A Wearable Device Application for Human-Object Interactions Detection

Michele Mazzamuto[1,2], Francesco Ragusa[1,2], Alessandro Resta[1], Giovanni Maria Farinella[1,2] and Antonino Furnari[1,2]
[1]*FPV@IPLAB, DMI - University of Catania, Italy*
[2]*Next Vision s.r.l. - Spinoff of the University of Catania, Italy*

Keywords: Egocentric Vision, Human-Object Interaction, Smart Glasses.

Abstract: Over the past ten years, wearable technologies have continued to evolve. In the development of wearable technology, smart glasses for augmented and mixed reality are becoming particularly prominent. We believe that it is crucial to incorporate artificial intelligence algorithms that can understand real-world human behavior into these devices if we want them to be able to properly mix the real and virtual worlds and give assistance to the users. In this paper, we present an application for smart glasses that provides assistance to workers in an industrial site recognizing human-object interactions. We propose a system that utilizes a 2D object detector to locate and identify the objects in the scene and classic mixed reality features like plane detector, virtual object anchoring, and hand pose estimation to predict the interaction between a person and the objects placed on a working area in order to avoid the 3D object annotation and detection problem. We have also performed a user study with 25 volunteers who have been asked to complete a questionnaire after using the application to assess the usability and functionality of the developed application.

1 INTRODUCTION

Wearable technologies continued to improve rapidly with the advances in sensors, communication technologies, and artificial intelligence over the past decade. Smart glasses for augmented reality are taking on an important role in the growth of wearable devices. The global augmented reality market size was estimated at USD 25.33 billion in 2021 and is expected to expand at a compound annual growth rate (CAGR) of 40.9% from 2022 to 2030. Companies are putting a strong emphasis on finding ways to exploit the potential of Augmented Reality (AR) technology. Providing a unique and interactive experience to end-users is expected to drive the growth of the market over the forecast period. The proliferation of handheld and wearable devices, such as smartphones and smart glasses, and the subsequent increase in the adoption of mobile AR technology to provide a more immersive experience are expected to contribute to the growth of the market as well.

Different devices such as Microsoft HoloLens 2, Nreal, Magic Leap 2 have already been launched in the market promising to be able to augment our perception of the real world with additional virtual elements (e.g., holograms, images, videos, audio). All these wearable glasses provide some standard capabilities such as spatial mapping, plane detection, hand and gaze tracker which are needed to enable the interaction with virtual elements and can be exploited to develop mixed reality applications using different platforms (e. g., Unity[1] and OpenXR[2]).

We argue that, for these devices to really be able to blend the real and virtual worlds, it is essential to integrate artificial intelligence algorithms which can understand human behavior in the real world. In particular, recognizing human-object interactions from the first-person perspective allows to build intelligent systems able to understand how humans interact with the world and consequently support them during their daily activities in different domains, including home scenarios (Damen et al., 2014), cultural sites (Cucchiara and Del Bimbo, 2014; Farinella et al., 2019; Mazzamuto et al., 2022) and industrial environments (Colombo et al., 2019; Ragusa et al., 2021). Since AR devices need to be able to put the real and virtual worlds in the same reference system, a 3D understanding of user-object interactions is fundamental.

Wearable systems that recognise human-object interactions in the 3D real world can be useful in workplaces, where they can support workers by providing a continuos training on how to use a specific ob-

[1]https://unity.com/
[2]https://www.khronos.org/openxr/

Mazzamuto, M., Ragusa, F., Resta, A., Farinella, G. and Furnari, A.
A Wearable Device Application for Human-Object Interactions Detection.
DOI: 10.5220/0011725800003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP*, pages 664-671
ISBN: 978-989-758-634-7; ISSN: 2184-4321

ject, or by monitoring the use of a specific machine to schedule mainteinance procedures and avoid unexpected machine downtime. This kind of systems can also support procedural works providing a control mechanism to monitor procedures and notify missing actions (Soran et al., 2015).

In this paper, we present a smart glass application to support workers by recognizing human-object interactions in an industrial context. Modeling human interactions with real objects in the 3D scene is not trivial. Current applications such as Vuforia Model Targets[3] implement a full 6 DOF object pose estimation pipeline, for which 3D bounding box labels around objects are considered. Since annotating objects with 3D bounding boxes is expensive, it is difficult to extend these kinds of systems to scenarios in which new object classes may appear. The time taken by an annotator to label a box is approximately 7 seconds (Papadopoulos et al., 2017), instead the time needed to annotate a 3D bounding box is significantly higher and generally requires the availability of 3D point clouds. The KITTI dataset (Geiger et al., 2012) reports that on average, annotating one full batch (240 frames) in 3D took approximately 3 hours. The SUN RGB-D dataset (Song et al., 2015) reported that 2,051 hours of annotation were required to label 64,595 3D object instances, around 114 seconds per instance.

To avoid the 3D objects annotation problem, we propose a system which needs only 2D bounding box annotations to understand egocentric human-object interactions in the 3D real world by leveraging the common software layer provided by AR platforms. In particular, the proposed system makes use of a 2D object detector to localize and recognize the objects present in the scene and exploits standard capabilities such as plane detector, virtual object anchoring and hand pose estimation to recognize human-object interactions.

As a test use case, we implement a system which can offer additional information regarding the correct use of that object. We deployed the proposed application on Nreal[4] smart glasses in the ENIGMA[5] laboratory, of the University of Catania. We have also organized a test campaign on the use of the application in which 25 participants tested the application and compiled surveys provide feedback on the functionality and usability of the system. We have analyzed the results of the surveys to understand strengths and where to improve the proposed application.

[3]https://library.vuforia.com/objects/model-targets

[4]https://www.nreal.ai/

[5]https://iplab.dmi.unict.it/ENIGMA/

2 RELATED WORK

The proposed application is related to different lines of research: human-object interaction detection, and smart glass software development. The following sections discuss some relevant works related to these research lines.

2.1 OpenXR

OpenXR is considered as the reference platform when developing on mixed reality. By allowing apps to run on a larger range of hardware platforms without having to port or rewrite their code, OpenXR aims to make the creation of AR/VR software easy. Without a cross-platform standard, VR and AR applications and engines must use each platform's proprietary APIs. New input devices need customized driver integration. In this work we have used Nreal device. Since at the moment Nreal does not support development with OpenXR, in the development of the presented application it was decided to use NRSDK.

2.2 NRSDK

NRSDK[6] is the platform used by Nreal to develop mixed reality applications. It provides a series of features that can be accessed via a high-level API, simplifying the creation of content. In the proposed system we exploited the following capabilities provided by the API:

- Hand Tracking;
- Plane Detection;

2.2.1 Hand Tracking

The NRSDK Hand Tracking capability tracks the position of key points of the hands and recognises hand poses in real time. Hand poses are recognized in the first-person view and used to interact with virtual objects immersively in the world. The system can track hands through the world coordinate system and annotate the position and orientation of 23 key points. Currently, it supports six hand poses (gestures) from either hand. In the proposed application, the hand's keypoint tracking allowed us to estimate the position of the hands in the 3D environment in order to estimate possible interactions with objects.

2.2.2 Plane Detection

Since Nreal Light is not equipped with a depth sensor to obtain 3D information on the surrounding world

[6]https://nrealsdkdoc.readthedocs.io/

useful to understand the distance of the objects respect the human, we developed an approach which derived this information using the Plane Detection feature of the NRSDK. In the proposed system, we obtain the depth of an object, associating the depth of the plane in which the object is placed. In this way, considering the 2D coordinates obtained by the object detector and the depth approximated with the plane detection we are able to predict the 3D position of objects.

We decided to use Barracuda to integrate an object detector, which we trained, into our application.

2.3 Human-Object Interacion (HOI and EHOI)

In recent years, many works have focused on the Human-Object Interaction detection task, considering both third and first person views. Human-object Interaction (HOI) detection strives to locate both the human and an object as well as identify complex interactions between them. Most existing HOI detection approaches are instance-centric, where interactions between all possible human-object pairs are predicted based on appearance features and coarse spatial information. The authors of (Gupta and Malik, 2015) annotated the COCO dataset with verbs to study HOI detection from the third point of view (V-COCO). The authors of (Gkioxari et al., 2017) proposed a three-branch method which estimates the verb of the interaction, the human position as well as the possible location of the object involved in the HOI exploiting both humans and objects features. The authors of (Chao et al., 2017) studied the problem of detecting HOI in static images, predicting a human and an object bounding box with an interaction class label using graph convolutional neural networks.

Other works look at the task of HOI detection from an egocentric perspective (EHOI) which is increasingly studied. The authors of (Nagarajan et al., 2018) proposed an approach to learn human-object interaction "hotspots" directly from video in a weakly supervised manner. The authors of (Nagarajan et al., 2020) introduced a model for environment affordances that is learned directly from egocentric video, linking the environment with the action performed.

To study this task from the egocentric point of view, many datasets have acquired and labeled considering different scenarios.

The authors of (Grauman et al., 2022) recently released EGO4D, a massive-scale egocentric data set and benchmark suite collected in 74 locations around the world and 9 countries, with over 3,670 hours of video of daily activities.

Figure 1: Nreal Light smart glass.

The authors of (Ragusa et al., 2021) proposed the MECCANO dataset as the first dataset of egocentric videos to study human-object interactions in industrial-like settings. This dataset has been acquired in an industrial-like scenario in which subjects built a toy model of a motorbike.

Since collecting and labeling large amounts of real images is challenging, the authors of (Leonardi et al., 2022) proposed a pipeline and a tool to generate photo-realistic synthetic First Person Vision (FPV) images automatically labeled for EHOI detection in a specific industrial scenario.

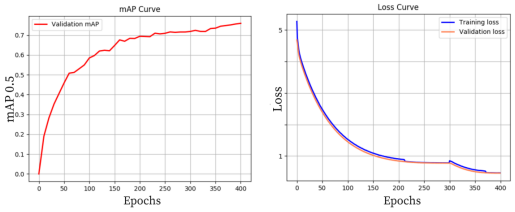
In this paper, we propose a system which combines a 2D object detector, hand tracking and plane finding modules in order to detect EHOIs.

3 PROPOSED SYSTEM

In this Section, we first discuss the hardware on which the proposed system has been developed (Section 3.1), then we present the application pipeline of the system (Section 3.2).

3.1 Hardware

Nreal Light (shown in Figure 1) is a smart glasses device for augmented and mixed reality created by the Nreal company for consumer users. Since the computation is done on an external unit (i.e., a comput unit provided by Nreal or an Android smartphone), this device is lighter and more comfortable with respect to other AR headsets. Software development on the Nreal glass was done via the Nebula platform. Nreal Light features several sensors, including two grayscale cameras for spatial computing, an RGB video camera for capturing frames with a resolution of 5 MegaPixel, microphones, speakers, accelerometer, gyroscope, sensor ambient, and proximity light (to detect if the glass is worn by the user). We opted for the adoption of this smart glass in this work for these features and given the high lightness and versatility of the device.



(a) mAp@50 of the trained detector. (b) Loss of the trained detector.

Figure 2: Loss and mAp of the detector.

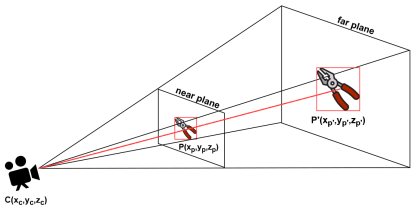


Figure 3: Relationship between 2D and 3D object coordinates.

Figure 4: Examples of classes composing the dataset.

3.1.1 Compute Unit

Nreal provides Nreal Light Developer Kit, which consists of the Computing Unit and the Controller. The unit weights 140g, is equipped with a Qualcomm Snapdragon 845 SoC, an 8 Qualcomm Kryo 385 64-bit core CPU and a Qualcomm Adreno 630 GPU. The storage is 64 GB, the operation system is Android 8. A OnePlus 8 was used to deploy the application as an alternative to the compute unit.

3.2 Application Pipeline

The proposed system includes five modules, which are described in the following subsections: 1) Frame Acquisition, 2) 2D Object Detection, 3) Plan detection, 4) 3D Object Tracking, 5) Hand Tracking and 5) Interaction Detection. Figure 5 shows the architecture of the proposed system.

3.2.1 Frames Acquisition

The acquisition of frames is performed through the RGB video camera every T seconds, through the GetTexture() method of the NRRGBCamTexture class, provided by the NRSDK suite. Each frame is acquired with a resolution of $1920x1080$ pixels. The acquisition period T is set empirically considering the computational time required to perform the objects detection in a frame. We set T=3 on the compute unit and T=2 on the One Plus 8 device.

3.2.2 2 D Object Detection

The proposed system needs to recognize objects wich can be manipulated by the user. Training a standard 2D detector is less demanding than training a 3D detector in terms of both computational time needed for the training phase and time and costs for the annotation process. Moreover, unlike data annotated with 3D Bounding boxes, there are many public dataset available with 2D annotations which allows to train 2D object detectors. We used a Tiny YOLOv4[7] object detector, which allows fast and accurate recognition of objects. Given an image, the model predicts a tuple (x, y, w, h, c) where x, y, w, h are the 2D Bounding box coordinates in the image and c represents the class of the detected object. We integrated the 2D object detector through the Barracuda library[8]. To train the YOLOv4 object detector, was used the dataset presented in (Leonardi et al., 2022), which is composed of images acquired in the industrial laboratory ENIGMA of the University of Catania, and identifies the same scenario considered in this project. A list of examples of the classes present is shown in Figure 4. We trained the model for 395 epochs. Figure 2b shows the mAP (mean Average Precision) on the validation set along with the number of training epochs. The mAP of the chosen epoch is equal to 0.775.

3.2.3 Plane Detection

Plane detection, is a feature available in the NRSDK, by the script PlaneDetector.cs present in the library. The script uses a Unity's prefab named PolygonPlaneVisualizer, which places markers of polygonal shape on the detected flat surfaces. In this way, it is

[7]https://models.roboflow.com/object-detection/yolov4-tiny-darknet

[8]https://docs.unity3d.com/Packages/com.unity.barracuda@1.0

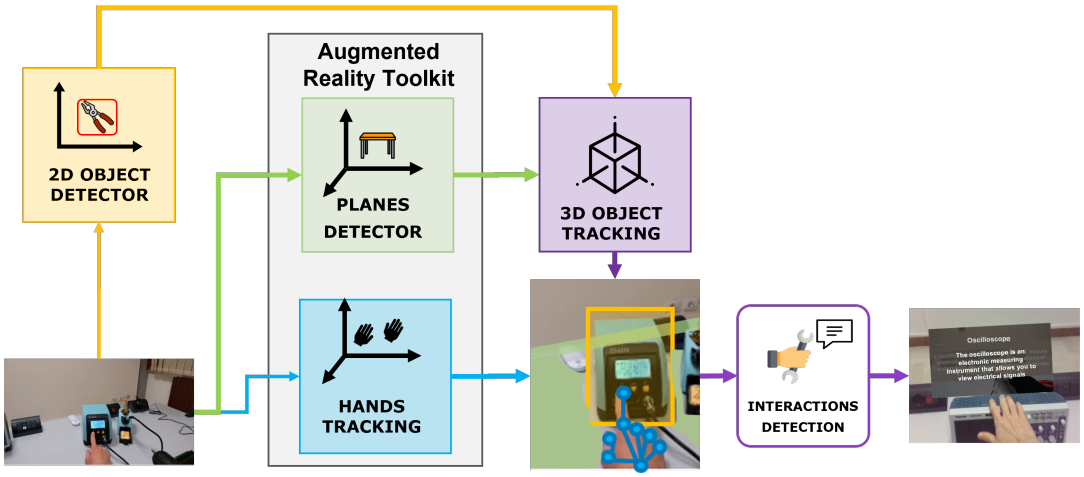


Figure 5: Overview of the full system.

possible to trace the 3D plane positions in the Unity space. Since this application is made for industrial operators, we decided to turn off the graphic rendering of the plans, for the benefit of greater visibility of the work area and less distraction for the operator. The accuracy of plane detection technology from NRSDK is generally high, but can be affected by various factors such as lighting and clutter in the environment. In situations with ample lighting and distinct planes, the detection algorithm tends to perform very well, accurately identifying and tracking planes. However, in conditions such as low light or cluttered environments, the accuracy may be compromised. The NRSDK offers a dependable and effective solution for plane detection in various situations.

3.2.4 3 D Object Tracking

This module takes as input the 2D Bounding boxes of the detected objects and the 3D coordinates of the detected plane and assigns to each object a 3D point. First, the x and y coordinates are extracted from the center point of bounding box of the identified object, creating the point $P_{2D}(x,y)$. Then the point P_{2D} is transformed into $P_{3D}(x,y,z)$ where the z coordinate is taken from the "near plane", that is a plane used in mixed reality to render 2D elements at a fixed distance. After that, a ray is generated from the 3D position of the camera that passes through $P(x,y,z)$. Considering the ray incidents on the detected plane ("far plane"), we obtain the point $P'(x',y',z')$ which represents the estimated 3D position of the detected object (see Figure 3). Once this 3D point has been identified, an invisible virtual gameobject is placed in its position.

3.2.5 Hand Tracking

Hand tracking is another feature made available by NRSDK. To activate this feature, it is required to import the prefab called "NRInput" into the application scene in Unity, and select the "Hands" value for the "Input Source Type" attribute. In this way, the system will enable the use of the hands as an input source, as an alternative to the controller (smartphone / computing unit). Once these steps are completed the hand tracking module is enabled, allowing to obtain the status of each hand, in terms of the performed gesture, position and rotation of each keypoint. This module allows to track the 3D position of both user's hands in the world respect to the camera.

3.2.6 Interaction Detection

To detect if a user is interacting with objects, the Euclidean distance between his hand, traced by hand tracking, and the various invisible gameobject that are currently tracking objects previously detected by the object detector (section 3.2.4) are checked at each frame.

Given the estimated 3D position of the objects and the user's hands present in the scene, the system predicts whether a human-object interaction is happening. For each hand, we calculate the Euclidean distance between the hand and each object present in the scene. If the distance is below the threshold of 2 centimeters, a HOI is happening and the system outputs the class of the active object involved in that interaction. This output will trigger an augmented reality service which will give to the user additional information on that object. The information is audiovisual. The user will hear an audio description relative to the interacted object and will see a short video tutorial in

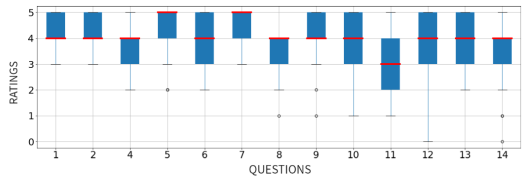


Figure 6: Results have been represented through boxplots. The median value is indicated in red.

mixed reality, showing a possible use of that object. A complete overview of the system is shown in Figure 5.

4 QUALITATIVE ANALYSIS

We asked 25 subjects to test the proposed system in the real industrial laboratory ENIGMA of the University of Catania. At the end of each session, we administrated a survey composed of 14 questions related to the usability, comfort, usefulness and possible privacy issues of the system. Table 1 reports the list of the questions included in the survey. Each question could be rated from 0 (extremely negative) to 5 (extremely positive). For the analysis, a custom version of the application was created that let the user interact with 5 objects through an audiovisual guide that provides instructions on what to do step by step. The five selected objects for this survey are: clamp, screwdriver, oscilloscope, oscilloscope probe and soldering iron base.

4.1 Results

In order to evaluate the general performance of the application, we visualize the results extracted from the surveys using boxplots (see Figure 6). Participants do not highlight any particular issues related to the analysis of the system (questions number 1-2). Each question on average has a score that varies between 4 and 5, except for question number 11 ("How do you rate the presence of wire and outdoor unit (smartphone)?"), which has a median of 3, indicating that the presence of the wire connected to the computing unit is not appreciated by the users. Questions number 6, 10, 12 and 13 have the highest interquartile range (IQR), suggesting that there is little agreement in the way the visual/text information is provided and the relevance of privacy issues in the proposed system.

We also analyzed the Spearman's rank correlation coefficient ρ of the answers which are statistically relevant (p-value < 0.05) (see Figure 7). The figure shows that the characteristics which significantly influence the overall positive judgment of the application (Question 1) are: the clarity of the instructions provided by the application (Question 7, $\rho = 0.66$), the response time of the interactions (Question 8, $\rho = 0.57$) and the number of detected interactions (Question 4, $\rho = 0.48$). It is also interesting to note how prolonged use over time (Question 14) is strongly correlated with the presence of the wire connected to the wearable device (Question 11, $\rho = 0.74$) and its weight (Question 9, $\rho = 0.58$). In particular, those who evaluates negatively the presence of the wire and the device's weight, do not express a positive opinion about the use of the device for prolonged periods. In contrast, those who positively evaluate the presence of the wire or the device's weight have no problems with prolonged use of the device.

Table 1: List of questions included in the survey. Each question is associated with an unique ID.

ID	Question
1	How satisfied are you overall with the experience?
2	How supportive do you think the technology demonstrated in this application prototype can really be?
3	Do you believe that the technology demonstrated in this prototype can be used in more complex systems?
4	How often has the system recognized the interactions?
5	How do you evaluate the information transmitted through sounds?
6	How do you evaluate the information transmitted through video?
7	How clear were the audio/video instructions included in the application about its use?
8	How satisfied are you with the system response time before receiving information on objects?
9	How do you rate the weight of the device?
10	How do you rate the visual rendering of the device?
11	How do you rate the presence of wire and outdoor unit (smartphone)?
12	Considering that the system does not save any visual data, but could keep higher level information obtained from the visual data, how much do you think the system respect the privacy of users?
13	How do you evaluate the information transmitted through text?
14	How do you evaluate a prolonged use over time?

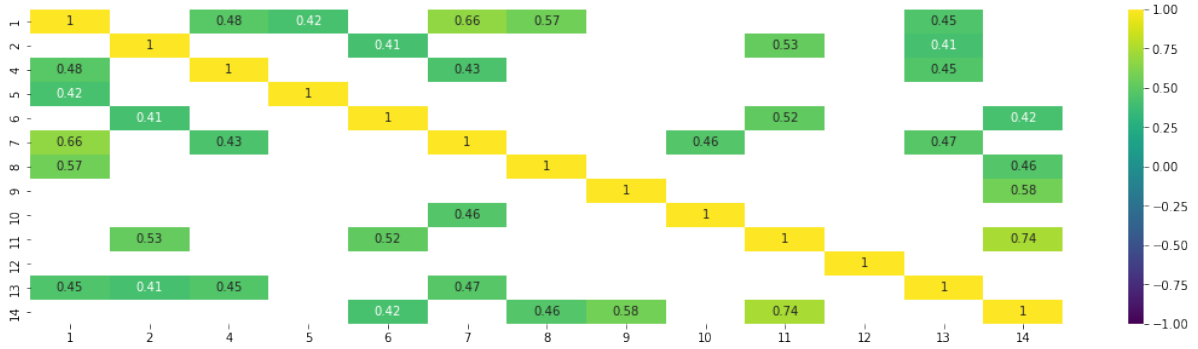


Figure 7: Filtered p-value$<$ 0.05 Spearman correlation.

5 CONCLUSIONS

In this paper, we presented a smart glass application that assists industrial employees understanding human-object interactions. To avoid the challenge related to 3D object annotation, we proposed a system that uses a 2D object detector to find and identify the objects in the scene and common features available on AR devices such as plane detector, virtual object anchoring, and hand tracking to predict how a human would interact with the objects. For qualitative evaluation purpose, we set up a test campaign for the application, in which the 25 volunteers tested the application and responded to a survey on the app's functionality and usability. The results suggest that approach presented in this work can be useful to develop applications helpful in manufacturing environments.

ACKNOWLEDGEMENTS

This research has been supported by Next Vision[9] s.r.l., by the project MISE - PON I&C 2014-2020 - Progetto ENIGMA - Prog n. F/190050/02/X44 – CUP: B61B19000520008 and by Research Program Pia.ce.ri. 2020/2022 Linea 2 - University of Catania.

REFERENCES

Chao, Y.-W., Liu, Y., Liu, X., Zeng, H., and Deng, J. (2017). Learning to detect human-object interactions.

Colombo, S., Lim, Y., and Casalegno, F. (2019). Deep vision shield: Assessing the use of hmd and wearable sensors in a smart safety device. PETRA '19, page 402–410, New York, NY, USA. Association for Computing Machinery.

Cucchiara, R. and Del Bimbo, A. (2014). Visions for augmented cultural heritage experience. *IEEE Multimedia*, 21.

Damen, D., Leelasawassuk, T., Haines, O., Calway, A., and Mayol-Cuevas, W. (2014). You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

Farinella, G., Signorello, G., Battiato, S., Furnari, A., Ragusa, F., Leonardi, R., Ragusa, E., Scuderi, E., Lopes, A., Santo, L., and Samarotto, M. (2019). *VEDI: Vision Exploitation for Data Interpretation*, pages 753–763.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gkioxari, G., Girshick, R., Dollár, P., and He, K. (2017). Detecting and recognizing human-object interactions.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S. K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E. Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Fuegen, C., Gebreselasie, A., Gonzalez, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolar, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Puentes, P. R., Ramazanova, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhu, Y., Arbelaez, P., Crandall, D., Damen, D., Farinella, G. M., Ghanem, B., Ithapu, V. K., Jawahar, C. V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H. S., Rehg, J. M., Sato, Y., Shi, J., Shou, M. Z., Torralba, A., Torresani, L., Yan, M., and Malik, J. (2022). Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*.

Gupta, S. and Malik, J. (2015). Visual semantic role labeling.

Leonardi, R., Ragusa, F., Furnari, A., and Farinella, G. M. (2022). Egocentric human-object interaction detection exploiting synthetic data.

[9]Next Vision: https://www.nextvisionlab.it/

Mazzamuto, M., Ragusa, F., Furnari, A., Signorello, G., and Farinella, G. M. (2022). Weakly supervised attended object detection using gaze data as annotations. In Sclaroff, S., Distante, C., Leo, M., Farinella, G. M., and Tombari, F., editors, *Image Analysis and Processing – ICIAP 2022*, pages 263–274, Cham. Springer International Publishing.

Nagarajan, T., Feichtenhofer, C., and Grauman, K. (2018). Grounded human-object interaction hotspots from video.

Nagarajan, T., Li, Y., Feichtenhofer, C., and Grauman, K. (2020). Ego-topo: Environment affordances from egocentric video.

Papadopoulos, D. P., Uijlings, J. R. R., Keller, F., and Ferrari, V. (2017). Extreme clicking for efficient object annotation.

Ragusa, F., Furnari, A., Livatino, S., and Farinella, G. M. (2021). The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain. In *IEEE Winter Conference on Application of Computer Vision (WACV)*.

Song, S., Lichtenberg, S. P., and Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576.

Soran, B., Farhadi, A., and Shapiro, L. G. (2015). Generating notifications for missing actions: Don't forget to turn the lights off! *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4669–4677.