virtual worlds

MDPI

Article

Avatar Detection in Metaverse Recordings

Felix Becker [1,*], Patrick Steinert [1], Stefan Wagenpfeil [2] and Matthias L. Hemmje [1]

[1] Faculty of Mathematics and Computer Science, University of Hagen, Universitätsstrasse 1, D-58097 Hagen, Germany; patrick.steinert@fernuni-hagen.de (P.S.); matthias.hemmje@fernuni-hagen.de (M.L.H.)
[2] Faculty of Business Computing and Software Engineering, PFH University of Applied Science, Weender Landstraße 3-7, D-37073 Goettingen, Germany; s.wagenpfeil@pfh.de
* Correspondence: herrnfelixbecker@gmail.com

Abstract: The metaverse is gradually expanding. There is a growing number of photo and video recordings of metaverse virtual worlds being used in multiple domains, and the collection of these recordings is a rapidly growing field. An essential element of the metaverse and its recordings is the concept of avatars. In this paper, we present the novel task of avatar detection in metaverse recordings, supporting semantic retrieval in collections of metaverse recordings and other use cases. Our work addresses the characterizations and definitions of avatars and presents a new model that supports avatar detection. The latest object detection algorithms are trained and tested on a variety of avatar types in metaverse recordings. Our work achieves a significantly higher level of accuracy than existing models, which encourages further research in this field.

Keywords: avatars; object detection; YOLO; artificial intelligence; convolutional neural networks; metaverse

Citation: Becker, F.; Steinert, P.; Wagenpfeil, S.; Hemmje, M.L. Avatar Detection in Metaverse Recordings. *Virtual Worlds* **2024**, *3*, 459–479. https://doi.org/10.3390/virtualworlds3040025

Academic Editor: Anton Nijholt

Received: 26 July 2024
Revised: 25 September 2024
Accepted: 22 October 2024
Published: 30 October 2024

1. Introduction

Recognized as a global trend in 2022 [1], the metaverse [2,3] is continuously growing [4]. Global crises, such as climate change and COVID, made it clear that digital technologies, such as video calls [5], present options for replacing in-person meetings or even providing effective virtual collaborations. This trend is likely to continue and makes the metaverse especially interesting, as it yields the potential to partially replace or at least support many in-person activities. Some of the largest companies worldwide have heavily invested in the metaverse [6–8], and public interest in this area is continuously increasing [3,9]. In recent years, numerous metaverse virtual worlds have emerged in the wild. One of the first is Second Life [10], which commenced in 2003. More recent metaverses with a high level of usage [4] are Decentraland [11], Roblox [12], and Fortnite [13]. Notably, Meta Horizon Worlds [14] is a virtual world based on Virtual Reality (VR) [15].

The metaverse is a concept that can be represented in various forms. In this paper, we follow the metaverse definition provided by Ritterbush and Teichmann: "Metaverse, a crossword of 'meta' (meaning transcendency) and 'universe', describes a (decentralized) three-dimensional online environment that is persistent and immersive, in which users represented by avatars can participate socially and economically with each other in a creative and collaborative manner in virtual spaces decoupled from the real physical world" [3]. The term avatar is based on the ancient Hindu concept of calling the physical representation of a Hindu god an avatar [16]. It is therefore not a simple placeholder but the actual representation of something in a different sphere of reality. If users create a custom character or are represented by a particular character, then this can be seen as their avatar in this world.

In these non-real or virtual worlds, there is usually at least one avatar, i.e., the representation of a user, who is taking control over the environment. These characters interact

with their environment. For a user, their own avatar can be shown in a first- or third-person view. Most 3D virtual worlds use a third-person view, while almost all VR-based virtual worlds use a first-person view. A world is typically shared with others, is persistent, to some degree, and reacts to actions happening in real-time [17].

User sessions in the metaverse can be recorded [18] as, for example, screen recordings, which we refer to as *metaverse recordings (MVRs)* [18]. Hence, the metaverse produces Multimedia Content Objects (MMCOs), i.e., images or videos. *MVRs* can serve a variety of use cases, including the creation of personal memories, such as lifelogs [19], the sharing of experiences with others [20], and the implementation of quality control in VR training [21]. Another significant use case emerges from the industrial metaverse, where the virtual world is employed for simulations to validate production lines or to generate training data for machine learning (ML) in autonomous driving [22–24].

In Multimedia Information Retrieval (MMIR) [25], avatar detection can be used to find *MVRs* in larger collections. For example, in VR training, a trainer could search recorded trainings for specific actions of avatars, or, in the case of ML training data, they could find examples for certain conditions. To search for *MVRs* with MMIR, a computer should be able to understand the content of *MVRs* and gain semantic information in virtual worlds [17]. Therefore, avatars are crucial elements that represent users in the virtual world. To achieve a semantic understanding, avatars must be detected and classified within *MVRs*.

We introduce the detection and classification of avatars within images and videos as a novel task termed *Avatar Detection*. *Avatar Detection* can be viewed as a specialized subset of object detection [26] for images and videos. One might argue that virtual world providers could offer semantic labeling or metadata for avatars during live gameplay, which we define as Scene Raw Data [27]. While this approach is considered feasible, current virtual worlds, e.g., Roblox [12,28] or Meta Horizon Worlds [14], do not provide such information for recordings. The task of avatar detection is relevant for searching and indexing large collections of images and videos within the metaverse. By incorporating this specific semantic information, the efficiency and accuracy of image and video retrieval processes can be enhanced, thus providing a robust framework for metaverse retrieval [29]. The ability to locate and classify avatars within collections would increase the effectiveness of search processes and enrich the metadata associated with digital content, thus offering substantial advancements in the management and utilization of MMCOs.

The need for research on avatar detection in virtual world recordings stems from the increasing relevance of these digital environments in both social and professional contexts, as well as the increasing number of streams and recordings of virtual worlds. Avatars serve as the primary medium through which users interact, representing their identities and actions within virtual worlds; therefore, the interest to search for avatars is a relevant capability for *MVR* retrieval systems. Further developments could use the detected avatars to identify them, similar to face identification for photos. Furthermore, the interactions of avatars are also interesting in terms of enabling searches. Interaction detection, e.g., human–object interaction or human-human interactions, rely on the detection of humans, in this case avatars, in virtual worlds. Hence, reliable avatar detection is relevant.

In MMIR, semantic information can be used to examine a situation based on the automatically extracted meta information. Semantic information describes information about the content, its meaning, and possibly even its meaning in relation to other content. For example, we can look at the statement that two avatars of the humanoid class are standing in close proximity to each other. The information can be annotated in the source image, shown in Figure 1, with two avatars interacting in an abstract location. The semantic information enables semantic search queries which, beyond a simple query for avatars, can search for avatars standing next to each other.

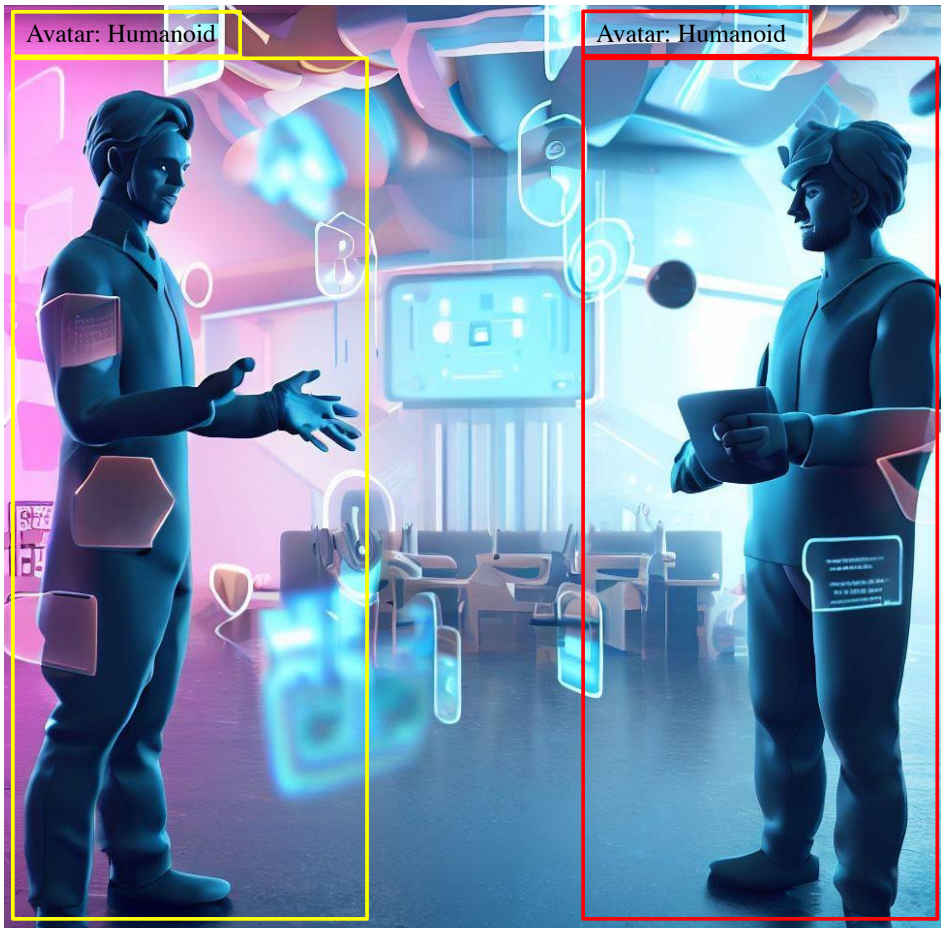

Figure 1. Highlighted by the yellow and red squares are two recognised avatars interacting with each other.

To create the semantic information, the algorithm receives an image as input, then automatically detects semantic information by locating avatars, drawing boxes around them, classifying them, and then noting their classes on the top of these boxes as *Avatar: Humanoid*.

This paper focuses on detecting whether or not some form of avatar is present and, if so, where it is located in the image. Therefore, additional information is extracted from the raw pixel data. To perform avatar detection, an appropriate algorithm must be found or implemented. Based on our research, a sufficient way to detect avatars in virtual worlds with artificial intelligence is unknown. This paper presents a method for *avatar detection* that demonstrates superior performance compared to the baseline methods.

In summary, avatar detection supports organizing and searching large collections of *MVRs*. Early experiments [30] show the limitations of existing object detection in regard to this task. We derived the problem statement that the object detection of avatars in *MVRs* is unknown. Our research addresses the research question of how avatars can be detected and classified in *MVRs*. The scientific approach used for this work follows the method described by Nunamaker et al. [31]. The application enables the researcher to present a clear and organized response to the research question. The Nunamaker framework connects the four disciplines of *Observation*, *Theory Building*, *System Development*, and *Experimentation*. These areas influence each other, with System Development being in the center.

The remainder of the paper is organized as follows. Section 2 presents the current state of the art and forms the baseline for our work. In Section 3, we present our modeling work to classify and detect avatars in *MVRs*. Section 4 describes the implementation of our model and the dataset, which are then used for the evaluation, presented in Section 5.

In the process of preparing this manuscript, we utilized AI-powered language tools, including DeepL and Writefull. These tools were employed to enhance the efficiency

of the writing process and to ensure clarity in presenting complex ideas. However, all interpretations, conclusions, and the final content of the article were thoroughly reviewed and validated by the human authors to maintain the integrity of the research.

2. Current State of the Art

Applying the scientific methodology of Nunamaker, this section presents our observations on the current state of the art.

2.1. Definitions and Characterizations of Avatars in Literature

Avatars represent the user in a virtual world [32,33]; however, they are more than just a tool to for interacting with the world and others. The ability to customize avatars allows the users to express themselves and is a relevant part of the immersive experience that the user engages in [32,34,35]. This, for example, contributes to a better learning experience [36]. This culminates in the realistic representation of facial expressions and gestures, which leads to an increased emotional and empathetic relationship with one's own and others' avatars [33]. The avatar concept also provides a form of anonymity, which provides a degree of liberty but also leads to the risks of misuse and abuse [9]. Digital identities connected to avatars have been put in place with the intention of helping to track people and holding them accountable [37]. With regard to the recognition of avatars, the infinite variations in their appearance and the lack of associated information about them in video footage represent a significant challenge.

According to Miao et al. [38], avatars in the virtual worlds of the metaverse lack a unified definition and taxonomy [38]. They suggest the following definition based on their empirical findings of different avatar definitions used in relevant papers: "We define avatars as digital entities with anthropomorphic appearance, controlled by a human or software, that are able to interact" [38]. They suggest a simple taxonomy that is two-by-two in dimension. The first dimension is form realism and the second one is behavioral realism. Realism in form is defined mainly by the level of anthropomorphism, which increases with realistic human appearance, movement, and spatial dimensions. The behavioral realism is determined by interactivity and the controlling entity. For the purposes of this discussion, it is sufficient to note that four simple characters can be created and are of the types *Simplistic Character*, which is low in form and behavior, *Superficial Avatar*, which is high in form but low in behavior, *Intelligent Unrealistic Avatar*, which is low in form but high in behavior, and *Digital Human Avatar*, which is high in form and behavior. They define a typology of avatars where the form realism part describes attributes that are relevant when looking at an avatar; these include the representation as a 2D or 3D model, its static or dynamic graphic, and human characteristics such as gender, race, age, and name [38].

Ante et al. [37] present a similar definition. They define an avatar of a user as "one or more digital representations of themselves in the digital world" [37]. At a high level, they classify them into the following types: *Customizable*, *Non-customizable*, *Self-representational*, *NonHuman* and *abstract*. They again see a high level of anthropomorphism, but they also include abstract and nonhuman characters, which can lack these attributes [37].

In consideration of the aforementioned classifications, it is evident that humanoid avatars (human) are a prominent subject of interest, particularly in the context of the metaverse. By the classification of Miao et al. [38] they are regarded as *Superficial Avatars* or *Digital Human Avatars*. By the classification of Ante et al. [37] they fit the categories of Customizable, Non-customizable and Self-representational. These humanoid avatars are most easily recognized by their silhouette, with four limbs, a torso, and a head, but also a face and clothing. All other detected avatars are then broadly subsumed with a residue class (*NonHuman*).

Anthropomorphism is described as an important feature of an avatar [37]. Using humanoid avatars makes it easier to assert human features to avatars and strengthens social interactions and general engagement in a virtual world [33]. This includes a simpler possibility of self-representing and identifying with an avatar. In addition, a higher level of

empathy, social connection, and satisfaction is reached [37]. The human-centered design approach [39] is another argument to focus on in regard to humanoid avatars.

In summary, a characterization of avatars as roughly either being *Human* or *NonHuman* is found, while *NonHuman* can still include anthropomorphic features at a lower level (but they are not required). This might make it harder to classify them as avatars than with *Human* avatars. Furthermore, there is no unified or widely used avatar characterization, and creating or extending an ontology is helpful in regard to modelling avatars.

The 256-Metaverse Records dataset [40,41] contains video-based *MVRs* collected in the wild from different metaverse virtual worlds. A sample of different avatars from the dataset is shown in Figure 2. The virtual worlds displayed all contain an avatar that is, at least, a self-representation of the user engaging in the virtual world. On the top left, a scene from Second Life [42] is displayed, followed by a snapshot of Roblox [12]. On the bottom left, a gathering in Fortnite [13] is shown, next to a scene in a restaurant in Meta Horizon Worlds. All avatars are close to a humanoid representation with different levels of abstraction. The Roblox sample displays a blocky toy-like representation and Second Life is close to a photorealistic representation, while Fortnite has a more realistic look that has some cartoonish elements. Finally, Horizon Worlds uses an oversimplified but still realistic look, but, at the same time, the avatars are missing their lower body and are free-floating. Even though there is an avatar-labeled training dataset, it is highly likely that the labels might require adaption or the videos must be converted to either images of specific sizes, formats, or something similar. Figure 2 shows different examples of avatars in the dataset. From the observations on the dataset made by the authors of this paper, virtual worlds employ *indicators* of different forms that hover over the avatars' heads, including text boxes, diamonds, or arrows. Examples of such indicators are shown in Figure 2, as seen in the white downward arrows in the lower left example or the names that hover above the heads in the upper left and lower right examples. However, such indicators are not guaranteed to be used in a virtual world or be visible in the scene.

Figure 2. Samples of the 256 Metaverse Recording dataset.

Steinert et al. [27] investigate the differences in the ontologies of common multimedia with *MVRs*. They also propose defining an avatar as a tuple of a name tag, referring to the described *Indicator*, and a character. Further, the characterization describing a character can be a text line, a 2D model, or a 3D model. Although Steinert et al. do not find an existing ontology containing an avatar, they propose extending the Large-Scale Concept Ontology for Multimedia (LSCOM) [43]. They name as a possible super-class the *perceptual agent* class. After reviewing the literature and visually inspecting *MVRs* with avatar data,

it might be easier to classify avatars when extending the found characterization. When something of type *Human* is found, it could be an avatar but could also be a simple representation of a human in a painting. For *honhuman* avatars it can be even harder. The author proposes including a *sign* in the recognition that refers to the described *Indicator* used to emphasize that a character is a player character. When such indicators are found in relation to the detection of a *human* or *NonHuman*, it may allow for easier identification of avatars. However, based on the observations in the wild, the described name tag is only one form of *Indicator* and no method and evaluation is presented.

Upon examination of the *MVRs*, it becomes evident that a significant proportion of nonhuman avatars are anthropomorphic animals or animal-like creatures. This observation has the potential to influence recognition, yet it is not reflected in existing categorization schemes.

However, the extension of the existing avatar classification by modeling the *Indicator* property, and the use in avatar detection remains a challenge.

2.2. Object Detection

The automatic detection of object instances in images and video, machine learning, in particular object detection, has proven to be efficient [44]. There are multiple algorithms from the field of supervised learning used for classification and localization, which might be applicable to the task of avatar detection. Neural networks are computational models inspired by the human brain [26], consisting of interconnected layers of nodes (neurons) that process data to recognize patterns and make predictions. Convolutional Neural Networks (CNNs) are a specialized type of neural network designed for image and video recognition, using convolutional layers to automatically detect features such as edges, textures, and shapes in visual data [26,45].

One can use existing object detection models to detect avatars. For example, avatars are similar to humans. Hence, a model successfully detecting humans could be used for avatar detection. This approach likely reduces the amount of training data needed [46], by using a combination of active learning and transfer learning. Transfer learning provides a pre-trained model that has been trained on a different dataset [46]. Active learning describes a selective annotation of only unlabeled training data with a high entropy; e.g., this could be determined by classification of the unlabeled data and then checking for data points with low probabilities assigned [46]. Therefore, only some training data have to be labeled.

A similar approach is used by Ratner et al. [47], showing three things when using their data programming framework within the field of weak supervision, where some is automatically labeled by simple solutions such as labeling function that are based on heuristics and therefore noisy, biased, or otherwise error-prone. First, Ratner et al. show that data programming can generate high-quality training datasets. Second, they demonstrate that LSTM models can be used in conjunction with data programming to automatically generate better training data. As a last point, they present empirical evidence that this is an intuitive and productive tool for domain experts.

These approaches outline that acquiring training data is not a simple or solved issue, because not even the labeling can be handled fully automatically. In theory, if the model that is planned to be trained works well, then this could also be regarded an automatic creator for labels of training data within the field of weak supervision. Other than that, these approaches might help to create more labels for training data, but still require generation of training data to label.

Other approaches that seem promising due to current success with image recognition are based on Convolutional Neural Networks (CNN) [48], basic CNNs have been extended and improved to models such as You Only Look Once (YOLO) [49] or Regional CNNs (R-CNNs) [50], which have proven to be useful for object detection. In short comparison, R-CNNs work in multiple steps which increases accuracy but reduces speed for live object

detection, while YOLO does all these steps at once which reduces complexity and increases speed, but slightly lowers detection accuracy.

The YOLO algorithm works by dividing an image into a grid, predicting bounding boxes and class probabilities for each region in a single pass through a neural network [26]. YOLO generalizes objects better compared to R-CNNs by a wide margin, e.g., after training on images of real humans, it is still able to detect abstract persons quite well in artworks. A high capability in abstraction might be a big advantage. At the same time it is really fast, allowing for a wider use case or less resource consumption due to its more simple and efficient modeling. Furthermore, it takes in the input of the entire picture including the background, which might be helpful to include contextual information, especially when trying to detect more abstract avatars. However, even if an avatar would look like something amorphic, YOLO has proven to detect unspecific objects like potholes [51,52].

YOLO's major shortcoming in accuracy is with exact detection location and multiple objects in proximity. This might limit the ability of the model when multiple avatars might be close to each other, but newer versions of YOLO are quite capable at reducing these issues [45]. In general, YOLO is an adaptable, fast, real-time object-detection method that achieves good accuracy in comparisons [45].

Our literature search could not find an approach that directly attempts to apply object detection on avatars in *MVRs*, but there are multiple highly potent candidate algorithms at hand. Some, such as a pre-trained YOLO model, might work quite well without further modification, since they are able to generalize well from human images to humanoid representations. Then, specialized training data is provided to such models. A remaining challenge is the modeling and implementation of an adapted YOLO model, specialized by transfer learning on the avatar class annotated *MVRs*.

2.3. Summary

The existing avatar classification is inadequate for object detection purposes. It includes irrelevant characteristics and fails to account for a crucial indicator used widely in metaverse virtual worlds to identify avatars. Although effective object-detection methods have been extensively researched and provide a solid foundation for training on avatar-specific classes, our literature review reveals a gap in object detection specifically tailored for avatars. In the next section, we present our modeling as a solution for these challenges.

3. Modeling

This section presents our modeling and design work. Based on the body of research, we modeled an avatar classification, and selected an ML model for Avatar Detection. We use the Unified Modeling Language (UML) [53] for our modeling work.

3.1. Avatar Classification

As presented, the existing classification of avatars lacks supporting object detection. We propose an avatar class model that can be deduced from the aforementioned research that includes two types of avatars *HumanAvatar* and *NonHumanAvatar*, visualized in the class diagram in Figure 3. The classes contain an extension part, which includes attributes of these classes via aggregation, adding *Indicator* to *Human* to form a *HumanAvatar* and to *NonHuman* to form a *NonHumanAvatar*.

The class model can be used in information systems and, therefore, has additional attributes. The core of the model is the `Avatar` class which contains a unique ID of the avatar displayed. The location given by `xCord` and `yCord` and the size of the box, centered around these coordinates, given by `xBox` and `yBox`. Measured in pixels, their data types are integer. The confidence value of a detector can be stored in the confidence attribute.

When looking at the extension part of the model, a human-like appearance is an essential attribute for *HumanAvatar* detection; therefore, it has exactly one association with the *human* class representing this attribute. Similarly, the *NonHuman* class is an essential part of the *NonHumanAvatar*. There might be issues with detecting avatars using only this

characteristic. One reason for error might be the detection of a human or human-like image or picture that is not an avatar. A typical issue could also be that an anthropomorphic appearance of a figure might be given but is not controlled by a human or machine user. The biggest issue might arise when trying to detect nonhuman avatars which feature little to no human-like appearances. To clarify, the *Human* and *NonHuman* classes are both anthropomorphic. *NonHuman* is not a simple negation of *Human* but rather a classification related to *Human* while not being *Human*, typically featuring less anthropomorphic features. Both classes represent a detected instance of an avatar in an *MVR*.

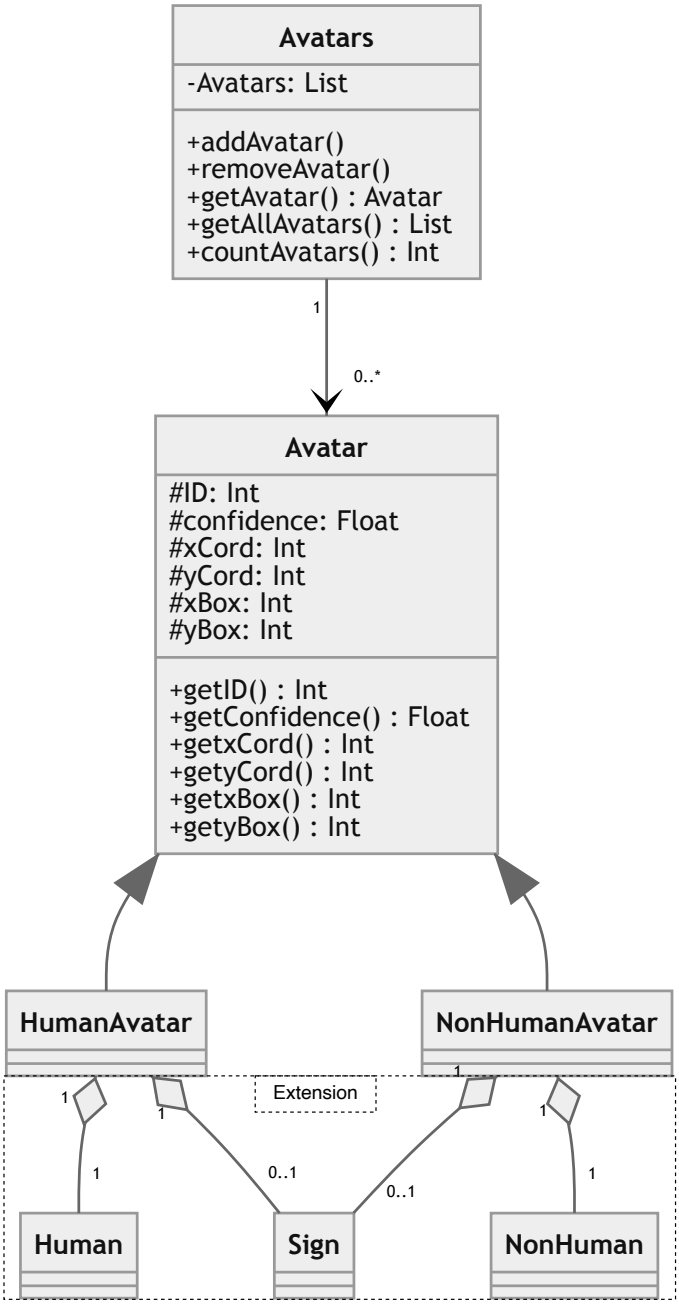

Figure 3. Information UML Class Diagram of *Avatars*.

The second attribute for avatar detection is the *Indicator*. Possible instances of an *Indicator* could be text boxes, i.e., name tags, symbols (such as arrows), or other indicators, which mostly hover above the head of an avatar. An *Indicator* might be especially useful for *NonHumanAvatar* classes if non-obvious representations, such as normal animals, are used for a *NonHuman* or, in general, if the shown avatar is very small in its representation.

Indicators may only appear in specific circumstances, i.e., depending on the distance to the avatar or the context in the virtual world. Including the second attribute, *Indicator*, in the class model, the aggregation of *Human* and *Indicator* make up the *HumanAvatar* class. The *Indicator* is kept optional, to respect a possible disappearance in context, which is modeled by the 1 to 0.1 relationship with the *Indicator*. Similarly, the *Indicator* is also added as an optional feature for the *NonHumanAvatar* class.

A common way to define such classifications is through ontologies, e.g., the RDF [54] -based ontology in Notation 3 (N3) [55] displayed in Listing 1. An *Avatar* can be added as a subclass of the perceptual agent in the existing LSCOM ontology. The *HumanAvatar* and *NonHumanAvatar* subclasses can also be added.

Including an *Indicator*, such as the proposed name tag [27], in conjunction with *Human* or *NonHuman* detection is thought to be a valid way to create the *HumanAvatar* and *NonHumanAvatar* class. Thus, it is included as a relation or, in RDF terms, a *Property*.

Listing 1. Information Model Formal Language Specification of *Avatars*.

```
@prefix : <https://github.com/JokerFelix/MasterThesisCode/AvatarOntology> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
# Define Classes
:Avatar rdf:type rdfs:Class .

: AvatarHuman rdfs:subClassOf :Avatar .
: AvatarNonHuman rdfs:subClassOf :Avatar .

:Human rdf:type rdfs:Class .
: NonHuman rdf:type rdfs:Class .
: Sign rdf:type rdfs:Class .

# Define Properties for Aggregation
: includesHuman rdf:type rdf:Property ;
    rdfs:domain :AvatarNonHuman ;
    rdfs: range: Human .

: includesSignForHuman rdf:type rdf:Property ;
    rdfs: domain: AvatarHuman ;
    rdfs: range : Indicator .

:includesSignForNonHuman rdf:type rdf:Property ;
    rdfs: domain: AvatarNonHuman ;
    rdfs: range : Indicator .

: includesNonHuman rdf:type rdf:Property ;
    rdfs:domain :AvatarHuman;
    rdfs: range :NonHuman .
```

The resulting class model presents four distinct classes which can be used in *Avatar Detection* for *MVRs*.

3.2. Avatar Detector Model

The state-of-the-art object detection mechanisms promise good results with proper transfer learning. Hence, our avatar detector, referred to as $ADET$ is based on an existing model trained with avatar training data. The selected algorithm family is YOLO. The training is based on a training and test dataset and a set of hyperparameters. These hyperparameters influence the model's exact structure and training process to converge to

a decent minimum of the loss function. Common hyperparameters include the learning rate, batch, and epoch sizes. For our training, the software performs one optimization step, using the optimizer selected for every batch and for every epoch, in the direction of steepest descent provided by the negative gradient, accessing the model's parameters and basing the magnitude of change on the learning rate.

YOLO is available in multiple network architectures and with different sets of hyperparameters. For *ADET*, the YOLOv7 standard model is selected with minor configuration changes to fit the amount and types of classes selected for avatar detection. The exact set of hyperparameters is then determined in the implementation by trial and error, also called hyperparameter tuning; however, as a starting point, a default set of hyperparameters is used again.

4. Implementation

This section describes our results in terms of the Nunamaker phase system's development. The results comprise a created dataset for training and evaluation of the object recognition, as well as the selection and parameterization of object recognition models.

4.1. ADET Dataset

Dataset Statistics: We created a dataset of 408 images sampled from the 256-Metaverse Records dataset [30]. We refer to this as *ADET-DS*. *ADET-DS* contains 716 labeled avatars, 478 instances of class *HumanAvatar*, and 238 instances of class *NonHumanAvatar*.

The original dataset consists of video files. The first step of selecting individual frames and marking their timestamps is tedious work. The 256 Recordings dataset is diverse in terms of the environments and avatars displayed while being in the video format, providing per second roughly 30 possible candidate frames to select. With the provided *MVRs* covering over 8 hours, approximately 882,719 candidate frames are provided. Hence, the annotator skipped through the footage at a higher speed or to specific time stamps. When investing only a little time on each frame, a clear indicator, such as a name tag, a health bar, a highlighting contour line around a character, or a symbolic indicator such as an arrow hovering over a character's head, is easier to recognize and allowed for faster recognition of an avatar, increasing the efficiency of the process. This supports the thesis that indicators are relevant in the recognition process. One risk is to rely too much on the presence of these indicators, creating an imbalanced and unrealistic dataset, because some avatar representations do not include this *Indicator* attribute. Thus, relying solely on these indicators when quickly screening through frames should be avoided.

For frame extraction and labeling, a custom Python script using FFmpeg [56] and LabelImg [57] is used. The annotation of frames is achieved manually by a human expert in the field. LabelImg is used to draw bounding boxes and label them with *HumanAvatar* and *NonHumanAvatar* as text names, as suggested by the classification. Figure 4 displays an example of the labeling process. The instances of cat and unicorn feature a clear *Indicator*-type indicator, including their names. The *NonHuman* classification is made because even though the form realism is low, it still features quite a bit of anthropomorphic features, such as a 3D model with arms, torso, legs and an overly big head. The race, even though not human, is still clearly present as a cat and a unicorn. They also feature a name, displayed with *Indicator*. Although they are probably most likely NPC controlled, they are classified as a *NonHumanAvatar*. If the race is more clearly that of a human, then it is classified as *Human* with a low form of realism. This example demonstrates that it is non-trivial to assign the correct class labels and that, while the indicator helps to identify avatars, it is sometimes not present or unique to an avatar.

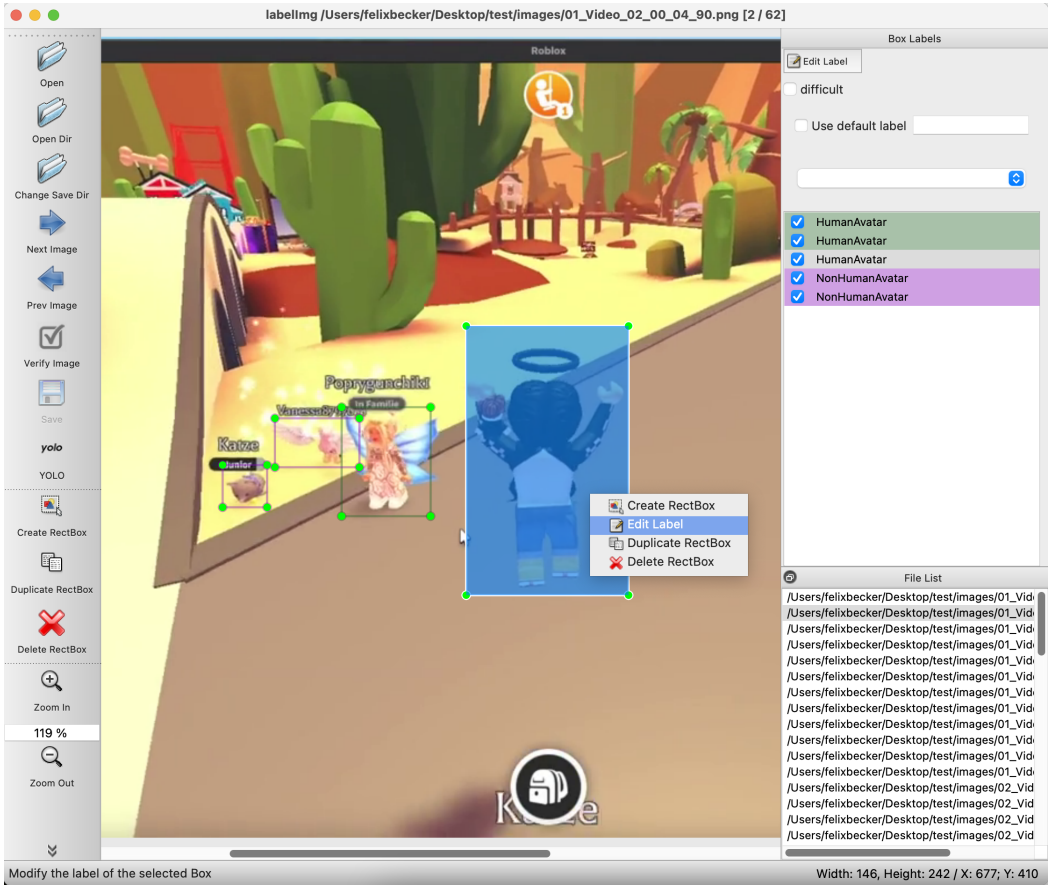

Figure 4. Example image annotation of avatars in *MVRs* with *LabelImg*.

4.2. Avatar Detector

The actual detection model is mostly the default YOLOv7 implementation [58]. The final implementation code is provided in [59,60].

The first YOLOv7 implementation created is called $YOLO^{base}$ or $YOLO^{base}$, representing the original vanilla YOLOv7 [58] model, and was trained using the COCO dataset [61]. Without any further training, it can simply detect *person*-type classes, which are then interpreted as avatar detections. The second created YOLOv7 implementation is referred to as $ADET$, representing the adapted YOLO model, and it specializes in transfer learning in regard to the two avatar classes of *HumanAvatar* and *NonHumanAvatar*. Here, $ADET$, or $ADET^{indicator}$, is the avatar detection implementation based on $YOLO^{base}$ and is trained with the avatar training data, yielding different weights and biases; it also features minimal adjustments for the hyperparameters of the P5 configuration [62] delivered with YOLOv7. The number of known classes was adopted for the two classes *HumanAvatar* and *NonHumanAvatar*. The learning rate $lr0$ was set to 0.001. A third model was trained, referred to as $ADET^{NI}$, which is identical to $ADET^{indicator}$ but was trained with modified test images in which the feature of type *Indicator* was removed; therefore, no indicator is visible. The same configuration of $ADET^{indicator}$ was used for training $ADET^{NI}$.

For the final training of both models, a batch size of 16, an image size of 640, and a maximum of 1300 epochs are used. The dataset *ADET-DS* was split in a 70/15/15 ratio for train/test/validation. During the hyperparameter tuning test that was used to compare the detection results of models trained with different sets of hyperparameters, the validation part was left unseen by the model until the final model was selected and evaluated.

Table 1 summarizes the different avatar detectors. $YOLO^{base}$, a plane yolov7 model which can detect persons, can be used as a baseline. $ADET^{indicator}$ can classify the presented classification model with indicators trained with avatar-specific data. In contrast, $ADET^{NI}$ is trained on the classification without the indicators. An example output of the detected

images is presented in Figure 5. It demonstrates that the avatar instances can be detected, but some avatars are missing, such as the cat, and other detection bounding boxes differ when compared to the annotations in Figure 4.

Table 1. Overview of the avatar detectors.

Detector	$YOLO^{base}$	$ADET^{indicator}$	$ADET^{NI}$
Base Algorithm	YOLOv7	YOLOv7	YOLOv7
Classes	MS COCO classes, e.g., Person	HumanAvatar, NonHumanAvatar	HumanAvatar, NonHumanAvatar
Training Dataset	MS COCO	*ADET-DS* with indicators in bounding box annotations	*ADET-DS* without indicators in bounding box annotations
Configuration	-	p5	p5
Learning rate	-	0.001	0.001

Figure 5. Example of detected avatar instances.

The implementation shows that existing object detection algorithms can be used to detect avatars. The limitation of the training data is noteworthy, with only 687 instances of class *HumanAvatar* and 687 instances of class *NonHumanAvatar*. Despite the research showing that transfer learning is achievable with 1500 images [51], the diversity of avatars suggests that more training data could yield better results. The next section provides a detailed evaluation of the effectiveness.

5. Evaluation

In this section, we present and discuss the results of our experiments for the avatar detector *ADET*. In addition, an ablation study analyzes the impact of the *Indicator* in regard to *ADET*.

5.1. Evaluation of the Avatar Detection

The following experiments evaluate the effectiveness of the avatar detector *ADET*. First, a baseline is created to compare the effectiveness of *ADET*.

5.1.1. Baseline

The first test is performed using the vanilla $YOLO^{base}$ detector to test the performance of detecting instances of the class Avatar. Since $YOLO^{base}$ is trained on COCO classes, all the labels of type Avatar in the validation dataset are renamed as person.

Dataset Settings: The $YOLO^{base}$ was trained with the COCO dataset. The *ADET-DS* test split was used for the evaluation.

Training and Inference: The $YOLO^{base}$ model was used pre-trained from the model zoo [58].

Metrics: The common metrics Precision, Recall, mean Average Precision (mAP) [44] at threshold T (mAP@T), and Intersection over Union (IOU) [44] are used to evaluate the performance of the models.

Results: When using *person* only as an approximation for both avatar classes, the model delivered an mAP@0.5 of 0.582. This is already a decent result, which also implies that, for a fixed IOU of 0.5, the AP is also the same as mAP since only one class is included. The Precision Recall (PR) curve is shown in Figure 6.

The default class *person* included in the COCO dataset is obviously also a good approximation for the avatar class *Human* on its own; however, overall, the results of the $YOLO^{base}$ are mediocre.

The $F1$ score is quite consistent for different threshold levels, with higher confidence values falling off, and it is optimal at 0.155, as shown in Figure 7.

An in-depth analysis of the predictions of the $YOLO^{base}$ model shows that the model rarely identified an avatar as a horse, cow, or similar class that can be regarded a subclass of *NonHuman* in the sense of our modeling. Most of the time, the *NonHuman* avatars are detected as a *person*.

With respect to the class *Indicator*, there are two detections of *frisbee* close to an avatar that are actually indicators belonging to an avatar. An example of detection is shown in Figure 8.

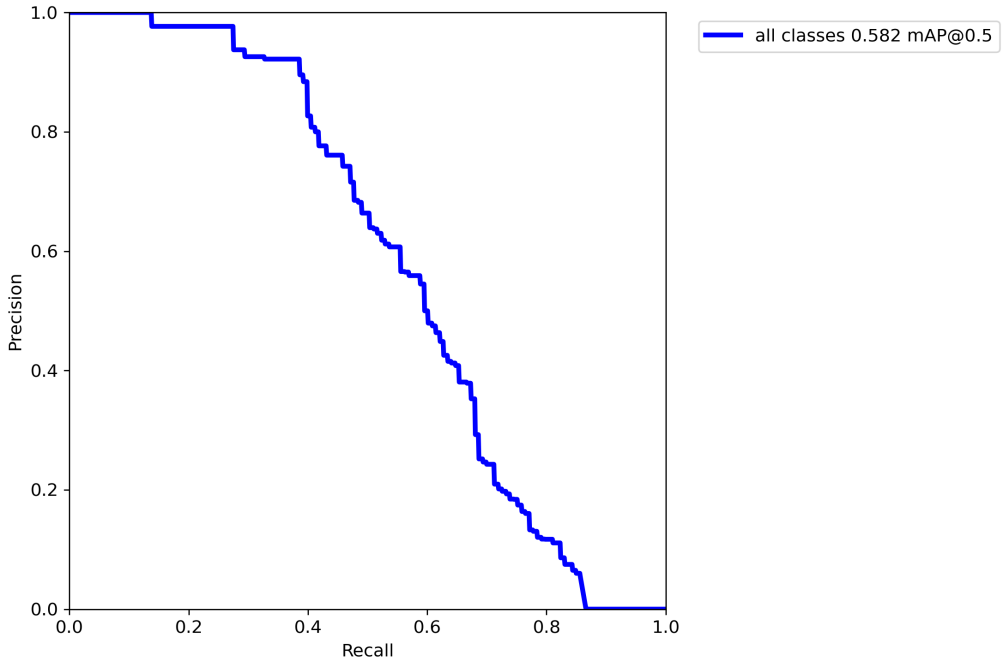

Figure 6. AP and mAP of $YOLO^{base}$ on Test Data.

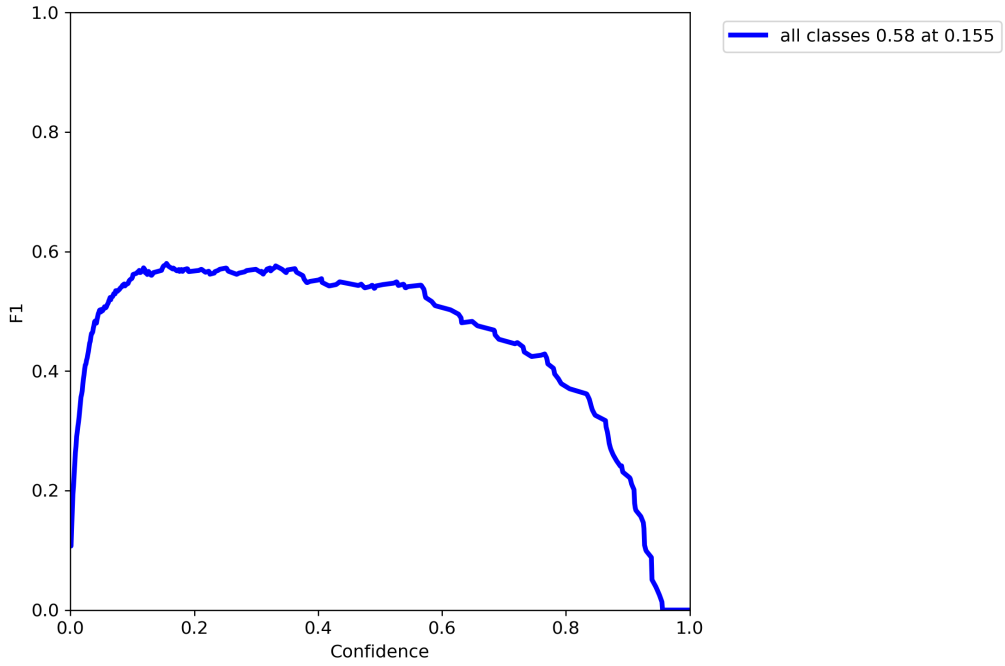

Figure 7. *F*1 Score for Different Thresholds of $YOLO^{base}$.

Figure 8. Predicted Avatars $YOLO^{base}$ on Test Data of *ADET-DS*.

These results support the idea of the anthropomorphic appearance of avatars and the idea of *NonHuman* representing something that is not actually a human but close to it. The results also show that animal classes cannot be suitable subclasses for *NonHuman*.

Indicators of type *Indicator* could usefully be included in an avatar characterization, but the standard recognition model shows only a weak indication of this. This might be caused by the training data classes themselves. There are no well-fitting classes of indicators in the basic COCO training data, such as text boxes, arrows, name tags, etc. The included classes of the COCO dataset, such as *street signs* and *frisbee*, are confusing in terms of avatar detection and give false positive results. Thus, the $YOLO^{base}$ model is not a good fit to differentiate avatar classes into *HumanAvatar* and *NonHumanAvatar*.

5.1.2. Avatar Detector

The adapted YOLO model, which specialized in transfer learning on the avatar class using the avatar-annotated *MVRs*, is presented next.

Dataset Settings: *ADET-DS* is used in the 70/15/15 train/test/val split.

Training and Inference: $ADET^{indicator}$ was used after being pre-trained with COCO from the YOLOv7 model zoo. Further training with avatar data was done for 1300 epochs

Using the $ADET^{indicator}$ detector, in contrast to the $YOLO^{base}$ detector, *HumanAvatar* and *NonHumanAvatar* can be detected separately.

Metrics: Previous metrics are used.

Results: The *mAP*@0.5 for both classes is at 0.825; the *HumanAvatar* class is at 0.905 and the *NonHumanAvatar* achieved an *AP* of 0.745. The classes *Human*, *NonHuman*, and *Indicator*, or any other class, are not included explicitly in the training.

The results are shown in Figure 9.

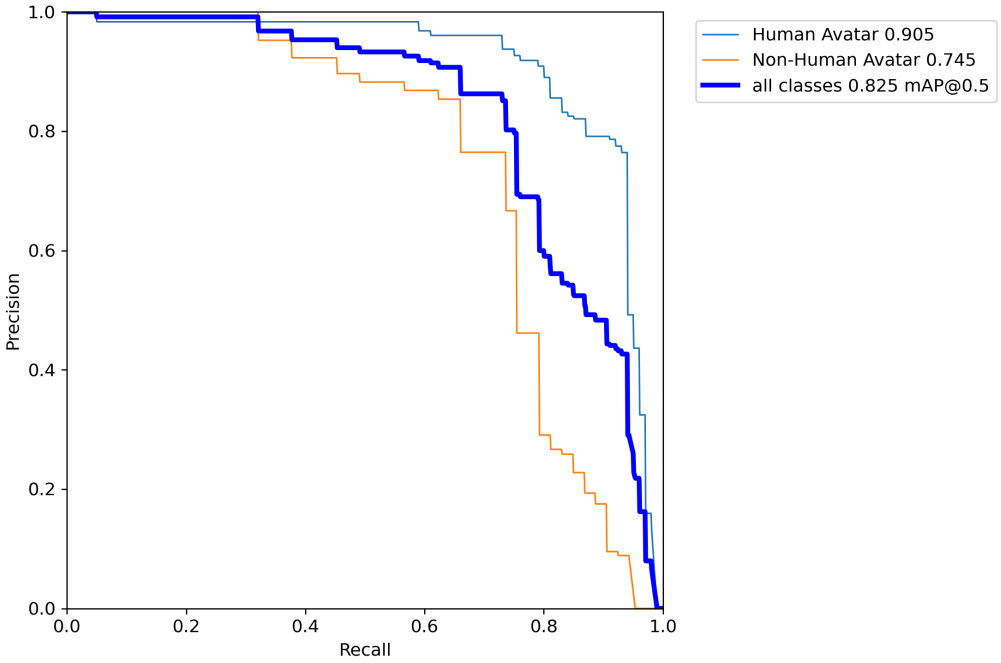

Figure 9. AP and mAP of $ADET^{indicator}$ on test data of *ADET-DS*.

The *F*1 score is quite consistent for different threshold levels and is optimal at 0.245, as shown in Figure 10.

An example of the detection is shown in Figure 11. The same sub-sample of test data is selected as what was used for $YOLO^{base}$. Only relevant objects such as *Avatars* are detected, especially *NonHumanAvatars* which are correctly classified.Detecting far-away or tiny avatars seems easier for the model, and, at the same time, the bounding boxes are comparable in precision.

Table 2 summarizes the key statistics comparing $ADET^{indicator}$ and $YOLO^{base}$. The adapted $ADET^{indicator}$ implementation prototype of *ADET* trained on the avatar-annotated *MVR* is clearly an overall improvement. There is a percentage point rise of 24.5 in *mAP*, indicating a relative improvement of 0.422.

The detection of the *NonHumanAvatar* class is weaker compared to the *HumanAvatar* class with an absolute delta of 0.160 percentage points; however, in contrast to $YOLO^{base}$, the detection is at least possible. For future work, the performance could be increased for this class by including more training data of *NonHumanAvatars*, as this class is more diverse.

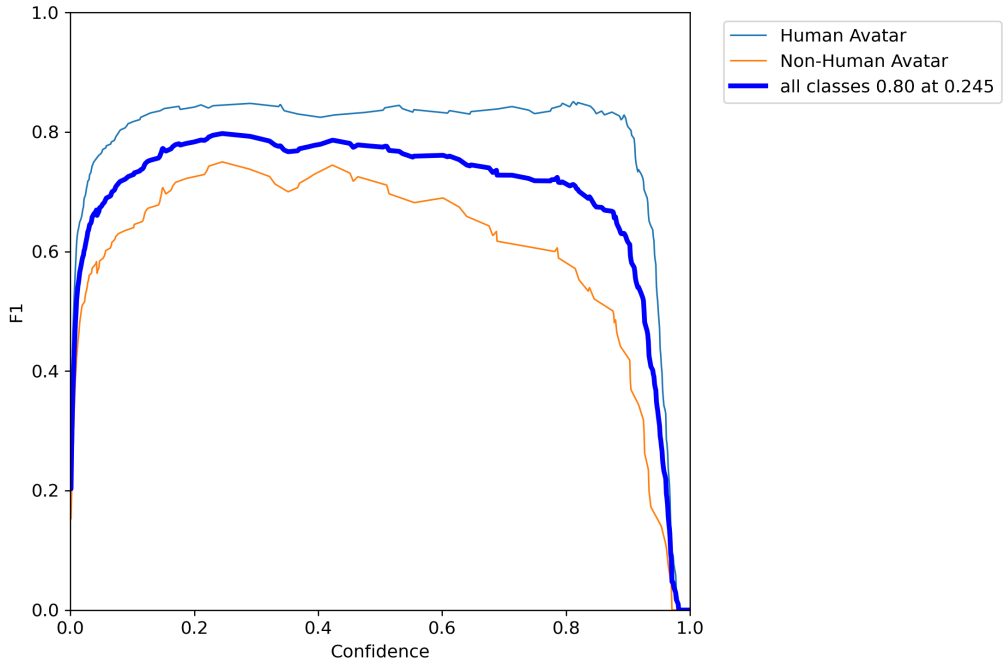

Figure 10. *F*1 Score for Different Thresholds of $ADET^{indicator}$.

Figure 11. Predicted Avatars $ADET^{indicator}$ on Test Data of *ADET-DS*.

Table 2. Comparison of $ADET^{indicator}$ and $YOLO^{base}$ Detection Metrics.

		AP ↑	mAP@0.5 ↑	F1@Optimum ↑
$ADET^{indicator}$	Class HumanAvatar	0.905		
$ADET^{indicator}$	Class NonHumanAvatar	0.745		
$ADET^{indicator}$	Both Classes		0.825	**0.800**
$YOLO^{base}$	Person		0.582	0.580

5.2. Ablation Study: Evaluating the Avatar Indicator

An ablation study investigates the performance of an AI system by removing certain components to understand the contribution of the component to the overall system. The

experiment is carried out to test whether the *Indicator* indicator class is meaningful to an avatar classification. Thus, the avatar-trained YOLO model $ADET^{indicator}$ is tested on data that include the *Indicator* indicator ($ADET^{indicator}$) and data that exclude it ($ADET^{NI}$). The idea is that these models can explicitly provide the important detected features that make up an avatar. Therefore, the complex implicit knowledge involved when a human labels these avatars is returned quantitatively by the avatar detection algorithm. If elements such as *Human* and *Indicator* are detected by the algorithm, this is a clear indicator that the suggested model includes relevant features of an avatar.

Dataset Settings: We split *ADAT-DS* into test/train/val and edited out indicators such as name tags or graphical symbols.

Training and Inference: $ADET^{NI}$ was used and was pre-trained with COCO from the YOLOv7 model zoo. Further training with avatar data was done for 1300 epochs.

Metrics: Previous metrics are used.

Results: In regard to testing $ADET^{indicator}$ while including the *Indicator* indicators first, the mAP@0.5 for both classes is at 0.825, the *HumanAvatar* class is at 0.905 and the *NonHumanAvatar* achieves an AP of 0.745. Subclasses such as *Human*, *NonHuman*, and *Indicator*, as well as any other class, are not included directly as subclass detections. As shown in Table 3, the results for mAP@0.5 dropped by 0.09 percentage points to 0.735, the *HumanAvatar* class dropped slightly by 2.4 percent, from 0.905 to 0.883, and the *NonHumanAvatar* class dropped in AP from 0.745 to 0.586. The decrease by 21.3 percent is quite significant and most relevant to the overall mAP drop. This shows that *Indicator* is an important element for *NonHumanAvatar* detections and that it slightly helps with *HumanAvatar* detections, as the same model showed inferior performance in regard to detecting avatars when the indicator class *Indicator* was not included for the same test images.

Table 3. Comparison with and without Indicator.

Model	Train Set	AP Human ↑	AP Non-Human ↑	mAP ↑
ADET	Indicator	**0.905**	**0.745**	**0.825**
ADET	No Indicator	0.883	0.586	0.735
YOLO	COCO	0.582 *	-	

* class person.

Two reasons may be responsible for this, the first being the large variability in the *NonHuman* class compared to the *Human* class. The second reason is the often observed similarity between *NonHuman* and *Human*. In future work, it might be interesting to identify more *NonHuman* classes and to repeat these tests for more diverse training and test data. The experiments allow for evaluating the avatar characterization suggested and, especially, the positive relevance of the *Indicator* class as required.

Furthermore, the results presented previously suggest that an AI-based avatar detector such as $ADET^{indicator}$ uses this indicator to classify an avatar. The classes *Indicator* and *NonHuman* are both detected for the *NonHumanAvatar* detections, while it is mostly *Human* that is detected with *HumanAvatar*. This supports the thesis that *Indicator* is a reasonable extension that helps to spot avatars.

In future work, utilizing more significant subclasses of the *NonHuman* class in the context of avatars might result in another meaningful extension of avatar classifications.

6. Discussion and Future Work

In this paper, we have explored the novel task of *avatar detection* in *MVRs*, presenting a significant contribution to the fields of MMIR and artificial intelligence. The presented avatar classification model provides a classification for object detection models and includes a novel approach to incorporating visual indicators in the classification.

Utilizing the YOLO algorithm, specifically the YOLOv7 architecture, the $ADET$ model was developed and fine-tuned to detect avatars within various virtual environments. This approach achieved significant improvements in accuracy of 0.825 mAP@0.5, particularly

compared to standard models such as $YOLO^{base}$, which only detected human-like avatars. The research demonstrates a notable improvement in detection accuracy that is mainly attributed to the specialized training on avatar-specific datasets. The inclusion of visual indicators, such as name tags and hovering arrows, significantly improves the performance of the model, highlighting its importance in the detection process. This model effectively addresses the complexities inherent in identifying avatars, particularly those with varying degrees of anthropomorphism and abstract features, thereby advancing the capabilities of current object-detection methodologies.

Despite these advancements, several limitations were observed. First, the reliance on indicators introduces a vulnerability in virtual worlds where such indicators are not always present, limiting the robustness of the model for nonhuman avatars in certain contexts. Second, the dataset may not fully represent the diversity of avatars found in all metaverse environments, particularly for nonhuman forms that exhibit a wide variety of shapes and behaviors.

The $ADET$ models provide avatar detectors that can be used in MVR-specific MMIR systems to recognize avatars in images and videos. This allows at least MVRs or segments to be differentiated into MVRs with and without avatars. The $ADET$ models can also be used as a foundation for human–object and human–human interaction detection.

Future research should aim to expand the training datasets to encompass a broader range of avatar types and virtual environments. This expansion is essential for improving the generalizability and robustness of the model, particularly for nonhuman avatars, which exhibit a wider variety of forms and characteristics. Moreover, addressing the model's current dependency on visual indicators should be a key focus, as reducing this reliance would make the model more adaptable to environments with fewer or no external cues. Additionally, refining the avatar classification schema to include more granular subclasses will enhance the detection accuracy and applicability of the model across different metaverse platforms. For MMIR use cases, the identification of an avatar, or at least the detection of individual avatars over a temporal segment in a video, could improve search results.

We also suggest exploring additional object detection algorithms, such as R-CNN or CLIP-based models, which may provide different strengths compared to YOLO-based models. These models could be tested against $ADET$ in future studies to identify potential areas of improvement, particularly in handling nonhuman avatars and reducing the reliance on indicators.

To conclude, the idea of this paper is to add avatar detections as semantic information non-manually to images taken in virtual worlds such as the metaverse. Our findings demonstrate that, by adapting existing models, such as YOLO, with specific avatar classifications, we achieved superior results compared to generic object detection models. This work lays the groundwork for future improvements and applications in regard to avatar detection in the metaverse and other virtual environments. Although there is room for further research, this work demonstrates that, by combining and adapting the existing model, improvements on the YOLO-COCO person-class baseline can be achieved.

Author Contributions: Conceptualization and methodology: F.B., P.S., S.W. and M.L.H. Software, validation, formal analysis, investigation, resources, data curation, writing: F.B. and P.S. Review, editing, and supervision: P.S., S.W. and M.L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gartner Inc. *Gartner Predicts 25% of People Will Spend at Least One Hour per Day in the Metaverse by 2026*; Gartner Inc.: Stamford, CT, USA, 2022.
2. Mystakidis, S. Metaverse. *Encyclopedia* **2022**, *2*, 486–497. [CrossRef]
3. Ritterbusch, G.D.; Teichmann, M.R. Defining the Metaverse: A Systematic Literature Review. *IEEE Access* **2023**, *11*, 12368–12377. [CrossRef]
4. KZero Worldwide. *Exploring the Q1 24' Metaverse Radar Chart: Key Findings Unveiled*; KZero Worldswide: Dubai, United Arab Emirates, 2024.
5. Karl, K.A.; Peluchette, J.V.; Aghakhani, N. Virtual Work Meetings During the COVID-19 Pandemic: The Good, Bad, and Ugly. *Small Group Res.* **2022**, *53*, 343–365. [CrossRef] [PubMed]
6. Meta Platforms, Inc. *Meta Connect 2022: Meta Quest Pro, More Social VR and a Look Into the Future*; Meta Platforms, Inc.: Menlo Park, CA, USA, 2022.
7. Takahashi, D. Nvidia CEO Jensen Huang Weighs in on the Metaverse, Blockchain, and Chip Shortage. *GamesBeat News*, 12 June 2021.
8. Apple Inc. Apple Vision Pro Available in the U.S. *Newsroom*, 2 February 2024.
9. INTERPOL. *Grooming, Radicalization and Cyber-Attacks: INTERPOL Warns of 'Metacrime'*; INTERPOL: Lyon, France, 2024.
10. Linden Lab. *Official Site Second Life*; Linden Lab: San Francisco, CA, USA, 2024.
11. Decentraland. Official Website: What Is Decentraland? 2020. Available online: https://decentraland.org (accessed on 9 June 2023).
12. Corporation, R. Roblox: About Us. 2023. Available online: https://www.roblox.com/info/about-us?locale=en_us (accessed on 3 November 2023).
13. Games, E. FAQ, Q: What Is Fortnite? 2023. Available online: https://www.fortnite.com/faq (accessed on 6 November 2023).
14. Meta Platforms, Inc. *Horizon Worlds | Virtual Reality Worlds and Communities*; Meta Platforms, Inc.: Menlo Park, CA, USA, 2023.
15. Wikipedia. Virtual World, 2023. Page Version ID: 1141563133. Available online: https://en.wikipedia.org/wiki/Virtual_world (accessed on 8 March 2023).
16. Lochtefeld, J.G. *The Illustrated Encyclopedia of Hinduism*; The Rosen Publishing Group, Inc.: New York, NY, USA, 2002.
17. Bartle, R. *Designing Virtual Worlds*; New Riders Games: Indianapolis, IN, USA, 2003.
18. Steinert, P.; Wagenpfeil, S.; Frommholz, I.; Hemmje, M.L. Towards the Integration of Metaverse and Multimedia Information Retrieval. In Proceedings of the 2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE), Milano, Italy, 25–27 October 2023; pp. 581–586. [CrossRef]
19. Ksibi, A.; Alluhaidan, A.S.D.; Salhi, A.; El-Rahman, S.A. Overview of Lifelogging: Current Challenges and Advances. *IEEE Access* **2021**, *9*, 62630–62641. [CrossRef]
20. Bestie Let's Play. Wir Verbringen Einen Herbsttag mit der Großfamilie!!/Roblox Bloxburg Family Roleplay Deutsch, 2022. Available online: https://www.youtube.com/watch?v=sslXNBKeqf0 (accessed on 28 February 2023).
21. Uhl, J.C.; Nguyen, Q.; Hill, Y.; Murtinger, M.; Tscheligi, M. xHits: An Automatic Team Performance Metric for VR Police Training. In Proceedings of the 2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE), Milano, Italy, 25–27 October 2023; pp. 178–183. [CrossRef]
22. Koren, M.; Nassar, A.; Kochenderfer, M.J. Finding Failures in High-Fidelity Simulation using Adaptive Stress Testing and the Backward Algorithm. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 5944–5949. [CrossRef]
23. Li, X.; Yalcin, B.C.; Christidi-Loumpasefski, O.O.; Martinez Luna, C.; Hubert Delisle, M.; Rodriguez, G.; Zheng, J.; Olivares Mendez, M.A. Exploring NVIDIA Omniverse for Future Space Resources Missions. In Proceedings of the Space Resources Week 2022, Luxembourg, 3–5 May 2022.
24. NVIDIA Corp. *NVIDIA DRIVE Sim*; NVIDIA Corp.: Santa Clara, CA, USA, 2024.
25. Rüger, S.; Marchionini, G. *Multimedia Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2010; OCLC: 1333805791.
26. Zhao, Z.Q.; Zheng, P.; Xu, S.T.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
27. Steinert, P.; Wagenpfeil, S.; Frommholz, I.; Hemmje, M.L. Integration of Metaverse Recordings in Multimedia Information Retrieval. In Proceedings of the ICSCA 2024, Bali Island, Indonesia, 1–3 February 2024; pp. 137–145. [CrossRef]
28. TSB Gaming Ltd. *The Sandbox Game—Own the Future. Play, Create, Earn*; TSB Gaming Ltd.: St. Julians, Malta, 2024.
29. Abdari, A.; Falcon, A.; Serra, G. Metaverse Retrieval: Finding the Best Metaverse Environment via Language. In Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval, Ottawa, ON, Canada, 2 November 2023; pp. 1–9. [CrossRef]
30. Steinert, P. 256-MetaverseRecordings-Dataset Repository, 2024. Original-Date: 2024-01-12T07:26:01Z. Available online: https://github.com/marquies/256-MetaverseRecordings-Dataset (accessed on 20 January 2024).
31. Nunamaker, J.; Chen, M. Systems development in information systems research. In Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences, Kailua-Kona, HI, USA, 2–5 January 1990; Volume 3, pp. 631–640. [CrossRef]
32. Anderson, J.; Rainie, L. The metaverse in 2040 Pew Research Center, Washington, DC USA. 2022. Available online: https://www.pewresearch.org/internet/2022/06/30/the-metaverse-in-2040/ (accessed on 20 January 2024).

33. Kim, D.Y.; Lee, H.K.; Chung, K. Avatar-mediated experience in the metaverse: The impact of avatar realism on user-avatar relationship. *J. Retail. Consum. Serv.* **2023**, *73*, 103382. [CrossRef]
34. Triberti, S.; Durosini, I.; Aschieri, F.; Villani, D.; Riva, G. Changing Avatars, Changing Selves? The Influence of Social and Contextual Expectations on Digital Rendition of Identity. *Cyberpsychol. Behav. Soc. Netw.* **2017**, *20*, 501–507. [CrossRef] [PubMed]
35. Fong, K.; Mar, R.A. What Does My Avatar Say About Me? Inferring Personality from Avatars. *Personal. Soc. Psychol. Bull.* **2015**, *41*, 237–249. [CrossRef] [PubMed]
36. Franceschi, K.; Lee, R.; Zanakis, S.; Hinds, D. Engaging Group E-Learning in Virtual Worlds. *J. Manag. Inf. Syst.* **2009**, *26*, 73–100. [CrossRef]
37. Ante, L.; Fiedler, I.; Steinmetz, F. Avatars: Shaping Digital Identity in the Metaverse. 2023. Available online: https://www.blockchainresearchlab.org/wp-content/uploads/2020/05/Avatars-Shaping-Digital-Identity-in-the-Metaverse-Report-March-2023-Blockchain-Research-Lab.pdf (accessed on 17 July 2023).
38. Miao, F.; Kozlenkova, I.V.; Wang, H.; Xie, T.; Palmatier, R.W. An Emerging Theory of Avatar Marketing. *J. Mark.* **2022**, *86*, 67–90. [CrossRef]
39. Mourtzis, D.; Panopoulos, N.; Angelopoulos, J.; Wang, B.; Wang, L. Human centric platforms for personalized value creation in metaverse. *J. Manuf. Syst.* **2022**, *65*, 653–659. [CrossRef]
40. Steinert, P.; Wagenpfeil, S.; Hemmje, M.L. 256-MetaverseRecords Dataset. 2023. Available online: https://www.patricksteinert.de/256-metaverse-records-dataset/(accessed on 3 January 2024).
41. Steinert, P.; Wagenpfeil, S.; Frommholz, I.; Hemmje, M. 256 Metaverse Recording Dataset. In Proceedings of the ACM Multimedia 2024, Melbourne, Australia, 28 October–1 November 2024.
42. Linden Research, Inc. SecondLife. 2023. Available online: https://secondlife.com (accessed on 3 November 2023).
43. Naphade, M.; Smith, J.; Tesic, J.; Chang, S.F.; Hsu, W.; Kennedy, L.; Hauptmann, A.; Curtis, J. Large-scale concept ontology for multimedia. *IEEE MultiMed.* **2006**, *13*, 86–91. [CrossRef]
44. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
45. Kaur, R.; Singh, S. A comprehensive review of object detection with deep learning. *Digit. Signal Process.* **2023**, *132*, 103812. [CrossRef]
46. Rauch, L.; Huseljic, D.; Sick, B. Enhancing Active Learning with Weak Supervision and Transfer Learning by Leveraging Information and Knowledge Sources. In Proceedings of the Workshop on Interactive Adaptive Learning co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Grenoble, France, 23 September 2022; pp. 27–42
47. Ratner, A.J.; De Sa, C.M.; Wu, S.; Selsam, D.; Ré, C. Data Programming: Creating Large Training Sets, Quickly. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016; Volume 29.
48. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 6999–7019. [CrossRef] [PubMed]
49. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475. [CrossRef]
50. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
51. Ukhwah, E.N.; Yuniarno, E.M.; Suprapto, Y.K. Asphalt Pavement Pothole Detection using Deep learning method based on YOLO Neural Network. In Proceedings of the 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 28–29 August 2019; pp. 35–40. [CrossRef]
52. Dharneeshkar, J.; Soban Dhakshana, V.; Aniruthan, S.A.; Karthika, R.; Parameswaran, L. Deep Learning based Detection of potholes in Indian roads using YOLO. In Proceedings of the 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–28 February 2020; pp. 381–385. [CrossRef]
53. Rumpe, B. *Modeling with UML*; Springer: Berlin/Heidelberg, Germany, 2016.
54. W3C. *RDF Model and Syntax*; W3C: Cambridge, MA, USA, 1997.
55. Wikipedia. Notation3, 2024. Page Version ID: 1221181897. Available online: https://en.wikipedia.org/wiki/Notation3 (accessed on 8 June 2024).
56. FFmpeg Project. FFmpeg. 2024. Available online: https://www.ffmpeg.org/ (accessed on 10 October 2023).
57. Lin, T.T. labelImg PyPI. 2021. Available online: https://pypi.org/project/labelImg/ (accessed on 21 January 2024).
58. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. WongKinYiu/yolov7. 2022. Available online: https://github.com/WongKinYiu/yolov7 (accessed on 22 January 2024).
59. Becker, F. JokerFelix/MasterThesisCode. 2023. Available online: https://github.com/JokerFelix/MasterThesisCode (accessed on 21 January 2024).
60. Becker, F. JokerFelix/Gmaf-Master. 2023. Available online: https://github.com/JokerFelix/gmaf-master (accessed on 21 January 2024).

61. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 740–755.
62. Wong, K.-Y. yolov7/data/hyp.scratch.p5.yaml at Main · WongKinYiu/yolov7. 2022. Available online: https://github.com/WongKinYiu/yolov7/blob/main/data/hyp.scratch.p5.yaml (accessed on 21 October 2024).