Plan van Aanpak

Maxim van Duin
Faculteit Digitale Media en Creatieve Industrie
Hogeschool van Amsterdam
Amsterdam, Nederland
maxim.van.duin@hva.nl

***Abstract*—Samenvatting hier**
***Index Terms*—keyword, keyword**

I. Introductie

Uit onderzoek blijkt dat 42% van de bedrijven in de industrie te kampen heeft met een personeelstekort. De verwachting is dat dit tekort binnen twee jaar zal oplopen tot 69% [25]. Daarnaast ervaart 22,4% van de bedrijven hinder bij het vinden van personeel vanwege een beperkt aantal werkzoekenden. Voor 25,9% vormt een gebrek aan scholing, kwalificaties of relevante ervaring bij potentiële werknemers een belangrijke belemmering [1]. Om dit probleem te ondervangen, investeert 24,4% van de bedrijven in trainingen en ontwikkelingsprogramma's voor hun medewerkers [2].

JOZ, een fabrikant van landbouwmachines, ervaart vergelijkbare uitdagingen op het gebied van personeelstekort en werving. Om de vaardigheden van werknemers te verbeteren, biedt JOZ eveneens online trainingen aan. Hoewel dit een stap in de juiste richting is, ontbreekt het deze trainingen aan interactiviteit. Werknemers hebben geen mogelijkheid om vragen te stellen of directe ondersteuning te krijgen.

Uit eerder onderzoek blijkt dat een RAG (Retrieval Augmented Generation)-systeem kan worden ingezet als virtuele les assistenten om leerlingen te ondersteunen en hun vragen te beantwoorden [4], [9], [19]. Door het gebruik van RAG kan een Large Language Model (LLM) relevante informatie uit de lesstof halen en dit gebruiken bij het genereren van een antwoord. Dit zorgt ervoor dat de antwoorden nauwkeuriger en contextspecifiek zijn [7].

RAG haalt zijn informatie gebruikelijk op uit documenten. Veel online trainingen en lessen worden echter aangeboden in de vorm van video's, waardoor een traditioneel RAG-systeem, dat zijn antwoorden baseert op documenten, niet voldoende is. Dit onderzoek bouwt voort op het werk van twee UvA-studenten, die het Context-Aware Knowledge Extraction (CAKE) framework hebben ontwikkeld (BRON TOEVOEGEN). Dit framework verbetert de prestaties van LLM's met behulp van Knowledge Extraction algoritmes en is ontworpen om contextuele kennis te extraheren en te benutten uit onder andere video's, audio en tekst.

Uit hun onderzoek blijkt echter dat de huidige methode voor het extraheren van informatie uit videobeelden niet optimaal werkt. Daarom wordt in dit onderzoek een nieuwe methode voorgesteld, waarbij gebruik wordt gemaakt van een objectdetectie- en actiedetectiemodel. Vervolgens wordt een methode ontwikkeld die de output van beide modellen combineert. Deze gegevens worden vervolgens opgeslagen in een knowledge graph en geïntegreerd in het bestaande CAKE-framework.

Het doel van dit onderzoek is om de effectiviteit van RAG-systemen bij het verwerken van videobeelden te verbeteren. Door objectdetectie en actiedetectie te combineren en de verkregen informatie op te slaan in een knowledge graph binnen het CAKE-framework, kunnen virtuele les assistenten binnen online trainingen en lessen in de industrie effectiever ondersteunen.

Om dit doel te bereiken wordt de volgende onderzoeksvraag onderzocht: *Hoe kunnen objectdetectie en actiedetectie worden ingezet om informatie uit videobeelden te extraheren en te integreren in een knowledge graph binnen het CAKE-framework, om de effectiviteit van virtuele les assistenten in online trainingen en lessen in de industrie te verbeteren?* Deze onderzoeksvraag wordt opgedeeld in de volgende deelvragen:

- Vraag1?
- Vraag2?
- Enz.

In hoofdstuk 2 wordt het theoretisch kader opgesteld, waarbij er literatuuronderzoek is gedaan naar onder andere ... . In hoofdstuk 3 wordt de voorgestelde onderzoeksmethode besproken. In hoofdstuk 4 staat de planning van het onderzoek centraal. Tot slot gaat hoofdstuk 5 in op de conclusie.

II. Theoretische kader

Dit onderzoek maakt deel uit van een groter project waarin informatie uit video's wordt opgeslagen in een knowledge graph en vervolgens wordt gebruikt in een Retrieval-Augmented Generation (RAG) systeem. Dit systeem maakt gebruik van een Large Language Model (LLM) om antwoorden te genereren op basis van gestructureerde kennis.

Om de basis van dit grotere onderzoek te begrijpen, worden in dit theoretisch kader de relevante concepten besproken. Eerst worden Large Language Models (LLM's), Retrieval-Augmented Generation (RAG) en knowledge graphs uitgelegd. Vervolgens wordt ingegaan op Graph Retrieval-Augmented Generation (GRAG), een techniek die knowledge graphs integreert in RAG-systemen. Daarna wordt het CAKE-framework beschreven waarin video's worden gebruikt in een GRAG-pipeline. Tot slot worden objectdetectie en actiedetectie besproken, twee technieken die essentieel zijn voor het extraheren van betekenisvolle informatie uit videobeelden.

A. Large Language Models (LLM's)

Large Language Models (LLM's) zijn modellen die zijn getraind op het begrijpen en genereren van menselijke taal [29]. Door het trainen op grote hoeveelheden tekstdata leren deze modellen complexe taalpatronen te herkennen en kunnen ze diverse taalgerelateerde taken uitvoeren [20].

De meeste LLM's zijn gebaseerd op de transformer-architectuur [6]. Deze architectuur maakt gebruik van het self-attention mechanism, waardoor modellen effectief afhankelijkheden die ver uit elkaar staan in de tekst vast kunnen leggen [28]. Sinds de introductie van de transformer-architectuur zijn verschillende LLM's ontwikkeld, zoals GPT (Generative Pre-trained Transformer) en BERT (Bidirectional Encoder Representations from Transformers).

LLM's hebben potentie om ingezet te worden in het onderwijs. Ze kunnen ondersteuning bieden bij het ontwikkelen van lees-, schrijf-, wiskunde- en taalvaardigheden. Daarnaast kunnen ze studenten voorzien van gepersonaliseerd oefenmateriaal, samenvattingen en uitleg, wat kan bijdragen aan betere leerprestaties en een verrijkte leerervaring [14].

B. Retrieval Augmented Generation (RAG)

LLM's worden getraind op grote, algemene datasets en beschikken daardoor over weinig tot geen domeinspecifieke kennis [3]. Om LLM's toch toegang te geven tot domeinspecifieke kennis, kan Retrieval Augmented Generation (RAG) worden ingezet. Daarnaast zorgt RAG ervoor dat LLM's minder hallucineren [16], wat een voordeel met zich mee brengt binnen het onderwijs. RAG kan worden onderverdeeld in de volgende basiscategorieën [7]:

1) Naive RAG: Naive RAG is de eenvoudigste vorm van RAG. Hierbij wordt tekst opgedeeld in kleinere stukken tekst, deze kleinere stukken tekst worden chunks genoemd. Deze chunks worden met een embedding model omgezet in vectorrepresentaties en vervolgens opgeslagen in een vectordatabase. Wanneer een gebruiker een vraag stelt aan het LLM, wordt deze vraag ook omgezet in een vectorrepresentatie. Vervolgens wordt deze vector vergeleken met alle vectoren in de vectordatabase. De K meest overeenkomende chunks worden opgehaald uit de vectordatabase en toegevoegd aan de prompt als context. Op basis van deze extra context kan het LLM een beter onderbouwd antwoord genereren [7].

2) Advanced RAG: Advanced RAG is een uitbreiding ten opzichte van Naive RAG. Hierbij worden twee stappen toegevoegd. Er wordt een pre-retrieval stap toegevoegd om de kwaliteit van de geïndexeerde tekst te verbeteren en de oorspronkelijke vraag van de gebruiker duidelijker en beter geschikt te maken voor de retrieval-taak. Daarnaast wordt er een post-retrieval stap toegevoegd om een overload aan irrelevante informatie te voorkomen. Dit proces richt zich op het selecteren van essentiële informatie, het benadrukken van de belangrijke onderdelen en het verkorten van de context die verwerkt moet worden [7].

Naast deze twee basiscategorieën zijn er nog vele uitbreidingen mogelijk op een RAG-framework.

Onderzoek toont aan dat het gebruik van RAG in het onderwijs diverse voordelen heeft. Zo blijkt uit het onderzoek van *Hicke et al.* [9] dat de inzet van RAG leidt tot een verbetering van 30% in de antwoordkwaliteit. Daarnaast toont het onderzoek van Alawwad et al. [4] aan dat de nauwkeurigheid van antwoorden op de testset met bijna 10% toeneemt door het gebruik van RAG. Verder geeft het onderzoek van *Ma et al*. [19] aan dat 78% van de gebruikers aangaven dat AI les assistenten hun leerproces verbeterden.

Samengevat laat onderzoek zien dat RAG inzetten in het onderwijs voordelen biedt. LLM's kunnen beter domeinspecifieke antwoorden geven en minder hallucinatie vertonen. Studies laten zien dat de kwaliteit van antwoorden omhooggaat en dat gebruikers ervaren dat hun leerproces verbetert.

C. Knowledge graphs

In plaats van vectordatabases kan RAG ook gebruikmaken van knowledge graphs. Knowledge graphs zijn structuren die informatie organiseren over entiteiten zoals objecten, gebeurtenissen, situaties en concepten. Ze worden gebruikt om complexe netwerken van gerelateerde informatie te modelleren, waardoor ze gebruikt kunnen worden om kennis uit af te leiden en over te dragen [10].

Er is geen universeel geaccepteerde definitie van knowledge graphs. In het paper *'Knowledge graphs'* wordt de volgende inclusieve definitie gehanteerd: *Knowledge graphs worden beschouwd als een grafiek van gegevens die bedoeld is om kennis over de echte wereld te verzamelen en over te dragen, waarbij de knooppunten belangrijke entiteiten vertegenwoordigen en de randen de relaties tussen deze entiteiten weergeven* [10].

Het ontwikkelen van knowledge graphs wordt doorgaans onderverdeeld in de volgende fases:

1) Specificatiefase: In deze fase wordt de algemene motivatie, doel en het onderwerp voor het maken van de knowledge graph vastgesteld.
2) Conceptualisatiefase: In deze fase wordt de verzamelde domein kennis gestructureerd in de vorm van tussen representaties.
3) Formalisatiefase: In deze fase wordt het conceptuele model van het domein omgezet in een, door machines interpreteerbaar model.
4) Integratiefase: In deze fase wordt het model uitgebreid met bestaande kennis.
5) Augementatiefase: In deze fase wordt de kwaliteit beoordeelt en de knowledge base verfijnd.

Met deze algemene stappen kan een knowledge base worden opgebouwd [12]. Deze informatie is belangrijk voor het begrijpen van een knowledge graph binnen het CAKE-framework.

D. Graph Retrieval-Augmented Generation (GRAG)

Graph Retrieval-Augmented Generation (GRAG) is een variant op RAG waarbij gebruikgemaakt wordt van knowledge graphs. RAG werkt zoals eerder uitgelegd door documenten op te halen op basis van tekstuele overeenkomst, maar houdt geen rekening met de onderlinge relaties van de chunks. GRAG

houdt hier wel rekening mee door gebruik te maken van knowledge graphs [11].

Tijdens de retrieval fase identificeert GRAG eerst de meest relevante knooppunten in de knowledge graph en selecteert hun directe buren. Vervolgens wordt met een soft pruning techniek irrelevante informatie verwijderd uit de subgraph [11].

De subgraph wordt vervolgens op twee manieren verwerkt door het LLM. De eerste manier is de graph view, waarbij representaties van de subgraphs worden geleerd als soft prompt om vast te leggen hoe teksten met elkaar verbonden zijn. De tweede manier is text view, waarbij de subgraphs worden omgezet in hiërarchische tekstbeschrijvingen als hard prompt om vast te leggen hoe de verbindingen worden verwoord [11].

Uit de resultaten blijkt dat GRAG beter presteert dan andere RAG-methoden en zelfs gefinetunede LLM's overtreft op alle metrics en datasets [11]. Dit laat dus zien dat GRAG een effectievere methode is voor het verbeteren van LLM-prestaties.

E. Context-Aware Knowledge Extraction (CAKE)

In het grotere onderzoeksproject wordt er een GRAG-pipeline opgezet om informatie uit video om een LLM antwoord te laten geven op basis van video's. In dit onderzoek wordt hier verder aan gewerkt. In figuur 1 is het deel van de pipeline te zien waar de video's worden omgezet in een knowledge graph.

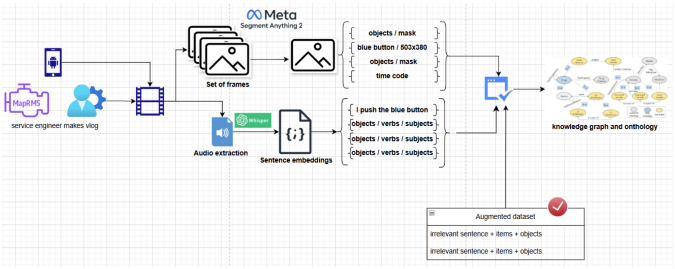

Fig. 1. Enter Caption

In dit proces worden de video's opgedeeld in audio en beeld. De audio wordt vervolgens getranscribeerd naar tekst met behulp van het wisper large-v2 model van openAI. De output is een tekstbestand. Deze tekst wordt vervolgens met SDPMChunker van Chonkie opgedeeld in kleinere chunks. De start- en eindtijd van wanneer deze chunk voorkomt in de video wordt bijgehouden. Vervolgens wordt deze informatie opgeslagen in een JSON-bestand. Nadat het audiodeel is uitgevoerd, wordt het beeld verwerkt. Het beeld van de video wordt op basis van de chunks die zijn gemaakt in het audiodeel gesegmenteerd. Vervolgens wordt in elk segment per drie seconden objecten gedetecteerd in het beeld met behulp van het YOLOv8-model. Deze informatie wordt opgeslagen in een JSON-bestand en wordt gekoppeld aan de bijbehorende chunk. Deze data wordt vervolgens omgezet in een knowledge graph.

Uit de resultaten blijkt dat het extraheren van de juiste informatie niet altijd voldoende werkt. Er worden vaak te veel niet-relevante objecten gedetecteerd en daarnaast worden de objecten ook regelmatig onjuist gelabeld.

Aangezien in dit onderzoek gefocust wordt op het verbeteren van het extraheren van visuele informatie uit de video's, wordt er in het theoretisch kader verder ingegaan op objectdetectie en actiedetectie.

F. Objectdetectie

Objectdetectie is een onderdeel van computervision en houdt zich bezig met het detecteren van objecten en een bepaalde klasse in afbeeldingen. Het doel van objectdetectie is het ontwikkelen van een model dat aangeeft welke objecten zich waar bevinden [30].

De meeste objectdetectiemodellen kunnen tegenwoordig ingedeeld worden in de klasse Two-stage detector en One-stage detector [30] [15].

De Two-stage detector verdeelt het detecteren van objecten in twee taken, het lokaliseren en het classificeren van het object. Eerst wordt de regio bepaald waar het object zich in bevindt en vervolgens wordt de regio geclassificeerd. Het voordeel van deze categorie is de hoge nauwkeurigheid, het nadeel is de lage detectiesnelheid [15]. Voorbeelden van Two-stage detector modellen zijn: Faster RCNN [8] en FPB [18].

De One-stage detector lokaliseert en classificeert objecten tegelijkertijd, zonder ze in twee afzonderlijke stappen te verdelen [15]. Voorbeelden van One-stage detector modellen zijn: YOLO [22] en DETR [5].

Het YOLO framework is een veelgebruikt framework en staat bekend om zijn goede balans tussen snelheid en accuracy [24]. Dit framework wordt ook vaak toegepast in de industrie [13], [21], [27].

Het recentste model in de YOLO-serie is YOLOv12. Voorheen maakten YOLO-modellen gebruik van convolutionele neurale netwerken (CNN), omdat op attention-based modellen de snelheid van CNN-gebaseerde modellen niet konden evenaren. YOLOv12 brengt hier verandering in door wél gebruik te maken van een attention-based model, terwijl het dezelfde snelheid behaalt als CNN-gebaseerde YOLO-modellen. Bovendien levert YOLOv12 betere prestaties dan alle voorgaande YOLO-modellen [26].

Een bijkomend voordeel is dat YOLOv12 een relatief compact model is. Afhankelijk van de gekozen variant varieert het aantal parameters tussen 2,6 miljoen en 59,1 miljoen [26].

Kort samengevat toont onderzoek aan dat YOLOv12 een van de snelste en best presterende objectdetectiemodellen is, met als bijkomend voordeel dat het relatief compact is.

G. Human Action Recognition (HAR)

Human Action Recognition richt zich op het begrijpen en labelen van menselijke acties [23]. Dit proces herkent acties op basis van video's waarin de volledige handeling is vastgelegd. Dit verschilt van action prediction, waarbij de actie wordt voorspeld op basis van een onvolledige video [17].

De meest gebruikte data voor HAR is RGB-video. Dit datatype heeft als voordelen dat het breed toepasbaar is en eenvoudig te verkrijgen en te gebruiken, bijvoorbeeld bij

lesvideo's. Echter, RGB-video heeft ook enkele nadelen. HAR-modellen die op deze data zijn getraind, zijn gevoelig voor factoren zoals kijkhoek, achtergrond en verlichting [23].

UITBREIDEN MET BESTE MODELLEN OP DIT MOMENT!!!

III. Onderzoeksmethode

Uit het onderzoek van de UvA-studenten blijkt dat het extraheren van informatie uit het beeld van de video's in het huidige proces nog onvoldoende werkt, waardoor de knowledge graph onvolledig blijft. Dit onderzoek richt zich op het verbeteren van het extraheren van informatie uit de beelden van de video's.

A. Objectdetectie

Om het huidige proces van het extraheren van informatie uit het beeld van video's te verbeteren, wordt de volgende oplossing voorgesteld. In de huidige methode wordt elke drie seconden op één frame objectdetectie uitgevoerd met het YOLOv8-model. Zoals al eerder genoemd, is uit eerder onderzoek echter gebleken dat deze aanpak niet optimaal presteert.

Om de prestaties te verbeteren, wordt voorgesteld om het ...-model te gebruiken en dit specifiek te fine-tunen op de meest voorkomende objecten in de video's van …, zonder de kennis van het model te verliezen. Deze kennis kan mogelijk relevant zijn bij het detecteren van andere algemene, relevante objecten in de video's.

Bij het fine-tunen van het …-model worden extra klassen toegevoegd die het model moet kunnen herkennen. Deze klassen zijn afkomstig uit de …-dataset. Tijdens het fine-tuneproces worden de volgende hyperparameters geoptimaliseerd totdat het optimale model is gevonden:

- Epochs: (Uitleg waarom dit belangrijk is)
- Hyperparameter 2:
- Enz.

Na de optimalisatie van het objectdetectiemodel wordt de aanpak verbeterd door gebruik te maken van Multi-Frame Tracking, in plaats van objectdetectie uit te voeren op één frame per drie seconden. Multi-Frame Tracking zorgt ervoor dat objecten in opeenvolgende frames niet telkens als nieuwe entiteiten worden behandeld. Zonder deze techniek zou een object in elk frame opnieuw worden gedetecteerd zonder een uniek ID. Door Multi-Frame Tracking behoudt een herkend object gedurende meerdere frames hetzelfde ID, wat de nauwkeurigheid verhoogt en ruis vermindert. Objectdetectie kan namelijk soms objecten missen in een frame, maar Multi-Frame Tracking kan deze gaten opvullen en zorgen voor een consistenter resultaat (BRON NOG TOEVOEGEN).

Daarnaast zorgt Multi-Frame Tracking voor een efficiëntere verwerking. Doordat objecten al een ID hebben, hoeft de detectie alleen te controleren of ze al bestaan, in plaats van ze telkens opnieuw te analyseren. Dit vermindert de rekenkosten en versnelt het proces aanzienlijk (BRON NOG TOEVOEGEN).

Een ander voordeel van Multi-Frame Tracking is dat het herhaling bij het opslaan voorkomt. Aangezien acties meestal langer duren dan één frame, zou het inefficiënt om per frame een object-actie relatie op te slaan. Multi-Frame Tracking zou hierbij kunnen bijdragen door bijvoorbeeld bij te houden wanneer een object-actie relatie begint en eindigt. Hierdoor wordt er niet onnodig dubbele data opgeslagen in de knowledge graph.

B. Actiedetectie

In het huidige proces wordt er alleen rekening gehouden met de objecten die in het beeld staan. Echter, is het ook belangrijk welke handelingen er met deze objecten worden uitgevoerd. Daarom wordt er voorgesteld om naast objectdetectie gebruik te maken van actiedetectie.

C. Data

D. Model

E. Evaluatie

F. Vergelijken baseline

IV. Planning

V. Conclusie

References

[1] Hogere werkdruk belangrijkste gevolg personeelstekort volgens ondernemers, August 2024.

[2] Centraal Bureau voor de . Drie kwart van de ondernemers probeert productiviteit te verhogen, May 2024.

[3] Ankush Agarwal, Sakharam Gawade, Amar Prakash Azad, and Pushpak Bhattacharyya. KITLM: Domain-Specific Knowledge InTegration into Language Models for Question Answering, August 2023. arXiv:2308.03638 [cs].

[4] Hessa A. Alawwad, Areej Alhothali, Usman Naseem, Ali Alkhathlan, and Amani Jamal. Enhancing textual textbook question answering with large language models and retrieval augmented generation. *Pattern Recognition*, 162:111332, June 2025.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers, May 2020. arXiv:2005.12872 [cs].

[6] Lizhou Fan, Lingyao Li, Zihui Ma, Sanggyu Lee, Huizi Yu, and Libby Hemphill. A Bibliometric Review of Large Language Models Research from 2017 to 2023. *ACM Transactions on Intelligent Systems and Technology*, 15(5):1–25, October 2024.

[7] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey, March 2024. arXiv:2312.10997 [cs].

[8] Ross Girshick. Fast R-CNN. pages 1440–1448. IEEE, 2015.

[9] Yann Hicke, Anmol Agarwal, Qianou Ma, and Paul Denny. AI-TA: Towards an Intelligent Question-Answer Teaching Assistant using Open-Source LLMs, December 2023. arXiv:2311.02775 [cs].

[10] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge Graphs. *ACM Computing Surveys*, 54(4):1–37, May 2022. arXiv:2003.02320 [cs].

[11] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. GRAG: Graph Retrieval-Augmented Generation, October 2024. arXiv:2405.16506 [cs].

[12] Ali Hur, Naeem Janjua, and Mohiuddin Ahmed. A Survey on State-of-the-art Techniques for Knowledge Graphs Construction and Challenges ahead. In *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 99–103, Laguna Hills, CA, USA, December 2021. IEEE.

[13] Muhammad Hussain. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, 11(7):677, June 2023.

[14] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274, April 2023.

[15] Ravpreet Kaur and Sarbjeet Singh. A comprehensive review of object detection with deep learning. *Digital Signal Processing*, 132:103812, January 2023.

[16] Jason Kirchenbauer and Caleb Barns. Hallucination Reduction in Large Language Models with Retrieval-Augmented Generation Using Wikipedia Knowledge, May 2024.

[17] Yu Kong and Yun Fu. Human Action Recognition and Prediction: A Survey, February 2022. arXiv:1806.11230 [cs].

[18] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, Honolulu, HI, July 2017. IEEE.

[19] Iris Ma, Alberto Krone Martins, and Cristina Videira Lopes. Integrating AI Tutors in a Programming Course, July 2024. arXiv:2407.15718 [cs].

[20] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large Language Models: A Survey, February 2024. arXiv:2402.06196 [cs].

[21] Nitin Rane. YOLO and Faster R-CNN object detection for smart Industry 4.0 and Industry 5.0: applications, challenges, and opportunities. *SSRN Electronic Journal*, 2023.

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, Las Vegas, NV, USA, June 2016. IEEE.

[23] Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. Human Action Recognition from Various Data Modalities: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022. arXiv:2012.11866 [cs].

[24] Juan Terven and Diana Cordova-Esparza. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, November 2023. arXiv:2304.00501 [cs].

[25] Janika Thielecke, Paul Couzy, and Gerben Hulsegge. Werkgevers Enquête Arbied 2024. 2024.

[26] Yunjie Tian, Qixiang Ye, and David Doermann. YOLOv12: Attention-Centric Real-Time Object Detectors, February 2025. arXiv:2502.12524 [cs].

[27] Liang Tianjiao and Bao Hong. A optimized YOLO method for object detection. In *2020 16th International Conference on Computational Intelligence and Security (CIS)*, pages 30–34, Guangxi, China, November 2020. IEEE.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. page 11, June 2017.

[29] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A Survey of Large Language Models, October 2024. arXiv:2303.18223 [cs].

[30] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object Detection in 20 Years: A Survey, January 2023. arXiv:1905.05055 [cs].