A GENERIC COMPUTER VISION TOOL FOR RECOGNISING HUMAN ACTIVITIES

J.J. Jacobs[1] & G.S. Nel[1*]

ARTICLE INFO

Article details
Presented at the 2nd International Conference on Industrial Engineering, Systems Engineering and Engineering Management, held from 2 to 4 October 2023 in Somerset West, South Africa

Available online 17 Nov 2023

Contact details
* Corresponding author
gsnel@sun.ac.za

Author affiliations
1 Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Stellenbosch, South Africa

ORCID® identifiers
J.J. Jacobs
https://orcid.org/0000-0002-8583-7277

G.S. Nel
https://orcid.org/0000-0002-0293-1234

DOI
http://dx.doi.org//10.7166/34-3-2956

ABSTRACT

A prominent application area in the domain of computer vision is human activity recognition, which involves automatically detecting people's actions from video footage. In this paper, a generic decision support tool capable of performing human activity recognition by employing computer vision is developed. The objective of the decision support tool is to facilitate the analysis of video footage by learning to identify human activities from video footage. The decision support tool facilitates the processing of raw data, the training of a computer vision model in respect of the processed data, and the deployment of the trained computer vision model in respect of unseen video footage. This paper details a computerised implementation of the decision support tool in respect of a benchmark data set and real-world data set involving the cash replenishment process of a South African bank.

OPSOMMING

'n Prominente toepassingsgebied in die veld van rekenaarvisie is menslike aktiwiteitsherkenning, wat behels dat mense se optrede outomaties vanaf videomateriaal geanaliseer word. In hierdie artikel word 'n generiese besluitsteunsinstrument ontwikkel wat in staat is om menslike aktiwiteitsherkenning uit te voer deur rekenaarvisie te gebruik. Die doel van die besluitondersteuningsinstrument is om die ontleding van videomateriaal te fasiliteer deur te leer om menslike aktiwiteite uit videomateriaal te identifiseer. Die besluitondersteuningsinstrument fasiliteer die verwerking van rou data, die opleiding van 'n rekenaarvisiemodel ten opsigte van die verwerkte data, en die ontplooiing van die opgeleide rekenaarvisiemodel ten opsigte van ongesiene videomateriaal. Hierdie vraestel beskryf 'n gerekenariseerde implementering van die besluitsteuninstrument ten opsigte van 'n maatstafdatastel en werklike datastel wat die kontantaanvullingsproses van 'n Suid-Afrikaanse bank behels.

1. INTRODUCTION

Contemporary advances in the domain of machine learning and the increased capabilities of computer hardware have led to the computational viability and proliferation of *computer vision* (CV) – i.e., the field of study that algorithmically enables computers to 'see' and comprehend a physical environment (typically expressed using image-based data such as photos and videos). Multiple tasks may be performed on visual data with the aid of CV, such as object detection [1]–[3], pose estimation [4]–[6], and activity recognition [7]–[9], the last of which is the focal point of this research. *Human activity recognition* (HAR) can be defined as the field of study that aims to recognise human activities from multiple observations of actions performed by key subjects in their environments [10]. HAR methodologies may be applied to a variety of problems, ranging from ambient assisted living for elderly people to surveillance monitoring, sports analytics, and even behavioural analysis [10].

\Although many CV-based approaches to HAR deliver state-of-the-art performance on benchmark data sets, they lack the utility to be applied to different data sets without considerable modification. The primary aim in this project is to design and develop a generic *decision support tool* (DST) that can employ CV models to identify human activities from video footage. The proposed DST, called the *computer vision human activity recognition tool* (CV-HART), employs a deep-learning-based CV model to perform the recognition of activities. The model is first trained and evaluated on a benchmark HAR data set in order to verify the working of the CV-HART. A case study is also performed on data provided by the industry partner attached to this project (i.e., a large South African retail bank) in order to demonstrate further the utility of the CV model and the CV-HART. The case study focuses on the task of automating the monitoring of the bank's cash replenishment process at automated teller machines (ATMs).

The remainder of this paper is organised as follows: first, the literature relevant to the work carried out in this project (i.e., deep learning, HAR, and DSTs) is discussed in Section 2, which is followed by a detailed discussion in Section 3 of the design and development of the proposed CV-HART. In Section 4, the tool's verification in respect of a widely used benchmark data set is addressed. The validation of the CV-HART is discussed in Section 5, which involves its application to a real-world case study of the industry partner. The paper is then concluded in Section 6 with a summary of its contributions and possible extensions for future work. Supplementary material is relegated to Appendix A.

2. LITERATURE REVIEW

In this section, the domains relevant to the work presented in this paper are discussed, namely deep learning, HAR, and DSTs. A brief discussion of similar work (with respect to the proposed CV-HART) is also presented.

2.1. Deep learning

Deep learning is a term used to describe *artificial neural networks* (ANNs) that comprise many hidden layers [11], and represents the most prolific approach to CV. One specific algorithm, and perhaps the most influential algorithm in the field of CV, is the *convolutional neural network* (CNN) [12]. Although the foundational work of CNNs can be traced back to the conceptual work of Hubel and Wiesel [13] in the 1960s and the first computerised implementation thereof by Fukushima's *Neocognitron* [14] in the 1980s, it is arguably the paper by Le Cun *et al.* [12], entitled "Handwritten digit recognition: Applications of neural network chips and automatic learning", that can be regarded as the most profound. In the early 2010s, Le Cun *et al.* simplified the Neocognitron architecture and applied the method of *backpropagation* to train the model efficiently in a supervised learning fashion at scale, ushering in an era of CNN proliferation.

CNNs are specifically designed to process data that are characterised by spatial dependence, such as images comprising a grid-like structure. CNNs replace the conventional ANN operation of matrix multiplication (as is the case, for example, in *multi-layer perceptrons*) with a so-called *convolution* operation for efficient feature extraction [11], [15]. There are three central notions that underpin a CNN's working and that contribute to their innate computational capabilities, namely *sparse connectivity*, *parameter sharing*, and *equivariance to translation* [11]. The ResNet model architecture from He *et al.* [16] is one of the most influential architectures in the field of CV. A number of popular deep-learning architectures are based on, or draw inspiration from, the fundamental concepts of ResNets. One of the most influential is the You Only Look Once (YOLO) algorithm and, even more prolific, its second version, YOLOv2 [17]. The YOLOv2 algorithm draws inspiration from ResNets by employing a so-called *passthrough* layer, similar to the identity mappings employed by ResNets, to combine features from various resolutions and abstraction layers.

In the context of HAR, a number of architectures employ ResNets. Gowda *et al.* [18] use two ResNet backbone networks in their *SMART frame selector* algorithm for action recognition. Similarly, Qiu *et al.* [19] use ResNet backbones in their *local and global diffusion* model, achieving state-of-the-art activity recognition results. Another powerful HAR model is proposed by Tran *et al.* [20] that employs an adapted ResNet model with modifications to the architecture's dimensionality for improved performance.

2.2. Human activity recognition

HAR may be formally described as the task of recognising a human activity based on information from various sensors [10]. Sensors in the domain of HAR may include cameras, wearable sensors, and sensors placed throughout an environment. HAR is widely regarded as a difficult task, as actions may consist of a

single atomic movement, such as jumping, or a combination of movements, such as following a recipe. Additional complexity may be ascribed to recognising similar activities that are performed in different settings or by different people. There is a considerable variety of HAR applications in the literature, examples of which include ambient assisted living for elderly people [21]–[23], security and surveillance [24]–[26], sports analysis [27]–[29], and behaviour analysis [30]–[32].

HAR can be separated into two main branches, *sensory-based* and *vision-based* methods, the latter of which is the focus of this study. As the name suggests, vision-based sensing employs images and videos to recognise activities performed by humans. Images or videos can be captured or recorded using various devices such as cell phone cameras, sports cameras, and surveillance cameras. While vision-based activity recognition can deliver a favourable performance, a few key drawbacks are worth considering, such as its expensive nature, its increased computational complexity, and privacy issues that are applicable to certain vision-based action recognition methods [10].

Various computational domains for performing HAR have been proposed in the literature. Examples of these approaches include *action recognition* [18], [33], *pose estimation* [34], [35], and *spatiotemporal action localisation* (STAL) [36]–[38], the last of which is the focal point. While action recognition aims to identify what action is performed in a video, STAL aims not only to detect what action occurs but also to identify when the action occurs and the spatial location in the frame(s). A powerful STAL architecture is the *You Only Watch Once* [39] (YOWO) architecture. YOWO, as shown in Figure 1, consists of two different network architectures, referred to as *backbone* networks. A 2D backbone extracts spatial information while a 3D backbone extracts temporal information from previous frames concurrently. The information from both the 2D and the 3D networks is combined using a channel fusion and *attention* mechanism. The success of various YOWO applications warrants its inclusion in this paper.

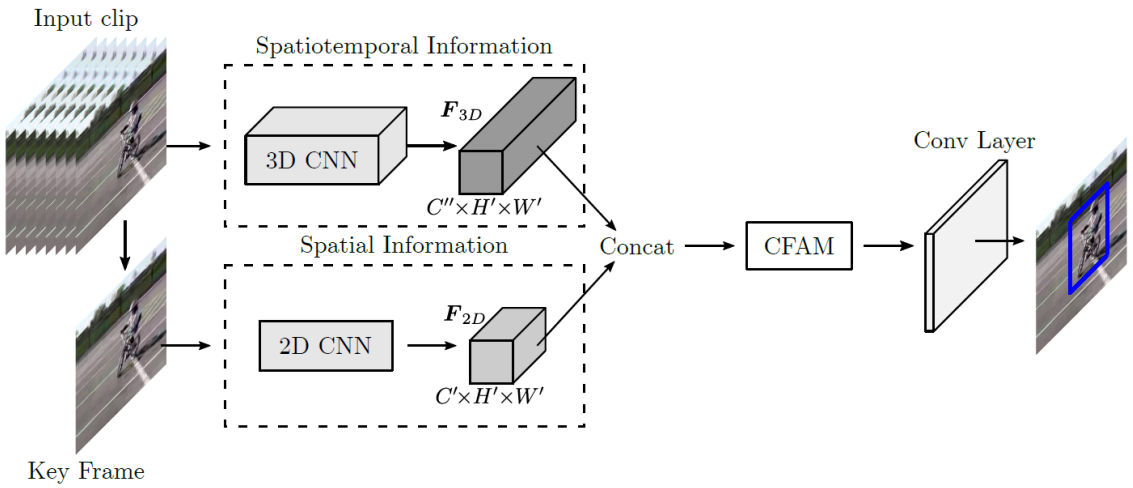

Figure 1: YOWO model architecture (adapted from Köpüklü *et al*. [39])

2.3. Decision support tools

Decision support systems, through which DSTs are contextualised, can be described as computerised technology-based solutions that employ a data-driven approach to modelling decision-making problems [40]. DTSs are designed to facilitate the decision-making process and to support the decision-maker while facilitating and automating various intermediate processes. In the discussion that follows, the main concepts pertaining to the specific type of DST proposed in this paper are addressed.

2.3.1. Components

Shim *et al*. [40] report that a conventional DST comprises three main facets, namely a *database* component with database management capabilities, a *modelling* (or *functional*) component combined with a *model management system* (MMS), and, finally, a *user interface* (UI).

Database component

The database component of a DST, which may also be described as a *database management system* (DBMS), is dedicated to creating, storing, removing, and changing data entries that are used as part of the DST. A database can be described as a collection of data that represent aspects of the real world, each of which is logically coherent with inherent meaning, and is designed, built, and populated for a specific purpose [41]. A DBMS can employ one or more database types; examples of prominent database types are *relational* databases, *graph* databases, *temporal* databases, *geographic* databases, and *hierarchical* databases, to name a few [42], [43]. The *hierarchical* database model is the most relevant database type, given the scope of this research, attributable to the database structures employed by the benchmark data set that was considered – i.e., UCF101 [44].

Modelling component

Various types of model can be used in DSTs, such as optimisation models, simulation models, and data-driven ML models [45]. This functional component also employs an MMS. An example of model management that may be performed is a 'what if' analysis, which performs various tests to determine the model output based on different inputs and scenarios. When multiple models and sub-models are used simultaneously, the MMS is also responsible for combining the models' results. It is common to have multiple models in a competing fashion, according to which different models are presented with the same inputs, whereafter the results of the different models are compared so as to select the best-performing model.

User interface component

The UI component is the primary (and typically the only) interaction point between the user and the DST. So it is important for the UI to be well-designed and therefore easy to use. A UI comprises two parts, the first of which pertains to the input, which enables users to input commands, data, and instructions, whereas the second part pertains to the output, which enables the system to communicate with the user by requesting more or correct information, showing errors, and providing solutions to the problems specified by the user [45]. UIs can be categorised into various types, such as *menu-based*, *form-based*, *question-and-answer*, *command line*, and – the most popular – *graphical* UIs [41], [46].

2.3.2. Data flow

For a DST to provide decision support effectively, the different components of the DST must communicate with one another by passing information and data between components. A frequently used approach to illustrating the flow of information in a system is a *data flow diagram* (DFD), which represents the inputs, processes or components, and outputs of the system under consideration [45]. A large system can be represented by an overview DFD that can be further decomposed into a series of stacked DFDs that contain more detail of the various sub-components. A DFD can be created by using combinations of only four elements, as shown in Figure 2.

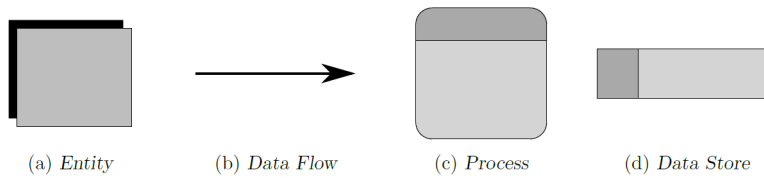

Figure 2: Elements of a DFD (adapted from Kendall and Kendall [45])

The *entity* element represents external entities that interact with the system, such as system users. The data *flow* element indicates the direction in which data flow between entities and processes. The processes employed by the system to transform the input are represented by the *process* element. Finally, the *data store* element represents the data stores used for data storage purposes in respect of input by the user and other data such as results generated by the DST.

2.4. Related work

In their paper entitled “Implementation of an anomalous HAR system”, Shreyas *et al.* [47] presented a HAR system that focuses on anomaly detection. The authors aimed to detect anomalous behaviour automatically and in real time, such as theft, abuse, fights, and accidents to name a few) from CCTV footage by employing HAR. Shreyas *et al.* performed a case study on the UCF-101 crime data set, which contains 13 anomaly classes. A 3D CNN feature extractor was employed to learn and extract spatiotemporal features. One of the limitations of the HAR system presented by Shreyas *et al.* was that specific actions were not detected; instead an anomaly score was computed.

A second study – in a paper entitled “Human activity recognition system from different poses with CNN” by Atikuzzaman *et al.* [48] – presented a HAR system that was capable of detecting and recognising different classes of human activities. The authors created their own data set comprising five human action classes with a combination of 5 648 frames captured by either a laptop camera or a CCTV camera. The activity recognition component employed a custom CNN architecture with six convolutional layers and a three-layer perceptron. Atikuzzaman *et al.* achieved a pose extraction accuracy of 99.86%, followed by near-perfect precision and recall scores for the actions detected from the extracted images. The limitations of the proposed HAR system include a lack of action diversity and an inability to detect actions for multiple people in one frame.

3. COMPUTER VISION HUMAN ACTIVITY RECOGNITION TOOL

A high-level overview of the CV-HART’s design (comprising three components and a GUI) is shown in Figure 3. For the sake of brevity, discussions of the database and the GUI components are omitted. Graphical illustrations of the GUI, however, are presented in Appendix A.

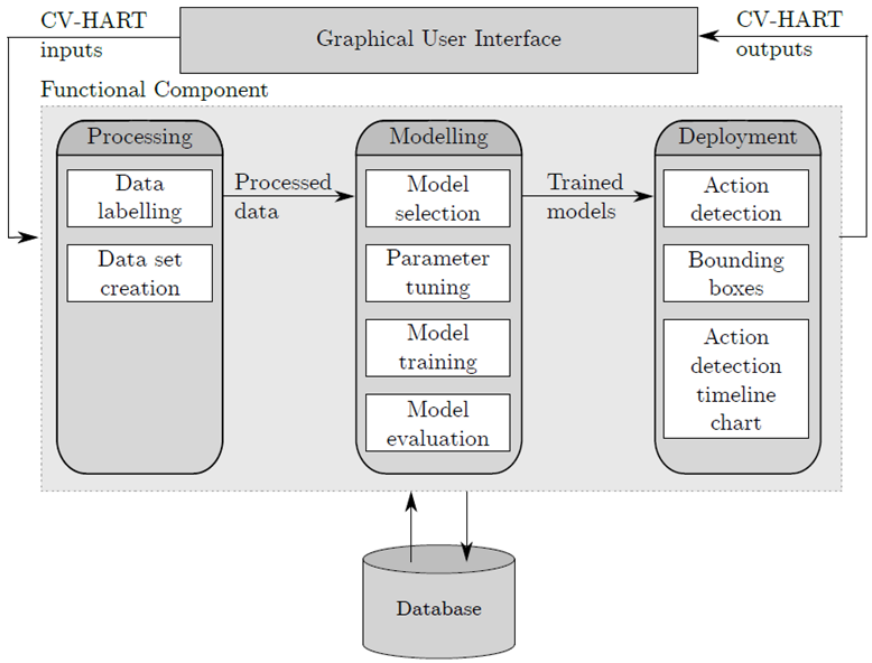

Figure 3: A graphical DFD representation of the proposed CV-HART

3.1. Processing component

The processing component of the CV-HART comprises two sub-processes, also referred to as *modules*. A user can choose to input labelled or unlabelled data. A module evaluates and translates the decisions made by the user to the system regarding whether a user inputs new data or employs currently available data, and whether the data are labelled or unlabelled. If a user chooses to create a new project and to upload new data, additional information about the data is required, such as the number of action classes, the

names of those action classes, and the name of the project. This information is then used to create an empty database in the correct structure for the project.

If a user inputs unlabelled data or wishes to label already inputted data, the user can use a data labelling module for this purpose. The video files provided by the user are all saved to a central raw data folder. The raw video files are selected and labelled iteratively, after which the labelled data are saved as images and text files containing annotations for the labelled actions. A data labelling tool called *DarkLabel* [49], an open-source utility program for image and video labelling, is employed as part of the labelling module. The annotation files are created per frame with a single label file containing all the actions labelled in that frame. The data labelling process is an iterative process, according to which each frame is presented to the user through the GUI, and the user provides bounding boxes (shown in Figure 4) for all the actions in the frame — a process that is repeated until the user is satisfied with the number of labelled frames.

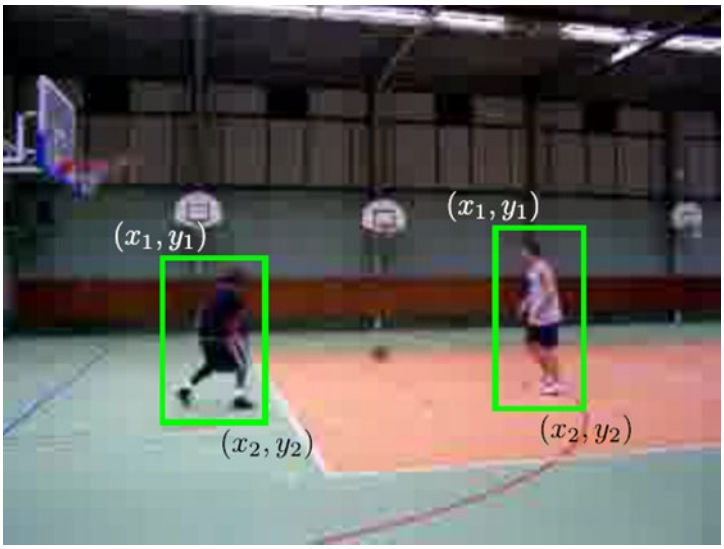

Figure 4: Example frame of bounding boxes drawn around persons performing an action

If a user wishes to crop the images in the data set, the CV-HART provides a module for this purpose. This reduction of the image size reduces the computational time required by removing areas in which no actions occur. In an iterative manner, the most extreme labels are created – i.e., the labels closest to the edges of the images. The extreme labels are then presented for consideration. The user can choose to crop the images with respect to the outside edges of the extreme labels, to provide different margins for cropping, or not to crop at all. The refined data set (including labels) is then saved to a database.

Next, a module separates the data into training data and testing data (the latter of which is applicable to the deployment component). Another folder is subsequently created containing the ground-truth values, which collectively represent the correct labels of the testing data set. These ground-truth label files are employed during evaluation in order to calculate the performance of the trained models. The last step of the labelling module involves generating so-called *anchor boxes*. Anchor boxes can be described as possible bounding box suggestions, based on prior knowledge that guides the model towards improved starting points.

3.2. Modelling component

Once the data processing is complete, inputs pertaining to the modelling component of the CV-HART are provided by the user. The user can decide to train a model either from an existing trained model state or from a completely new randomly initiated state. A model can be any deep-learning model that is capable of HAR; however, CNN-based architectures and other deep-learning architectures (designed explicitly for HAR), such as the YOWO architecture, are suggested. In the case of model training, the user is prompted to specify model parameter values. A list of possible parameters (typical value ranges or options) is shown in Table 1.

Table 1: YOWO model parameters and values/ranges

Parameter	Possible values/ranges	Source
Learning rate	0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001	[11]
Batch size	8, 16, 32, 64, 128, 256, 512	[50]
Clip length (frames)	6, 18	[39]
Backbone training state	pre-trained, untrained	[39]
Backbone models	ResNet18, ResNet50, ResNet101, ResNeXt101	[39]

After parameter values and options have been selected, a deployable model is created and trained. Feedback (e.g., training performance) is continually presented via the user interface. After the model training procedure has been concluded, model performance is evaluated. A popular performance metric known as *frame mean average precision* (abbreviated as 'frame mAP') is used in this paper. Frame mAP is typically used to evaluate the performance of object detection models, as it provides a robust measure of both model accuracy and localisation in a video (or a sequence of frames) [51].

3.3. Deployment

Upon establishing a satisfactory model through iterative parameter tuning and training, deployment can be performed, during which the user can apply a trained CV model to new unseen video data (i.e., testing data) in order to perform STAL. The deployment data types can be either video segments or untrimmed video streams. Typical deployment decisions include selecting a trained CV model, specifying data types, and choosing one or more deployment tasks. After a user has provided the data on which to perform STAL, the model creates action detection files that contain the detected actions (also saved in a database).

So-called *action detection timeline* (ADT) charts can be generated that provide an intuitive visualisation of both the specific detected actions and the corresponding temporal information. These ADT charts facilitate the verification of whether detected actions occur in the correct order. ADT charts also provide algorithmic performance insight, such as when the model prediction constantly alternates between two detected classes. An example of an ADT-generated chart is shown in Figure 15 (in Appendix A). The action detections are also visualised as bounding boxes superimposed on the video frames. The corresponding prediction class *Biking*, the confidence score of the class prediction 0.76, and the bounding box, are also displayed. A visualised action detection for a cycling video is shown in Figure 18 (in Appendix A).

4. VERIFICATION

In order to verify the proposed CV-HART, a computerised instantiation is implemented in respect of an open-source benchmark data set, the results of which are presented in this section.

4.1. Case study data background

In order to verify the workings of the CV-HART, a verification case study is performed to confirm that the algorithmic results produced by the CV-HART implementation are similar to those of Köpüklü *et al.* [39] – i.e., the original paper in which the YOWO architecture is proposed. To this end, the same benchmark data set (i.e., UCF101-24 [44]) and the original model parameter values (summarised in Table 2) are employed. The UCF101-24 data set is a subset of the UCF101 dataset, containing only 24 of the 101 possible classes. The video clips have a frame rate of 25 frames per second and a frame size of 320 x 240 pixels. An example frame for each class is presented in Figure 5. Each of the 24 classes comprises 25 groups of videos (i.e., groups are based on the actors and the backgrounds), and each group contains between four and seven video clips. The extent to which these scenes differ in the different groups is illustrated in Figure 6, in which five additional background settings or actors are shown for the Basketball, Fencing, PoleVault, and SoccerJuggling classes.

4.2. CV-HART verification implementation

The structured approach of *systems and software testing* (propounded by Kendall and Kendall [45]) is adopted for verification purposes. For the sake of brevity, however, the focus is placed only on stage four, namely *full systems testing with live data*, in respect of the modelling component exclusively. 'Live data' refers to the actual data for which the output is known – i.e., the benchmark data set. The best-performing

YOWO trained model, as provided by Köpüklü *et al*. [39], is imported into the CV-HART for evaluation purposes. The frame mAP values are calculated by presenting the UCF101-24 data set to the model. A summary of the results that are obtained is presented in Table 3. Köpüklü *et al*. reported a frame mAP value of 87.2% for the benchmark model in the original paper, which is almost identical to the value obtained by using the CV-HART with the same data set. The CV-HART can therefore be deemed verified because of the similarities between the results of the tool's implementation and those of the original authors.

Table 2: Parameter settings and values used by Köpüklü *et al*. [39] for the YOWO architecture

Parameter	Setting/value
2D backbone	19-layer YOLO
3D backbone	101 layer ResNeXt
training state	pre-trained
optimiser	Stochastic gradient descent
learning rate	0.0001
image size	224 x 224
clip duration	16 frames
mini-batch size	8
batch size	16
learning rate scheduler	Halve the learning rate at 30 000, 40 000, 50 000, and 60 000 iterations

Figure 5: Example frames for all 24 classes of the UCF101-24 [44] data set

Figure 6: Six example frames for each of the Basketball, Fencing, PoleVault, and SoccerJuggling classes of the UCF101-24 [44] data set

Table 3: AP values obtained by applying the CV-HART to the Köpüklü *et al*. [39] benchmark model

Class	AP	Class	AP	Class	AP	Class	AP
Basketball	64.84	Fencing	93.73	PoleVault	76.83	SoccerJuggling	96.74
BasketballDunk	90.7	FloorGymnastics	95	RopeClimbing	97.61	Surfing	96.83
Biking	89.8	GolfSwing	92.34	SalsaSpin	89.14	TennisSwing	86.93
CliffDiving	88.07	HorseRiding	99.8	SkateBoarding	88.69	TrampolineJumping	82.96
CricketBowling	72.83	IceDancing	80.89	Skiing	77.11	VolleyballSpiking	85.19
Diving	98.16	LongJump	63.42	Skijet	97.34	WalkingWithDog	88.52
						Average frame mAP	**87.23**

An additional verification step involves training a new model from scratch and subsequently comparing the algorithmic performance. To this end, the original parameter values employed by Köpüklü et al. are used again (shown in Table 2). The following computational resources are employed for the model training: a computing cluster comprising eight computer nodes with three identical graphics processing units per node; Nvidia Tesla T4, Nvidia Quadro RTX 4 000, and Nvidia Quadro RTX 6 000 GPUs; and for each node, two Intel Xeon Gold 5218 central processing unit processors with at least 376 GB of RAM per computer. The following modelling results (in respect of frame mAP) are obtained for the five respective epochs (in ascending order): 83.15, 82.05, 85.72, 84.52, and 83.86. The best model is therefore obtained after epoch three, with a frame mAP score of 85.72.

A summary of a performance comparison between the benchmark and trained models is presented in Table 4. The performance achieved by the CV-HART model is similar to that of the benchmark model; the overall performance is only 1.51% inferior. Eight of the 24 classes show an increase in AP: the LongJump and Basketball classes show significant increases of 8.23% and 23.35% respectively. The other 16 classes, however, experience a reduction in performance, eight instances of which correspond to a reduction larger than 5%. The largest reductions occur in the Skiing and SkateBoarding classes – i.e., 8.10% and 14.68% respectively. The performance differences are not entirely unexpected, as the original authors did not provide their random seeds for model weight initialisation. Overall, performance is mostly similar, and therefore further empirical proof of verification is demonstrated.

Table 4: **Performance comparison between benchmark and newly trained model**

Class	Benchmark	Trained Model	Difference
Basketball	64.84	93.19	28.35
BasketballDunk	90.7	90.84	0.14
Biking	89.8	86.38	−3.42
CliffDiving	88.07	84.81	−3.26
CricketBowling	72.83	68.71	−4.12
Diving	98.16	98.02	−0.14
Fencing	93.73	91.82	−1.91
FloorGymnastics	95	94.76	−0.24
GolfSwing	92.34	86.44	−5.90
HorseRiding	99.8	99.81	0.01
IceDancing	80.89	74.68	−6.21
LongJump	63.42	71.65	8.23
PoleVault	76.83	80.55	3.72
RopeClimbing	97.61	93.77	−3.84
SalsaSpin	89.14	82.53	−6.61
SkateBoarding	88.69	74.01	−14.68
Skiing	77.11	69.01	−8.10
Skijet	97.34	95.22	−2.12
SoccerJuggling	96.74	90.87	−5.87
Surfing	96.83	97.3	0.47
TennisSwing	86.93	81	−5.93
TrampolineJumping	82.96	77.23	−5.73
VolleyballSpiking	85.19	85.55	0.36
WalkingWithDog	88.52	89.11	0.59
frame mAP	**87.23**	**85.72**	**−1.51**

5. VALIDATION

In order to demonstrate the practical utility of the verified CV-HART, it is applied to a real-world case study.

5.1. Case study background

According to the South African Banking Association, about 30 000 ATMs are located across South Africa [52]. These ATMs are situated at on-site (bank) locations and at off-site locations, such as malls and petrol stations. ATMs are constantly replenished. Independent *cash in transit* (CIT) service providers are typically contracted to perform these ATM cash replenishments.

The industry partner attached to this project, a large South African retail bank, developed strict procedural steps and rules to be adhered to by the CIT personnel. Currently, surveillance monitoring is carried out manually, which is prone to human error, inefficiencies, and unnecessary expenditure [53]. In an attempt to automate this monitoring process, the application of CV to perform HAR is considered. The main objectives of this case study involve determining CV-HART's algorithmic performance in this problem context to demonstrate the extent of its potential utility.

5.2. Case study data

The data provided correspond to several months of video footage recorded at an undisclosed location. A snapshot of the footage showing the camera angle and layout of the specific room is shown in Figure 7. Deliberation with the industry partner resulted in the following action classes being deemed important: (1) EnterRoom, (2) OpenSafe, (3) CloseSafe, (4) UseComputer, (5) HandleCash, and (6) ExitRoom.

Figure 7: Snapshot of ATM cubicle video data

5.3. Implementation of CV-HART validation

A sensitivity analysis is employed to determine the sensitivity of a model's performance in respect of parameter variations. Values deemed to be of a small, medium, and large magnitude are considered for each numerical parameter. In the case of non-numerical parameters, a selection of the most popular approaches in the literature is used. A baseline model is established (based on the parameters of Köpüklü et al. [39], as shown in Table 2) after which individual changes are made to each parameter. A summary of the different parameter combinations is presented in Table 5.

Table 5: Parameter combinations employed during the sensitivity analysis

Combination	Learning rate	Batch size	Clip length	Training state	3D Backbone model
Baseline	0.0001	8	16	pre-trained	ResNeXt101
1	0.00001	8	16	pre-trained	ResNeXt101
2	0.001	8	16	pre-trained	ResNeXt101
3	0.0001	16	16	pre-trained	ResNeXt101
4	0.0001	32	16	pre-trained	ResNeXt101
5	0.0001	64	16	pre-trained	ResNeXt101
6	0.0001	8	8	pre-trained	ResNeXt101
7	0.0001	8	16	untrained	ResNeXt101
8	0.0001	8	16	pre-trained	ResNet18
9	0.0001	8	16	pre-trained	ResNet50
10	0.0001	8	16	pre-trained	ResNet101

Ten epochs are used for training, except in the case of combination seven, which evaluates an untrained backbone. Accordingly, the untrained model is trained in ten different training runs, each comprising ten epochs. This approach accounts for the stochasticity associated with training randomly initialised weights. The respective random seeds are provided for replication purposes; they are: 3, 8, 17, 18, 30, 40, 57, 59, 69, 94.

A summary of the final results for each parameter combinations is shown in Table 6, where the frame mAP values for the best epoch for each parameter combination are shown. Only one parameter combination, combination 3, improves on the baseline performance. Combinations 4, 9, and 10 achieve admirable performance (greater than 90.0). From the results, it can also be reported that the model exhibits notable sensitivity to the training state parameter, followed by the learning rate and batch size parameters. The best-performing model corresponds to combination 3, and so is employed for the remainder of this case study. A summary of class-specific performance is presented in Table 7. The performance is consistent across each class. OpenSafe is the only class with a score less than 90%. The best-performing class, UseComputer, shows a near-perfect score of 99.61%.

Table 6: Best frame mAP value for each parameter combination

Parameter varied	Combination	Frame mAP
learning rate	1	85.28
	2	88.48
	3	93.76
batch size	4	91.71
	5	84.00
clip duration	6	86.13
training state (best state)	7	79.96
	8	87.98
3D backbone	9	92.21
	10	90.41

Table 7: Individual class APs from the best-performing parameter combination (combination 3)

Class	AP
CloseSafe	92.00
EnterRoom	96.50
ExitRoom	94.24
HandleCash	91.19
OpenSafe	89.00
UseComputer	99.61
frame mAP	93.76

In an attempt to glean additional insight from the algorithmic performance, an average bounding box is calculated and analysed for each class, as shown in Figure 8. As expected, boxes 1 and 2 – i.e., EnterRoom and ExitRoom – are markedly similar, as are boxes 3 and 4 – i.e., OpenSafe and CloseSafe. Box 5 – i.e., UseComputer – is the most distinct box, while box 6 – i.e., HandleCash – overlaps with boxes 3, 4, and 5 (the last of them being the most notable). The best- and worst-performing classes naturally correspond to the largest and smallest overlap respectively in respect of bounding boxes.

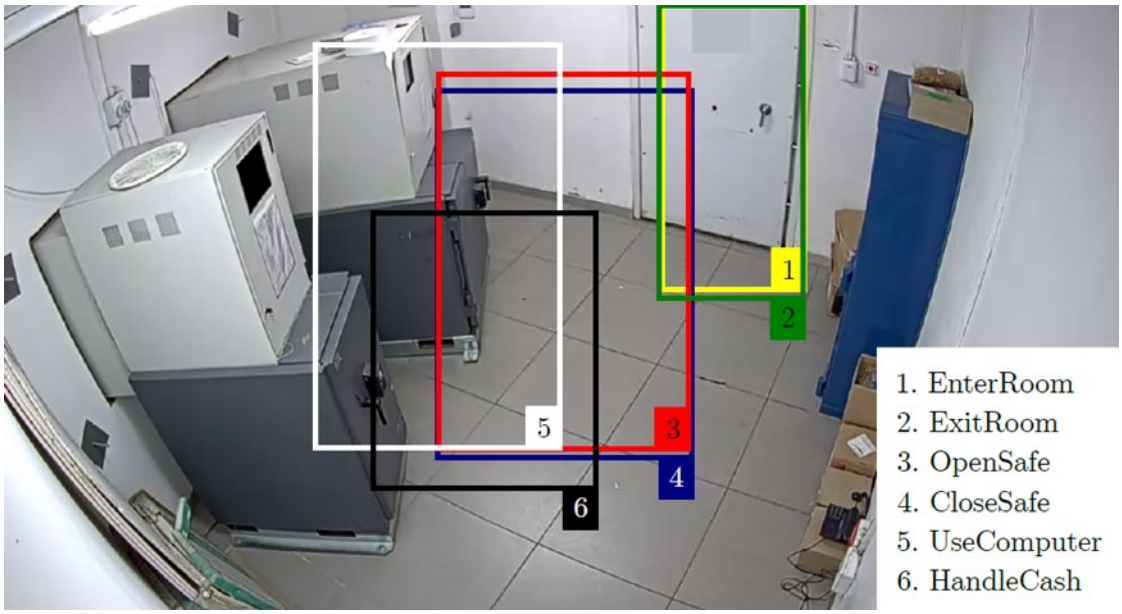

Figure 8: The average bounding boxes for each class of the validation case study

A confusion matrix (shown in Figure 9) is also constructed in an attempt to gain additional insight into the model's performance. It can be seen that the model does, in fact, confuse classes moderately often, with the ExitRoom class being incorrectly predicted in more than 34% of the predictions. Overall, the performance is favourable, as confirmed by the main diagonal.

Further insight can be inferred from the ADT chart, shown in Figure 10. Upon comparing the detected ADT chart with the ground truth data, it is evident that the model detects actions markedly well. The frames with the largest number of incorrect class predictions occur at frame number 5 000; and it is interesting to note that those frames, and other occurrences of incorrect detection, correspond to instances during which no labelled actions of interest occur. The model then continually searches for actions, and the model identifies different actions that do not occur at that time.

To validate the CV-HART further, subject matter experts (SMEs) and key industry partner stakeholders were interviewed in the light of the obtained results. These SMEs were Dr Buitendag [54] (manager of insurance data and analytics), Mr van Staden [53] (team leader at the cash investigations department), and Mr Gerber [55] (projects engineer at the company). All three SMEs stated notable satisfaction with the algorithmic performance results obtained by the CV-HART. Buitendag noted that, although the model is complex (in its scale) and that real-time application could prove difficult at first, he confirmed that deploying the CV-HART on batches of live footage could prove practically beneficial. Both Buitendag and Van Staden agreed that a larger number of classes could result in improved utility. To this end, Gerber also suggested that decomposing classes further into smaller sub-classes – such as expanding OpenSafe to EnterSafeKey and OpenSafeDoor – could add value. Van Staden confirmed that the CV-HART could demonstrably aid process monitoring. The ADT charts were highlighted in particular, given their practically intuitive nature.

In summary (and based on the quantitative studies and the qualitative SME discussions), the following validation-specific conclusion can be reasonably drawn in respect of the proposed CV-HART: the developed tool (which employs HAR techniques based on CV and STAL) represents a successful proof-of-concept in respect of high-quality algorithmic performance and practical utility.

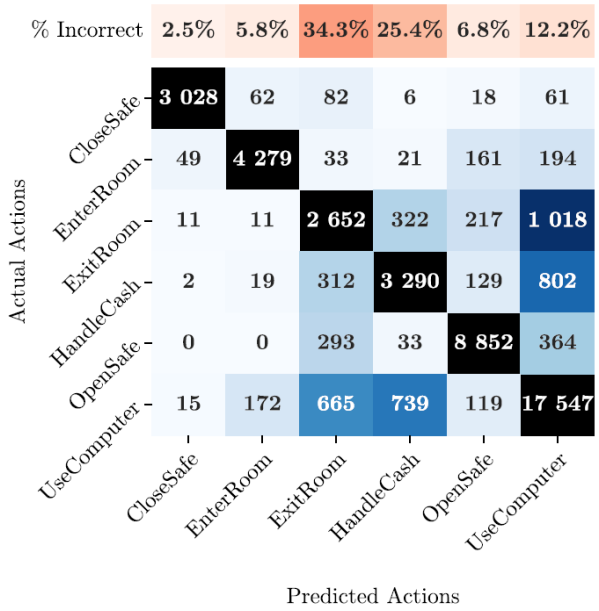

Figure 9: Confusion matrix in respect of case study predictions

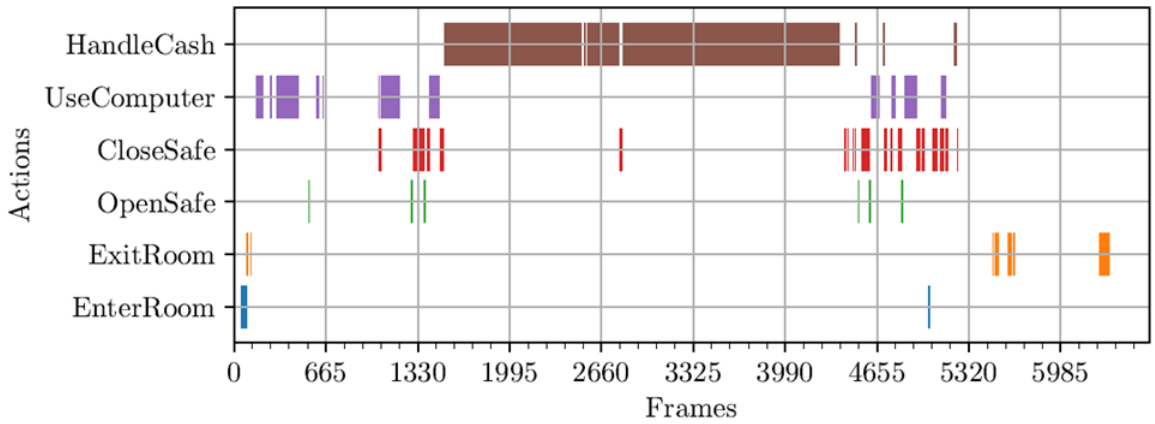

Figure 10: ADT chart depicting the detected actions in a deployment video

6. CONCLUSION

In pursuit of the overarching goal of this project, a generic DST for HAR was designed, developed, and implemented. The novel CV-HART was developed by incorporating and synthesising three fields of research, namely DL, HAR, and DSTs, as reviewed in Section 2. The proposed DST was not only designed conceptually, but also implemented practically. In Section 4, the implementation of the CV-HART with respect to a prominent benchmark STAL data set – i.e., the UCF101-24 data set [44] – was discussed. The aim of the verification case study was to determine whether the YOWO architecture, as it was employed in the CV-HART, was able to achieve a performance score similar to that of the original authors [39]. The CV-HART was verified, and it was shown that a markedly similar performance score could be achieved on the UCF101-24 data set.

Aside from the CV-HART's successful application to the verification case study, the generic capabilities of the approach were also demonstrated. To this end, the application of the CV-HART to a real-world case study was performed and documented in Section 5. The case study data set constituted video surveillance footage provided by an industry partner. Impressive performance was achieved, as exemplified by a frame mAP of 93.76%, confirming both that the CV-HART is generic in its applicability and that it delivers a highly

favourable performance. To the best of the author's knowledge, this HAR use case is novel, and therefore represents a development of the respective DL and HAR bodies of knowledge.

Possible avenues for future work include improving the practical utility of the CV-HART by developing additional functionality in respect of automatically detecting process breaches by means of, for example, rule-based heuristics for process conformance. Other avenues include an investigation into the application of the CV-HART to additional domains (beyond sports activities and replenishment surveillance footage) so as to showcase further the generic nature of the CV-HART. Other domains of interest include detecting the activities of patients in a hospital or in an old-age home for anomaly detection. Another potential avenue for future work is to improve YOWO architectural inefficiencies (e.g., batch frame loading) or to create a semi-automatic data labelling tool for improved usability.

REFERENCES

[1] W. Ouyang *et al.*, "DeepID-Net: Object detection with deformable part based convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1320–1334, 2016.

[2] W. Rattanapitak and S. Wangsiripitak, "Vision-based system for automatic detection of suspicious objects on ATM," *International Conference on Computer Analysis of Images and Patterns*, pp. 570–581, 2015.

[3] H. Sako, T. Watanbe, H. Nagayoshi, and T. Kagehiro, "Self-defense-technologies for automated teller machines," *International Machine Vision and Image Processing Conference (IMVIP 2007)*, pp. 177–184, 2007.

[4] X. Chen and A. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," *Advances in Neural Information Processing Systems*, 2014, pp. 1–9.

[5] E. Nishani and B. Çiço, "Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation," *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, 2017, pp. 1-4. doi: 10.1109/MECO.2017.7977207

[6] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653-1660. doi: 10.1109/CVPR.2014.2142014

[7] S. Cao and R. Nevatia, "Exploring deep learning based solutions in fine grained activity recognition in the wild," *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016, pp. 384–389.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725-1732. doi: 10.1109/CVPR.2014.2232014

[9] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," *IEEE International Conference on Computer Vision*, 2017, pp. 5783–5792.

[10] Z. Hussain, Q. Z. Sheng, and W. E. Zhang, "A review and categorization of techniques on device-free human activity recognition," *Journal of Network and Computer Applications*, vol. 167, 102738, 2020.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, vol. 1. Cambridge, MA: MIT Press, 2016.

[12] Y. Le Cun *et al.*, "Handwritten digit recognition: Applications of neural network chips and automatic learning," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 41–46, 1989.

[13] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[14] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognition*, vol. 15, no. 6, pp. 455–469, 1982.

[15] C. C. Aggarwal, *Neural networks and deep learning*. Cham: Springer, 2018.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778. doi: 10.1109/CVPR.2016.902016

[17] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[18] S. N. Gowda, M. Rohrbach, and L. Sevilla-Lara, "Smart frame selection for action recognition," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 1451–1459.

[19] Z. Qiu, T. Yao, C.-W. Ngo, X. Tian, and T. Mei, "Learning spatio-temporal representation with local and global diffusion," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12056–12065.

[20] D. Tran, H. Wang, L. Torresani, J. Ray, Y. Le Cun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.
[21] Y. Hbali, S. Hbali, L. Ballihi, and M. Sadgal, "Skeleton-based human activity recognition for elderly monitoring systems," *IET Computer Vision*, vol. 12, no. 1, pp. 16–26, 2018.
[22] M. Martinez, L. Rybok, and R. Stiefelhagen, "Action recognition in bed using BAMs for assisted living and elderly care," IEEE, 2015, pp. 329–332.
[23] L. Schrader *et al.*, "Advanced sensing and human activity recognition in early intervention and rehabilitation of elderly people," *Journal of Population Ageing*, vol. 13, pp. 139–165, 2020.
[24] M. Babiker, O. O. Khalifa, K. K. Htike, A. Hassan, and M. Zaharadeen, "Automated daily human activity recognition for video surveillance using neural network," *2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA*), 2017, pp. 1–5.
[25] S. Habib *et al.*, "Abnormal activity recognition from surveillance videos using convolutional neural network," *Sensors*, vol. 21, no. 24, p. 8291, 2021.
[26] M. Popa, L. Rothkrantz, Z. Yang, P. Wiggers, R. Braspenning, and C. Shan, "Analysis of shopping behavior based on surveillance system," *IEEE*, 2010, pp. 2512–2519.
[27] D. Deotale, M. Verma, and Suresh, P., "Human activity recognition in untrimmed video using deep learning for sports domain," *ICICNIS 2020*, 2020, pp. 596–607.
[28] A. Nadeem, A. Jalal, and K. Kim, "Automatic human posture estimation for sport activity recognition with robust body parts detection and entropy Markov model," *Multimedia Tools and Applications*, vol. 80, no. 14, pp. 21465–21498, 2021.
[29] K. Rangasamy, M. A. As'ari, N. A. Rahmad, and N. F. Ghazali, "Hockey activity recognition using pre-trained deep learning model," *ICT Express*, vol. 6, no. 3, pp. 170–174, 2020.
[30] L. Chen, X. Liu, L. Peng, and M. Wu, "Deep learning based multimodal complex human activity recognition using wearable devices," *Applied Intelligence*, vol. 51, no. 6, pp. 4029–4042, 2021.
[31] J. Han *et al.*, "Cbid: A customer behavior identification system using passive tags," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2885–2898, 2015.
[32] Y. Xu and T. T. Qiu, "Human activity recognition and embedded application based on convolutional neural network," *Journal of Artificial Intelligence and Technology*, vol. 1, no. 1, pp. 51–60, 2021.
[33] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," *2017 IEEE Conference on Computer Vision and Pattern Recoginition*, 2017, pp. 6299–6308.
[34] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 466–481
[35] A. Zeng *et al.*, "DeciWatch: A simple baseline for 10x efficient 2D and 3D pose estimation," *ECCV*, 2022.
[36] C. Gu *et al.*, "Ava: A video dataset of spatio-temporally localized atomic visual actions," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6047–6056.
[37] X. Peng and C. Schmid, "Multi-region two-stream R-CNN for action detection," *European Conference on Computer Vision*, 2016, pp. 744–759.
[38] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Action tubelet detector for spatio-temporal action localization," *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4405–4413.
[39] O. Köpüklü, X. Wei, and G. Rigoll, "You only watch once: A unified CNN architecture for real-time spatiotemporal action localization," *arXiv preprint arXiv:1911.06644*, 2019.
[40] J. P. Shim, M. Warkentin, J. F. Courtney, D. J. Power, R. Sharda, and C. Carlsson, "Past, present, and future of decision support technology," *Decision Support Systems*, vol. 33, no. 2, pp. 111–126, 2002.
[41] R. Elmasri and S. B. Navathe, *Fundamentals of database systems*, 6th ed. Boston, MA: Springer, 2010.
[42] I. Foster, R. Ghani, R. S. Jarmin, F. Kreuter, and J. Lane, *Big data and social science: A practical guide to methods and tools*, 2nd ed. Boca Raton, FL: Chapman and Hall/CRC, 2021.
[43] P. Revesz, *Introduction to databases: From biologial to spatio-temporal*. London: Springer, 2010.
[44] K. Soomro, A. R. Zamir, and M. Shah, *UCF101: A dataset of 101 human actions classes from videos in the wild*. Orlando, FL: University of Central Florida, 2012.
[45] D. J. Power, "Supporting business decision making," in D. J. Power, *Decision support systems: Concepts and resources for Managers*. Westport, CT: Quorum Books, 2002, pp. 1–20.
[46] K. E. Kendall and J. E. Kendall, *Systems analysis and design*, 9th ed. Saddle River, NJ: Pearson, 2013.
[47] D. G. Shreyas, S. Raksha, and B. G. Prasad, "Implementation of an anomalous human activity recognition system," *SN Computer Science*, vol. 1, no. 3, pp. 1–10, 2020.

[48] M. Atikuzzaman, T. R. Rahman, E. Wazed, M. P. Hossain, and M. Z. Islam, "Human activity recognition system from different poses with CNN," *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2020, pp. 1–5.
[49] Dark Label, "DarkLabel," Feb. 01, 2021. https://github.com/darkpgmr/DarkLabel (accessed Mar. 01, 2022).
[50] P. M. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets," *Information Technology and Management Science*, vol. 20, no. 1, pp. 20–24.
[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[52] The Banking Association South Africa, "ATM hygiene measures," *The Banking Association South Africa*, Jul. 05, 2021. https://www.banking.org.za/news/atm-hygiene/ (accessed Jun. 01, 2022).
[53] J. van Staden, "Personal communication," Interview, 2022.
[54] S. Buitendag, "Personal communication," 2022.
[55] H. Gerber, "Personal communication," 2022.

APPENDIX A

This section contains a number of graphical depictions of the GUI that was developed as part of the CV-HART.

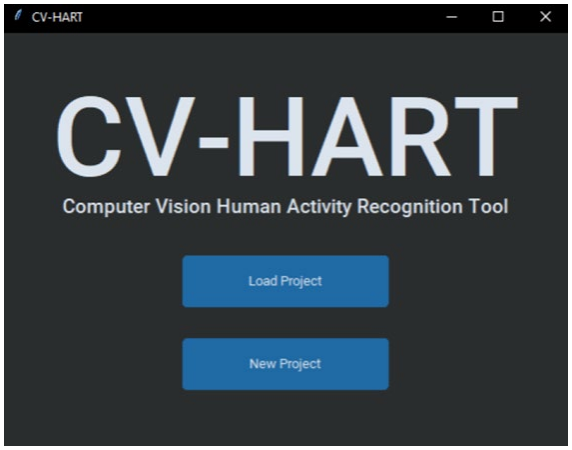

Figure 11: Main CV-HART window

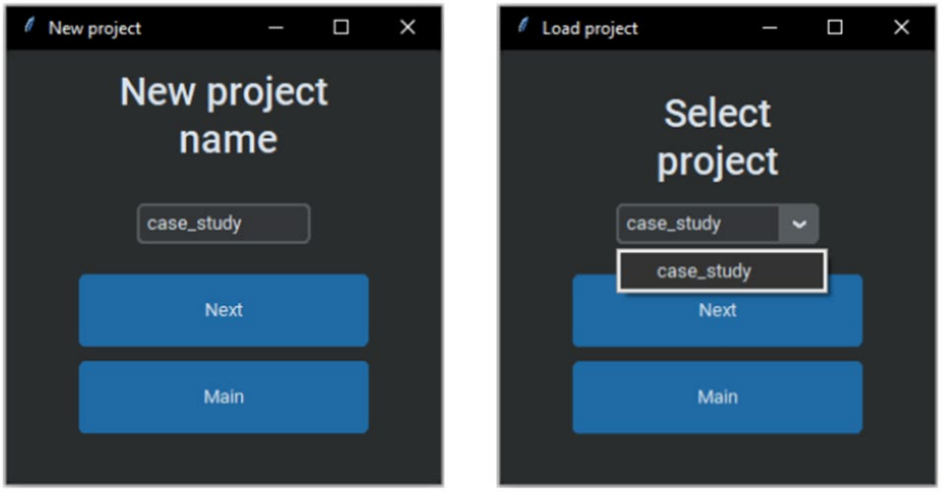

Figure 12: New project window (left) and load project window (right)

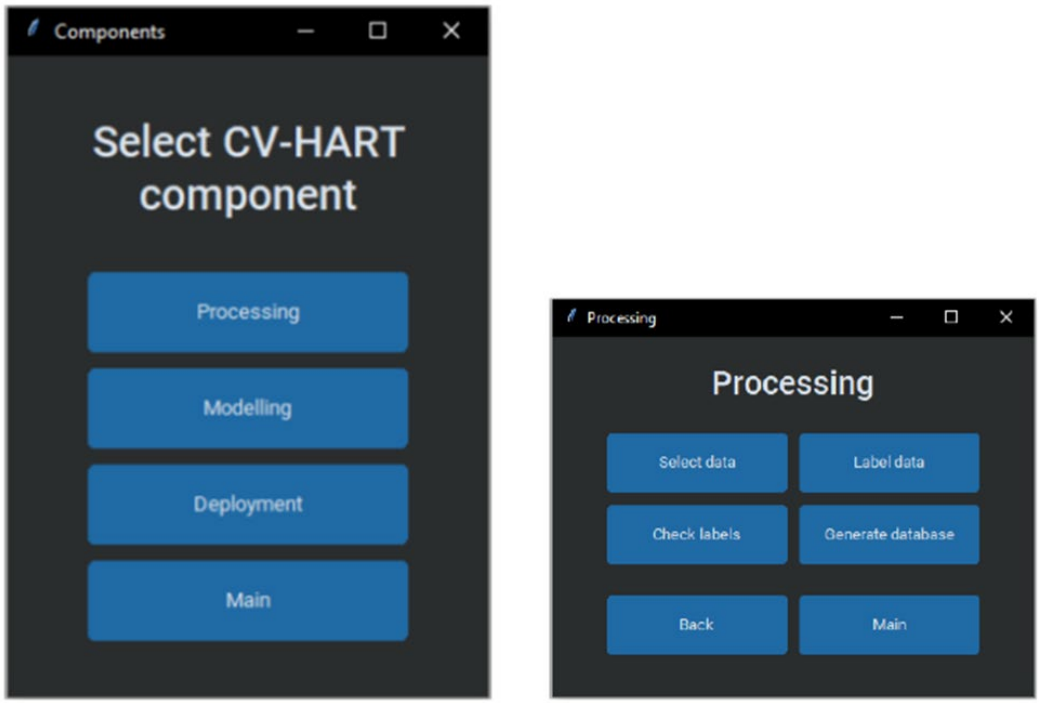

Figure 13: Components window (left) and processing window (right)

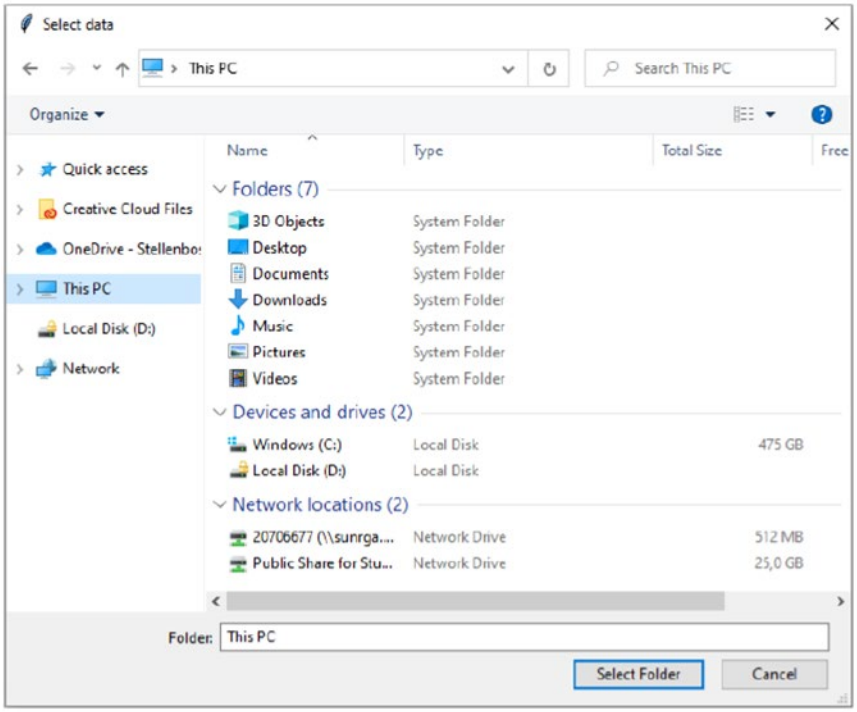

Figure 14: Select data window

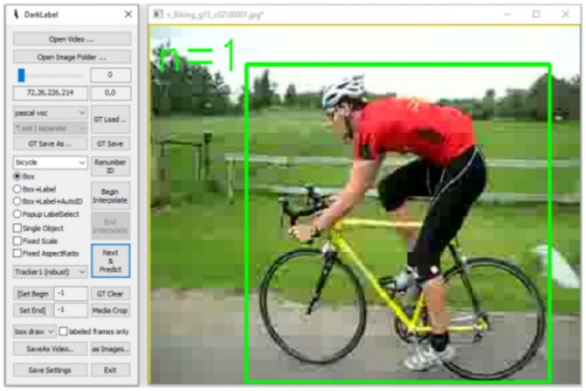

Figure 15: DarkLabel window

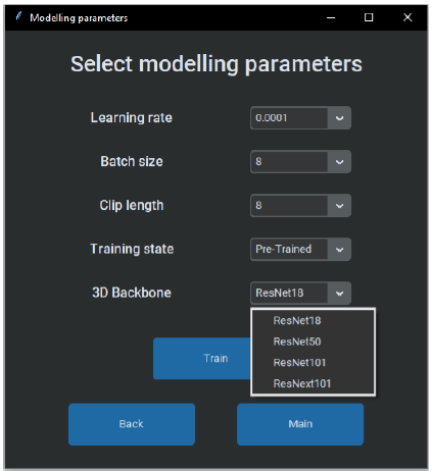

Figure 16: Model parameters window

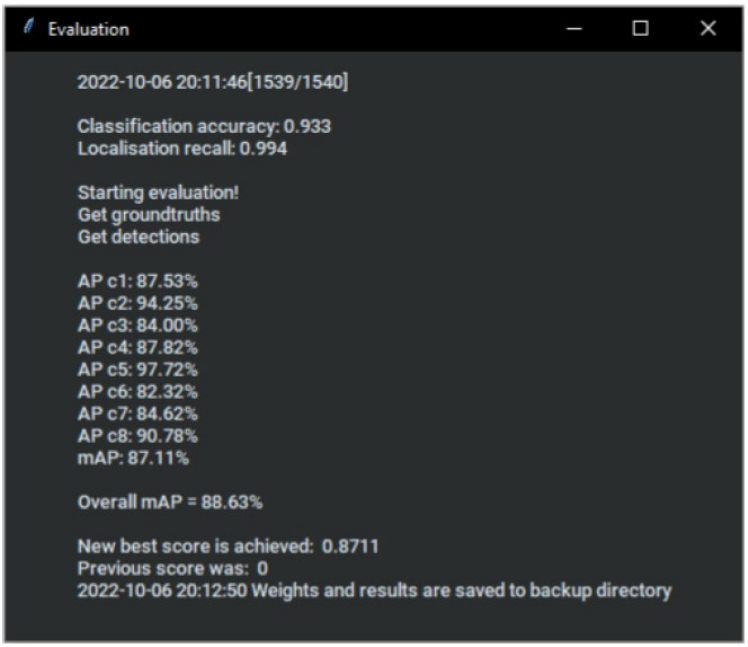

Figure 17: Evaluation window

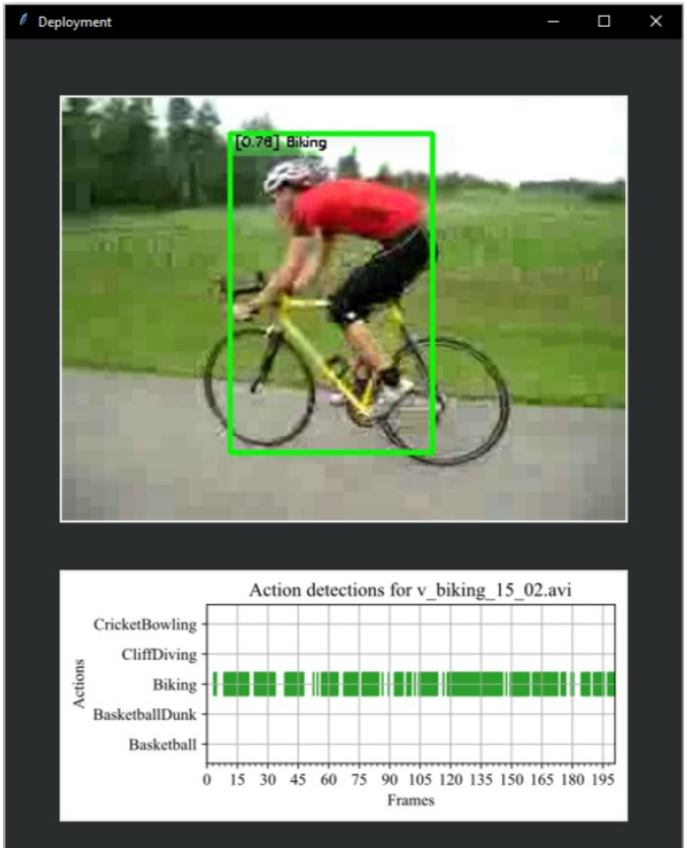

Figure 18: Deployment window