Fachhochschule Aachen

Analyse und Lösungsansätze einer High-Speed-Bilderkennung bei Kickertischen zur Vorbereitung einer KI-gesteuerten Spielsteuerung

1. Prüfer: Prof. Dr. Philipp Rohde
2. Prüfer: Andre Gilerson

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die Seminararbeit mit dem Thema

Analyse und Lösungsansätze einer High-Speed-Bilderkennung bei Kickertischen zur Vorbereitung einer KI-gesteuerten Spielsteuerung

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Ich verpflichte mich, ein Exemplar der Seminararbeit fünf Jahre aufzubewahren und auf Verlangen dem Prüfungsamt des Fachbereiches Medizintechnik und Technomathematik auszuhändigen.

Name: Jannis De Maré

Aachen, den 21.12.2022

Unterschrift der Studentin / des Studenten

Jannis De Mare

Digital unterschrieben von Jannis De Mare
Datum: 2022.12.21 15:13:39 +01'00'

Kurzfassung

Bilderkennungsalgorithmen, die Objekte in Echtzeit erkennen können, wurden in den letzten Jahren stark erforscht. Diese Arbeit bestimmt den aktuellen Stand der Technologie, um dessen Eignung für eine High-Speed-Bilderkennung bei Kickertischen zu überprüfen. Für die automatische Spielsteuerung mit echten Motoren muss eine Bilderkennung die Position des Balls in unter 4 Millisekunden bestimmen. Aktuelle Deep-Learning-Algorithmen ermöglichen solche Geschwindigkeiten auf neuster Hardware mit einem Kompromiss bei der Genauigkeit. Der YOLOv7 Algorithmus erzielt aufgrund seiner effizienten Architektur bei diesen Geschwindigkeiten die höchste Genauigkeit. Ein Einsatz zusammen mit dem DeepSORT-Algorithmus ist sinnvoll, um auch bei mehreren Bällen gleichzeitig die Bewegung zu verfolgen. Die getrackten Positionsdaten der Bälle können in einem nächsten Schritt als time-series-Daten an ein Machine Learning Modell übergeben werden, um so die Trajektorie möglichst genau zu bestimmen. Mit diesen Informationen kann eine Spielsteuerung Entscheidungen über die Positionierung des Torhüters auf dem Kickerspielfeld treffen und Torschüsse abwehren. Der YOLOv7 Algorithmus bietet zudem Möglichkeiten zum parallelen Tracking der Spielfiguren, wodurch ausreichend Informationen gesammelt werden können, um eine vollautomatische KI-gesteuerte Spielsteuerung zu ermöglichen.

Inhaltsverzeichnis

1. Einleitung

Am Fraunhofer Institut für Produktionstechnologie wird ein Demonstrator zum Thema 5G in der Produktion entwickelt. Für eine gute Veranschaulichung der niedrigen Latenz und hohen Datenrate dieser Technologie, wurde die Idee entwickelt, einen Kickertisch mit einer automatischen Spielsteuerung auszustatten und die Datenübertragung über 5G zu realisieren. So sollen menschliche SpielerInnen die Möglichkeit haben, gegen einen computergesteuerten Gegner anzutreten. Das Spielfeld wird von einer oder mehreren Kameras beobachtet und die generierten Bilddaten schnellstmöglich ausgewertet, um an erster Stelle die Position des Balls zu bestimmen. Diese Anwendung erfordert das Zusammenspiel von verschiedensten State of the Art Lösungen für Bilderkennung, Datenverarbeitung, Motoren, etc. Im Rahmen dieser Arbeit werden die Anforderungen einer solchen Anwendung an eine Bildverarbeitung analysiert und Lösungsansätze für die Objekterkennung eines Kicker-Balls gegeben. Genauer wird untersucht, welcher der aktuell verfügbaren High-Speed-Bilderkennungsalgorithmen für diese Anwendung geeignet ist und wie das Verhältnis von Geschwindigkeit und Genauigkeit ist. Das in den letzten Jahren rasant weiterentwickelte Feld der Bilderkennung macht viele zuvor undenkbare Anwendungen möglich. Diese Arbeit zeigt, was mit dem aktuellen Stand der Technik möglich ist, um darauf basierende Lösungsansätze für eine automatische Spielsteuerung zu entwickeln. Die diskutierten Inhalte finden ebenfalls viel Bedeutung in Bereichen wie dem autonomen Fahren, wo High-Speed-Bilderkennung sich als essenziell erweist. Außerdem können am Fraunhofer IPT erforschte Produktionsketten vom Einsatz dieser Technologien profitieren. Der Aufbau dieser Arbeit beinhaltet zuerst die Grundlagen der historischen und modernen Verfahren für Bild- bzw. Objekterkennung. Anschließend werden aktuell verfügbare Algorithmen verglichen. Im zweiten Kapitel werden die Anforderungen einer automatischen Spielsteuerung an eine High-Speed-Bilderkennung analysiert. Basierend auf den dort gesetzten Zielen, wird ein Lösungsansatz für das zuvor genannte Szenario geboten bzw. nachvollziehbar entwickelt.

2. Grundlagen

In diesem Kapitel werden zuerst die wichtigsten Konzepte, die der Demonstrator veranschaulichen soll, erklärt. Anschließend werden die Grundlagen von klassischer und moderner Bilderkennung veranschaulicht und die wichtigsten Begriffe, die zur Performance-Bewertung benötigt werden, erläutert. Die Objekterkennung ist von künstlicher Intelligenz und Supervised Learning geprägt, deshalb werden diese Themen angeschnitten und grob erklärt, bevor die besten Echtzeit-fähigen Algorithmen gezeigt werden. Zuletzt wird noch kurz erklärt, wie die Lokalisierung bzw. das Tracking von Objekten mit Algorithmen gelöst werden kann.

2.1. Industrie 4.0

Der Schwerpunkt der Forschung am Institut für Produktionstechnologie liegt momentan auf *Industrie 4.0*. So heißt das Zukunftsprojekt, welches die umfassende Digitalisierung und Vernetzung der industriellen Produktion anstrebt.[1] Die vierte industrielle Revolution ermöglicht es, große Produktionsketten zu vernetzen und mit einem zuvor nicht vorhandenen Maß an Flexibilität die Produktion effizienter zu gestalten.[2]

2.2. 5G-Netze in der Produktion

Im Kontext von Industrie 4.0 wird seit einigen Jahren das Potential von 5G-Netzen erforscht. Es ist vor allem bei der Vernetzung von Echtzeitsystemen sehr nützlich, da eine hohe Datenrate und sehr niedrige Latenz möglich sind und das, ohne alle Geräte verkabeln zu müssen. Im industriellen Kontext, wo die Distanzen typischerweise keine 1000 Meter überschreiten, sind Latenzen von unter einer Millisekunde zu erwarten. Datenraten sind bis zu 10 000 Mbit/s möglich. Ebenfalls können auf engem Raum zahlreiche Geräte gleichzeitig betrieben werden, ohne eine Beeinträchtigung der Leistung befürchten zu müssen.[3]

2.3. Bilderkennungsmethoden

Bilderkennung wird aktuell noch stark erforscht. Bei der Wahl von Methoden ist zu berücksichtigen, dass einige Monate später bereits eine performantere Methode verfügbar sein kann. Die modernen Bilderkennungsmethoden basieren alle auf mathematischen Modellen. Digitale Bilder werden als Zahlen gespeichert[4]. So hat jeder Pixel einen Wert, üblicherweise zwischen 0 und 255. Dieser Wert legt die Helligkeit an diesem Punkt fest, bei monochromen Bildern[5] steht 0 i.d.R. für schwarz und 255 für weiß. So können ganze Bilder in Programmen effizient als ein- oder zweidimensionales Array dargestellt werden. Dieses lässt sich auch auf das bekannte RGB-Farbspektrum[6] erweitern, indem Werte für rot, grün und blau zwischen 0 und 255 gespeichert werden. Bilderkennungsmethoden lassen sich aufteilen in klassische

[1] (Fraunhofer IPT)
[2] (SAP)
[3] (Fraunhofer IPT)
[4] (Codeburst)
[5] (Merriam-Webster)
[6] (Pramoditha)

Bildverarbeitungsmethoden und moderne Deep-Learning-Netzwerke[7]. In den folgenden Abschnitten werden die Vor- und Nachteile beider Ansätze gegenübergestellt, um das Verständnis für den State of the Art entwickeln zu können. Anschließend werden die Echtzeit-fähigen Bilderkennungsalgorithmen erwähnt und ihre Funktionsweise erklärt.

2.3.1. Klassische Bilderkennung (OpenCV)

Mit klassischen Bilderkennungsverfahren können Bilder schnell und ohne große Rechenleistung durch Verwenden von Filtern und ähnlichen Verfahren ausgewertet werden. Hierzu wird meist als erstes ein Preprocessing benötigt. Beim Vorverarbeiten wird oft die Helligkeit korrigiert oder der Kontrast angepasst, sodass die Features (dt. Merkmale), die das zu erkennende Objekt ausmachen, sich besser erkennen lassen. Beim nächsten Schritt, dem Feature Extraction, werden Merkmale des zu erkennenden Objektes extrahiert. Dies kann eine Kantenerkennung sein oder aber auch komplexere Transformationen des Bildes (Schnelle Fourier Transformation etc.).[8] Es gibt keine Richtlinien für die richtige Vorverarbeitung oder das Anwenden von Filtern. Es werden Anfangs vernünftige Vermutungen aufgestellt und durch “trial and error” wird oftmals eine gut funktionierende Kombination von Filtern gefunden. In einem konkreten Beispiel konnte am Fraunhofer IPT durch Anwenden eines Threshold-Filters[9] in Kombination mit morphologischen Operationen[10] ein Organ in einem OCT-Scan einer Maus erkannt und so ausgemessen werden.

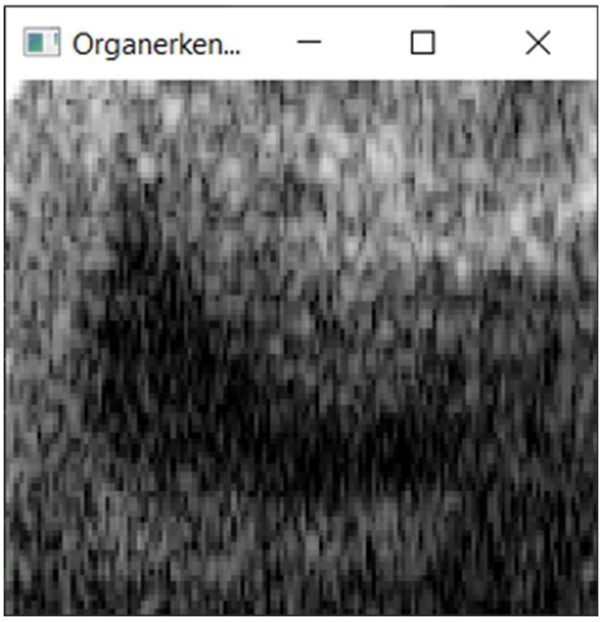

Abbildung 1 OCT-Scan mit Organ einer Maus, unbearbeitet

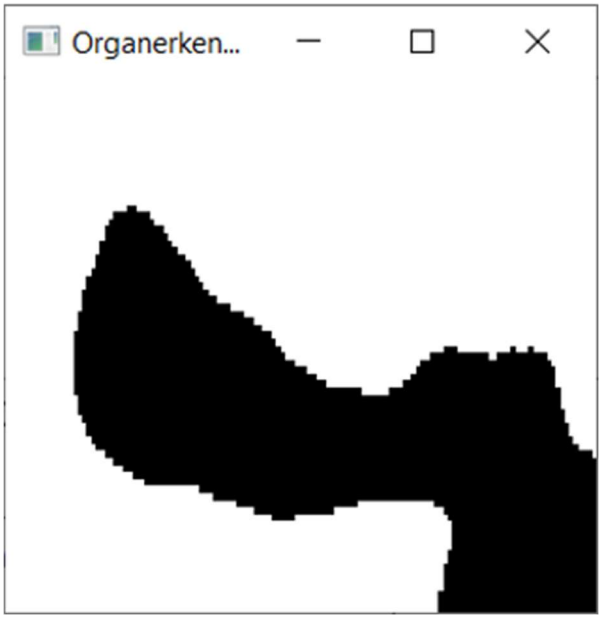

Abbildung 2 OCT-Scan mit angewandten Threshold-Filtern, Organ in schwarzer Fläche

Dieses Programm wurde mit Hilfe von OpenCV realisiert. OpenCV (Open Computer Vision) ist die bekannteste Bibliothek für Bildverarbeitung und Computer Vision. Sie ist verfügbar für die

[7] (Boesch)
[8] (Mallick)
[9] (Tutorialspoint)
[10] (OpenCV)

Programmiersprachen C, C++, Python und Java. Die Vorteile bei der Umsetzung einer Bilderkennung mit traditionellen Methoden sind die gute Performance und einfache Umsetzung. Es werden keine historischen Bilddaten mit Annotationen benötigt, wie um ein neuronales Netz zu trainieren. Allerdings sind diese Methoden sehr fehleranfällig in komplexeren Szenarien. Für dieses Problem existiert keine direkte Lösung, denn wenn ein Programm geschrieben wurde, bspw. um die runde Form von einem Ball zu erkennen, wird dieses Programm den Ball nicht mehr erkennen können, sobald auch nur ein kleiner Teil des Balls verdeckt ist. Schatten und andere Lichtverhältnisse, die nicht bei der Programmierung berücksichtigt wurden, verursachen ebenfalls Ungenauigkeiten. Diese Probleme sind nur in Verbindung mit erheblichem Aufwand zu umgehen. Bei Deep-Learning-basierten Verfahren lassen sich solche Fälle deutlich besser berücksichtigen.

2.3.2. Künstliche Intelligenz in Bilderkennung

Deep Learning basierte Bilderkennungsverfahren sind seit 2014 State of the Art. Deep Learning ist eine Machine Learning Klasse, die in den letzten Jahren stark an Popularität gewann. Die meisten Deep-Learning-Architekturen basieren auf neuronalen Netzen, die die Funktionsweise des menschlichen Gehirns imitieren. Dieser und folgende Abschnitte orientieren sich, sofern nicht anders angezeigt, am Buch “Deep Learning in Object Detection and Recognition”. Ein Ausschnitt beschreibt die grundsätzliche Funktionsweise von künstlichen neuronalen Netzen (engl. **A**rtificial **N**eural **N**etwork (ANN)) unabhängig von Bilderkennungsanwendungen:

> *The earliest models of ANNs are simple linear models and associated the output value y with the input x. In other words, these models want to learn a function given a training set of samples, in which each sample is a pair of an input value and output value. For instance, the perceptron, proposed in 1957 [45], learns a linear model in a supervised way, which can be represented as f (x, w) = wx + b. This mathematical model mimics the operation of neurons in humans’ brain. Therefore, deep learning consisting of many perceptrons can also be considered as a generalization of a linear or logistic regression and imitate the functions of brains.* (Pang et al. 2)

Jedes Perzeptron hat Gewichte, die während des Trainingsprozess angepasst werden, um den kleinstmöglichen **M**ean **S**quared **E**rror (MSE) zu erzielen[11]. Der MSE beschreibt die durchschnittliche quadrierte Abweichung der vorhergesagten Daten zu den tatsächlichen Werten.

$$MSE = \frac{\sum(y_i - \hat{y}_i)^2}{n}$$

Formel zur Berechnung des Mean Squared Error

Perzeptronen finden im 2-dimensionalen Raum die Gerade mit dem geringsten MSE. In höheren Dimensionen werden sie zur Hyperebene.[12]

[11] (Frost)
[12] (Bialonski)

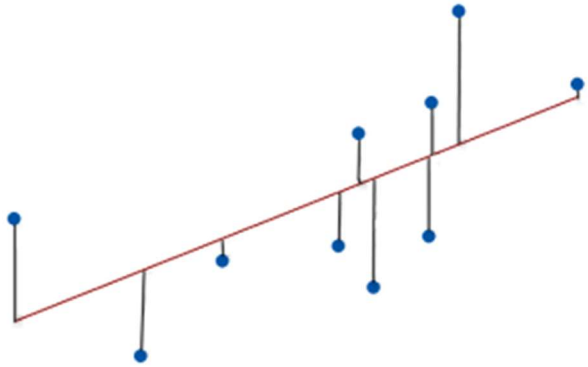

Abbildung 3 Mean Squared Error in linearer Regression

Neuronale Netze bestehen aus mehreren Schichten (engl. Layers) der Perzeptronen, die alle mit den Perzeptronen der nächsten Schicht vernetzt sind[13]. Dabei gibt es einen Input-Layer und einen Output-Layer mit mehreren versteckten Schichten dazwischen. Eine Visualisierung der Struktur verdeutlicht, wieso es als Netz bezeichnet wird.

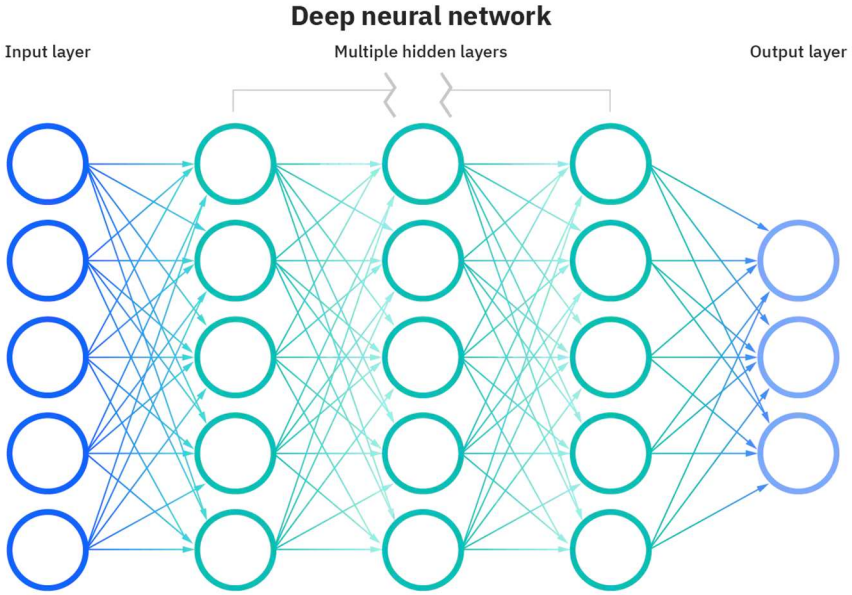

Abbildung 4 schichtenartiger Aufbau von Deep Neural Networks

Diese Struktur ist sehr komplex, aber ein trainiertes neuronales Netz kann Daten mit einer so hohen Geschwindigkeit auswerten, dass nur ein Bruchteil der von Menschen benötigten Zeit zur Auswertung benötigt wird. Sie sind außerdem gegen die Schwachstellen von klassischer Bilderkennung (Okklusion, veränderte Belichtung, etc.) deutlich resistenter und werden daher als erste Wahl für die meisten fortgeschrittenen Bilderkennungs-Anwendungen angesehen. Der stärkste zurückhaltende Faktor von Deep-Learning-Algorithmen ist aktuell die verfügbare Rechenleistung[14]. Da diese mit rasantem Tempo anwächst, werden sie ebenfalls immer leistungsfähiger. Um solche zuverlässigen Algorithmen zu erhalten, werden große Mengen

[13] (IBM)
[14] (Boesch)

Trainingsdaten benötigt, an denen der Algorithmus lernen kann, welche Objekte zu erkennen sind.

2.3.2.1. Supervised Learning

Zur Objekterkennung mit neuronalen Netzen wird Supervised Learning genutzt. Im Gegensatz zum Unsupervised Learning wird dem Algorithmus die Information gegeben, was in den gelieferten Daten zu erkennen ist. Diese Art eignet sich bestens, um mit neuronalen Netzen Vorhersagen treffen zu können. Unsupervised Learning hingegen eignet sich, um bspw. ohne weiteren Kontext aus Daten das Kaufverhalten von Nutzern auf Webseiten zu beschreiben und so Marketingkampagnen entwickeln zu können.[15]

2.3.3. Objekterkennung

Beim Supervised Learning für Bilderkennung bestehen die Daten aus Bildern, die alle einzeln gelabelt wurden. Labels sind Informationen zu den Objekten, die in einem Bild zu erkennen sind[16]. Sie werden meist in einer separaten Textdatei oder als Metadaten gespeichert. Diese Informationen können einfache Namen zu Objekten sein oder zusätzlich Positionsdaten enthalten.

Abbildung 5 Ein trainierter Algorithmus erkennt Objekte und deren Position in einem Bild. Zum Training müssen diese Annotationen erst von Menschen angebracht werden, bevor ein Algorithmus sie erkennen kann.

Der Prozess des Labelings ist zeit- und kostenintensiv. Wenn ein Algorithmus mit ausreichend gelabelten Bildern gefüttert wird, lernt er, die Merkmale der gelabelten Objekte zu erkennen und

[15] (Pang et al. 156)
[16] (Sydorenko)

kann mit hoher Genauigkeit Vorhersagen über deren Anwesenheit oder sogar Position treffen. Dies bietet ein enormes Potential zur Automatisierung von unzähligen Anwendungen. Um Bilderkennung allgemein zugänglich zu machen, haben Firmen Datensätze mit Millionen gelabelter Objekte erstellt (z.B. Microsoft mit dem COCO Dataset). Dadurch können Netze bereits vortrainiert werden, sodass anschließend nur noch eine geringe Menge Daten notwendig ist, um ein spezielles Objekt erkennen zu können.[17]

2.3.3.1. Algorithmen im Vergleich

Bei der Verwendung von neuronalen Netzen gibt es 2 prominente Ansätze. One-stage- und two-stage-Detektoren. Bei den zweistufigen Methoden werden die zu erkennenden Objekte bzw. deren Merkmale erst extrahiert und anschließend klassifiziert. Das bedeutet, dass die Objekte erkannt werden und ihre Größe erst danach approximiert wird. Diese Methoden benutzen ein Region-Proposal, bei dem mit konventionellen Bilderkennungsmethoden Regionen vorgeschlagen werden (oft ca. 2000 Regionen), in denen sich mit erhöhter Wahrscheinlichkeit Objekte befinden könnten. R-CNN (**R**egion-Based **C**onvolutional-**N**eural-**N**etwork), “Fast RCNN” und “Faster RCNN” haben sich dank hoher Genauigkeit als State of the Art in der Bilderkennung etabliert.[18] Diese sind aufgrund ihres Aufbaus relativ langsam. Das R-CNN braucht 49 Sekunden für die Auswertung eines einzelnen Bilds.[19] Mit dem Faster RCNN ist die Reduzierung auf 0.2 Sekunden zwar bereits drastisch, aber noch kein Vergleich zu einstufigen Methoden.[20] Diese treffen Vorhersagen für die Position und Klasse aller Objekte gleichzeitig. So schaffen bekannte Algorithmen, wie OverFeat, YOLO, SSD und RetinaNet deutlich höhere Geschwindigkeiten auf Kosten der Genauigkeit.

R-CNN	49 Sekunden
Faster RCNN	0.2 Sekunden
YOLO-Algorithmus	6.2 Millisekunden

Um im Weiteren die Genauigkeit der Algorithmen im Verhältnis zur Geschwindigkeit betrachten zu können, sollten einige Begriffe noch genauer erläutert werden.

- **Bounding Box** ist eine meist rechteckige Fläche, die die Positions eines Objekts markiert.
- **Intersection of Union (IoU)**[21] ist ein Wert zwischen 0 und 1, der die Überlappung der vorhergesagten Boundingbox und der tatsächlichen Bounding Box beschreibt. Eine vollständige Überlappung erhält somit den Wert 1. Wenn die Boxen sich gar nicht berühren, ist der Wert 0.

[17] (Pang et al. 2-5)
[18] (Boesch)
[19] (Girshick)
[20] (Ren)
[21] (Anwar)

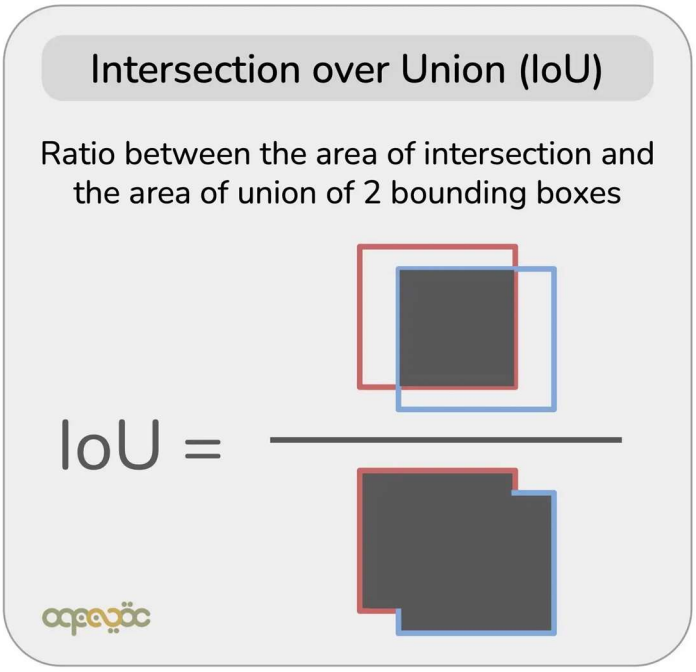

Abbildung 6 Intersection of Union (IoU) ist der Wert für die Überlappung der Vorhersage mit der tatsächlichen Objektposition

- **True Positive:** Die vorhergesagte Klasse eines Objekts innerhalb einer Bounding Box stimmt mit dem tatsächlichen Objekt überein. Ebenfalls muss die IoU einen zu wählenden Schwellenwert erreichen oder überschreiten, um als positiv erkannt zu werden.
- **False Positive:** Ein Objekt wurde vorhergesagt, allerdings ist dieses dort nicht vorhanden, ein anderes Objekt ist an dieser Stelle (Bsp. Prediction = Cat, Picture = Dog) oder die IoU ist zu klein.
- **False Negative:** Ein Objekt wurde an einer Stelle nicht erkannt, obwohl es erkannt werden sollte.
- **True Negative:** Ein Objekt wurde nicht erkannt, weil es nichts zu erkennen gab.
- $$Precision = \frac{Correct\ Predictions}{Total\ Predictions} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$
- **Klassen:** Name der Objekte, die erkannt werden sollen.
- **Mean average Precision:** Der Durchschnitt aller durchschnittlichen Genauigkeiten der Klassen.

2.3.3.2. Real-Time Algorithmen

Echtzeit-fähige Bilderkennungsalgorithmen widmen sich unter anderem durch das autonome Fahren einer der kritischsten Aufgaben des Themenfeldes. Die Anforderungen an die Genauigkeit und Geschwindigkeit sind verhältnismäßig hoch. Deshalb ist der Fortschritt in den letzten zwei Jahren in diesem Bereich stark vorangetrieben worden. Eine Übersicht der Average Precision von performanten Algorithmen der letzten 2 Jahre ist in Abbildung 7 zu sehen.

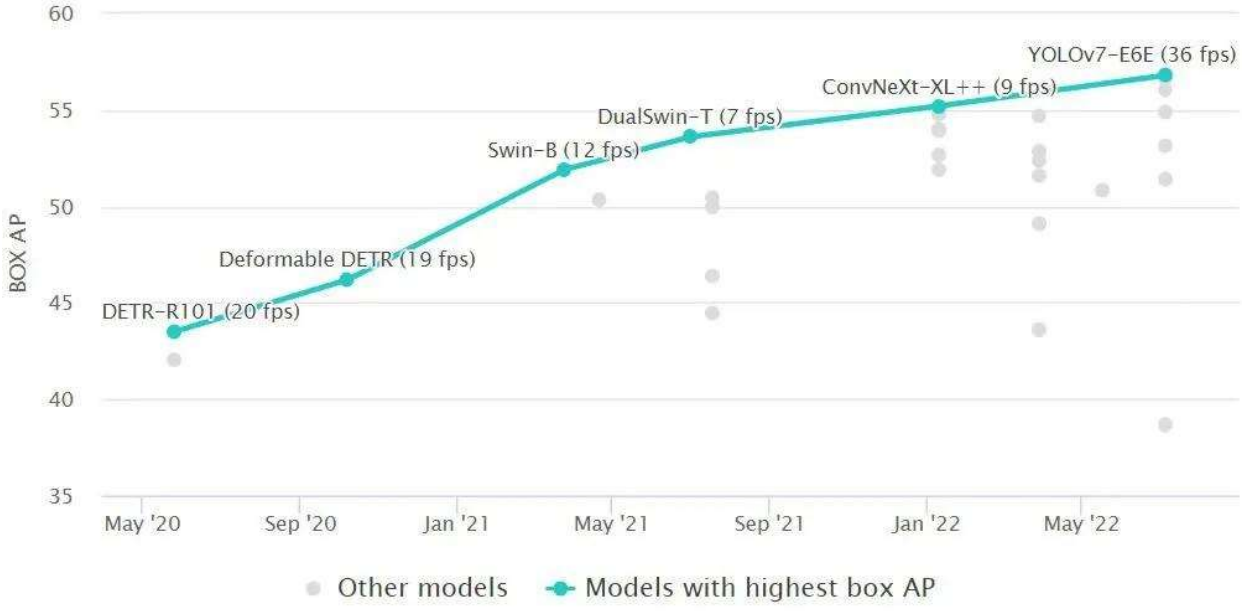

Abbildung 7 Der Anstieg der Genauigkeit von Echtzeit-Bilderkennungsalgorithmen in den letzten zwei Jahren. Benchmark auf COCO-Dataset.

2.3.3.3. YOLOv7-Algorithmus[22]

Im Folgenden wird der Fokus auf den YOLO-Algorithmus gelegt, da er sich als den mit Abstand schnellsten und dafür verhältnismäßig genauesten Algorithmus differenziert hat.[23] Dies wird nochmals in Abbildung 8 sichtbar. **Y**ou **O**nly **L**ook **O**nce ist ein einstufiger Bilderkennungsalgorithmus, der seit 2016 immer weiterentwickelt wurde. Er ist besonders beliebt wegen seinem besonders guten Verhältnis von Genauigkeit zu Geschwindigkeit. Er wird hier genauer als andere Algorithmen betrachtet, da er den State of the Art repräsentiert und sich im Verlauf dieser Arbeit als besonders geeignet erweisen wird. Am 6. Juli 2022 wurden das wissenschaftliche Paper und der Open-Source Code für die Version 7 (YOLOv7) veröffentlicht. Der standardmäßige YOLOv7 Algorithmus weist bei der Evaluation am MS COCO min-val Subset eine Genauigkeit von 51.2% mit einer Auswertungsdauer von 6.2 Millisekunden auf. Diese Teilmenge des Datensatzes wird benutzt, um die Leistung von Objekterkennungsalgorithmen auszuwerten. Die Genauigkeit liegt zwar “nur” bei knapp 50%, allerdings ist zu beachten, dass dieser Datensatz einen hohen Schwierigkeitsgrad hat. Eine Average Precision von 50% wurde erstmals 2019 erreicht und bis heute konnte noch kein Algorithmus mehr als 65% erreichen, wobei die meisten der genauen Algorithmen keine Echtzeit-Fähigkeit beabsichtigen. Ein genaueres Bild der zu erwartenden Genauigkeit liefert eine Studie, in der YOLOv3 Menschen mit einer **T**rue **P**ositive **R**ate (TPR) von 95% erkennen konnte und das, obwohl das Modell nur mit Frontalansichten von Menschen trainiert wurde.[24] Die Falschklassifizierungsrate (**F**alse **P**ositive **R**ate oder FPR) lag bei gerade mal 0.2%. Die Auswertungsdauer von 6.2 ms ist das Äquivalent von 161 Bildern pro Sekunde (FPS). Dies schafft er dank einer Reduzierung der benötigten Parameter auf 36.9 Millionen im Vergleich zu vorherigen Versionen. Trotz weniger Parametern ist die Genauigkeit weiterhin angestiegen.

[22] (Wang)
[23] (Tan)
[24] (Ahmad)

Ebenfalls zu erwähnen ist, dass der YOLOv7 Algorithmus in der Regel Bilder mit einer Größe von 640 x 640 Pixel verarbeitet.

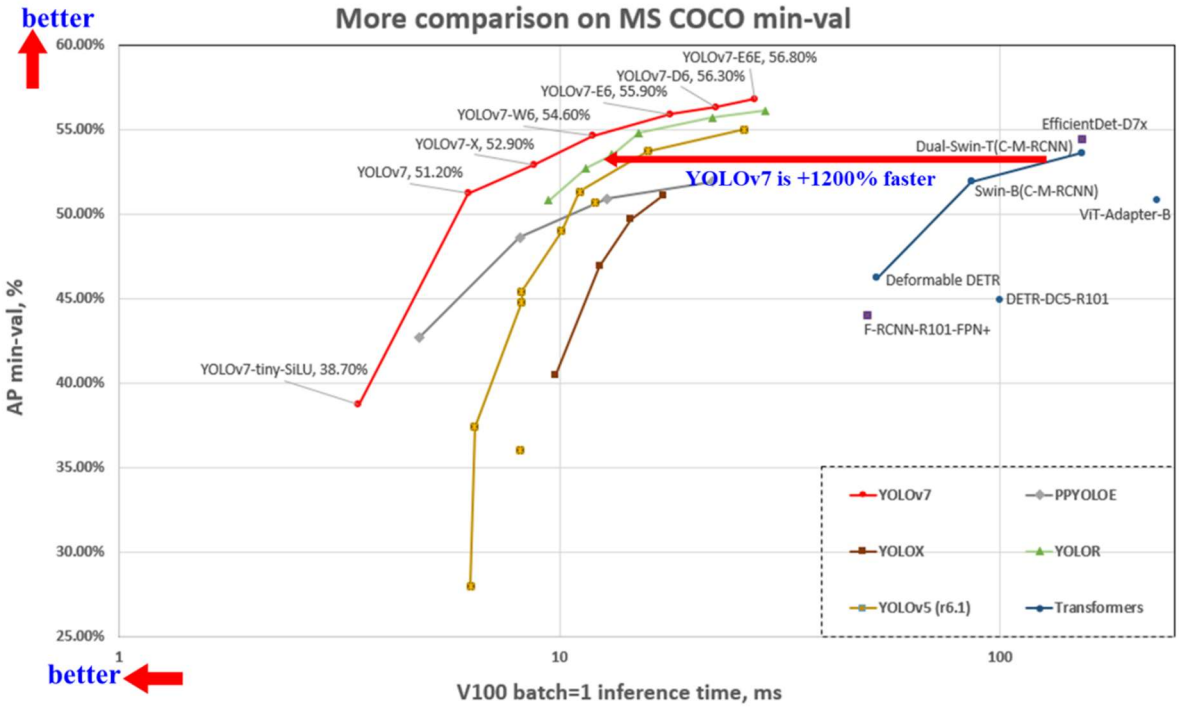

Abbildung 8 YOLOv7 im Vergleich zu anderen Algorithmen. Auswertungsdauer gegenüber durchschnittlicher Genauigkeit.

Die Version 7 beweist eine 12 mal bessere Performance als Transformator-Modelle und erreicht dennoch eine höhere Genauigkeit. Sie bietet außerdem noch weitere, für verschiedene Anwendungsfälle angepasste Modelle. YOLOv7-tiny ist beispielsweise dank noch geringerer Parameteranzahl für Edge AI, d.h. mobile Rechner, optimiert[25]. Es wird zwar nur noch eine durchschnittliche Genauigkeit von 38.7% erreicht, allerdings sind damit Geschwindigkeiten von 286 FPS möglich. Eine detaillierte Erklärung der YOLO Architektur würde den Rahmen dieser Arbeit sprengen, allerdings ist eine high-level Übersicht bereits ausreichend, um eine Entscheidung für den bestgeeigneten Algorithmus treffen zu können.

[25] (Yeung)

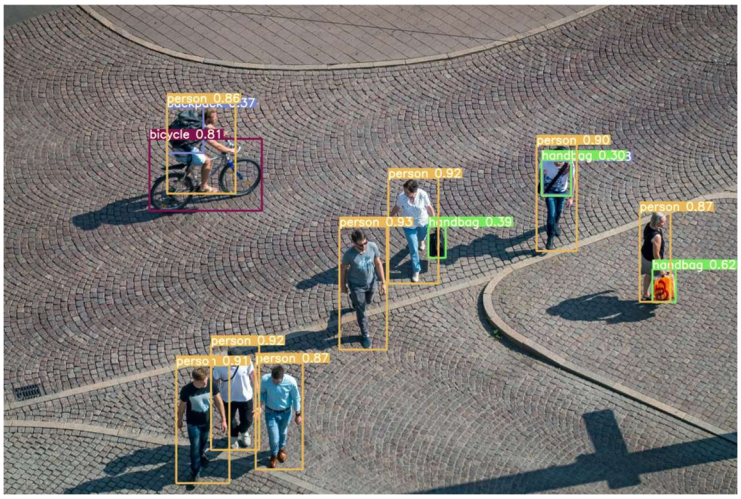

Abbildung 9 YOLOv7 erkennt Fußgänger und Fahrradfahrer

YOLO ist vor allem für sein Backbone, das E-ELAN genannt wird, bekannt. Das **E**xtended **E**fficient **L**ayer **A**ggregation **N**etwork ermöglicht das kontinuierliche Lernen, ohne den originalen Lernpfad zu zerstören. Dies ist bei vielen Algorithmen ein historisches Problem, da durch neu gelernte Informationen alte zerstört werden können. Außerdem implementiert die neueste Version ein effizientes Compound Model Scaling, das eine Anpassung an verschiedene Anwendungsfälle ermöglicht.[26]

2.4. Objektlokalisierung

Die vom YOLO Algorithmus erkannten Bounding Boxes liefern Positionsdaten. Der **S**imple **O**nline and **R**ealtime **T**racking (SORT) Algorithmus wurde entwickelt, um mit den von YOLO gelieferten Daten Personen zu tracken. Hierfür werden ähnliche Bounding Boxes von einem Bild zum nächsten mit Hilfe eines Kalman Filters “gematched”, sodass jede Person eine ID erhält und die Trajektorie bzw. der Gehweg aufgezeichnet werden kann. Dieses Verfahren lässt sich ebenfalls auf andere Objekte ausweiten, da nur die Positionsdaten benötigt werden.[27] Wenn eine Person allerdings kurzzeitig nicht zu sehen ist, erhält sie eine neue ID. Dieses Problem wird von DeepSORT durch Verwenden eines KI-Modells, das Ähnlichkeiten der Personen vergleicht, gelöst.

[26] (Boesch)
[27] (Yang)

3. Analyse und Lösungsentwurf

Damit das Fraunhofer IPT einen Demonstrator entwickeln kann, der die Konzepte von Industrie 4.0 und 5G in der Produktion anschaulich darstellen kann, wurde die Idee des automatisierten Kickertischs entwickelt. Menschliche SpielerInnen, bzw. ein Team, sollen die Möglichkeit haben, gegen einen computergesteuerten Gegner anzutreten. Dieser Teil der Arbeit analysiert die zur Umsetzung entstehenden Anforderungen mit Fokus auf die Anforderungen der Bilderkennung (zur Positionsbestimmung des Balls), um darauf aufbauend die im ersten Kapitel beschriebenen Methoden in einem vollständigen Konzept zu verwerten. Das Ziel des entstandenen Entwurfs ist es, die Umsetzung einer High-Speed-Bilderkennung bestmöglich vorzubereiten. Hierfür eignet sich eine Bottom-Up-Herangehensweise, da die benötigte Geschwindigkeit der Positionsbestimmung einen direkten Einfluss auf die Wahl der Algorithmen hat und die Anforderungen der Bildverarbeitung einen Einfluss auf die Wahl der Kamera haben.

3.1. Zielsetzung

Da das Fraunhofer IPT beim Einsatz von 5G bereits viel Erfahrung gesammelt hat (siehe “5G Industry Campus Europe[28] Research Infrastructure), will es seine Expertise demonstrieren.[29] Aus diesem Grund wird ein Demonstrator entwickelt, der 5G während einer echtzeitkritischen Anwendung implementiert. Dies bietet eine hervorragende Gelegenheit für Kunden, Partner und andere Interessenten, ohne tieferes technisches Verständnis einen Einblick in die Möglichkeiten der erforschten Technologie erhalten. Gute Demonstratoren wecken in erster Linie Interesse und schaffen es gleichzeitig, größere Konzepte anschaulich darzustellen. Dies ist der Fokus bei der Definition der Anforderungen.

3.2. Anforderungsanalyse

Um das übergeordnete Ziel eines anschaulichen und spannenden Demonstrators zu erreichen, sollte der Kickertisch in physikalischer Form vorhanden sein und nicht simuliert werden. Die Kicker-Figuren des nicht-menschlichen Teams sollen von Motoren bewegt werden. Dadurch wird die Komplexität des Demonstrators um einige Herausforderungen erweitert. Die Umsetzungsschritte dieses gesamten Projektes lassen sich gut voneinander trennen:

- **Entwicklung eines Tracking-Systems, das die Position des Balls bestimmen kann**
- Programmierung einer zuverlässigen Positionsvorhersage für den Ball
- Anschließen und Programmieren von Motoren
- Programmierung und ggfs. Training einer künstlichen Intelligenz, die entscheidet, wie die Spieler bewegt werden müssen um möglichst wenige Schüsse ins eigene Tor zu lassen
- Zukünftig Weiterentwicklung der Taktiken der KI, um auch selber Punkte erzielen zu können und nicht nur Bälle abzuwehren

Da jede dieser Aufgaben einen erheblichen Arbeitsaufwand erfordert, beschreibt diese Seminararbeit die Analyse der Echtzeit-Lokalisierung des Balls, das ‘Tracking-System’, und die

[28] (Fraunhofer IPT)
[29] (Scheiter)

daraus entwickelten Lösungsansätze. Ziel dieser Arbeit ist es, ein Verständnis der benötigten und verwendeten Technologien zu schaffen, um Entscheidungen für die bestmögliche Umsetzung des Systems treffen zu können.

3.2.1. Allgemeine Anforderungen

Einige allgemeine Anforderungen an das Projekt wurden von den Projektleitern gegeben, woraus sich genauere Anforderungen ableiten lassen. Diese sind: (1) ein Kickertisch soll durch Anschließen von Motoren automatisiert werden; (2) eine oder potentiell zwei Kameras sollen angeschlossen werden; (3) die Bilder sollen schnellstmöglich ausgewertet werden, um die Position des Balls zu bestimmen; (4) die Ballposition und seine Trajektorie sollen über ein 5G-Modul an einen anderen Rechner übertragen werden; (5) der andere Rechner steuert die Motoren der Spielstangen. Ebenfalls sollten, abgesehen vom externen Eingriff durch nicht-menschliche “Spieler”, keine Spielregeln verletzt werden.[30] Diese Arbeit behandelt ausschließlich Punkt (3). Für eine zielführende Analyse der schnellstmöglichen Bildauswertung und Positionsbestimmung mussten die Anforderungen an diese Teilaufgabe nochmals verfeinert werden.

3.2.2. Resultierende Anforderungen

Im Austausch mit Wissenschaftlern und Experten der Felder Bilderkennung und Automatisierung am Fraunhofer IPT wurde entschieden, dass klassische Bildverarbeitungsmethoden (bspw. mit OpenCV) zu fehleranfällig sind für die erwartete Spielumgebung. Hauptgrund für diese Entscheidung ist die nicht vorhandene Flexibilität beim Einsatz. Weitere Gründe wurden bereits im Kapitel Klassische Bilderkennung beschrieben und werden nochmals im Entwurf begutachtet. Stattdessen sollten Deep-Learning-Methoden eingesetzt werden. Die maximale Dauer der Bildverarbeitung lässt sich ebenfalls abschätzen, wodurch viele Algorithmen aus der Betrachtung herausfallen. Möglichst viele Schüsse auf das Tor abzufangen, ist das erste Ziel der Automatisierung, daher wird das Worstcase-Szenario für den Torhüter bzw. für Bildauswertung und Spielsteuerung betrachtet.

3.2.2.1. Berechnung der optimalen Auswertungsdauer

Die kürzeste Reaktionszeit, die der Torhüter zum Anpassen seiner Position hat, lässt sich aus dem Abstand zu anderen Spielfiguren und der höchstmöglichen Geschwindigkeit berechnen. Der Abstand zwischen den einzelnen Spielstangen ist auf minimal 14 cm und maximal 15 cm genormt.

[30] (foosball)

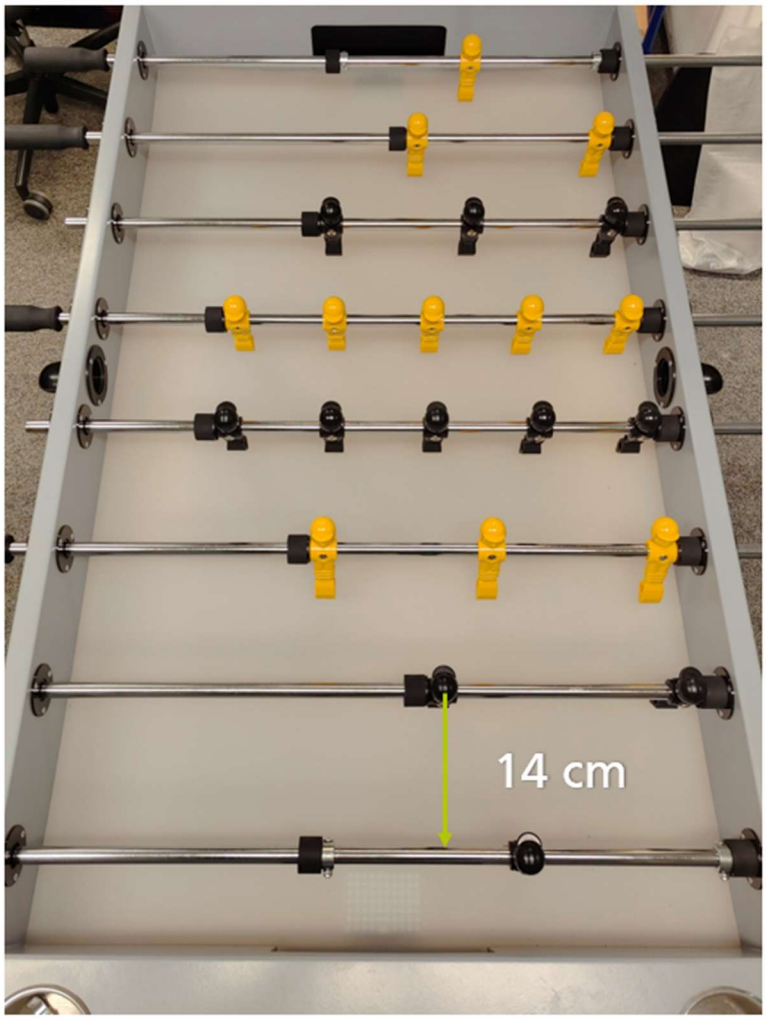

Abbildung 10 Dimensionen eines Kicktischs

Die nächste gegnerische Spielstange liegt somit zwar mindestens 28 cm vom Torwart entfernt, allerdings ist es zu erwarten, dass in vereinzelten Fällen ein Schuss von einem der eigenen, nur 14 cm entfernten Abwehrspieler abgelenkt wird. Beim sogenannten Snake Shot können Geschwindigkeiten von bis zu 60 km/h, also 16.67 m/s, erreicht werden[31]. Somit lässt sich die Zeit vom Ablenken des Schusses bis zur Ankunft beim Torwart berechnen:

$$\frac{14\ cm}{1667\ cm/s} = 0.0083983\ s\ =\ 8.3\ ms$$

Die Übertragung der Daten über 5G ist dank extrem niedriger Latenz (vergleichbar mit Ethernet-Kabeln) vernachlässigbar, allerdings sollte für die Ansteuerung der Motoren möglichst viel Zeit übrigbleiben. Aus diesem Grund wurde eine Zeit von 4 ms als Zielwert für die Bildaufnahme plus Bildauswertung festgelegt. Es ist zu vermuten, dass das Szenario, in dem diese Reaktionszeit notwendig ist, nicht häufig vorkommen wird. Dennoch ist das Ziel, den allgemeinen State of the Art und somit die höchstmöglichen Reaktionszeiten zu demonstrieren. Wieso ein Zielwert von 4 ms besonders ambitioniert ist, wird im Laufe des Entwurfs deutlich.

[31] (Koopmann)

Allerdings würde bereits eine Verarbeitungsdauer von 20 ms die durchschnittliche menschliche Reaktionszeit von durchschnittlich 100-250 ms bei weitem übertreffen.[32]

3.2.3. Erwähnung Optitrack Kameras

Anschließend ist noch zu erwähnen, dass es bereits High-Speed-Kameras mit eingebauten Tracking-Möglichkeiten gibt, die genutzt werden, um in Filmen Special-Effects in eine 3-dimensionale Umgebung einzubauen. Diese wurden bereits von verschiedenen Internet-Persönlichkeiten wie umfunktioniert, um Bälle zu lokalisieren.[33] Allerdings bringt dies Einschränkungen mit sich. Eine Verwendung dieser Kameras würde bedeuten, dass das Spiel abgeändert werden muss, indem an den Bällen reflektierende Punkte angebracht werden. Ebenfalls würde dies den Zweck des Demonstrators verfehlen, da der Preis solcher Kameras hoch und die Einbindung umständlich ist, sodass der Bezug zu Industrieanwendungen verloren gehen könnte. Somit wird vorzugsweise auf ein Tracking-System mit Open-Source Einbindungsmöglichkeiten gesetzt. Ebenfalls wird das Know-How zur Problemlösung der Fraunhofer Gesellschaft nochmals bewiesen.

3.3. Entwurf eines Lösungskonzepts

Da diese Arbeit das Ziel verfolgt, Lösungsansätze für die High-Speed-Bilderkennung zu entwickeln, werden diese in den folgenden Abschnitten vorgestellt. Die Priorität liegt hierbei selbstverständlich auf der Geschwindigkeit der Bilderkennung. Nichtsdestotrotz spielen auch Benutzerfreundlichkeit, bzw. Ease of Use, und Ausbaufähigkeit eine ausschlaggebende Rolle. Der automatische Torhüter stellt den ersten Meilenstein dar, wobei die vollautomatische Spielsteuerung das langfristige Ziel ist.

3.3.1. Vergleich Bilderkennung: klassisch vs. Deep Learning basiert

Klassische Bilderkennungsmethoden basieren auf Mustererkennung zur Klassifizierung von Objekten. Die Parameter der verwendeten Filter werden so lange angepasst, bis dass eine hohe Genauigkeit bei der Erkennung erzielt werden kann. Allerdings sind diese Filter sehr fehleranfällig, wenn z.B. die Lichtverhältnisse von der ursprünglichen Trainingsumgebung abweichen. Da Demonstratoren oft an unterschiedlichen Orten präsentiert werden, ist davon auszugehen, dass die Lichtverhältnisse sehr variabel sind. Möglicherweise könnte das Licht plötzlich härter sein, wodurch mehr Schatten geworfen werden. Andernfalls könnte auch ein menschlicher Spieler einen Schatten werfen, wodurch eine klassische Bildverarbeitung stark verwirrt oder unbrauchbar wird. Ebenfalls ist Okklusion (teilweise verdeckte Objekte) ein übliches Problem. Wenn der Ball beispielsweise von einer der Spielstangen verdeckt wird, dann ist evtl. nur noch der obere Rand erkennbar. Eine klassische Bildverarbeitung scheitert hier, da der Ball in dem Fall nicht mehr rund, sondern mondförmig ist. All diese Probleme wurden in der Vergangenheit schon, durch aufwändige Erweiterungen der Algorithmen, von Computer Vision Experten gelöst. Allerdings bietet die Deep-Learning-basierte Bilderkennung effizientere und zuverlässigere Methoden. Das Training der neuronalen Netze kann zwar ebenfalls aufwändig sein, allerdings bleibt das Troubleshooting der Mustererkennung erspart und es bleibt

[32] (Human Benchmark)
[33] (Stuff Made Here)

anschließend ein ebenso performanter und zuverlässiger Algorithmus, der außerdem noch leicht erweitert werden kann.

3.3.2. Performante Deep Learning Modelle im Vergleich

Bei der Betrachtung der State of the Art Deep-Learning-Modelle der letzten 9 Jahre, sticht ein Algorithmus dank hoher Geschwindigkeit und dennoch hoher Genauigkeit besonders hervor. Der YOLOv7 Algorithmus kann 161 Bilder pro Sekunde verarbeiten mit einer durchschnittlichen Genauigkeit von 51.2%. Das sind 6.2 ms pro Bild. Der Zielwert für die Bildauswertung wurde auf 4 ms festgelegt, daher ist der Großteil der anderen Algorithmen nicht geeignet. Eine Auswertungsdauer mit herkömmlichen Grafikkarten (GPU) von unter 4 ms ist nur mit dem YOLOv7-tiny Modell erreichbar (286 FPS = 3.5 ms). Dieses ist für Edge-GPUs entworfen worden und ist dank weniger Parametern etwas “leichter” als die anderen YOLOv7 Modelle. Ein Blick auf die Grafik und die Tabellen des YOLOv7-Papers lassen die Überlegenheit dieser Modelle nicht weiter in Frage stellen.

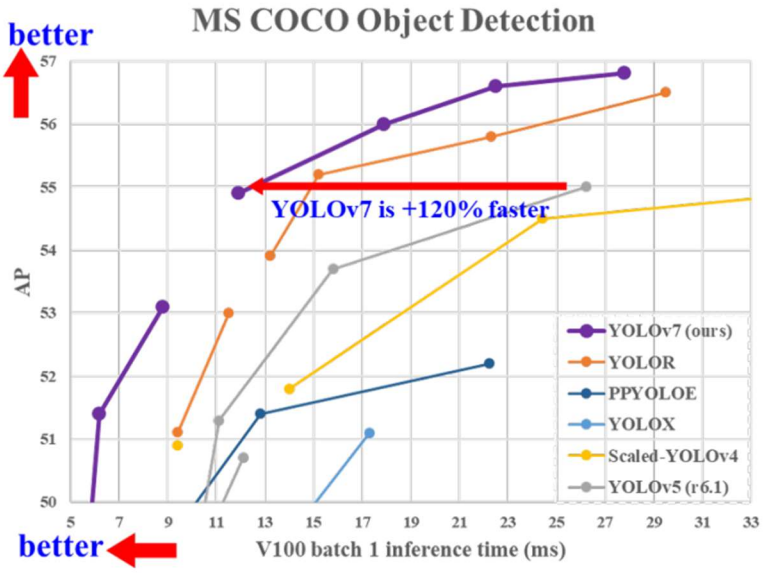

Abbildung 11 YOLOv7 im Vergleich zu anderen real-time Objekterkennungsalgorithmen deutlich überlegen

Jeder Algorithmus mit höherer Genauigkeit ist langsamer als YOLOv7. Genauso sind die wenigen Algorithmen mit höherer Geschwindigkeit deutlich ungenauer. Es ist deutlich, dass YOLOv7 die aktuell besten Eigenschaften für die Umsetzung einer High-Speed Bilderkennung bietet. Die Benutzerfreundlichkeit ist ebenfalls gegeben. Das bereitgestellte GitHub-Repository liefert am MS COCO Datensatz vortrainierte Gewichte. Diese können mit im Terminal ausführbaren Python-Skripten auf individuelle Anwendungen trainiert werden. Dies bedeutet, dass nur eine verhältnismäßig geringe Menge gelabelter Bilder notwendig sind, um eine für einen bestimmten Anwendungsfall angepasste Bilderkennung zu trainieren. Natürlich gilt hier, dass mehr Daten besser sind, aber bereits wenige Bilder können reichen. Der YOLOv7-tiny-Algorithmus ist somit bis auf weiteres der bestgeeignete Algorithmus für die Bilderkennung im Fall des Kickertischs. Bibliotheken wie OpenCV bieten Möglichkeiten, die trainierten Gewichte von neuronalen Netzen zu importieren, um mit diesen ohne großen Aufwand eine eigene Computer-Vision-Anwendung erstellen zu können. Eine genauere Aussage über zu wählende technische Details bei der Umsetzung ist erst während einer (evtl. testweise simulierten) Umsetzung möglich.

3.3.3. Positionsbestimmung

Die Positionsbestimmung des Balls wird größtenteils über die Position der Bounding Boxes vom YOLOv7 Algorithmus übernommen. Das Zentrum einer Bounding Box wird aufgrund der runden Form von Bällen auch als das Zentrum des Kickerballs erwartet. Sobald der Ball an zwei unterschiedlichen Positionen nacheinander erkannt wurde, ist eine Aussage über seine Geschwindigkeit und Richtung möglich. Um kleine Fehler in der Bilderkennung auszuschließen, können auch noch mehr Positionen einbezogen werden, um eine genauere Aussage über das Ziel des Balls treffen zu können. Im Austausch mit Machine Learning Experten erwies sich der Einsatz von Machine Learning Methoden als äußerst sinnvoll. So könnten die Positionsdaten als Time-Series-Daten an ein Machine-Learning-Modell übergeben werden. Wenn ausreichend Positionsdaten gesammelt wurden, kann dieses Modell lernen, wo die Grenzen des Spielfeldes sind und mit welcher Geschwindigkeit und Richtung ein Ball an der Wand abprallen wird. Dies könnte sich als äußerst nützlich erweisen. Weiterhin bieten SORT und vor allem DeepSORT Möglichkeiten, mehrere Bälle zur gleichen Zeit zu verfolgen und als einzelne Entitäten zu betrachten. So kann die Trajektorie jedes Balls aufgezeichnet werden, auch wenn ein Ball kurzzeitig verdeckt ist. DeepSORT verwendet ein KI-Modell, um Ähnlichkeiten von Objekten zu vergleichen, daher wäre es sinnvoll, den Bällen ein unterschiedliches Aussehen zu geben. Dies kann beispielsweise durch Verwenden unterschiedlicher Farben erreicht werden. Falls DeepSORT sich als ungeeignet erweist, kann auch durch einen simplen Einsatz von OpenCV Methoden die Farbe der unterschiedlichen Bälle durch Benutzung der Positionsdaten ausgelesen und verfolgt werden, um so Verwechslungen zu umgehen.

3.3.4. Verwendung mehrerer Kameras

Mit einer zuverlässigen Bilderkennung, intelligenten Algorithmen zur Bestimmung der Trajektorie und einer zuverlässigen, motorischen Spielsteuerung ist bereits ein kompetenter Spielgegner zu erwarten. Um die Vorteile der Digitalisierung noch weiter auszunutzen, ist der Einsatz von zusätzlichen Kameras allerdings sinnvoll. Andere Umsetzungen von automatischen Kickertischen verwenden oftmals eine unter der Spielplatte eingebaute Kamera.[34] Dies ist in diesem Fall aufgrund des Aufbaus des Kickertisches nicht möglich. Außerdem bewegen die meisten Menschen ihre Köpfe während dem Spiel. Dies kann eine Kamera auch nicht eigenständig. Wenn eine oder mehr zusätzliche Kameras angeschlossen werden, bietet dies die Möglichkeit, die Ballposition mit hoher Konfidenz zu bestimmen, auch wenn diese in einem Kamerabild schwer zu erkennen ist. Auch in gut erkennbaren Szenarien bietet dies zusätzliche Sicherheit. Ebenfalls wäre eine dreidimensionale Interpretation des Spielfelds möglich, wodurch ein sich in der Luft befindender Ball lokalisiert werden könnte. Der Einsatz von zusätzlichen Kameras wäre realisierbar, indem der YOLO Algorithmus parallel für mehrere Datenströme ausgeführt wird. Dies erfordert eine höhere Rechenleistung, daher sollten Nutzen und Kosten abgewogen werden. Die Positionsdaten von zusätzlichen Kameras können durch einfache lineare Operationen auf einen einheitlichen Raum „gematched“ werden. D.h., dass im Voraus die Position und Ausrichtung der Kameras zueinander überprüft werden, sodass die Positionsdaten einfach zwischen den verschiedenen geometrischen Räumen transformiert werden können.

[34] (Bosch)

3.3.5. Auswahl der Kamera(s)

Bei der Wahl einer Kamera ist an erster Stelle auf die Framerate (dt. Bildwiederholrate) zu achten. Allerdings sollte diese selbstverständlich auch eine ausreichend hohe Auflösung haben und eine Linse, die den gesamten Kickertisch abbilden kann. Laut den Anforderungen soll die Aufnahme und Verarbeitung der Bilder so schnell wie möglich (< 4 ms) laufen. Somit ist eine möglichst hohe Framerate zu wählen, allerdings ist hier ein zusätzlicher Faktor zu beachten. Denn da der YOLO Algorithmus i.d.R. Raten von ca. 250 FPS schaffen kann, ist eine viel höhere Framerate nicht zwingend förderlich. Während der Implementierung des Algorithmus sollte eine realistische Performance erforscht werden. So besteht die Möglichkeit, eine Framerate zu wählen, die der doppelten Verarbeitungsgeschwindigkeit des Algorithmus entspricht. Dies kann in einigen Fällen die Latenz zwischen Aufnahme des Bilds und Übertragung der Daten verkürzen, da (wie bei Monitoren auch) Kameras mit höherer Framerate, oft auch eine geringere Latenz bei der Datenverarbeitung haben. Es ist davon abzuraten, eine Framerate zu wählen, die nicht der exakten oder doppelten Performance des Algorithmus entspricht, da es sonst zu Wartezeiten bei der Verarbeitung kommen kann. Dies ist zu befürchten, wenn die Bilder ankommen, bevor sie verarbeitet werden können.

3.4. Bewertung der Lösungsansätze

Eine Gesamtdauer der Positionsbestimmung von unter 4 Millisekunden ist nur mit geringer Wahrscheinlichkeit umsetzbar. Einerseits brauchen Kameras Zeit zum Aufnehmen und Verschicken von Bildern, andererseits ist zum jetzigen Zeitpunkt noch nicht sicher, wie gut die zur Bildauswertung verfügbare Hardware sein wird. Es ist davon auszugehen, dass keine NVIDIA V100 GPU (wie im YOLOv7 Paper verwendet) verfügbar sein wird. Trotzdem legt diese Arbeit die Grundlage zur Umsetzung der schnellstmöglichen Bilderkennung mit großem Potenzial zur Erweiterung des Anwendungsszenarios. Mit dem YOLOv7 Algorithmus wurde der zum Anwendungsfall passendste Algorithmus gewählt, der klassische Methoden der Bildverarbeitung bei weitem übertrifft. Mit diesem ist eine Bildauswertungsdauer (d.h. ohne die Dauer der Bildaufnahme) von unter 4 ms realistisch erreichbar. Die Umsetzung der gewählten Methoden ist noch nicht durchgeführt worden und daher ist ihre Zuverlässigkeit noch nicht getestet. Allerdings kann mit Zuversicht gesagt werden, dass der gewählte Algorithmus im Anwendungsgebiet der High-Speed-Objekterkennung keine direkten Konkurrenten mit vergleichbarer Genauigkeit und Geschwindigkeit hat.

3.4.1. Potenzial für Skalierung

In Hinblick auf eine vollautomatische Spielsteuerung sind viele Erweiterungen des in dieser Arbeit entworfenen Konzepts möglich. Eine nützliche Funktionalität der Bilderkennung wäre die Erkennung der Ausrichtung und Position der Spielstangen. Dies ist eine Funktionalität, die YOLO dank Pose-Estimation liefert. Wenn ein Machine Learning Modell zusätzlich die Positionen der einzelnen Spielfiguren erhält, können genauere Vorhersagen zur Bewegung der Bälle getroffen werden. Zusätzlich können Taktiken zum Schießen von Toren entwickelt werden.

4. Fazit

Diese Arbeit hat das Thema der High-Speed-Bilderkennung im Kontext einer automatischen Spielsteuerung für Kickertische bearbeitet. Hierzu konnte bestimmt werden, dass eine Auswertung der Bilddaten in unter 4 Millisekunden geschehen sollte. Dies bedeutet, dass ein Bilderkennungsalgorithmus mit einer Geschwindigkeit von über 250 Bildern pro Sekunde notwendig ist, um ein Bottleneck der Bilderkennung in jedem möglichen Fall zu vermeiden. Aktuelle State of the Art High-Speed-Algorithmen (besonders YOLOv7) bieten bereits Möglichkeiten der Objekterkennung und -lokalisierung mit Geschwindigkeiten von 286 Bildern pro Sekunde unter Verwendung der NVIDIA V100 Grafikkarten. Der You-Only-Look-Once-Algorithmus kann trotz seiner konkurrenzlosen Geschwindigkeit mithalten mit der hohen Genauigkeit der meisten nicht-Echtzeit-orientierten Algorithmen. Die Performance dieses Algorithmus im Falle der Erkennung von Kickerbällen konnte noch nicht evaluiert werden, aber wird hoch eingeschätzt, da die Bälle grundsätzlich als leicht zu erkennende Objekte gelten. Bei der Entwicklung eines Konzepts für Objekterkennung ist die durchgehende Weiterentwicklung der Algorithmen zu beachten, sodass sich der State of the Art oft ändert. Eine Umsetzung der Lokalisierung ist mit dem YOLOv7 Algorithmus in Kombination mit DeepSORT sinnvoll und eignet sich als Thema für eine Bachelorarbeit. Diese Algorithmen bieten großes Potenzial für Erweiterungen der Features, sodass in Zukunft auch eine vollautomatische Spielsteuerung möglich ist.

Quellenverzeichnis

Ahmad, Misbah. “Overhead View Person Detection Using YOLO” IEEEXplore, 12 October 2019, https://ieeexplore.ieee.org/abstract/document/8992980. Accessed 25 October 2022.

Anwar, Aqeel. “What is Average Precision in Object Detection & Localization Algorithms and how to calculate it?” *Towards Data Science*, 13 May 2022, https://towardsdatascience.com/what-is-average-precision-in-object-detection-localization-algorithms-and-how-to-calculate-it-3f330efe697b. Accessed 11 December 2022.

Bialonski, Stephan. *Einführung in Machine Learning*. Fachhochschule Aachen 2022.

Boesch, Gaudenz. “Object Detection in 2022: The Definitive Guide - viso.ai.” *Viso Suite*, 2022, https://viso.ai/deep-learning/object-detection/. Accessed 7 December 2022.

Boesch, Gaudenz. “YOLOv7: The Most Powerful Object Detection Algorithm (2022 Guide) - viso.ai.” *Viso Suite*, 2022, https://viso.ai/deep-learning/yolov7-guide/. Accessed 3 December 2022.

Bosch. “Tischkicker mit Künstlicher Intelligenz.” *Bosch Global*, 2018, https://www.bosch.com/de/stories/kick-it-like-bosch/. Accessed 15 October 2022.

Codeburst. “How Computers Store Images. Images are at the source of nearly… | by Greg.” *codeburst*, 12 October 2020, https://codeburst.io/how-are-images-stored-on-a-computer-353ac16b6d8f. Accessed 3 December 2022.

foosball. “USTSA Foosball Rules of Play.” *Foosball.com*, 2012, https://www.foosball.com/learn/rules/ustsa/. Accessed 22 October 2022.

Fraunhofer IPT. “5G-Industry Campus Europe - Fraunhofer IPT.” *Fraunhofer IPT*, 1 August 2019, https://www.ipt.fraunhofer.de/en/projects/5g-industry-campus-europe.html. Accessed 12 December 2022.

Fraunhofer IPT. “5G-Technologie - Fraunhofer IPT.” *Fraunhofer IPT*, 2020, https://www.ipt.fraunhofer.de/de/angebot/digitalisierung/5g.html. Accessed 12 December 2022.

Fraunhofer IPT. “Industrie 4.0 - Fraunhofer IPT.” *Fraunhofer IPT*, 2020, https://www.ipt.fraunhofer.de/en/trends/industrie40.html. Accessed 21 December 2022.

Frost, Jim. “Mean Squared Error (MSE) - Statistics By Jim.” *Statistics by Jim*, 2022, https://statisticsbyjim.com/regression/mean-squared-error-mse/. Accessed 3 December 2022.

Girshick, Ross. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 22 October 2014. *arxiv.org*, https://arxiv.org/pdf/1311.2524.pdf. Accessed 9 December 2022.

Human Benchmark. “Reaction Time Test.” *Human Benchmark*, 2022, https://humanbenchmark.com/tests/reactiontime. Accessed 10 December 2022.

IBM. “What are Neural Networks?” *IBM*, 17 August 2020, https://www.ibm.com/cloud/learn/neural-networks. Accessed 5 December 2022.

Koopmann, Anna-Lena. “Kickern: Mit diesen Tricks werden Sie Tischfußball-Profi.” *Esquire*, 24 February 2020, https://www.esquire.de/kickern-tricks-schusstechniken-tischfussball. Accessed 20 October 2022.

Mallick, Satya. “Image Recognition and Object Detection : Part 1.” *LearnOpenCV*, 14 November 2016, https://learnopencv.com/image-recognition-and-object-detection-part1/. Accessed 12 December 2022.

Merriam-Webster. “Monochrome Definition & Meaning.” *Merriam-Webster*, 17 December 2022, https://www.merriam-webster.com/dictionary/monochrome. Accessed 12 December 2022.

OpenCV. “Eroding and Dilating.” *OpenCV*, 2022, https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html. Accessed 13 December 2022.

OptiTrack. “Cameras.” *OptiTrack*, 2022, https://www.optitrack.com/cameras/. Accessed 15 December 2022.

Pang, Yanwei, et al., editors. *Deep Learning in Object Detection and Recognition*. Springer Singapore, 2019. Accessed 2 December 2022.

Pramoditha, Rukshan. “How RGB and Grayscale Images Are Represented in NumPy Arrays.” *Towards Data Science*, 4 December 2021, https://towardsdatascience.com/exploring-the-mnist-digits-dataset-7ff62631766a. Accessed 12 December 2022.

Ren, Shaoqing. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 6 January 2016. *arxiv.org*, https://arxiv.org/pdf/1506.01497.pdf. Accessed 9 December 2022.

SAP. “What is industry 4.0? | Definition, technologies, benefits.” *SAP*, 2022, https://www.sap.com/insights/what-is-industry-4-0.html. Accessed 10 December 2022.

Scheiter, Nina. “Infrastructure.” *5G-Industry Campus Europe*, 29 June 2021, https://5g-industry-campus.com/infrastructure/. Accessed 12 December 2022.

Stuff Made Here. “I made a 100MPH flying hoop.” *YouTube*, 5 June 2022, https://www.youtube.com/watch?v=xHWXZyfhQas. Accessed 10 October 2022.

Sydorenko, Iryna. “Introduction to Labeled Data: What, Why, and How.” *Label Your Data*, 14 September 2020, https://labelyourdata.com/articles/introduction-to-labeled-data-what-why-and-how. Accessed 6 December 2022.

Tan, Lu. *Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identifcation*. 22 November 2021. *BMC*, https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-021-01691-8. Accessed 5 December 2022.

Tutorialspoint. “OpenCV - Adaptive Threshold.” *Tutorialspoint*, 2019, https://www.tutorialspoint.com/opencv/opencv_adaptive_threshold.htm. Accessed 9 December 2022.

Wang, Chien-Yao. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. 6 July 2022. *arxiv.org*, https://arxiv.org/pdf/2207.02696v1.pdf. Accessed 10 December 2022.

Yang, Feng. *Video object tracking based on YOLOv7 and DeepSORT*. *arxiv.org*, https://arxiv.org/pdf/2207.12202.pdf. Accessed 11 December 2022.

Yeung, Tiffany. “What Is Edge AI and How Does It Work?” *NVIDIA Blog*, 17 February 2022, https://blogs.nvidia.com/blog/2022/02/17/what-is-edge-ai/. Accessed 10 December 2022.

Abbildungsverzeichnis