Avatar Detection in Metaverse Recordings

Felix Becker * , Patrick Steinert , Stefan Wagenpfeil , Matthias L. Hemmje

Posted Date: 31 July 2024

doi: 10.20944/preprints202407.2512.v1

Keywords: Avatars; Object Detection; YOLO; Artificial Intelligence; Convolutional Neural Networks; Metaverse

Article

Avatar Detection in Metaverse Recordings

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

[1] Faculty of Mathematics and Computer Science, University of Hagen, Universitätsstrasse 1, D-58097 Hagen, Germany, psteinert@acm.org (P.S.), Matthias.hemmje@fernuni-hagen.de (M.H.)

[2] Faculty of Business Computing and Software Engineering, PFH University of Applied Science, 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, with growing collections. An essential element of the metaverse and its recordings are 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 supporting avatar detection. The latest object detection algorithms are trained and tested on a variety of avatar types in metaverse recordings. Our work achieves a significant 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

1. Introduction

Recognized as a global trend in 2022 [1], the metaverse [2][3, p.12368] is continuously growing [4]. Global crises, such as climate change and COVID, made it clear that digital technology, such as e.g., video calls [5], is an option to replace in-person meetings or even to provide effective virtual collaboration. 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 is continuously increasing [3, p.12368][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, p.12373]. The term avatar is based on the ancient Hindu concept, which calls a physical representation of a Hindu god an avatar [16, p.73-73]. 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 some 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, his 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, to some degree persistent and reacting to actions happening in real time [17, p.4ff].

User sessions in the metaverse can be recorded [18], i.e. as a screen recording, which we refer to as *Metaverse Recordings (MVRs)* [18]. Hence, the metaverse produces Multimedia Content Objects

 doi:10.20944/preprints202407.2512.v1

(MMCO), 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 case of ML training data, to 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, p.1]. Therefore, avatars are crucial elements that represent users in the virtual world. To achieve 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, p.1]. This task is relevant for searching and indexing vast 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 [27]. 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.

In MMIR, semantic information can be used to reason about the situation based on the automatically extracted meta information. Semantic information describes information about the content, its meaning, and possibly even about the meaning in relation to other content. This allows, for example, 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.

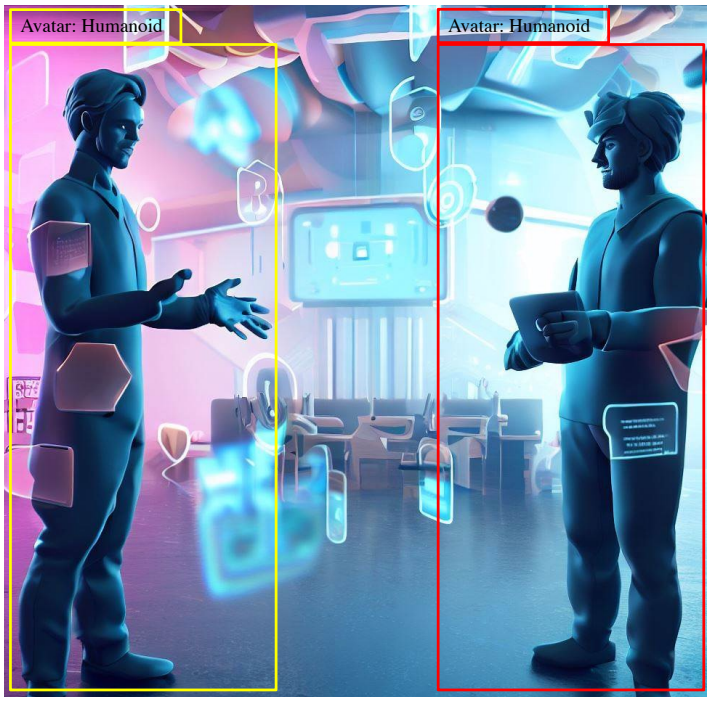

Figure 1. Two Detected Avatars Interacting.

 doi:10.20944/preprints202407.2512.v1

To create the semantic information, the algorithm receives an image as input, then automatically detects semantic information by locating avatars drawing boxes around them and classifying them noting their classes on 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, the 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 a baseline.

The scientific approach used for this work follows the method described by Nunamaker [28, p.94]. The application enables the researcher to present a clear and organized response to the research question. The Nunamaker framework connects the four disciplines *Observation*, *Theory Building*, *System Development* and *Experimentation*. These areas influence each other, with System Development 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.

2. Current State-of-the-Art

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

2.1. Definitions and Characterizations of Avatars in Literature

According to Miao et al. [29, p.71f], Avatars in virtual worlds of the metaverse lack a unified definition and taxonomy [29, p.71f]. 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" [29]. They suggest a simple taxonomy two-by-two in dimension. The first dimension is form realism, 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 type, *Simplistic Character*, low in form and behavior, *Superficial Avatar*, high in form but low in behavior, *Intelligent Unrealistic Avatar*, low in form but high in behavior, and *Digital Human Avatar*, 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, they include the representation as a 2D or 3D model, its static or dynamic graphic, and human characteristics, such as gender, race, age, and name. The excerpt is presented in Figure 2 [29, p.72].

 doi:10.20944/preprints202407.2512.v1

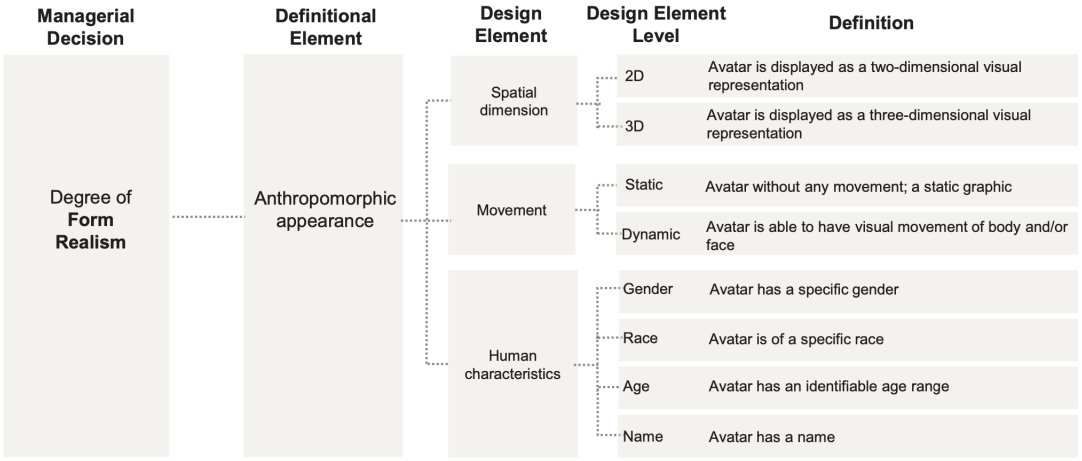

Figure 2. Avatar Typology of Form Realism [29, p.72].

Ante et al. [30, p.11] present a similar definition. They define an avatar of a user as "one or more digital representations of themselves in the digital world" [30, p.11]. At a high level, they classify them into the following types: *Customizable*, *Non-customizable*, *Self-representational*, *Non-human* and *Abstract*. They again see a high level of anthropomorphism, but also include abstract and non-human characters, which can lack these attributes [30, p.16].

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. [29, p.56] they are regarded as *Superficial Avatar* or *Digital Human Avatar*. By the classification of Ante et al. [30, p.16] they fit Customizable, Non-customizable and Self-representational. These humanoid avatars are most easily recognized by their silhouette, with four limbs, a torso, and 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 [30, p.5]. Using humanoid avatars makes it easier to assert human features to avatars and strengthens social interaction and general engagement in a virtual world [31]. 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. [30, p.20f.] The Human-Centered Design approach [32, p.1106] is another argument to focus on humanoid avatars.

In summary, a characterization of avatars roughly either being *Human* or *NonHuman* is found, while *NonHuman* can still include anthropomorphic features at a lower level, but 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 to model avatars.

The 256-Metaverse Records dataset [33,34] contains video-based *MVRs* collected in wild from different metaverse virtual worlds. A sample of different avatars from the dataset is shown in Figure 3. The virtual worlds displayed all contain an avatar, at least the self-representation of the user engaging in the virtual world. On the top left, a scene from Second Life [35] 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 [36]. All avatars are close to a humanoid representation, with different levels of abstraction. The Roblox sample displays a blocky toy-like representation, Second Life is close to a photorealistic representation, Fortnite has a more realistic look that has some cartoonish elements, and 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 data set, it is highly likely that the labels might require adaption or the videos must be converted, either to images of specific size, format or similar. Figure 3 shows different examples of avatars in the dataset.

 doi:10.20944/preprints202407.2512.v1

From the observation of the dataset made by the authors of this paper, virtual worlds employ *indicators* hovering over the avatar heads of different forms, including text boxes diamonds, or arrows.

Steinert et al.[37, p.4] investigate the differences in ontologies of common multimedia with *MVRs*. They also propose to define an avatar as a tuple of a nametag, the *Indicator*, and a character. Further, the characterization describes 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 to extend the Large-Scale Concept Ontology for Multimedia (LSCOM) [38, p.81f.]. 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 characterization found. 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 *NonHuman* avatars it can be even harder. The author suggests including a *Sign*, which refers to the *indicator*, when found in relation to a detection of *Human* or *NonHuman*, to allow easier identification of avatars. However, based on the observations in the wild, the described nametag 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 non-human 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.

Figure 3. Samples of the 256 Metaverse Recording dataset.

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 [39]. 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.

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 [40], 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 [40]. 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

 doi:10.20944/preprints202407.2512.v1

then checking for data points with low probabilities assigned [40]. Therefore, only some training data have to be labeled.

A similar approach is used by Ratner et al. [41, p.7], 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 data sets. 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) [42], basic CNNs have been extended and improved to models such as You Only Look Once (YOLO) [43] or Regional CNNs (R-CNNs) [44, p.1], 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.

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 [45,46].

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.

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) [47, p.2f.] for our modeling work.

 doi:10.20944/preprints202407.2512.v1

3.1. Avatar Classification

As presented, the existing classification of avatars lacks supporting object detection. We propose a 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 4. 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*.

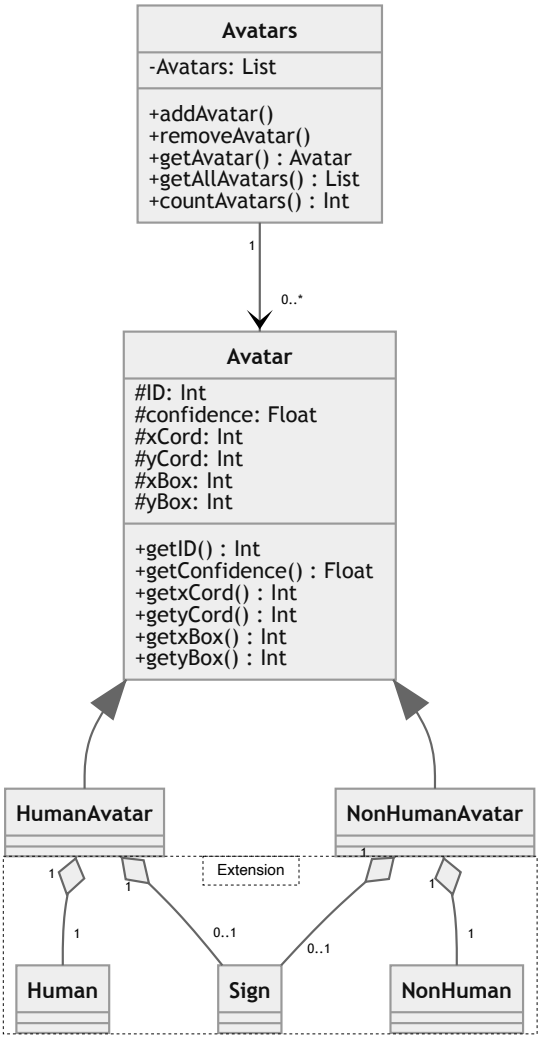

Figure 4. Information Model Class Diagram of *Avatars*.

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 an 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 anthropomorphic appearance of a figure might be given, but it is not controlled by a human or machine user. The biggest issue might arise when trying to detect non-human avatars, which feature

 doi:10.20944/preprints202407.2512.v1

little to none human-like appearance. To clarify the *Human* and *NonHuman* classes, both are meant to be anthropomorphic, *NonHuman* is not the simple negation of *Human*, it is rather meant as related to *Human*, but not a *Human*, typically featuring less anthropomorphic features. Both classes represent a detected instance of an avatar in an *MVR*.

The second attribute for avatar detection is the *Indicator*. Possible instances of an *Indicator* could be text boxes, i.e. nametags, symbols, i.e. arrows, or other indicators, mostly hovering above the head of an avatar. An *indicator* might be especially useful for *NonHumanAvatar* classes if non-obvious representations such as a normal animal is used for *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 context in the virtual world. Including the second attribute, the *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 to *NonHumanAvatar* class.

A common way to define such classifications are ontologies. As an example, the RDF [48] based ontology in Notation 3 (N3) [49] displayed in Listing 1. At least the *Avatar* can be added as a subclass of the perceptual agent to the existing LSCOM ontology. The *HumanAvatar* and *NonHumanAvatar* subclasses can also be added.

Including the *Indicator*, such as the proposed name tag [37, p.4], in conjunction with a *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;
```

 doi:10.20944/preprints202407.2512.v1


```
    rdfs: range :NonHuman .
```

The resulting class model presents 4 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 models 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 models parameters, and basing the magnitude of change on the learning rate.

YOLO is available in different 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, but as a starting point again a default set of hyperparameters is used.

4. Implementation

This section describes our results of the Nunamaker phase systems development. The results comprise a created dataset for training and evaluation of 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 Recordings dataset [50]. We refer to this as *ADET-DS*. *ADET-DS* contains 687 instances of class *HumanAvatar* and 327 instances of class *NonHumanAvatar*.

The original data set 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 their environments and avatars displayed, while being in 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 characters 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. One risk is to rely too much on the presence of these indicators creating an imbalanced and unrealistic data set, because some avatar representations do not include this *Indicator* attribute. Thus, only looking for these indicators when quickly screening through the frames should be avoided.

For frame extraction and labeling, a custom python script using FFmpeg [51] and LabelImg [52] 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 5 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

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 July 2024 doi:10.20944/preprints202407.2512.v1

a *NonHumanAvatar*. If the race is more clearly that of a human, then it is classified as *Human*, with a low form realism. This example demonstrates that it is non-trivial to assign the correct class labels, the indicator helps to identify avatars, but it is sometimes not present or unique to an avatar.

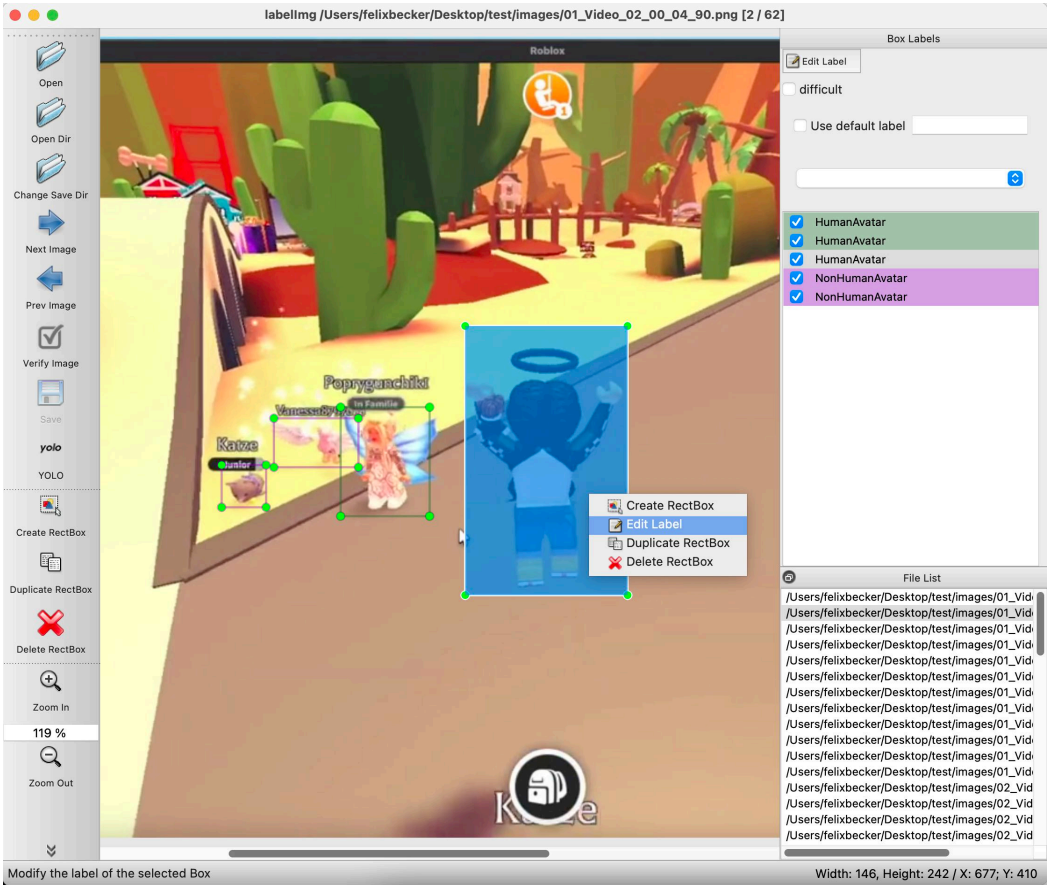

Figure 5. Example image annotation of avatars in *MVRs* with *LabelImg*.

4.2. Avatar Detector

The actual detection model is mostly the default YOLOv7 implementation [53]. The final implementation code is provided at [54,55].

The first YOLOv7 implementation created is called $YOLO^{base}$ or $YOLO^{base}$, representing the original vanilla YOLOv7 [53] model, trained using the COCO dataset [56, p.2]. Without any further training, it can simply detecting *person* type classes, which are then interpreted as avatar detections. The second YOLOv7 implementation created is referred to as $ADET$, representing the adapted YOLO model, specialized by transfer learning on two avatar classes *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, and features minimal adjustments for the hyperparameters of the P5 configuration [57] delivered with YOLOv7. The number of known classes was adopted to 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, hence, no indicator is visible. The same configuration of $ADET^{indicator}$ has been used for training $ADET^{NI}$.

For both model's final trainings, a batch size of 16, 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 detection results of models trained

 doi:10.20944/preprints202407.2512.v1

with different sets of hyperparameters, the validation part was left unseen by the model until the final model was selected and evaluated.

An example output of the detected images is presented in Figure 6. 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 5.

Figure 6. Example of detected avatar instances.

The implementation shows that existing object detection algorithms can be used to detect Avatars. 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* 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 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 on COCO dataset. The $ADET - DS$ test split was used for the evaluation.

 doi:10.20944/preprints202407.2512.v1

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) [39] at threshold T (mAP@T) and Intersection over Union (IOU) [39] 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 7.

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

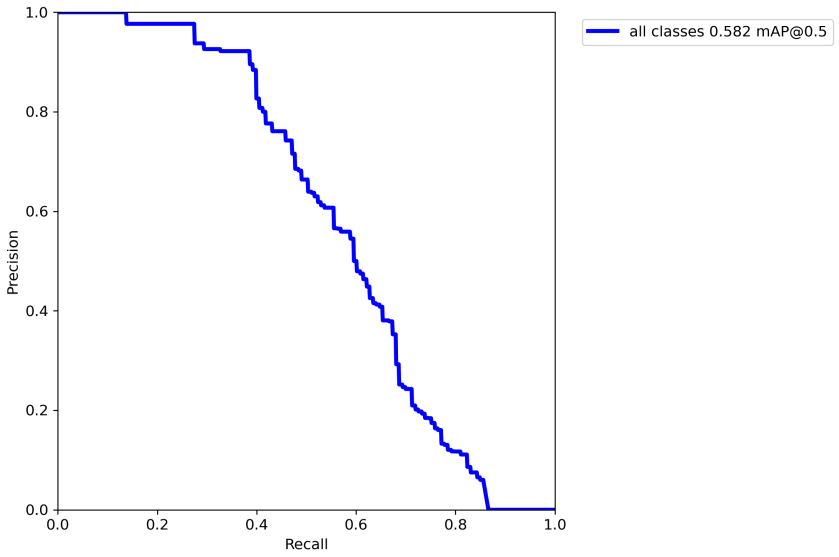

Figure 7. AP and mAP of $YOLO^{base}$ on Test Data.

The $F1$ score is quite consistent for different threshold levels falling of for higher confidence values and is optimal at 0.155 shown in Figure 8.

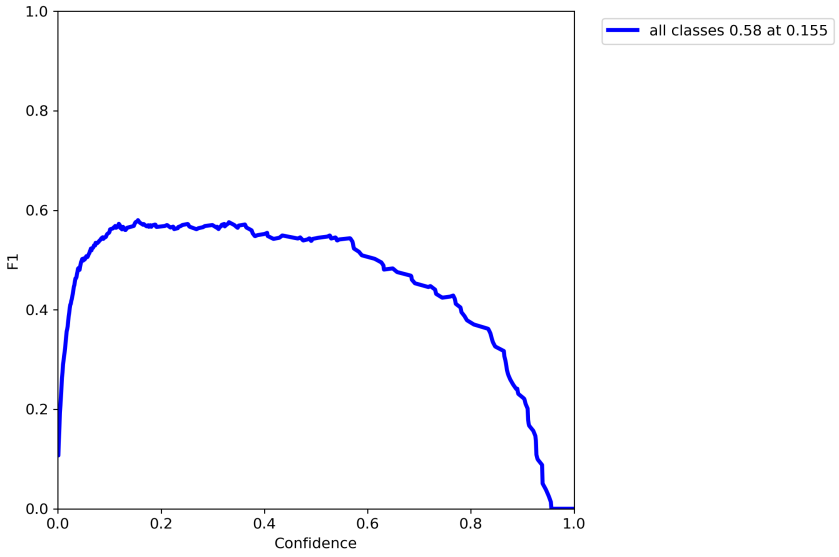

Figure 8. $F1$ Score for Different Thresholds of $YOLO^{base}$.

 doi:10.20944/preprints202407.2512.v1

An in-depth analysis of the predictions of the $YOLO^{base}$ model shows that rarely the model 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 9.

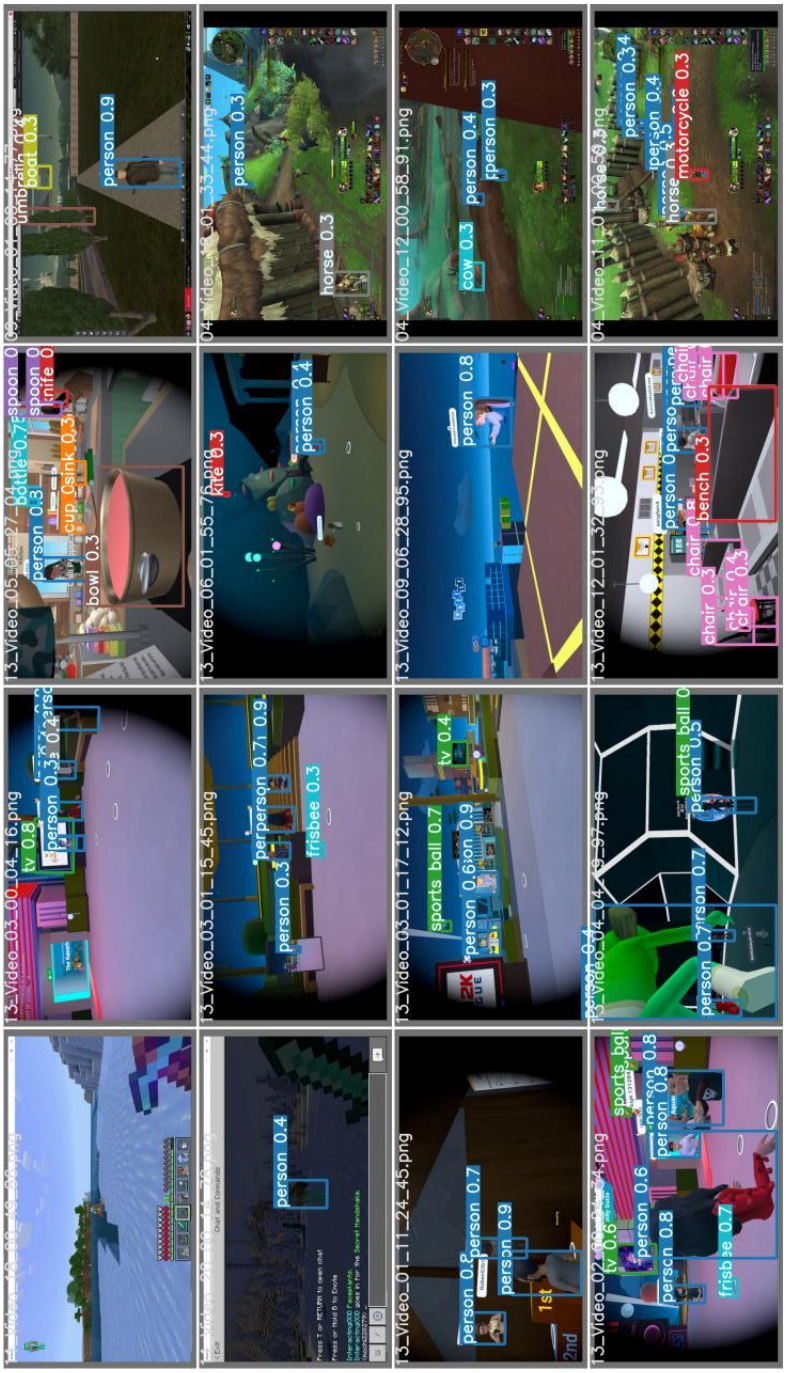

Figure 9. Predicted Avatars $YOLO^{base}$ on Test Data of ADET-DS.

These results support the idea of anthropomorphic appearance of avatars and the idea of *Non-Human* representing something that is not actually a human, but close to it.These results support the idea of the anthropomorphic appearance of avatars and the idea that the class *NonHuman* represents

 doi:10.20944/preprints202407.2512.v1

something that is not really a human, but comes close to one. 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 itself. 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 the 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, specialized by 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 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 now.

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 10.

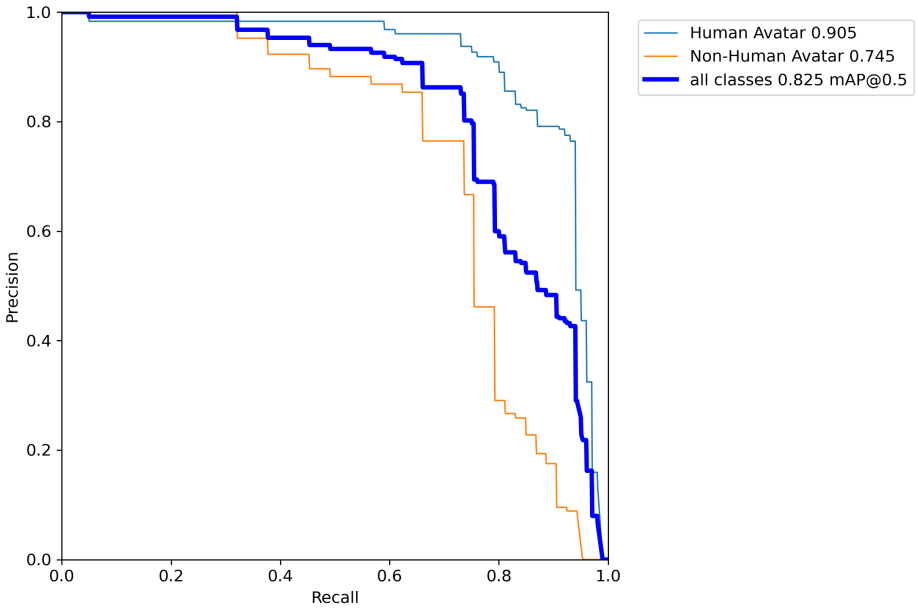

Figure 10. AP and mAP of $ADET^{indicator}$ on test data of ADET-DS.

The $F1$ score is quite consistent for different threshold levels and is optimal at 0.245 shown in Figure 11.

 doi:10.20944/preprints202407.2512.v1

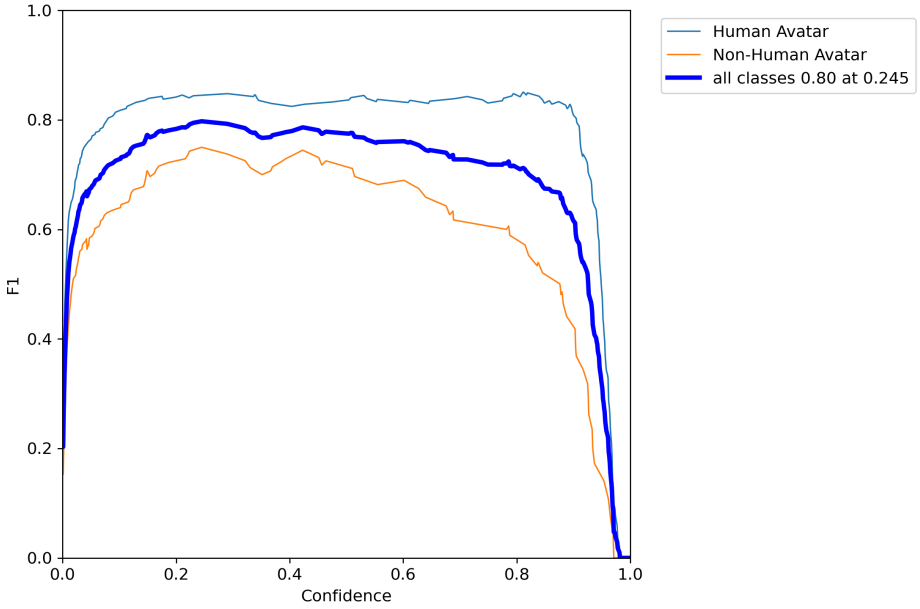

Figure 11. $F1$ Score for Different Thresholds of $ADET^{indicator}$.

An example of the detection is shown in Figure 12. The same sub-sample of test data is selected as for $YOLO^{base}$. Only relevant objects such as *Avatars* are detected, especially *NonHumanAvatars* are now correctly classified and detected at all. Detecting far-away or tiny avatars seems easier for the model, and at the same time, the bounding boxes are comparable in precision.

 doi:10.20944/preprints202407.2512.v1

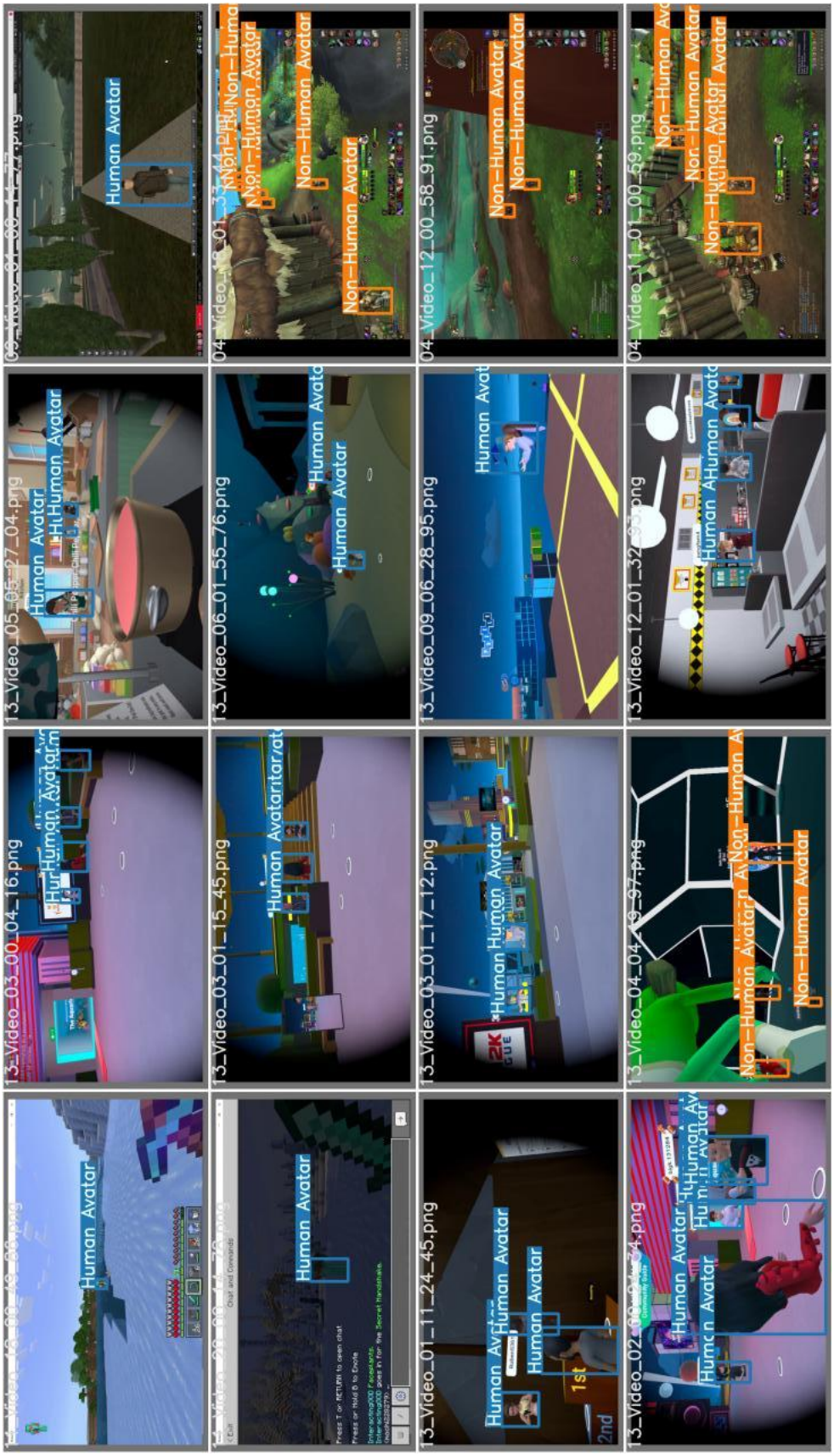

Figure 12. Predicted Avatars $ADET^{indicator}$ on Test Data of ADET-DS.

 doi:10.20944/preprints202407.2512.v1

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, a relative improvement of 0.422.

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

Table 1. Comparison of $ADET^{indicator}$ and $YOLO^{base}$ Detection Metrics.

Measure	$YOLO^{base}$	$ADET^{Indicator}$	Abs. Delta	Rel. Delta
mAP@0.5	0.582	0.825	+0.245	+0.422
Class HumanAvatar		0.905		
Class NonHumanAvatar		0.745		
F1@Optimum	0.580	0.800	+0.220	+0.379

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 including the *Indicator* indicator ($ADET^{indicator}$) and excluding it ($ADET^{NI}$). The idea is that these models can provide explicitly 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 $ADAT - DS$ in test/train/val, edited out the indicators such as nametags or graphical symbols.

Training and Inference: $ADET^{NI}$ was used 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: Testing $ADET^{indicator}$ 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* achieved an AP of 0.745. Subclasses such as *Human*, *NonHuman*, and *Indicator*, or 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, *Indicator* is an important element for *NonHumanAvatar* detections, while slightly helping with *HumanAvatar* detections, because the

 doi:10.20944/preprints202407.2512.v1

same model detected avatars worse, when the indicator class *Indicator* is 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 might be responsible for this, first the large variability in the *NonHuman* class compared to the *Human* class. Second, the often observed similarity between *NonHuman* and *Human*. As 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 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 class *Indicator* and *NonHuman* are both detected for the *NonHumanAvatar* detections, while mostly *Human* is detected with *HumanAvatar*. This supports the thesis that *Indicator* is a reasonable extension that helps spot avatars.

As future work, more significant subclasses to the *NonHuman* class in the context of avatars might be another meaningful extension of avatar classification.

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 multimedia information retrieval and artificial intelligence. The presented avatar classification model provides a classification for object detection models and includes the novel approach to incorporate visual indicators in the classification.

Utilizing the YOLO algorithm, specifically the YOLOv7 architecture, the $ADET$ model was developed, which is fine-tuned to detect avatars within various virtual environments. The research demonstrates a notable improvement in detection accuracy, mainly attributed to the specialized training on avatar-specific datasets. 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.

The results indicate that the $ADET$ model, especially the $ADET^{indicator}$ variant, achieves a substantial improvement in detection accuracy compared to the baseline models. 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.

Future research should aim to expand the training datasets to encompass a broader range of avatar types and virtual environments. This expansion is essential to improve the generalizability and robustness of the model, particularly for *non-human avatars*, which exhibit a wider variety of forms and characteristics. 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 further future work, other algorithms such as R-CNNs or CLIP-based [59] models can also be implemented and compared to the YOLO-based approach presented.

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. Although there is room for further research, this work shows, by combining and adapting the existing model, an improvement against the baseline of the YOLO person-class is achieved.

 doi:10.20944/preprints202407.2512.v1

References

1. Gartner Inc.. Gartner Predicts 25% of People Will Spend At Least One Hour Per Day in the Metaverse by 2026, 2022.
2. Mystakidis, S. Metaverse. *Encyclopedia* **2022**, *2*, 486–497. doi:10.3390/encyclopedia2010031.
3. Ritterbusch, G.D.; Teichmann, M.R. Defining the Metaverse: A Systematic Literature Review. *IEEE Access* **2023**, *11*, 12368–12377. doi:10.1109/ACCESS.2023.3241809.
4. KZero Worldwide. Exploring the Q1 24' Metaverse Radar Chart: Key Findings Unveiled - KZero Worldswide, 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 Research* **2022**, *53*, 343–365. doi:10.1177/10464964211015286.
6. Meta Platforms, Inc.. Meta Connect 2022: Meta Quest Pro, More Social VR and a Look Into the Future, 2022.
7. Takahashi, D. Nvidia CEO Jensen Huang weighs in on the metaverse, blockchain, and chip shortage, 2021.
8. Apple Inc. Apple Vision Pro available in the U.S. on February 2, 2024.
9. INTERPOL. Grooming, radicalization and cyber-attacks: INTERPOL warns of 'Metacrime', 2024.
10. Linden Lab. Official Site Second Life, 2024.
11. Decentraland. Official Website: What is Decentraland? https://decentraland.org, 2020. Online; accessed 09-June-2023.
12. Corporation, R. Roblox: About Us. https://www.roblox.com/info/about-us?locale=en_us, 2023. Online; accessed 03-Nov-2023.
13. Games, E. FAQ, Q: What is Fortnite? https://www.fortnite.com/faq, 2023. Online; accessed 06-Nov-2023.
14. Meta Platforms, Inc.. Horizon Worlds | Virtual Reality Worlds and Communities, 2023.
15. Wikipedia. Virtual world, 2023. Page Version ID: 1141563133.
16. Lochtefeld, J.G. *The Illustrated Encyclopedia of Hinduism*; The Rosen Publishing Group, Inc. New York, 2002.
17. Bartle, R. *Designing Virtual Worlds*; New Riders Games, 2003.
18. Steinert, P.; Wagenpfeil, S.; Frommholz, I.; Hemmje, M.L. Towards the Integration of Metaverse and Multimedia Information Retrieval. 2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE); IEEE: Milano, Italy, 2023; pp. 581–586. doi:10.1109/MetroXRAINE58569.2023.10405728.
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. doi:10.1109/ACCESS.2021.3073469.
20. Bestie Let's play. Wir verbringen einen Herbsttag mit der Großfamilie!!/Roblox Bloxburg Family Roleplay Deutsch, 2022.
21. Uhl, J.C.; Nguyen, Q.; Hill, Y.; Murtinger, M.; Tscheligi, M. xHits: An Automatic Team Performance Metric for VR Police Training. 2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE); IEEE: Milano, Italy, 2023; pp. 178–183. doi:10.1109/MetroXRAINE58569.2023.10405600.
22. Koren, M.; Nassar, A.; Kochenderfer, M.J. Finding Failures in High-Fidelity Simulation using Adaptive Stress Testing and the Backward Algorithm. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); IEEE: Prague, Czech Republic, 2021; pp. 5944–5949. doi:10.1109/IROS51168.2021.9636072.
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. 2022.
24. NVIDIA Corp. NVIDIA DRIVE Sim, 2024.
25. Rüger, S.; Marchionini, G. *Multimedia Information Retrieval*; Springer, 2010. OCLC: 1333805791.
26. Zhao, Z.Q.; Zheng, P.; Xu, S.T.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems* **2019**, *PP*, 1–21. doi:10.1109/TNNLS.2018.2876865.
27. Abdari, A.; Falcon, A.; Serra, G. Metaverse Retrieval: Finding the Best Metaverse Environment via Language. Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval; ACM: Ottawa ON Canada, 2023; pp. 1–9. doi:10.1145/3606040.3617445.
28. Nunamaker, J.; Chen, M. Systems development in information systems research. Twenty-Third Annual Hawaii International Conference on System Sciences, 1990, Vol. 3, pp. 631–640 vol.3.
29. Miao, F.; Kozlenkova, I.V.; Wang, H.; Xie, T.; Palmatier, R.W. An Emerging Theory of Avatar Marketing. *Journal of Marketing* **2022**, *86*, 67–90, [https://doi.org/10.1177/0022242921996646]. doi:10.1177/0022242921996646.

 doi:10.20944/preprints202407.2512.v1

30. Ante, L.; Fiedler, I.; Steinmetz, F. Avatars: Shaping Digital Identity in the Metaverse. https://www.blockchainresearchlab.org/wp-content/uploads/2020/05/Avatars-Shaping-Digital-Identity-in-the-Metaverse-Report-March-2023-Blockchain-Research-Lab.pdf, 2023. Online; accessed 17-July-2023.
31. Kim, D.Y.; Lee, H.K.; Chung, K. Avatar-mediated experience in the metaverse: The impact of avatar realism on user-avatar relationship. *Journal of Retailing and Consumer Services* **2023**, *73*, 103382.
32. Mourtzis, D.; Panopoulos, N.; Angelopoulos, J.; Wang, B.; Wang, L. Human centric platforms for personalized value creation in metaverse. *Journal of Manufacturing Systems* **2022**, *65*, 653–659.
33. Steinert, P.; Wagenpfeil, S.; Hemmje, M.L. 256-MetaverseRecords Dataset. https://www.patricksteinert.de/256-metaverse-records-dataset/, 2023.
34. Steinert, P.; Wagenpfeil, S.; Frommholz, I.; Hemmje, M.L. 256 Metaverse Recordings Dataset; In Publication: Melbourne, 2025.
35. Linden Research, I. SecondLife. https://secondlife.com, 2023. Online; accessed 03-Nov-2023.
36. Meta. Frequently Asked Questions, Q: What is Horizon Worlds? https://www.meta.com/en-gb/help/quest/articles/horizon/explore-horizon-worlds/horizon-frequently-asked-questions/, 2022. Online; accessed 06-Nov-2023.
37. Steinert, P.; Wagenpfeil, S.; Frommholz, I.; Hemmje, M.L. Integration of Metaverse Recordings in Multimedia Information Retrieval.
38. Naphade, M.; Smith, J.; Tesic, J.; Chang, S.F.; Hsu, W.; Kennedy, L.; Hauptmann, A.; Curtis, J. Large-scale concept ontology for multimedia. *IEEE MultiMedia* **2006**, *13*, 86–91. doi:10.1109/MMUL.2006.63.
39. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proceedings of the IEEE* **2023**, *111*, 257–276. doi:10.1109/JPROC.2023.3238524.
40. Rauch, L.; Huseljic, D.; Sick, B. Enhancing Active Learning with Weak Supervision and Transfer Learning by Leveraging Information and Knowledge Sources. *IAL@ PKDD/ECML* **2022**.
41. Ratner, A.J.; De Sa, C.M.; Wu, S.; Selsam, D.; Ré, C. Data Programming: Creating Large Training Sets, Quickly. Advances in Neural Information Processing Systems; Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; Garnett, R., Eds. Curran Associates, Inc., 2016, Vol. 29.
42. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems* **2022**, *33*, 6999–7019. doi:10.1109/TNNLS.2021.3084827.
43. 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. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Vancouver, BC, Canada, 2023; pp. 7464–7475. doi:10.1109/CVPR52729.2023.00721.
44. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. 2016, pp. 779–788. doi:10.1109/CVPR.2016.91.
45. Ukhwah, E.N.; Yuniarno, E.M.; Suprapto, Y.K. Asphalt Pavement Pothole Detection using Deep learning method based on YOLO Neural Network. 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA); IEEE: Surabaya, Indonesia, 2019; pp. 35–40. doi:10.1109/ISITIA.2019.8937176.
46. J, D.; V, S.D.; S A, A.; R, K.; Parameswaran, L. Deep Learning based Detection of potholes in Indian roads using YOLO. 2020 International Conference on Inventive Computation Technologies (ICICT); IEEE: Coimbatore, India, 2020; pp. 381–385. doi:10.1109/ICICT48043.2020.9112424.
47. Rumpe, B. *Modeling with UML*; Springer, 2016.
48. W3C. RDF Model and Syntax, 1997.
49. Wikipedia. Notation3, 2024. Page Version ID: 1221181897.
50. Steinert, P. 256-MetaverseRecordings-Dataset Repository, 2024. original-date: 2024-01-12T07:26:01Z.
51. FFmpeg project. FFmpeg, 2024.
52. Lin, T.T. labelImg PyPI. https://pypi.org/project/labelImg/, 2021. Online; accessed 21-January-2024.
53. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. WongKinYiu/yolov7. https://github.com/WongKinYiu/yolov7, 2022. Online; accessed 22-January-2024.
54. Becker, F. JokerFelix/MasterThesisCode. https://github.com/JokerFelix/MasterThesisCode, 2023. Online; accessed 21-January-2024.
55. Becker, F. JokerFelix/gmaf-master. https://github.com/JokerFelix/gmaf-master, 2023. Online; accessed 21-January-2024.

 doi:10.20944/preprints202407.2512.v1

56. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. Computer Vision – ECCV 2014; Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T., Eds.; Springer International Publishing: Cham, 2014; pp. 740–755.
57. Kin-Yiu, Wong. yolov7/data/hyp.scratch.p5.yaml at main · WongKinYiu/yolov7, 2022.
58. Wong, K.Y. WongKinYiu/yolov7, 2024. original-date: 2022-07-06T15:14:06Z.
59. Nguyen, T.N.; Puangthamawathanakun, B.; Caputo, A.; Healy, G.; Nguyen, B.T.; Arpnikanondt, C.; Gurrin, C. VideoCLIP: An Interactive CLIP-based Video Retrieval System at VBS2023. In *MultiMedia Modeling*; Dang-Nguyen, D.T.; Gurrin, C.; Larson, M.; Smeaton, A.F.; Rudinac, S.; Dao, M.S.; Trattner, C.; Chen, P., Eds.; Springer International Publishing: Cham, 2023; Vol. 13833, pp. 671–677. Series Title: Lecture Notes in Computer Science, doi:10.1007/978-3-031-27077-2_57.