebildern an ihre Grenzen stößt. Leider haben die benutzten Datensätze bisher keine große Ansammlung von Menschen beinhaltet. Auch der Aspekt der Privatsphäre bei der Objekterkennung wurde im Vergleich zu Bildern im sichtbaren Spektrum dabei noch nicht betrachtet.

2.2 RetinaNet

Zur Objekterkennung wurde das RetinaNet benutzt. Da es auf Bildern im sichtbaren Bereich trainiert ist, wird es mit einem eigens erstellten Datensatz nachtrainiert.

Das RetinaNet ist ein One-Stage-Detector welcher aus zwei Komponenten aufgebaut ist, einem Feature-Pyramid-Network(FPN) auf einem ResNet und zwei aufgabenspezifische Subnetzwerke.

Das FPN 2.2(b) augmentiert das ResNet 2.2(a), um die Ergebnisse zu verbessern, da ohne Augmentierung die Präzision des Netzwerkes wesentlich geringer ist[23]. Das ResNet bekommt als Eingabe ein Bild, welches es vorverarbeitet. Dann nimmt das FPN die Ausgabe der letzten Schicht vom ResNet. Jede FPN-Schicht bekommt zusätzlich Informationen von jeweils der gleich großen Schicht aus dem ResNet. In jeder Schicht vom FPN wird die Information vom ResNet und die Eingabe der vorherigen Schicht kombiniert und an die Subnetze weitergegeben. Die Nutzung eines FPN hat zur Folge, dass die Features der Objekte in verschiedenen Auflösungen erkannt werden[23].

Die Subnetzwerke sind aufgeteilt in ein Klassifikations- und ein "Box RegressionNetzwerk. Das Klassifikations-Netzwerk, siehe 2.2(c), klassifiziert Objekte aus den Schichten der Pyramide. Parallel dazu läuft ein Box Regression Subnetzwerk, welches die genaue Boxen-Position der Objekte bestimmt[23]. Ergebnis des Verfahrens ist die Position der Boxen mit entsprechender Klassifizierung.

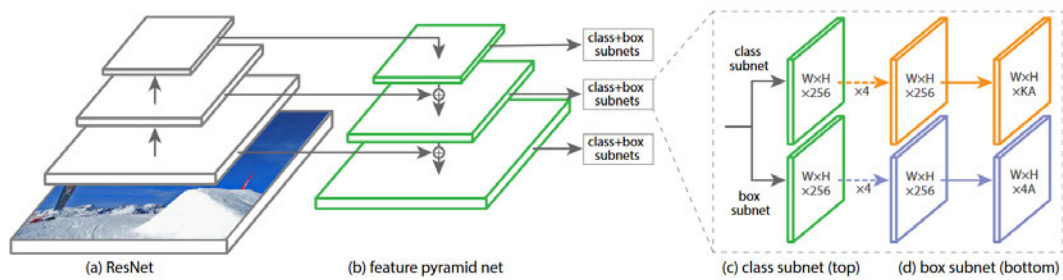


Abbildung 2.2: RetinaNet Architektur[23]

2.3 Labeling

Zum Annotieren der Bilder und Einzeichnen der Boxen wird das Programm LabelImg von Tzutalin[27] benutzt. Es ermöglicht das händische Annotieren der Bilder mit den entsprechenden Klassen und deren Position im Bild. Anschließend erstellt es eine XML-Datei mit den entsprechenden Bildern in einem gängigen Format, welches auch von ImageNet benutzt wird. Aus diesen annotierten Bildern kann dann ein Trainings-/ Evaluierungsdatensatz erstellt werden, mit welchem das Netzwerk trainiert werden kann.

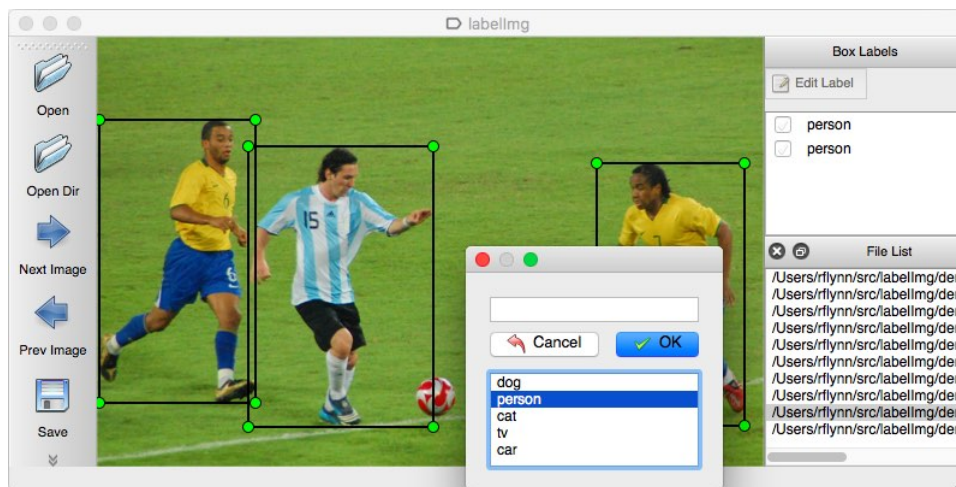


Abbildung 2.3: LabelImg Programm [27]

3 Entwurf

Ziel ist es, mithilfe eines antrainierten Netzwerkes Personen automatisch im Bild zu erkennen und ihre Positionsboxen zu erstellen. Es wird versucht, Teilnehmer in Fußgängerbereichen zu fotografieren. Dazu soll eine Wärmebildkamera und eine Kamera benutzt werden. Die Kamera dient dabei nur als Referenz, um die Wärmebilder richtig zu klassifizieren. Die Wärmebilder sollen zum Trainieren und Evaluieren des Netzwerkes genutzt werden. Für die Objekterkennung soll hierfür das RetinaNet benutzt werden. Wie von Marina Ivašić-Kos in "Human Detection in Thermal Imaging Using YOLO"[21] beschrieben, wird das Netzwerk mit einem eigenen Datensatz nachtrainiert, um bessere Ergebnisse zu erzielen. Versucht wird Personen, Fahrradfahrer und Hunde in Fußgängerbereichen zu erkennen. Auch soll der Hintergrund möglichst immer unterschiedlich sein, um einen Einsatz an vielen Orten zu ermöglichen.

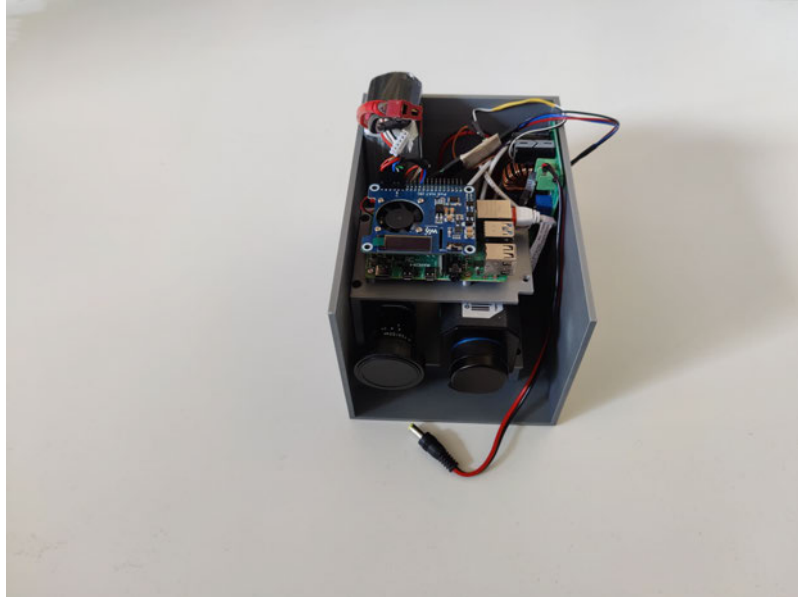


Abbildung 3.1: Kameragehäuse

Für die Aufnahme mit der Kamera und der Wärmebildkamera wird ein System gebaut, welches man in Abbildung 3.1 sieht. Das Kontextdiagramm 3.2 zeigt, wie die einzelnen Komponenten miteinander verbunden sind. Die Kameras werden über ein Netzwerk angesprochen, in welchem sie über den Ethernet Switch mit einem RaspberryPi verbunden sind. Der RaspberryPi verfügt zudem über einen angeschlossenen DHT22 Sensor, der Daten zur Außentemperatur sammelt. Diese können später dazu genutzt werden, um den Einfluss der Außentemperatur auf die Wärmebilder aufzuzeigen. Die Stromversorgung findet über den Ethernet Switch statt mit Ethernet Kabeln. Der Ethernet Switch selbst wird mit einem Power Supply Unit (PSU) Akku betrieben. So ist es möglich, unabhängig von der Stromversorgung Bilder aufzunehmen.

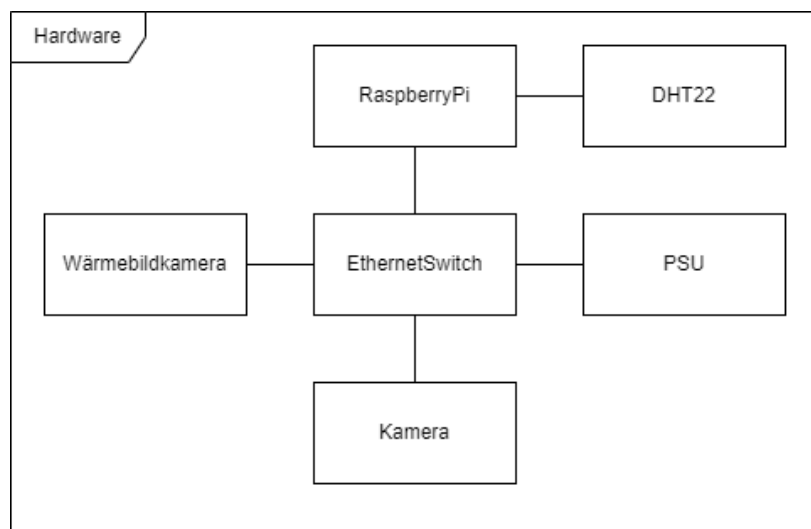


Abbildung 3.2: Kontextdiagramm der Hardware zur Aufnahme der Bilder

3.1 Anforderungen

Es müssen Bilder von Teilnehmern in Fußgängerbereichen gemacht werden. Dazu soll für jede Aufnahme ein Wärmebild und ein normales Bild zum gleichen Zeitpunkt aufgenommen werden. Die normalen Bilder dienen dabei nur als Referenz, um das korrekte Klassifizieren der Wärmebilder zu garantieren, da es besonders bei Gruppen von Personen schwierig ist, diese auseinander zu halten. Die Objekte auf den Wärmebildern sollen mit Boxdetection erkannt werden. Die Erkennung geschieht mit einem neuronalen Netz.

3.2 Hardware

Zum Sammeln der Daten werden zwei Kameras verwendet. Diese sind in einem Gehäuse untergebracht (siehe Abbildung 3.1), welches für die Aufnahmen in der Hand gehalten wird. Es beinhaltet einen 14,8V Lithium-Ionen Akku mit 5500mAh, einen DC zu DC Step Up Voltage Converter, einen Ethernet Switch mit Power over Ethernet, einer Basler a2A1920-51gcPRO Kamera mit einem Kowa LM3NCM | 1/1.8"3.5mm 2MP C-Mount Objektiv, eine Flir A35 Wärmebildkamera, eine Status-LED, einen Druckknopf, einen DHT22 Temperatur- und Luftfeuchtigkeitssensor, einen RaspberryPi 4 und einen PoE Hat für diesen.

3.2.1 Gehäuse

Das Gehäuse wurde in Autodesk Fusion360[8] modelliert und in PLA gedruckt. Es besteht aus dem Außengehäuse und zwei inneren Teilen, die miteinander verschraubt werden. Der tp-link TL-SG1005P Switch wird auf dem Boden des Gehäuses platziert. Darüber wird das erste innere Teil, an dem beide Kameras verschraubt sind, platziert und angeschraubt. Die Linsen der Kameras haben eine Distanz von 5,6mm zueinander. Über den Kameras wird das letzte Teil mit dem RaspberryPi verschraubt.

3.2.2 RaspberryPi 4

Zum Aufnehmen und Speichern der Bilder wird ein RaspberryPi 4 verwendet. Das Modell hat 8GB Arbeitsspeicher und verwendet eine 128GB große MicroSD Karte, auf der die Bilder gespeichert werden. [5]

3.2.3 DHT 22

Um die Außentemperatur während der Aufnahmen zu messen, wird ein Digital-output relative humidity & temperature sensor/module (DHT22) verwendet. Es kann die Temperatur in einem Bereich von $-40^{\circ}C$ bis $80^{\circ}C$ mit einer Genauigkeit von $\pm 0,2^{\circ}C$ und die Luftfeuchtigkeit in einem Bereich von 0% bis 100% mit einer Genauigkeit von $\pm 5\%$ bestimmen. Abfragen an den Sensor können maximal alle 2 Sekunden erfolgen.[25]

3.2.4 Flir A35

Zur Aufnahme der Infrarotbilder wird eine Flir A35 verwendet. Die Flir A35 hat eine Auflösung von 320 mal 256 Pixel. Sie kann Objekte in einem Wärmespektrum von $-40^{\circ}C$ bis $550^{\circ}C$ erkennen. Die Genauigkeit der Temperaturangabe liegt dabei bei mindestens $\pm 5^{\circ}C$ bzw. $\pm 5\%$ des Messwerts, je nachdem welcher Wert größer ist. Sie hat einen FoV von 48° mal 39° . Videos können mit bis zu 60 Bildern pro Sekunde aufgenommen werden [16].

3.2.5 Basler

Für die Aufnahmen der Bilder im sichtbaren Bereich wird eine Basler a2A1920-51gcPRO verwendet. Sie hat eine Auflösung von 1920 mal 1200 Pixel.[2] Für das Objektiv wird ein Kowa LM3NCM | 1/1.8"3.5mm 2MP Objektiv benutzt. Es hat einen FoV von 89° mal $73,8^{\circ}$. [4]

3.2.6 Akku

Beim Akku handelt es sich um einen 4 Zellen-Lithium-Ionen Akku mit einer Kapazität von $5500mAh$ bei einer Spannung von $14,8V$. Die maximale Dauerbelastung des Akkus liegt bei $25C$.

3.2.7 Ethernet Switch

Um die beiden Kameras und den RaspberryPi zu verbinden und mit Strom zu versorgen, wird ein Gigabit Ethernet Switch mit PoE benutzt. Bei seiner maximalen Leistung von $65W$ kann mit dem verwendeten Akku alles unter Volllast mindestens eine Stunde lang betrieben werden. [1]

3.2.8 DC Step up

Um die benötigte Spannung von $53,5V$ für den Ethernet Switch zu erzeugen, wird ein DC Step-up Konverter benutzt. Der Step-up Konverter akzeptiert eine Eingangsspannung von $12 - 60V$ und hat eine einstellbare Ausgangsspannung von $12 - 80V$. [3]

3.3 Software

Die Software ist in zwei Teile aufgeteilt, einmal die Software zum Aufnehmen der Bilder und der andere Teil zum Trainieren und Evaluieren des Netzwerkes. Beide Teile sind in Python geschrieben und benutzen die Version 3.8.

3.3.1 Aufnahme der Bilder

Auf dem RaspberryPi läuft das Betriebssystem Ubuntu 20.04.4 LTS (Focal Fossa), das 64-bit Server Image [13]. Um auf die Kameras zuzugreifen, müssen die entsprechenden Libraries benutzt werden. Für die Kamera ist es die PyPylon Library von Basler[7]. Sie ermöglicht es, die Kamera zu akquirieren und sowohl Bilder als auch Videos aufzunehmen. Werte wie die Belichtungszeit werden von der Kamera automatisch gesetzt. Einzig die Paketgröße wird kleiner gesetzt, da der RaspberryPi nicht in der Lage ist, JumboFrames zu verarbeiten. Für die Wärmebildkamera wird die PySpin Library von Flir verwendet mit der Version 2.4.0.143[17]. Da diese bei der Bedienung einen großen Overhead hat, wird zur vereinfachten Bedienung `simple_pyspin` verwendet[22]. Bei der Wärmebildkamera muss die Bildgröße angepasst werden, da ansonsten die Bilder in einem größeren Format abgespeichert werden als der Sensor aufnimmt, was einen schwarzen Balken am unteren Bildrand erzeugt. Um mit dem DHT22 Sensor zu kommunizieren, wird die Library `adafruit_dht22` verwendet[6]. Dabei ist darauf zu achten, dass ein Zugriff maximal alle 2 Sekunden erfolgt. Um die Aufnahme der Bilder starten und beenden zu können, wird ein Druckknopf verwendet, welcher an einen der Pins des RaspberryPi's angeschlossen ist. Eine LED, die ebenso mit dem RaspberryPi verbunden ist, wird zur Statusanzeige benutzt.

Bei den Bildern und der Temperatur ist es wichtig, dass diese möglichst synchron zueinander aufgenommen werden. So halten die Bilder den gleichen Zeitpunkt fest, nur in unterschiedlichen Wellenlängen, einmal im sichtbaren Spektrum und einmal im infraroten Spektrum. Die Temperaturmessung bildet die aktuelle Außentemperatur zum Zeitpunkt der Aufnahme ab.

3.3.2 Objekterkennung

Für die Objekterkennung wird TensorFlow 2.3.0 [26], mit der TensorFlow Object Detection API[19] verwendet. Für das Ausgangsnetz wird das RetinaNet50 (auch SSD Resnet50 FPN genannt) benutzt[23]. Um das Netzwerk zu trainieren, muss der Datensatz zuerst annotiert werden. Dazu wird der “LabelImg” von Tzutalin benutzt[27]. Die annotierten Bilder müssen anschließend in ein für TensorFlow verständliches Format umgewandelt werden. Um die Bilder in ein “.record” Format umzuwandeln, wird das Beispiel Skript “generate_tfrecord.py” verwendet. Anschließend kann das Netzwerk auf dem neuen Datensatz trainiert werden. Dafür wird das Skript “model_main_tf2.py” von der Object Detection API benutzt[19]. Das Skript ermöglicht es auch, parallel zum Training Metriken für das Netz zu erstellen, welche später in der Auswertung genutzt werden. Für den Aufbau der Objekterkennung wurde nach der Anleitung “Training Custom Object Detector” von Lyudmil Vladimirov vorgegangen[28].

4 Durchführung

Für das Sammeln der Daten werden die Kameras in der Hand gehalten. So ist gewährleistet, dass der Hintergrund sich verändert und das Netzwerk nicht auf einen spezifischen Hintergrund angewiesen ist. Zusätzlich wird die Kamera bei den Aufnahmen geschwenkt, um noch mehr unterschiedliche Umgebungen zu haben. Aufgenommen werden die Bilder auf dem Campus der Hochschule für Angewandte Wissenschaften (HAW) beim Berliner Tor 7 in Hamburg. Die Standorte im HAW Campus werden dabei regelmäßig gewechselt. Die Kameras nehmen alle 2 Sekunden Bilder auf. Parallel dazu wird die Außentemperatur aufgezeichnet.

4.1 Daten

Die Bilder sind mit der beschriebenen Hardware auf dem Campus-Gelände aufgenommen. Bilder sind nur von Fußgängern, Fahrradfahrern und Hunden gemacht worden. Auf den Bildern sind keine fahrenden oder stehenden Autos mit laufendem Motor zu sehen. Ebenso wurden keine Rollerfahrer, Rollstuhlfahrer oder Motorradfahrer aufgenommen. Kinderwagen sind zwar vorhanden, jedoch wird nicht versucht, diese zu erkennen. Kinder werden zu den Personen gezählt und als solche klassifiziert. Insgesamt wurden 236 Bilder gemacht.

4.1.1 Temperatureinfluss auf die Daten

Die Daten wurden bei einer Außentemperatur von durchschnittlich $13,3^{\circ}C$ und $6,2^{\circ}C$ aufgenommen. Der Sensor hat eine Messungenauigkeit von $\pm 0,2^{\circ}C$. Außerdem schwanken die aufgenommenen Temperaturen während der Aufnahmen um $1,2^{\circ}C$, was zu einer Messungenauigkeit von $\pm 0,8^{\circ}C$ führt.



Abbildung 4.1: Personen bei $5,8^{\circ}\text{C}$
Außentemperatur

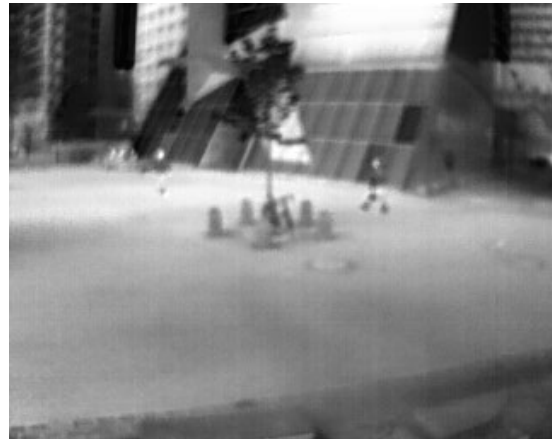


Abbildung 4.2: Personen bei $12,8^{\circ}\text{C}$
Außentemperatur

Auf den Abbildungen 4.1 und 4.2 sind Wärmebildaufnahmen von Personen bei unterschiedlichen Außentemperaturen zu sehen. In Abbildung 4.2 sind zwei Personen, welche sich etwas weiter weg von der Kamera befinden. Die Umgebung ist im Vergleich zu diesen Personen deutlich wärmer. Dadurch erscheinen die Personen dunkler. Besonders die rechte der beiden Personen ist im Vergleich zu der Umgebung wesentlich dunkler. Im Vergleich zu den Personen auf der Abbildung 4.1 sind die Personen auf Abbildung 4.2 bei einer Temperaturdifferenz von ca. $7,1^{\circ}\text{C}$, schwieriger zu erkennen. Es muss damit gerechnet werden, dass sich Personen bei niedrigeren Temperaturen besser erkennen lassen als bei wärmeren. Es liegen jedoch keine Daten vor, ob bei noch höheren Temperaturen der Kontrast immer weiter abnimmt oder der Kontrast wieder zunimmt, da dann die Umgebung insgesamt wärmer ist, als die Personen.

4.1.2 Probleme

Größere Personengruppen sind schwerer zu erkennen. Der Kontrast zwischen Personen ist bei Wärmebildern im Vergleich zu normalen Bildern sehr viel schlechter. So sind Personen auf Abbildung 4.3 schlechter zu erkennen als auf Abbildung 4.4. Besonders einzelne Personen sind in Gruppen schwierig zu erkennen. Dieses kann sowohl an dem Kontrast zwischen den Personen als auch an der geringen Auflösung der Wärmebilder liegen.



Abbildung 4.3: Wärmebild einer Gruppe



Abbildung 4.4: Bild im sichtbaren Spektrum einer Gruppe

Ein weiteres Problem können Fahrradfahrer darstellen, welche unter gewissen Bedingungen auch als Personen erkannt werden könnten. Die Fahrräder selber erscheinen sehr dunkel auf den Bildern, siehe Abbildung 4.5. In kälteren Umgebungen könnte der Kontrast zum Fahrrad nicht ausreichen.



Abbildung 4.5: Bild eines Fahrradfahrers

Einen Einfluss auf die Erkennbarkeit von Personen hat die Kleidung, die sie tragen. Je besser die Kleidung isoliert, desto dunkler erscheinen sie auf den Bildern bei den gemessenen Temperaturen. So ist der Oberkörper bei Personen ohne Jacke deutlich kontrast-

reicher als bei Personen mit Jacke. Besonders stark sieht man den Einfluss der Isolierung bei Hunden. So ist ein Hund mit langem Fell 4.7 deutlich schwerer zu erkennen als ein Hund mit kurzem Fell 4.6.



Abbildung 4.6: Wärmebild von einem Hund mit kurzen Fell

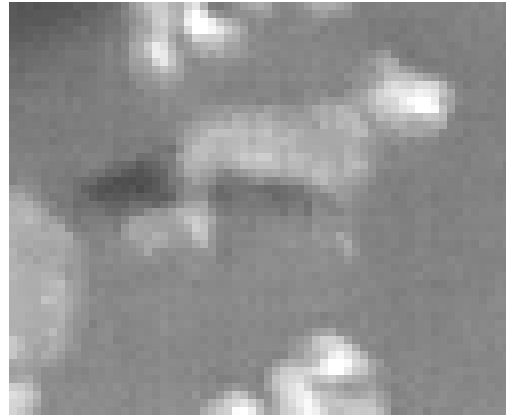


Abbildung 4.7: Wärmebild von einem Hund mit langem Fell

4.2 Training

Zum Trainieren des Netzwerkes wird nach der Anleitung "Training Custom Object Detector"[28] vorgegangen. Benutzt wird TensorFlow 2 mit Python 3.8. Nach dem Aufsetzen der nötigen der Umgebung werden die Bilder klassifiziert. Anschließend werden diese klassifizierten Bilder benutzt, um das Netzwerk zu trainieren.

4.2.1 Klassifizieren der Bilder

Die zu erkennenden Objekte werden auf den Wärmebildern markiert und entsprechend ihrer Klasse klassifiziert. Da es unter Umständen schwierig sein kann, Personen auf den Wärmebildern zu erkennen, werden die Bilder von der normalen Kamera beim Klassifizieren mitbenutzt. Sollte ein zu klassifizierendes Objekt auf den Wärmebildern nicht zu erkennen sein, so wird dieses nicht markiert.

4.2.2 Trainieren des Netzwerkes

Das benutzte SSD Resnet50 FPN Netzwerk verwendet eine Auflösung von 640×640 Pixel. Da die Wärmebildkamera eine Auflösung von 320×256 Pixel hat, wird die Eingabeschicht des Netzwerkes auf 320×256 Pixel runterskaliert.

Zum Trainieren wird der gesammelte Datensatz aufgeteilt in Bilder zum Evaluieren und zum Trainieren. 30 Bilder werden dabei zum Evaluieren des Netzwerkes benutzt und 206 Bilder zum Trainieren. Die Anzahl der Klassen wurde auf 3 reduziert, da es entsprechend nur drei zu erkennenden Objekte gibt, Personen, Hunde und Fahrradfahrer. Trainiert wird das Netzwerk für 25000 Schritte mit einer BatchSize von 12.

5 Auswertung der Netzwerk-Metriken

Am Ende des Trainings wird das Netzwerk mit Tensorboard ausgewertet. Es erstellt Metriken über die Performance des Netzwerkes. Die Abbildung 5.1 zeigt die mittlere Durchschnittspräzision(mAP) des Netzwerkes über den Trainingsverlauf auf den Evaluierungsdaten an.

Die mAP wird zusammengesetzt aus der Durchschnittspräzision (AP) aller Klassen. Bestimmt wird die AP aus der Recall-Differenz und der Präzision für alle Boxenüberlappungsschwellenwerte (IOU). Die IOU ist eine Rate, die angibt, wie viele der geschätzten Boxen mit ihrer eigentlichen Position und Größe übereinstimmt, sie wird für unterschiedliche Schwellenwerte bestimmt. Der Recall gibt an, wie viele der insgesamt zu erkennenden Objekte erkannt wurden. Die Präzision gibt an, wie viele der erkannten Objekte tatsächlich richtig erkannt wurden[18].

Das Netz verbessert sich bis Schritt 13000, danach stagniert die Verbesserung. Ab Schritt 16000 lässt sich sogar eine leichte Abwärtstendenz feststellen.

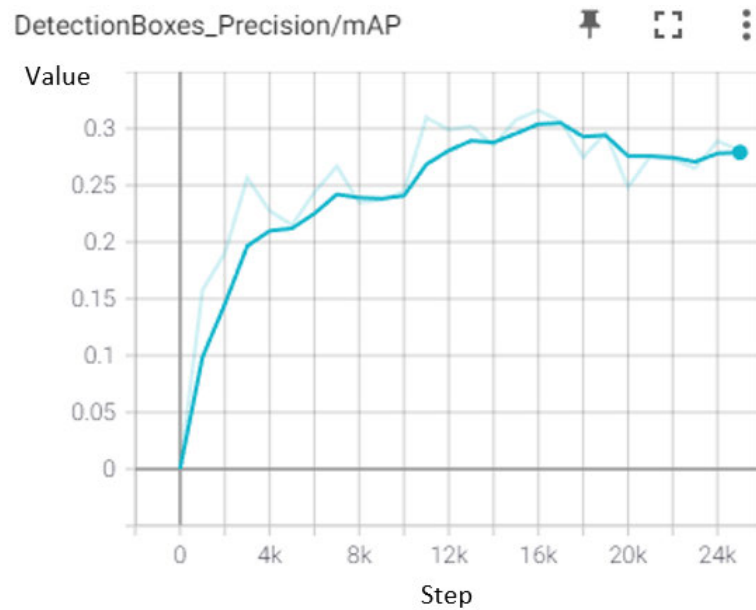


Abbildung 5.1: mAP des Netzes über den Trainingsverlauf

Die Metrik lässt sich weiter in 3 verschiedene Komponenten aufteilen, die durchschnittliche Präzision für große, mittelgroße und kleine Objekte. Kleine Objekte sind Boxen, die kleiner als 32 mal 32 Pixel sind. Mittelgroße Objekte sind Boxen zwischen den Größen 32 mal 32 und 96 mal 96 Pixel. Große Objekte sind alle, die noch größer als 96 mal 96 Pixel sind. Durch die geringe Auflösung der Kamera und den großen Kameraabstand zu den Objekten gibt es keine Bilder, auf denen Objekte unter die Kategorie groß fallen.

In den Abbildungen 5.2 und 5.3 ist zu erkennen, dass mittelgroße Objekte um 30% besser zu erkennen sind. Die Präzision für mittelgroße Objekte verbessert sich ab 9000 Schritten nicht mehr merklich. Während für kleine Objekte die Präzision bei ungefähr 16000 Schritten am höchsten ausfällt, verschlechtert sich die Präzision in den folgenden Schritten.

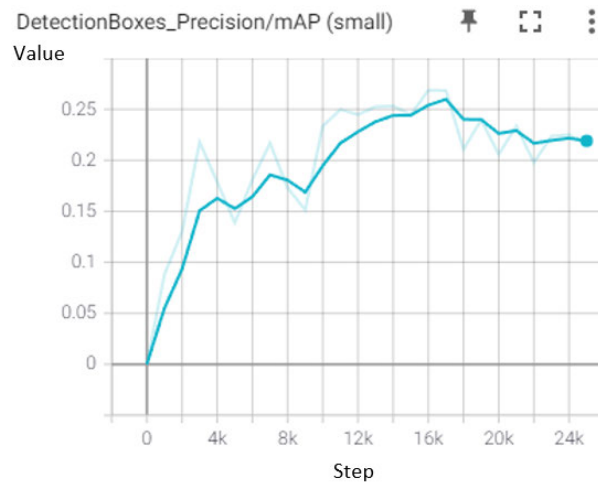


Abbildung 5.2: mAP des Netzes für kleine Objekte

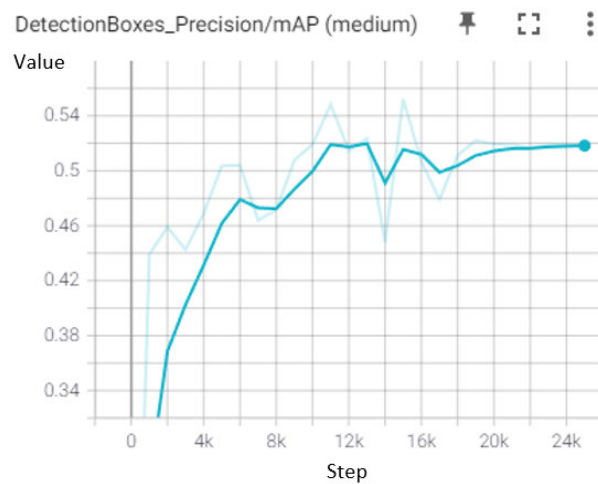


Abbildung 5.3: mAP des Netzes für mittelgroße Objekte

Auf den Abbildungen 5.4 und 5.5 ist die prozentuale Anzahl der Boxen zu sehen, welche eine Boxenüberlappung von mindestens 50% beziehungsweise 75% haben. Ungefähr die Hälfte aller erkannten Objekte haben am Ende des Trainings eine Überlappung von mindestens 50% erreicht (Abbildung 5.4). Jedoch kann man auch hier an dem Graphen erkennen, dass die Präzision gegen Ende des Trainings abnimmt. Besonders stark kann man diese Abnahme anhand der Abbildung 5.5 erkennen. So war die höchste Präzision

bei ungefähr 35%, fällt danach jedoch auf unter 25%. Dies deutet darauf hin, dass weiteres Training nicht mehr zu einer Verbesserung, sondern zu einer Verschlechterung der Erkennung führt.

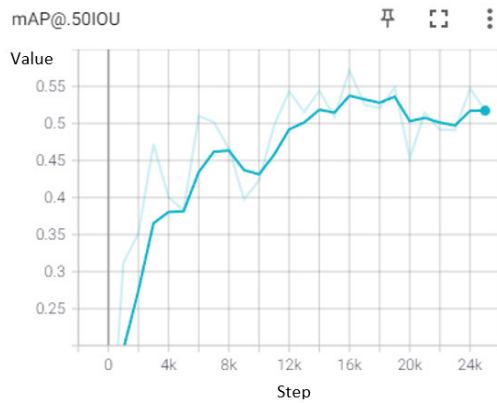


Abbildung 5.4: 50% Überlappung

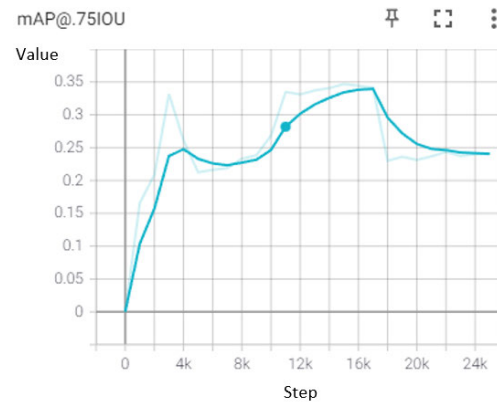


Abbildung 5.5: 75% Überlappung

In Abbildung 5.6 sieht man die Fehlerrate für die Lokalisierung der Boxen. Die Fehlerrate gibt an, wie viele Fehler das Netzwerk bei der Auswertung der Bilder macht. Idealerweise wird diese Fehlerrate über die Zeit immer kleiner. Betrachtet man die Fehlerraten bei der Lokalisierung der Boxen, sieht man, dass die beiden Fehlerraten in der Y-Achse versetzt sind. Die Evaluierung erzielt wesentlich schlechtere Werte als das Training. Idealerweise sollte der Abstand zwischen den beiden Fehlerraten so gering wie möglich sein. Trotzdem ist zu sehen, dass sich beide Fehlerraten im Verlauf ähnlich verhalten.

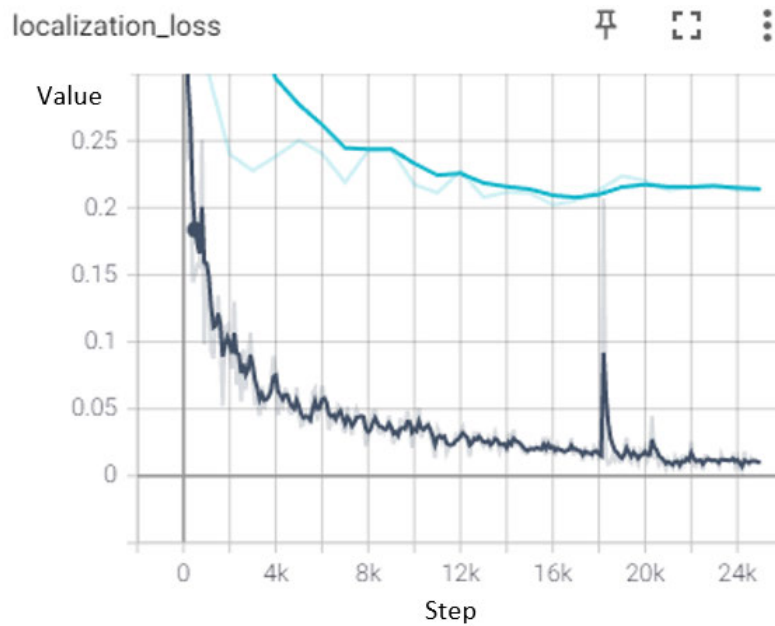


Abbildung 5.6: Lokalisierung Fehlerrate über den Trainingsverlauf.
Hellblau Evaluierungsdaten, Dunkelblau Trainingsdaten

Bei der Klassifikationsfehlerrate (Abbildung 5.7) konvergiert diese stetig nach unten. Die Fehlerrate für die Evaluierungsdaten verbessert sich bis ca. 12000 Schritte, verschlechtert sich dann über den restlichen Trainingsverlauf wieder. Dies deutet darauf hin, dass für die Klassifikation der Bilder eine Überanpassung des Netzwerkes erfolgt ist. Das heißt, es lernt auf den Trainingsdaten die Objekte immer besser zu klassifizieren, kann dieses aber nicht auf Bilder außerhalb der Trainingsdaten anwenden.

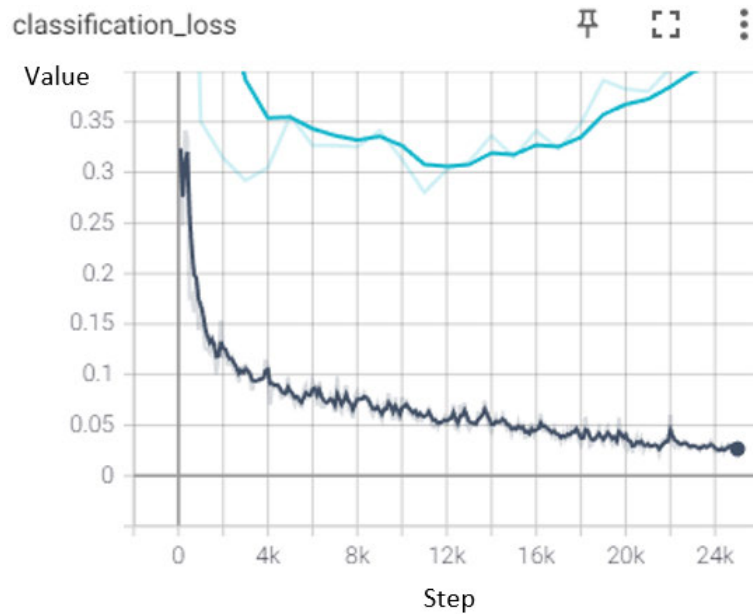


Abbildung 5.7: Klassifikation Fehlerrate über den Trainingsverlauf.
Hellblau Evaluierungsdaten, Dunkelblau Trainingsdaten

Wie man in den Abbildungen in diesem Kapitel sehen kann, gibt es durchaus Verbesserungspotential beim Training des Netzwerkes. So sieht man am Netzwerk eine mögliche Überanpassung während des Trainingsverlaufes. Der Überanpassung kann man entgegenwirken, indem man das Training frühzeitig beendet, sobald sich das Netzwerk verschlechtert. So zeigt Abbildung 5.1, dass eine Trainingsdauer von nur 16000 Schritten eine bessere Präzision hat als eine Trainingsdauer von 25000 Schritten. Eine weitere Möglichkeit, der Überanpassung entgegenzuwirken, ist die Nutzung eines größeren Datensatzes.

6 Ergebnisse

Dieses Kapitel befasst sich mit den Ergebnissen der Netzwerkklassifikation. Außerdem werden gewisse Eigenheiten bei der Objekterkennung von Wärmebildern beschrieben. Auch werden Probleme behandelt, welche bei diesem Verfahren auftreten können.

6.1 Ergebnisse des Netzwerkes

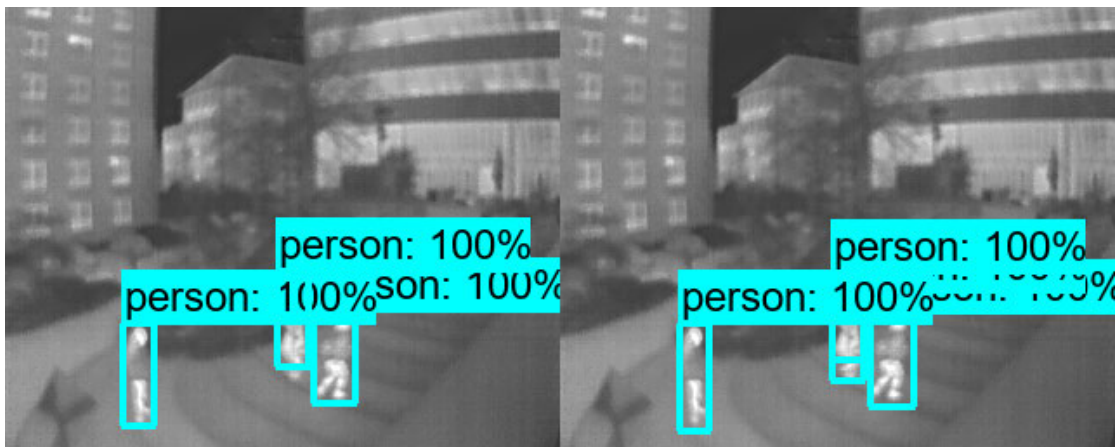


Abbildung 6.1: Links erkannte Personen nach 25000 Schritten, rechts Referenzbild

Auf der Abbildung 6.1 sieht man links die vom Netzwerk klassifizierte Objekte und rechts die handannotierte Referenz. Das Netzwerk klassifiziert mit einer Sicherheit von 100% drei Personen auf dem Bild. Jedoch umfasst die vom Netzwerk erstellte Box bei der mittleren Person nicht deren Füße. Schaut man sich die Szene aufgenommen von einer normalen Kamera an, so sieht man in Abbildung 6.2, dass sich in der Mitte noch eine weitere Person befindet, welche von der Person vor ihr teilweise verdeckt wird. Diese wurde vom Netzwerk nicht als eigene Person erkannt, sondern als Teil des Menschen vor

ihr. In Abbildung 6.3, welches Abbildung 6.1 ohne Annotationen darstellt, sieht man, dass die beiden Personen in der Mitte im Wärmebild deutlich schlechter auseinander zu halten sind. An diesem Beispiel kann man sehen, dass es sich schwierig gestaltet, sich überlappende Personen auf Wärmebildern voneinander abzugrenzen.



Abbildung 6.2: Referenzbild zu 6.1



Abbildung 6.3: Wärmebild ohne Annotationen

Kleine Objekte erkennt das Netzwerk jedoch nur mit niedriger Gewissheit. So sieht man auf Abbildung 6.4, dass alle 5 Personen erkannt werden. Dagegen sieht man in den Annotationen, dass die Vorhersagegewissheit für die Personen zwischen 52% und 75% liegt, siehe Tabelle A.1. Es lässt vermuten, dass kleinere Objekte für das Netzwerk schwerer zu erkennen sind. Dies unterstützt Abbildung 6.1, wo erkannte mittelgroße Objekte mit einer Gewissheit von 100% bestimmt werden. Dem widerspricht jedoch Abbildung 6.4. Dort nimmt die Person 1 mit der größten Vorhersagegewissheit die kleinste Bildfläche ein, während Person 5, welche eine größere Fläche einnimmt, die schlechteste Vorhersagegewissheit hat.

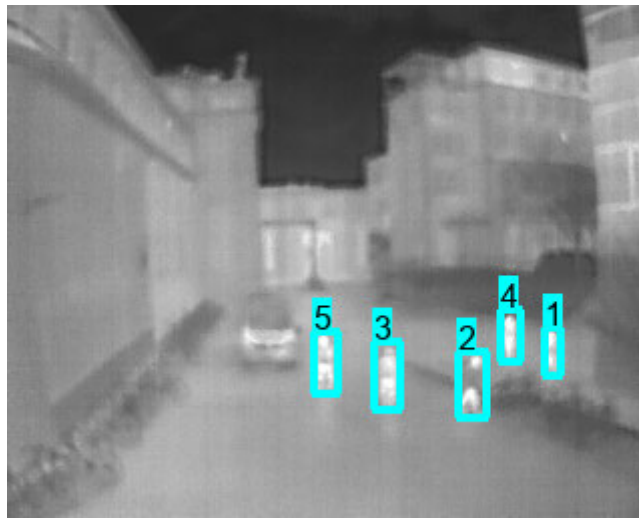


Abbildung 6.4: Personen weiter entfernt; siehe Tabelle A.1

Bei der Erkennung von Fahrradfahrern ist eines der Probleme die geringe Anzahl an Fahrradfahrern im Datensatz. So zeigt Abbildung 6.5 zwar einen Fahrradfahrer, das Netzwerk zeichnet die Box jedoch nur über die Person selbst und nicht auch über das Fahrrad, obwohl das Fahrrad sich gut vom Hintergrund abhebt. Der Grund hierfür kann sein, dass das Netzwerk Boxen in Personenform präferiert, da diese den Großteil der Trainingsdaten ausmachen. In den Trainingsdaten wurden Fahrradfahrer jedoch immer mit dem gesamten Fahrrad markiert. Auch ist die Vorhersagegewissheit vom Netzwerk, mit dem es das Objekt als Fahrradfahrer bestimmt, mit 40% recht niedrig.

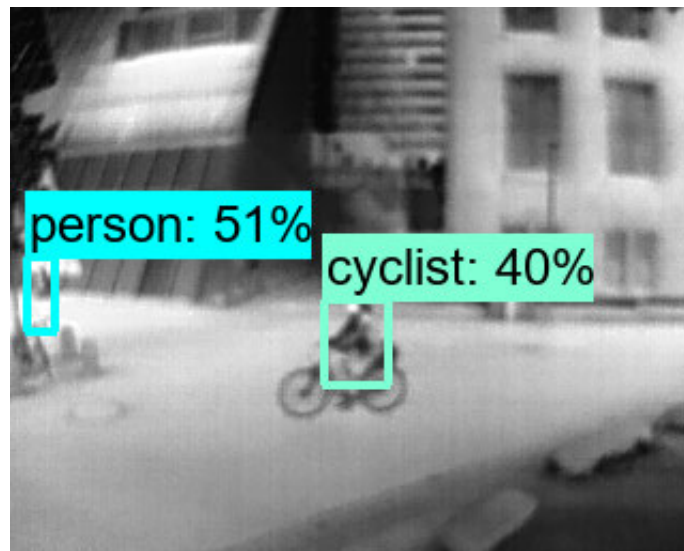


Abbildung 6.5: Fahrradfahrer

Die Personengruppe auf Abbildung 6.6 konnte bis auf zwei Ausnahmen vollständig erkannt werden. Die Vorhersagegewissheit liegt für die Objekte 1 bis 17 bei über 90%. Ausnahmen sind Personen, welche verdeckt und/oder nur abgeschnitten im Bild zu sehen sind. Dies sieht man bei den Boxen 18 bis 22. In der Abbildung kann man neben der Personengruppe einen Hund sehen. Dieser ist mit einer grünen Box markiert. Er hat im Gegensatz zu anderen Objekten einen geringeren Kontrast zu dem Hintergrund. Trotzdem kann das Netzwerk den Hund mit einer Sicherheit von 99% erkennen. Grund hierfür sind vermutlich die klar erkennbaren Umrisse des Hundes. So gibt es bei dem Hund keine Überlappungen mit anderen Objekten.

Trotz der im Kapitel Training bereits erwähnten Erkennbarkeit von Hunden mit kurzem Fell zeigt ein direkter Vergleich von Abbildung 6.7 mit Abbildung 6.6, dass das Netzwerk den Hund mit langem Fell besser erkannt hat. Der Hund in der Abbildung 6.7 wird fälschlicherweise als Person erfasst. Zwar weist der Hund in der Abbildung 6.7 durch sein kürzeres Fell einen höheren Kontrast zum Hintergrund auf, jedoch hilft dieses nicht bei der Erkennung. So wird der Hund in der Abbildung 6.6 auch als solcher mit einer hohen Vorhersagegewissheit identifiziert. Deutlich wichtiger für die Erkennbarkeit scheint, dass ein deutlicher Umriss des Objektes zu erkennen ist und das Objekt nicht mit anderen Objekten überlappt.

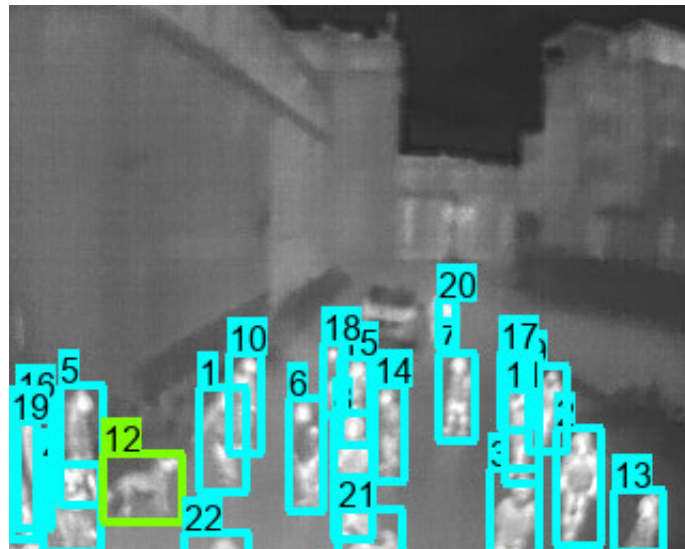


Abbildung 6.6: Gruppe von Personen und ein Hund; siehe Tabelle A.2

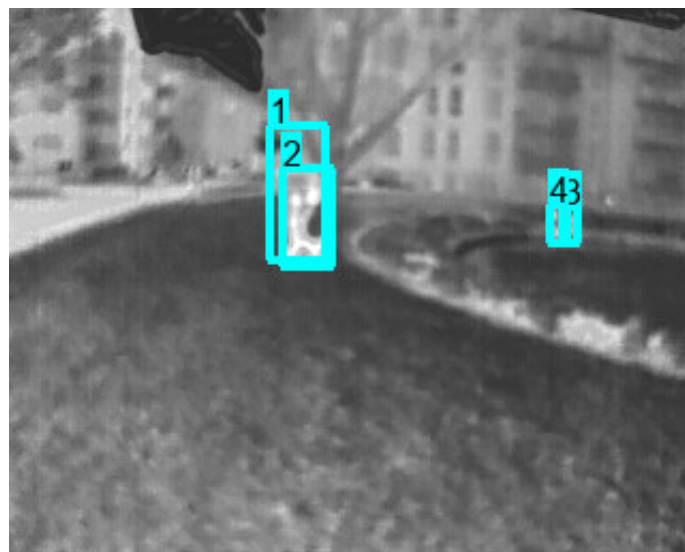


Abbildung 6.7: Hund mit seinem Besitzer; siehe Tabelle A.3

Die Abbildung 6.8 zeigt die Präzision des RetinaNet50 auf dem COCO Datensatz bei unterschiedlichen Auflösungen. Der COCO-Datensatz beinhaltet insgesamt 91 Kategorien an Objekten[24]. Die relevanteste Auflösung hierbei ist 400 mal 400 Pixel, da diese der Auflösung der Wärmebilder am nächsten kommt.

In Abbildung 6.8 sieht man unter AP, dass die durchschnittliche Präzision bei 30,5% liegt. Zu vermerken ist, dass bei der AP die mAP, also die AP über alle Klassen gemeint ist. Verglichen dazu wurde beim nachtrainierten Netzwerk eine Präzision von ungefähr 28% erreicht, siehe Abbildung 5.1.

Das RetinaNet50 bei einer Auflösung von 400 mal 400 Pixel scheint auf dem COCO Datensatz ähnlich präzise zu sein, wie das für diese Arbeit nachtrainierten Netzwerk auf dem eigens erstellten Datensatz bei einer Auflösung von 320 mal 256 Pixel. In Abbildung 6.8 kann man sehen, dass eine höhere Auflösung eine bessere Durchschnittspräzision auf Kosten der Berechnungsdauer erzielt.

Jedoch ist ein Vergleich des in dieser Arbeit benutzten Netzwerkes schwierig, da hier deutlich weniger Klassen verwendet werden und es keinen direkten Auflösungsvergleich in Abbildung 6.8 gibt. Weiter ist der Informationsgehalt der Bilder pro Pixel geringer, da die Farbbilder aus dem COCO Datensatz über drei Werte pro Pixel verfügen, während Wärmebilder im Gegensatz nur einen Wert pro Pixel haben.

depth	scale	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L	time
50	400	30.5	47.8	32.7	11.2	33.8	46.1	64
50	500	32.5	50.9	34.8	13.9	35.8	46.7	72
50	600	34.3	53.2	36.9	16.2	37.4	47.4	98
50	700	35.1	54.2	37.7	18.0	39.3	46.4	121
50	800	35.7	55.0	38.5	18.9	38.9	46.3	153

Abbildung 6.8: RetinaNet50 auf dem COCO Datensatz[23]

6.2 Privatsphäre

Laut der DSGVO Artikel 4[11] dürfen, um die Privatsphäre zu gewährleisten, keine personenbezogenen Daten erhoben werden. Personenbezogene Daten sind alle Daten, die sich auf eine “identifizierte oder identifizierbare natürliche Person[...] beziehen”[15]. Eine Person wird als identifizierbar angesehen, wenn mehrere besondere Merkmale der Person identifiziert werden können. So dürfen keine Daten zu vererbten oder erworbenen Eigenschaften von Personen erhoben werden. Diese Merkmale können Narben, Haarfarbe, Hautfarbe, Augenfarbe, biologisches Geschlecht, etc. sein.

Auf der Abbildung 6.9 ist eine Gruppe von Personen zu sehen. Es ist jedoch nicht möglich, Haarfarbe, Hautfarbe oder Augenfarbe zu bestimmen. Auch sind mögliche Tattoos

oder Narben auf den Bildern nicht sichtbar. Anhand der Körperform könnte in manchen Fällen das Geschlecht der Person bestimmt werden. Jedoch reicht eine Bestimmung des Geschlechts allein nicht aus, um als personenbezogene Daten zu gelten, da es nicht möglich ist, eine Person nur anhand ihres biologischen Geschlechtes eindeutig zu identifizieren.

Identifikationsmethoden von Personen mittels Videoaufnahmen wären denkbar. Diese wurden hier jedoch nicht betrachtet, da nur individuelle Bilder verwendet wurden.



Abbildung 6.9: Wärmebild einer Gruppe von Personen

7 Schlusswort

In dieser Arbeit wurde versucht, auf Wärmebildern Personen, Hunde und Fahrradfahrer in Fußgängerbereichen zu erkennen. Dafür wurde eine eigene Hardware entworfen, die Wärmebilder, normale Bilder und die dazugehörige Außentemperatur gleichzeitig aufnimmt. Mit dieser Hardware wurden am HAW-Campus an verschiedenen Standorten Daten gesammelt. Trotz der kleinen Datenmenge konnte gezeigt werden, dass es allein mit Wärmebildern möglich ist, Personen, Hunde und Fahrradfahrer größtenteils zu erkennen. Bei den Ergebnissen wurde festgestellt, dass die unterschiedliche Außentemperatur von ungefähr $6,2^{\circ}\text{C}$ und $13,3^{\circ}\text{C}$ keinen merklichen Unterschied auf die Erkennbarkeit der Objekte hat, obwohl eine niedrigere Außentemperatur für einen höheren Kontrast sorgt. Zwar ist der Kontrast zum Hintergrund wichtig, jedoch hat eine möglichst klare Unterscheidung zwischen den Objekten einen wesentlich größeren Einfluss auf die Erkennbarkeit der Objekte. Insgesamt hat das Netzwerk gezeigt, dass es in vielen Situationen Objekte lokalisieren kann. Auch mit Gruppen, partiell sichtbaren Objekten und verdeckten Objekten, kann das Netzwerk umgehen.

Verbesserungspotential gibt es noch bei der Klassifizierung der Objekte. Es konnte anhand von Abbildung 5.7 gezeigt werden, dass es zu einer möglichen Überanpassung des Netzwerkes gekommen ist. Dies könnte mit einer größeren Datenmenge oder einer kürzeren Trainingsdauer verbessert werden.

Zum Thema Privatsphäre wurde mit der DSGVO[15] begründet, dass es sich bei den Wärmebildern nicht um personenbezogene Daten handelt, da diese die Feststellung von bestimmten Merkmalen einer Person nicht im genügend großen Umfang zulassen, um sie zu identifizieren.

Für die zukünftige Forschung wäre eine nähere Betrachtung der Temperatur sinnvoll. So ist der Temperaturunterschied von den Daten mit 7°C zu gering, um Aussagen über einen ganzjährigen Einsatz von Wärmebildkameras für die Objekterkennung zu treffen.

Bilder an mehreren Standorten über ein Jahr hinweg aufzunehmen, würde helfen Daten von größeren Temperaturbereichen zu sammeln. So bleibt die Frage, wie sich die Objekterkennung im Winter bei Minusgraden oder im Sommer bei über 25°C verhält, unbeantwortet. Auch eine Wärmebildkamera mit einer höheren Auflösung wäre sinnvoll. So würden vor allem kleinere Objekte in den Konturen höhere Details aufweisen, was die Objekterkennung vereinfachen und dadurch die Präzision erhöhen würde. Eine höhere Auflösung müsste jedoch auch im Hinblick auf die Privatsphäre neu betrachtet werden, da diese möglicherweise zu einer Erkennung von personengebundenen Merkmalen führen könnte.

Ebenfalls interessant zu betrachten wäre der Performancevergleich von Objekterkennungsnetzwerken in Bezug auf Wärmebilder. Dabei könnte festgestellt werden, ob bei sichtbaren Bildern besser performende Netze ihre bessere Leistung bei Wärmebildern beibehalten, oder ob andere Netzwerke auf diesen Bildern bessere Ergebnisse erzielen können. Für diesen Zweck wäre es sinnvoll, einen Datensatz ähnlich zu COCO für Wärmebilder zu erstellen, um eine Vergleichbarkeit zwischen verschiedenen Netzwerken zu ermöglichen.

Literaturverzeichnis

- [1] LINK tp (Hrsg.): *5-Port Gigabit Desktop Switch with 4-Port PoE+*. – URL [https://static.tp-link.com/upload/product-overview/2021/202112/20211216/TL-SG1005P\(UN\)4.0%20%20TL-SG1005LP\(UN\)1.0_Datasheet.pdf](https://static.tp-link.com/upload/product-overview/2021/202112/20211216/TL-SG1005P(UN)4.0%20%20TL-SG1005LP(UN)1.0_Datasheet.pdf). – Zugriffsdatum: 2022-05-11
- [2] BASLER (Hrsg.): *a2A1920-51gcPRO - Basler ace 2*. – URL <https://www.baslerweb.com/de/produkte/kameras/flaechenkameras/ace2/a2a1920-51gcpro/>. – Zugriffsdatum: 2022-05-11
- [3] ANGEER (Hrsg.): *ANGEER 12-60V to 12-80V 600W 10A DC-DC Wandler Stromversorgung Step-up Konverter Modul Netzteil*. – URL <https://www.amazon.de/ANGEER-Wandler-Stromversorgung-Konverter-Netzteil/dp/B07RL9QD9M>. – Zugriffsdatum: 2022-05-11
- [4] KOWA (Hrsg.): *LM3NCM | 1/1.8"3.5mm 2MP C-Mount Objektiv*. – URL <https://www.kowa-lenses.com/lm3ncm-2mp-industrieobjektiv-c-mount>. – Zugriffsdatum: 2022-05-11
- [5] LTD., Raspberry Pi T. (Hrsg.): *Raspberry Pi 4 Computer*. – URL <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf>. – Zugriffsdatum: 2022-05-11
- [6] *Adafruit Python DHT Sensor Library*. Mai 2022. – URL https://github.com/adafruit/Adafruit_Python_DHT. – Zugriffsdatum: 2022-06-01. – original-date: 2014-06-07T04:40:36Z
- [7] *Getting Started*. Mai 2022. – URL <https://github.com/basler/pypylon>. – Zugriffsdatum: 2022-02-06. – original-date: 2018-03-05T14:35:19Z
- [8] AUTODESK: *Fusion 360 | Cloudbasierte 3D-Software für CAD, CAM, CAE und PCB*. – URL <https://www.autodesk.de/products/fusion-360/overview>. – Zugriffsdatum: 2022-06-06

- [9] BENEVOLO, Clara ; DAMERI, Renata P. ; D’AURIA, Beatrice: Smart Mobility in Smart City. In: TORRE, Teresina (Hrsg.) ; BRACCINI, Alessio M. (Hrsg.) ; SPINELLI, Riccardo (Hrsg.): *Empowering Organizations*. Cham : Springer International Publishing, 2016, S. 13–28. – ISBN 978-3-319-23784-8
- [10] BERRYBASE: *Adafruit MLX90640 24x32 IR Wärmebildkamera Breakout*. – URL <https://www.berrybase.de/sensoren-module/kameramodule/adafruit-mlx90640-24x32-ir-w-228-rmebildkamera-breakout>. – Zugriffsdatum: 2022-05-27
- [11] BERRYBASE: *Adafruit TTL Seriell JPEG Kamera mit NTSC-Video*. – URL <https://www.berrybase.de/sensoren-module/kameramodule/adafruit-ttl-seriell-jpeg-kamera-mit-ntsc-video>. – Zugriffsdatum: 2022-05-27
- [12] BUNDESREPUBLIK DEUTSCHLAND: § 4 BDSG – Videoüberwachung öffentlich zugänglicher Räume | BDSG (neu) 2018. – URL <https://dsgvo-gesetz.de/bdsg/4-bdsg/>. – Zugriffsdatum: 2022-05-27
- [13] CANONICAL: *Ubuntu 20.04.4 LTS (Focal Fossa)*. – URL <https://releases.ubuntu.com/20.04/>. – Zugriffsdatum: 2022-03-01
- [14] DAVIS, James W. ; KECK, Mark A.: A Two-Stage Template Approach to Person Detection in Thermal Imagery. In: *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION’05) - Volume 1* Bd. 1, Januar 2005, S. 364–369
- [15] EUROPÄISCHE UNION: *Art. 4 DSGVO – Begriffsbestimmungen*
- [16] FLIR, Teledyne: *FLIR A35 | Teledyne FLIR*. – URL <https://www.flir.de/products/a35/>. – Zugriffsdatum: 2021-09-13
- [17] FLIR, Teledyne: *Spinnaker SDK*. – URL <https://www.flir.de/products/spinnaker-sdk?vertical=machine+vision&segment=iis>. – Zugriffsdatum: 2022-02-06
- [18] HENDERSON, Paul ; FERRARI, Vittorio: *End-to-End Training of Object Class Detectors for Mean Average Precision*. März 2017. – Pages: 213. – ISBN 978-3-319-54192-1
- [19] HONGKUN YU, Xianzhi Du Yeqing Li Abdullah Rashwan Le Hou Pengchong Jin Fan Yang Frederick Liu Jaeyoun K. ; LI, Jing: *TensorFlow Model Garden*. <https://github.com/tensorflow/models>. 2020

- [20] ISMAGILOVA, Elvira ; HUGHES, Laurie ; RANA, Nripendra P. ; DWIVEDI, Yogesh K.: Security, Privacy and Risks Within Smart Cities: Literature Review and Development of a Smart City Interaction Framework. In: *Information Systems Frontiers* (2020). – URL <https://doi.org/10.1007/s10796-020-10044-1>
- [21] IVAŠIĆ-KOS, Marina ; KRIŠTO, Mate ; POBAR, Miran: Human Detection in Thermal Imaging Using YOLO. In: *Proceedings of the 2019 5th International Conference on Computer and Technology Applications*. New York, NY, USA : Association for Computing Machinery, April 2019 (ICCTA 2019), S. 20–24. – URL <https://doi.org/10.1145/3323933.3324076>. – Zugriffsdatum: 2022-05-27. – ISBN 978-1-4503-7181-0
- [22] KLECKNERLAB: *simple_pyspin*. Mai 2022. – URL https://github.com/klecknerlab/simple_pyspin. – Zugriffsdatum: 2022-02-06. – original-date: 2019-12-20T18:12:14Z
- [23] LIN, Tsung-Yi ; GOYAL, Priya ; GIRSHICK, Ross ; HE, Kaiming ; DOLLÁR, Piotr: Focal Loss for Dense Object Detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2020), Nr. 2, S. 318–327
- [24] LIN, Tsung-Yi ; MAIRE, Michael ; BELONGIE, Serge ; HAYS, James ; PERONA, Pietro ; RAMANAN, Deva ; DOLLÁR, Piotr ; ZITNICK, C. L.: Microsoft COCO: Common Objects in Context. In: FLEET, David (Hrsg.) ; PAJDLA, Tomas (Hrsg.) ; SCHIELE, Bernt (Hrsg.) ; TUYTELAARS, Tinne (Hrsg.): *Computer Vision – ECCV 2014*. Cham : Springer International Publishing, 2014 (Lecture Notes in Computer Science), S. 740–755. – ISBN 978-3-319-10602-1
- [25] LIU, Thomas ; AOSONG ELECTRONICS CO., Ltd (Hrsg.): *Digital-output relative humidity temperature sensor/module DHT22 (DHT22 also named as AM2302)*. – URL <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>. – Zugriffsdatum: 2022-05-11
- [26] TENSORFLOW: *tensorflow*. – URL <https://github.com/tensorflow/tensorflow>. – Zugriffsdatum: 2022-06-02
- [27] TZUTALIN: *LabelImg*. Mai 2022. – URL <https://github.com/tzutalin/labelImg>. – Zugriffsdatum: 2022-05-29. – original-date: 2015-09-17T01:33:59Z

- [28] VLADIMIROV, Lyudmil: *Training Custom Object Detector*. – URL <https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/training.html>. – Zugriffsdatum: 2022-01-26
- [29] WILIAM K PURVES, Gordon H. Orians H. Craig H.: *Biologie*. 7. Auflage. Heidelberg : Elsevier, Spektrum, Akad. Verl., 2006. – ISBN 3827416302
- [30] ZHOU, Xinyi ; GONG, Wei ; FU, WenLong ; DU, Fengtong: Application of deep learning in object detection. In: *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, Mai 2017, S. 631–634

A Anhang

ID	Class	Value
1	person	75%
2	person	71%
3	person	67%
4	person	66%
5	person	52%

Tabelle A.1: Tabelle zu Abbildung 6.4

ID	Class	Value
1	person	100%
2	person	100%
3	person	100%
4	person	100%
5	person	100%
6	person	100%
7	person	100%
8	person	100%
9	person	100%
10	person	100%
11	person	100%
12	dog	99%
13	person	99%
14	person	97%
15	person	95%
16	person	94%
17	person	91%
18	person	60%
19	person	54%
20	person	46%
21	person	45%
22	person	31%

Tabelle A.2: Tabelle zu Abbildung 6.6

ID	Class	Value
1	person	98%
2	person	89%
3	person	56%
4	person	52%

Tabelle A.3: Tabelle zu Abbildung 6.7

Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.



Ort

Datum

Unterschrift im Original