PeerJ Computer Science

Utilizing the YOLOv8 model for accurate hand recognition with complex background

Budhi Kristianto[1], Christine Dewi[1], Hindriyanto Dwi Purnomo[1], Kristoko Dwi Hartomo[1] and Siti Zaiton Mohd Hashim[2]

[1] Department of Information Technology, Faculty of Information Technology, Satya Wacana Christian University, Salatiga, Central Java, Indonesia
[2] Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia

ABSTRACT

Background: The recognition of human hands is essential in the pre-processing stage of several computer vision tasks, as they are actively involved in these actions. This task encompasses hand posture estimation, hand gesture recognition, human activity analysis, and related activities. The human hand shows a wide range of motion and experiences many morphological changes. The presence of numerous individuals in a limited area complicates the precise identification of distinct hand movements, while some hands display a wide variety of motion capabilities. This research's motivation is to open up new opportunities to solve the problems above.
Methods: This article provides a concise analysis of convolutional neural network (CNN)-based object detection algorithms, notably emphasizing the YOLOv8n and YOLOv8s models trained for 50 and 100 epochs. This research examines various object detection algorithms, including ones specifically utilized for hand identification. Furthermore, our proposed method is trained and evaluated on the Oxford Hand Dataset and EgoHand Dataset using the YOLOv8 framework. Performance measures are employed to assess and quantify critical data, including the number of Giga Floating-Point Operations Per Second (GFLOPS), the mean average precision (mAP), and the detection duration.
Results: The results of our experiments show that utilizing YOLOv8n with a training period of 100 epochs produces a more reliable conclusion than other previously published methods. In the training phase, the model exhibited a mean Average Precision (mAP) of 86.7% for the Oxford Hand Dataset and 98.9% for the EgoHand Dataset. Moreover, YOLOv8n with 100 epochs surpasses the maximum average score (mAP) relative to prior research for both datasets.

Subjects Artificial Intelligence, Computer Vision, Data Mining and Machine Learning, Optimization Theory and Computation, Neural Networks
Keywords Deep learning, Machine learning, Hand detection, YOLOv8, Convolutional neural network, Object detection, Hand gesture recognition

Submitted 5 November 2024
Accepted 4 September 2025
Published 20 October 2025

Corresponding authors
Budhi Kristianto, budhik@uksw.edu
Christine Dewi, christine.dewi@uksw.edu

Academic editor
Martina Iammarino

Additional Information and Declarations can be found on page 21

DOI **10.7717/peerj-cs.3244**

INTRODUCTION

The use of one's hands in everyday life is important for a variety of reasons, including communication with other people and engagement with one's surroundings. To accurately discern hand gestures and other human activities, it is imperative to diligently check the precise positioning and motion of an individual's hands while they are being recorded in

writing (*Xu et al., 2020*). The ability to accurately identify and discern hands depicted in images and videos holds significant potential for enhancing various visual processing tasks, including but not limited to the comprehension of gestures and scenes (*Gopikha & Balamurugan, 2023*). The presence of numerous hand variations depicted in images poses a challenge in identifying hands within uncontrolled scenarios (*Narasimhaswamy et al., 2019*). The hand can adopt various orientations, shapes, and sizes. The presence of occlusion and motion blur accentuates the distinct visual characteristics shown by hand (*Dewi & Juli Christanto, 2022*). Several applications of computer vision, particularly human-computer interaction, face considerable difficulties while using in cluttered surroundings (*Rapp, Curti & Boldi, 2021*; *Ashiquzzaman et al., 2020*), sign language recognition (*Shin et al., 2021*), and hand action analysis (*Knights et al., 2021*).

Noteworthy progress has been achieved in the field of hand position estimation and gesture detection in constrained environments in recent years, leading to a notable level of development and refinement. Hand-related applications in un-constrained environments are expected to be a significant trend soon. In the given conditions, the recognition of hands in an uncontrolled setting presents a significant obstacle in the field of hand-related work. Implementing a high-precision hand recognition method is of utmost importance for applications focused on hand-related tasks and functions inside surroundings with restricted limitations. Nevertheless, the use of deep learning techniques for gesture identification may result in a significant reduction in recognition accuracy when confronted with intricate background interferences, such as variations in skin tones included within the gesture image. This is because complex background interferences might make it difficult to distinguish between different people's gestures. The complexity of the hand detection job is strongly correlated with the wide array of hand appearances, which include changes in hand morphology, skin pigmentation, orientation, size, and partial obstruction, among several other attributes. This can pose a significant challenge to the execution of the task (*Guan et al., 2021*). To improve results on the hand identification task, we can use the common information supplied in the training signal for the hand appearance reconstruction task as an inductive bias (*Alam, Islam & Rahman, 2022*). The "hand appearance reconstruction task" involves training a model to recreate the visual characteristics of a hand, such as its shape, texture, and color, from input data. This process requires the model to learn detailed representations of hand features, which can be beneficial for tasks like hand identification.

The YOLO model is named after the maxim "You Only Look Once," referring to the fact that it can complete object recognition tasks with just one forward pass of the neural network rather than requiring multiple passes. YOLOv8 is not merely a more efficient iteration of its forerunner. It is the most up-to-date state-of-the-art model in the YOLO family and includes several architectural upgrades, such as a new backbone network, loss function, anchor-free detecting head, *etc.* (*Dillon et al., 2023*). YOLOv8's latest version has the same architecture as its predecessors 6, but it introduces numerous improvements compared to the earlier versions of YOLO. These improvements include a new neural network architecture that utilizes both Feature Pyramid Network (FPN) and Path Aggregation Network (PAN), as well as a new labeling tool that simplifies the annotation

process. Additionally, this latest version has the same architecture as its predecessors 6. This labeling tool has several helpful features, such as labeling shortcuts, auto labeling, and customized hotkeys (*Lou et al., 2023*). Because these features work together, it is now much simpler to annotate photos to train the model. For the FPN to function, the spatial resolution of the input image is gradually lowered while the number of feature channels is progressively raised. This leads to the production of feature maps that can find things on a variety of scales and resolutions because of the process. On the other hand, the PAN design uses skip connections to aggregate features from multiple tiers of the network. The network will be able to better collect features at many scales and resolutions if this is done, which is essential for reliably detecting objects of varying sizes and forms (*Ultralytics, 2022*; *Rossoshansky, 2023*).

Research gap and work intention

Hand gesture identification in intricate backgrounds poses considerable difficulties due to elements such as fluctuating illumination conditions, obstructions, and background disarray. For example, it's difficult to identify object with very bright background or with chaotic background that has similar cross section of colors. Although prior research has used diverse deep learning models for hand gesture detection, numerous algorithms show deficiencies in effectively finding motions inside intricate contexts. Furthermore, there is a dearth of research concentrating on the use of YOLOv8 models specifically designed for hand motion identification in intricate backgrounds. To address this gap, we explore the application of YOLOv8 models, specifically YOLOv8n and YOLOv8s, for hand gesture recognition tasks. By training these models on comprehensive datasets and evaluating their performance, we aim to enhance the accuracy and reliability of hand gesture recognition systems in complex backgrounds.

The following is the most important contribution that can be made from conducting this research: (1) this research contains a synopsis of the YOLOv8 family of object identification algorithms, which includes YOLOv8n and YOLOv8s with 50, 100, and epochs, as well as a brief discussion of each of these variants. (2) Many different types of object detectors are employed in this study. Metrics of performance keep a close eye on essential pieces of information, such as the average mean accuracy (mAP), the intersection over union (IoU), and the number of Giga Floating-Point Operations Per Second (GFLOPS). (3) Our proposed technique is trained and tested on the Oxford Hand Dataset and EgoHand Dataset using the YOLOv8 framework.

The outline of the article is as follows. 'Related Works' presents a comprehensive overview of the existing research articles in the field, along with a detailed explanation of the method employed in this study. The results of the experiments are presented in 'Methodology'. The findings are discussed in 'Experiments and Results', while 'Discussion' provides an outline of the conclusions drawn from the study and suggests potential avenues for future research.

RELATED WORKS

Convolutional neural network (CNN) used for hand recognitions

Convulational neural networks (CNNs) have proved extraordinary performance in image identification tasks due to their capacity to automatically learn and extract relevant features from images. This ability has enabled CNNs to achieve this level of success. After trying out several assorted color spaces, *Girondel, Bonnaud & Caplier (2006)* found that the Cb and Cr channels in the YCbCr color space were particularly effective for the skin recognition job. The Gaussian mixture model was proposed by *Sigal, Sclaroff & Athitsos (2004)*, and it performed exceptionally well under a wide variety of illumination conditions. Because precise hand detection is necessary for a wide variety of applications, *Mittal, Zisserman & Torr (2011)* developed a method that makes use of several movable parts.

Hand detection is a computer vision technique that involves showing and finding human hands in images or videos. It has various applications across different domains. Some of the popular hand detection applications include (1) The detection of hands plays a pivotal role in the realm of gesture recognition, as it helps the identification and interpretation of hand gestures. These gestures, in turn, have the potential to exert control over various devices, facilitate interaction with user interfaces, and even help the translation of sign language (*Adhikari et al., 2023*). (2) Human-computer interaction (HCI): hand detection can enhance natural user interfaces by allowing users to interact with computers, smartphones, or other devices through hand movements and gestures (*De Souza Vieira, Ribeiro Filho & De Salles Soares Neto, 2021*). (3) Augmented reality (AR): in AR applications, hand detection enables users to interact with virtual objects or manipulate the virtual environment using their hands (*Marrahi Gomez & Belda-Medina, 2023*). (4) Virtual reality (VR): like AR, hand detection is used in VR to create a more immersive experience, allowing users to interact with the virtual world using their hands (*Tran et al., 2023*). (5) Sign language translation: hand detection combined with natural language processing can be used to interpret sign language and translate it into spoken or written language (*Farooq, Mohd Rahim & Abid, 2023*). (6) Gaming: hand detection is used in motion-controlled games, where users can control the game characters or perform in-game actions using hand gestures (*Meriläinen, 2023*). (7) Biometrics: hand detection can be used as a biometric authentication method, identifying individuals based on the unique characteristics of their hands (*Azhar et al., 2023*). (8) Robotics: in robotics, hand detection helps robots understand and respond to human gestures, allowing for more intuitive human-robot interactions. (8) Health and rehabilitation: hand detection can be used in healthcare settings for monitoring and analyzing hand movements in patients during rehabilitation exercises or evaluating conditions related to hand dexterity (*Mengash et al., 2023*).

Furthermore, *Núñez et al. (2018)* employed a neural network in conjunction with a long short-term memory (LSTM) network to discern three-dimensional hand motions by using the temporal characteristics of a skeletal structure (*Xia & Xu, 2022*). The computer vision field has seen a discernible surge in the level of attention dedicated to CNN-based detection algorithms as a subject of research. The situation in question can be attributed to the

capacity of networked systems to acquire more profound and advanced features. The application of CNNs enables proficient resolution of the challenges associated with multi-scale and diverse rotations, as previously mentioned.

Current scholarly investigations have prioritized three primary domains to enhance the efficacy of object-detecting systems. The principles can also be applied to the task of hand detection using CNN-based methods (*Dai, Fan & Dewi, 2023*). In this section, we shall explain the three main directions. The primary and essential stage in this method involves altering the fundamental architecture of these networks. The second principal aim is to improve the data's potential through the augmentation of training data quality and diversity. The statement pertains to the determination of the second principal direction (*Dewi et al., 2021*). The opportunity for further study is clear in the use of proxy tasks to enhance object detection representations, particularly in reasoning and other top-down processes (*Dewi et al., 2022*). Our efforts are mostly focused on this third primary direction. We are now able to incorporate data that is readily accessible across the world into our detection system because of advances in hand appearance reconstruction (*Chen et al., 2020*; *Dewi et al., 2023*).

METHODOLOGY

YOLOv8 architecture

The fundamental structure of YOLOv8 is like that of YOLOv5, with the C3 module being substituted by the C2f module, which is derived from the Cross-Stage Partial Network (CSP) concept. The C2f module in YOLOv8 was developed by incorporating the Efficient Layer Aggregation Network (ELAN) concept from YOLOv7 and combining it with C3. This integration aimed to enhance the gradient flow information in YOLOv8 without compromising its lightweight design. In the final stage of the backbone architecture, the prevailing Spatial Pyramid Pooling-Fast (SPPF) module continued to be employed. This was followed by a sequential application of three Waxpools, each with a size of 5×5. Subsequently, the output of each layer was concatenated to ensure accurate detection of objects at different scales, while maintaining a lightweight design.

Within the cervical region, YOLOv8 utilizes the Path Aggregation Network-Feature Pyramid Network (PAN-FPN) feature fusion methodology to augment the amalgamation and use of feature layer information across different scales. The neck module in YOLOv8 was constructed by the researchers through the use of two up-sampling approaches, several C2f modules, and the final decoupled head structure. The technique of detaching the head in YOLO was previously used by YOLOv8 in the latter portion of the neck. Through the integration of confidence and regression boxes, a heightened level of accuracy was achieved. The YOLOv8 model shows the ability to support several iterations of YOLO and shows the adaptability to smoothly switch between different versions as needed. Moreover, the program has a remarkable level of adaptability as it has the ability to operate on various hardware architectures, encompassing both central processing units (CPUs) and graphics processing units (GPUs). The convolutional neural network (CNN) illustrated in Fig. 1 comprises three primary elements, including convolution, batch normalization, and Sigmoid Linear Unit (SiLU) activation functions.

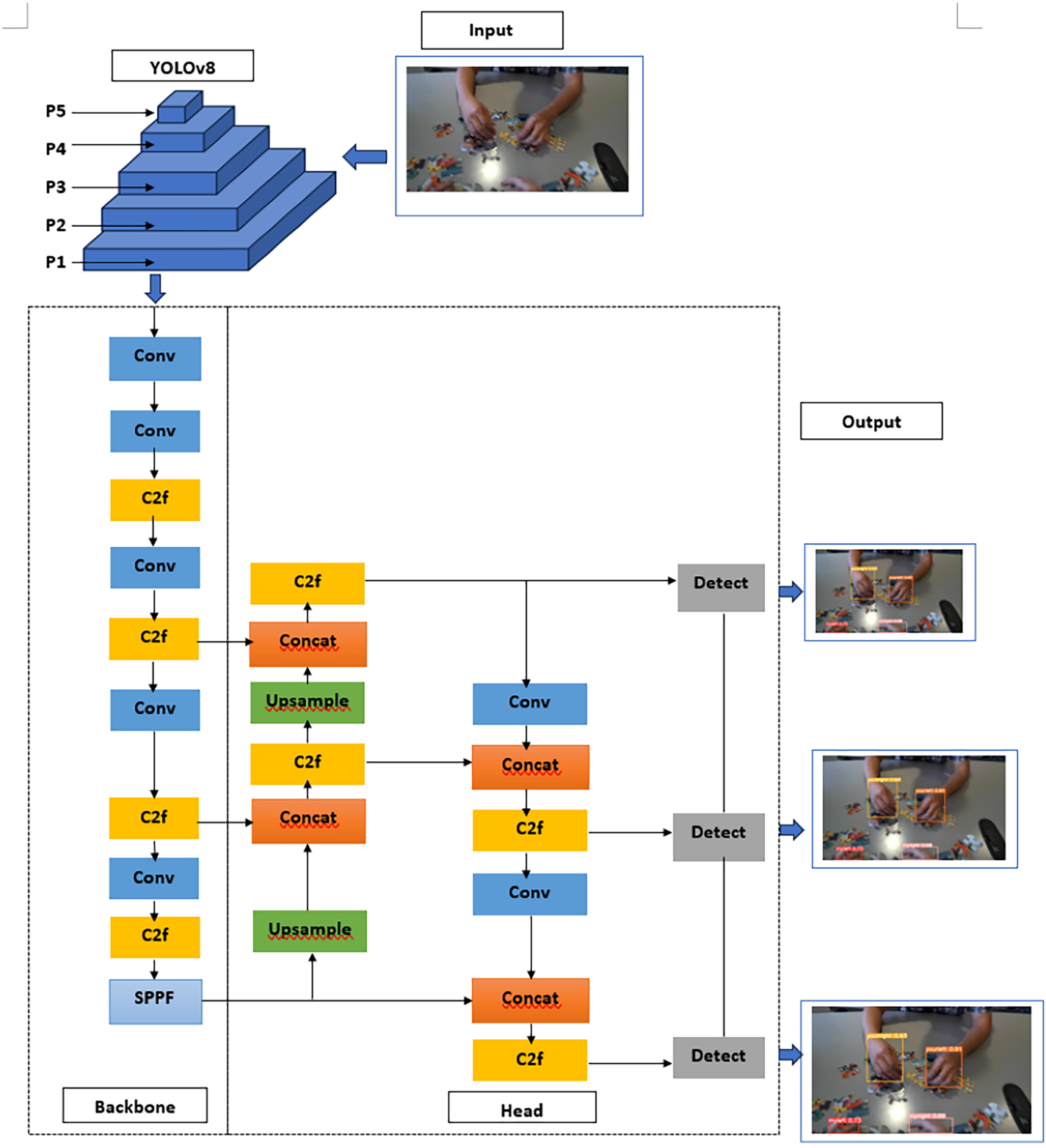


Figure 1 The architecture of YOLOv8. Full-size DOI: 10.7717/peerj-cs.3244/fig-1

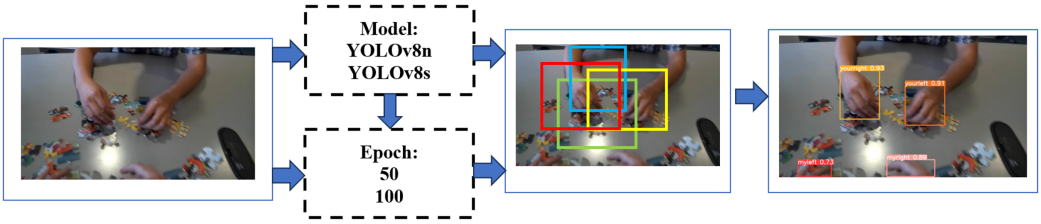


Figure 2 Research procedures. The images of people's faces were obtained from a publicly available database (the Oxford Hand Dataset). Full-size DOI: 10.7717/peerj-cs.3244/fig-2

The YOLOv8 network can be divided into the input network, the backbone network, and the head network, as shown in the structural diagram (*Dong et al., 2022*). SiLU was employed as the activation function in YOLOv8, just like it was in YOLOv7 and YOLOv5, respectively. There was a recommendation made that we make use of the ELAN module. The cardinality of the network was augmented, randomized, and merged to consistently enhance its learning capabilities while preserving the original gradient trajectory. This was achieved without altering the channel count in our feature map ensembles from their original design. The preservation of the same channel count eased the accomplishment of this goal. Ultimately, the quantity of channels seen on the output of the ELAN module is twice as many as the number of channels seen on the input. The max-pooling operation that was conducted by the top branch of the MaxPooling (MP) module resulted in a reduction of fifty percent in both the dimensions of the feature map and the number of channels. Following the completion of the primary convolution, the lower branch decreased the length and width of the feature map by a factor of fifty each, while simultaneously increasing the kernel size and stride by a factor of one and two, respectively. Both tree tiers have been combined into one. After all that effort, we were finally able to produce a feature map that had input and output channels of equal size (*Ultralytics, 2022*). Hence, the present research initially employed a 1×1 convolution to reduce dimensionality, followed by a 3×3 convolution for down sampling. This approach effectively minimizes computational load. In this procedure, the Maxpool layer and depth-wise separable convolution were combined. This approach has the potential to effectively compensate for the loss of information that occurs during the down-sampling process of each item.

The research workflow is depicted in Fig. 2. In this study, the hand detection procedure was employed, using photos sourced from the Oxford hand dataset (*Mittal, Zisserman & Torr, 2011*) as the input data. Subsequently, we are engaging in training our dataset using the advanced YOLOv8n and YOLOv8s architectures, employing 50 and 100 epochs for each respective model. Subsequently, we shall engage in a comprehensive examination and discourse about the outcomes derived from the training and testing stages of YOLOv8. This intricate process encompasses the meticulous computation of the bounding box through the use of Non-Maximum Suppression (NMS). We do not implement data processing other than that provided by the basic YOLOv8 model.

The Yolo labeling format is commonly used by annotation programs to generate output. This format organizes annotations for each image into a single text file. The annotation for each graphical element in an image is represented by a bounding box, also referred to as a "BBox" abbreviation, in the corresponding text file. The scale of the annotations has been changed to maintain proportionality with the image. The values of the annotations range from 0 to 1 (*Long et al., 2021*). In the computation that will be done using the YOLO format, Eqs. (1) through (6) will serve as the basis for the adjustment technique.

$$dw = 1/W. \tag{1}$$

$$x = \frac{(x_1 + x_2)}{2} \times dw. \tag{2}$$

$$dh = 1/H. \tag{3}$$

$$y = \frac{(y_1 + y_2)}{2} \times dh. \tag{4}$$

$$w = (x_2 - x_1) \times dw. \tag{5}$$

$$h = (y_2 - y_1) \times dh. \tag{6}$$

The variable H is employed to stand for the height of the image, while dh is utilized to signal the absolute height of the image. Similarly, the variable W is employed to signify the width of the image, while dw is used to stand for the absolute width of the picture.

EXPERIMENTS AND RESULTS

Computing infrastructure

We employed a Windows 11 Enterprise system that was equipped with an Intel(R) Core (TM) i9-12900H CPU 2.50GHz, 32 GB of RAM, and an NVIDIA RTX 3060 GPU to conduct this investigation. Additionally, we implemented simulations with Google Colab pro+.

Model evaluation

Ultralytics has employed the Binary Cross-Entropy with Logits Loss function, provided by PyTorch, to quantify the extent of loss experienced concerning both the class probability and the object score. This has been done to calculate the amount of loss that has occurred (*Zhao & Zhang, 2021*). A true positive (TP) refers to the count of instances where the model evaluation and the actual situation both indicate a positive outcome. On the other hand, a true negative (TN) denotes the count of instances where the model evaluation and the actual situation both indicate a negative outcome. The terms "TP" and "TN" are commonly abbreviated to denote true positive and true negative, respectively. In the context of statistical modeling, a false positive (FP) refers to a situation where the observed data does not align with the predicted value derived from the model. Conversely, a false negative (FN) arises when the observed data **does not align** with the predicted value derived from the model (*Dewi & Chen, 2022*).

$$Precision\ (P) = \frac{TP}{TP + FP}. \tag{7}$$

$$Recall\ (R) = \frac{TP}{TP + FN}. \tag{8}$$

$$Accuracy\ (Acc) = \frac{TP + TN}{TP + FN + FP + FN}. \tag{9}$$

The metrics encompassed in this analysis include Precision, Recall, and Accuracy. Within the set of metrics, Precision and Recall are formally defined in Eqs. (7) and (8), respectively. Subsequently, Accuracy is precisely delineated in Eq. (9), correspondingly (*Han & Zeng, 2022*; *Jiang et al., 2022*).

The integration across the precision function p(o) yields the arithmetic mean of the average precision (mAP) and the intersection over union (IoU) as depicted in Eqs. (10) and (11) correspondingly.

$$mAP = \int_0^1 p(0)do \tag{10}$$

where p(o) denotes the level of accuracy achieved by object detection. IoU determines the percentage of overlap between the bounding box of the prediction (pred) and the ground-truth value (gt) (*Arcos-García, Álvarez-García & Soria-Morillo, 2018*).

$$IoU = \frac{Area_{pred} \cap Area_{gt}}{Area_{pred} \cup Area_{gt}}. \tag{11}$$

Moreover, it is worth noting that FLOPS, which stands for Floating Point Operations Per Second, can be quantified and expressed in various degrees of precision. In our experimental setup, we have successfully incorporated the Gigaflops (Giga Floating Point Operations Per Second) metric, which denotes a computational speed of 10^9 FLOPS (Floating Point Operations Per Second). This achievement is evident in the equations referenced as Eq. (12) within our study.

$$FLOPS = cores \times \frac{cycles}{second} \times \frac{FLOPS}{cycle}. \tag{12}$$

Moreover, Eq. (13) (*Redmon et al., 2016*) shows the calculation of the YOLO loss functions.

$$\begin{aligned} Yolo\ Loss\ Function = {} & \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + \left(\mathrm{y} - \hat{y}_i\right)^2 \right] \\ & + \boldsymbol{\lambda_{coord}} \sum_{\boldsymbol{i=0}}^{\boldsymbol{s^2}} \sum_{\boldsymbol{j=0}}^{\boldsymbol{B}} \mathbb{I}_{\boldsymbol{ij}}^{\boldsymbol{obj}} \left[\left(\sqrt{\boldsymbol{w_i}} - \sqrt{\widehat{\boldsymbol{w}} \boldsymbol{i}} \right)^2 + \left(\sqrt{\boldsymbol{h_i}} - \sqrt{\widehat{\boldsymbol{h}}_{\boldsymbol{i}}} \right)^2 \right] \\ & \boldsymbol{the}\ + \sum_{\boldsymbol{i=0}}^{\boldsymbol{s^2}} \sum_{\boldsymbol{j=0}}^{\boldsymbol{B}} \mathbb{I}_{\boldsymbol{ij}}^{\boldsymbol{obj}} \left(\boldsymbol{C_i} - \widehat{\boldsymbol{C}}_{\boldsymbol{i}} \right)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} \left(C_i - \widehat{C}_i \right)^2 \\ & + \sum_{i=0}^{s^2} \mathbb{I}_i^{obj} \sum_{c \in classes} \left(p_i(\mathrm{c}) - \hat{p}_i(c) \right)^2. \end{aligned} \tag{13}$$

Assume that S is the image's total number of grid cells. The predicted class between each grid cell is denoted by c, and the number of bounding boxes that are expected to exist within each grid cell is denoted by B. Additionally, the sign $pi(c)$ denotes the confidence probability score. Within the framework of cell i, the variables x_{ij} and y_{ij} correspond to the coordinates of the anchor box's center. The variable hij represents the height of the box, while w_{ij} indicates its width. Additionally, C_{ij} denotes the confidence score associated with the box. The relative relevance of localization in the context of the job at hand is determined by using the weights $\lambda coord$ and $\lambda noobj$.

The YOLOv8 model employs the anchor-free approach as opposed to the Anchor-Based method, and it uses a dynamic TaskAlignedAssigner for its matching strategy. The alignment degree of the anchor level for each instance is computed by employing Eq. (14), where s stands for the classification score, u denotes the IOU value, and α and β are the weight hyperparameters. The algorithm strategically finds a set of m anchors, specifically those with the highest value (t), as positive samples within each instance. Conversely, it designates the remaining anchors as negative samples. Subsequently, the model undergoes training by optimizing the loss function.

$$t = s^{\alpha} \times u^{\beta}. \tag{14}$$

Following the enhancements, YOLOv8 has showed a marginal yet noteworthy 1% increase in accuracy compared to its predecessor, YOLOv5. Consequently, it has appeared as the most precise detector hitherto developed. One salient attribute of YOLOv8 exists in its intrinsic capacity for extensibility. The YOLOv8 framework has been meticulously designed to ensure compatibility across all iterations of YOLO. This remarkable feature allows researchers involved in YOLO projects to effortlessly assess and compare the performance of their respective models, thereby enhancing the convenience and efficiency of their work. This feature presents noteworthy benefits to the scholarly community. Consequently, the YOLOv8 iteration was selected as the benchmark model.

Oxford hand dataset

The Oxford hand dataset (*Mittal, Zisserman & Torr, 2011*) is a compendious and openly accessible collection of images depicting hands, which has been meticulously curated from a diverse array of publicly accessible image datasets.

The provided image holds annotations that highlight various instances of hands, which are readily discernible to human observers. Throughout the entirety of the dataset, a cumulative count of 13,050 instances of hands is seen. The training set assigns 11,019 data points to each hand instance, while the testing set only assigns 2,031 data points to each hand instance. During the whole process of data collection, there were no limitations put on the subjects' immediate environmental context, nor were there any restrictions placed on the subjects' attitudes or visibility. In addition, there were no restrictions placed on the at-attitudes or visibility of the subjects. The photos are accompanied by captions that name all the hands that are visible to the naked eye within the photographs. Human observers can distinguish between these hands. The annotations should be properly positioned around the wrist, while the bounding rectangles are not required to be aligned along any

Figure 3 Oxford hand dataset illustrations. The human features depicted in the figures are derived from publicly available datasets (Oxford Hand Dataset). Full-size DOI: 10.7717/peerj-cs.3244/fig-3

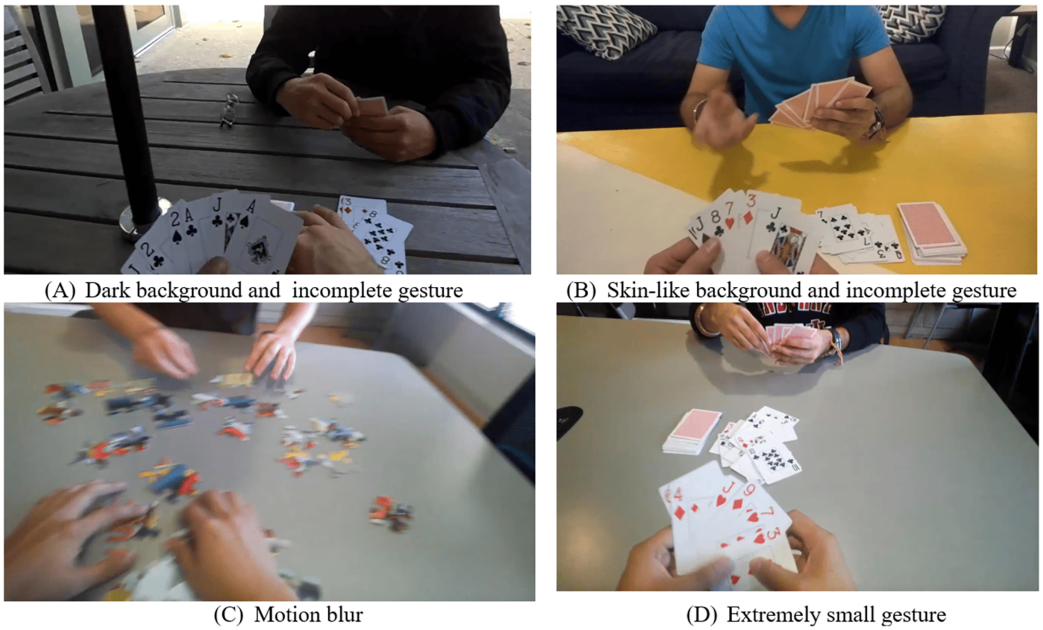
specific axis. The annotations in the 'annotations' folder are stored in the standard MATLAB ".mat" format, specifically being the coordinates for the four corners of the hand-bounding box.

The building is in the shape of a series of boxes, with indices represented by hand boxes. We execute the necessary data preparation on this dataset before exporting it in YOLO format. Annotations must be converted to the YOLOv8 format, which involves creating text files for each image with lines indicating object class and bounding box coordinates. The format is: <class_id> <x_center> <y_center> <width> <height>, with all values normalized between 0 and 1 relative to the image dimensions.

The training dataset comprises 70% of the total data, whilst the testing dataset constitutes the remaining 30%. Both sections contain depictions of diverse hand-held objects. Figure 3 depicts an illustrative instance of a picture derived from the Oxford hand dataset.

EgoHand dataset

The EgoHand dataset (*Bambach et al., 2015*) consists of 48 Google Glass videos of complex and first-person interactions, such as card games, chess, puzzles, and Jenga, between two individuals in various locations. There are a total of 130,000 video frames, of which 4,800 have been labeled and 15,053 instances of hands have been annotated. The EgoHand dataset includes a total of 48 videos saved in the MP4 (h264) format. Each video is a minute and a half long and has a resolution of 720 × 1,280 pixels at 30 frames per second. The labeled folder contains all the frames with their respective labels saved as 720 × 1,280 px JPEG files. The ground-truth labels are provided as MATLAB files and offer a



(A) Dark background and incomplete gesture

(B) Skin-like background and incomplete gesture

(C) Motion blur

(D) Extremely small gesture

Figure 4 Example of EgoHand dataset. Full-size DOI: 10.7717/peerj-cs.3244/fig-4

straightforward application programming interface for them. These masks are pixel-level and correspond to each hand type. We employ EgoHand Dataset to evaluate our proposed method YOLOv8 which consists of four classes (myleft, myright, yourleft, yourright). This dataset consists of complex backgrounds including : (1) dark background and incomplete gesture, (2) Skin-like background and incomplete gesture, (3) motion blur, and (4) extremely small gesture. Figure 4 shows an example of EgoHand Dataset. When dealing with a data set that consists of complex backgrounds for hand gesture recognition, it's important to preprocess the data effectively to improve model performance.

Training result

Within this segment, we shall try to furnish an all-encompassing elucidation of the training method and its corresponding outcomes. The elucidation of the training procedure for test batch 0 labels and test batch 0 predictions is depicted in Figs. 5 and 6.

The YOLOv8 model uses a genetic algorithm to autonomously generate the anchor boxes, thereby enhancing its ability for independent decision-making and adaptive optimization. The procedure is commonly denoted as "auto anchor" due to its inherent capability to autonomously recalibrate the anchor boxes, thereby enhancing their suitability for the given data, especially in cases where the default anchor boxes are deemed inadequate. The information provided is after incorporated into the k-means algorithm to generate k-means evolved anchor boxes. To imbue a network with profound supervision, one may opt to strategically introduce an auxiliary head within the intermediate layers at any given position. The auxiliary head can be effectively steered by using the shallow network weights and incorporating the assistance loss. In circumstances wherein the model weights would conventionally converge, this approach can still prove helpful for

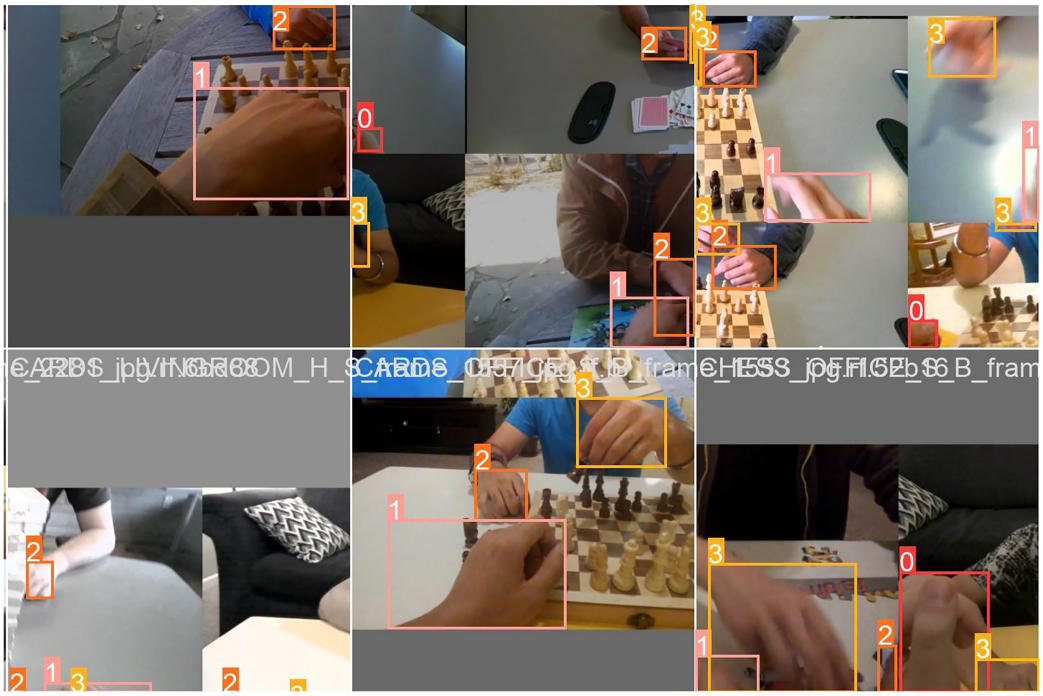


Figure 5 Batch 0 labels and batch 0 prediction for testing with Oxford Hand Dataset.
Full-size DOI: 10.7717/peerj-cs.3244/fig-5

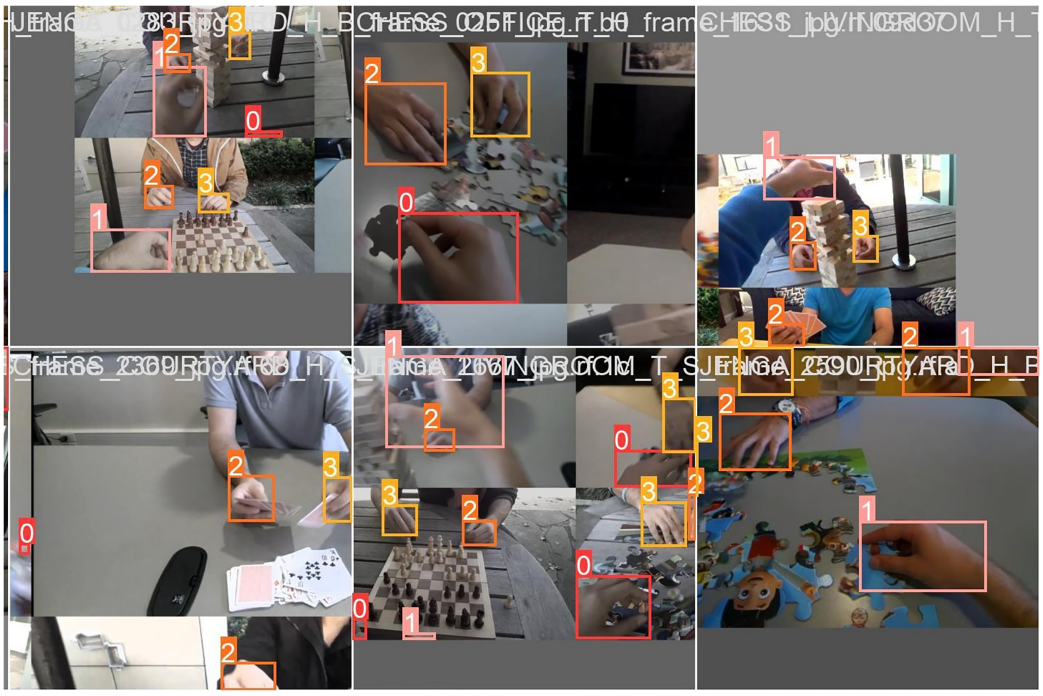


Figure 6 Batch 0 labels and batch 0 prediction for testing with EgoHand Dataset.
Full-size DOI: 10.7717/peerj-cs.3244/fig-6

effectuating modifications to the model. Within the architectural framework of YOLOv8, the training head is denoted as an auxiliary head, while the primary head assumes the responsibility of overseeing the production of the final output. The utilization of lead head prediction within the YOLOv8 model serves to offer guidance in generating hierarchical labels that progress from a general to a more specific level. The labels are subsequently utilized for auxiliary head acquisition and primary head acquisition, correspondingly.

Furthermore, the training process of YOLOv8 involves the merging of four separate images through concatenation. In addition, YOLOv8's training phase involves the combination of four distinct images. Following a random processing step during the splicing phase, each of the four distinct images has different dimensions and configurations than the others. The validation script will be employed to analyze our model. The 'task' setting enables users to personalize the evaluation of their model's performance, choosing between the whole training set, the validated test set, or the test set exclusively. The default location for storing results is the "runs/train" directory. In subsequent training sessions, a new directory called "experiment" is created within the "runs/train" directory. Each new directory is assigned a unique name, such as "exp1", "exp2", and so on. Please refer to the train and val.jpg files to observe the mosaics, labels, forecasts, and augmentation effects. It is noteworthy to mention that the training process necessitates the utilization of an Ultralytics Mosaic Data loader, which is a device designed to amalgamate four distinct photos into a unified mosaic. Upon completing 50 and 100 epochs of training, we proceeded to save the weight obtained from the model.

The discretionary nature of fine-tuning makes it the concluding stage of training. During this phase, the whole model that was previously constructed will be unfrozen and subjected to retraining using a significantly reduced learning rate, using our dataset. Significant improvements may be achievable by iteratively changing the pre-existing trained features to adapt to the new data. A hyperparameters configuration file is a file that stores various settings and values for the hyperparameters of a machine learning model or algorithm. Hyperparameters are parameters that are set before the learning process begins and affect the behavior and performance of the model during training. When the learning rate is significantly decreased compared to conventional settings in a machine learning model, it can have several implications on the model's training process and overall performance. The weights will be initialized with the most recently saved values from the preceding stage. Following the established convention in PyTorch, the trained model has been saved using the .pt file extension.

The mean average precision at a threshold of 0.5 (mAP@0.5) will be assessed during the training process to evaluate the proficiency of our detector in showing objects in the validation dataset. A higher value of mAP@0.5 signifies enhanced performance. The dataset written in YAML Ain't Markup Language is a vital component in the training process of YOLOv8. This file contains a list of class names and the corresponding data locations used for training and validation. To ensure proper determination of the positions of the images, labels, and classes in the training script, it is essential to include the file path as an argument. The dataset is already populated with the necessary data.

Table 1 Training effectiveness of YOLOv8 using the Oxford hand dataset.

Epoch	Model	Images	Class	Labels	P	R	mAP@.5
50	Yolov8n	1,223	All	2,898	0.776	0.687	0.758
100	Yolov8n	1,223	All	2,898	0.883	0.784	0.867
50	Yolov8s	1,223	All	2,898	0.762	0.706	0.749
100	Yolov8s	1,223	All	2,898	0.768	0.695	0.749

Table 2 Training effectiveness of YOLOv8 using the EgoHand dataset.

Epoch	Model	Images	Class	Labels	P	R	mAP@.5
100	Yolov8n	480	All	1,515	0.975	0.968	0.989
100	Yolov8n	480	myleft	262	0.949	0.962	0.985
100	Yolov8n	480	myright	384	0.984	0.971	0.993
100	Yolov8n	480	yourleft	455	0.983	0.971	0.988
100	Yolov8n	480	yourright	450	0.984	0.967	0.99

Table 1 presents a comprehensive overview of the training procedure for YOLOv8n and YOLOv8s, encompassing 50 and 100 epochs, respectively. The YOLOv8n model, trained in over 100 epochs, exhibits notable performance metrics. It achieves a precision of 88.3%, a recall of 78.4%, and a mean average precision (mAP) of 86.7%. The training process requires approximately 8.616 hours, and the resulting model size is 142.1 MB. Moreover, when YOLOv8s is trained for 100 epochs, it demonstrates a precision value of 76.8%, a recall value of 69.5%, and a mean average precision (mAP) of 74.9%.

The training approach for YOLOv8n with 100 epochs is laid out in detail in Table 2, which also provides an overview of the process. The mAP for all classes is 98.9%, 'myleft' achieves 98.5%, 'myright' 99.3%, 'yourleft' 98.8%, and 'yourright' 99%. Based on the empirical evidence obtained from the two datasets, it was found that the performance of the model reached its maximum level after 100 epochs. This observation suggests that the number of epochs has a substantial impact on the outcome of the training process. As the duration of the period increases, there is a corresponding increase in performance level; however, this improvement comes at the cost of longer processing time. The training graph of YOLOv8 for 100 epochs is depicted in Fig. 7.

Furthermore, there exists the possibility of obtaining the precision-recall curve, which is retained persistently after each validation process. Figure 8 illustrates the accuracy and recall metrics of YOLOv8n throughout 100 epochs. In this study, we evaluate the efficacy of the Oxford Hand Dataset and EgoHand Dataset through simulations employing the YOLOv8n and YOLOv8s models. We employ specific metrics to quantify the performance of the dataset in these simulations.

DISCUSSION

According to the findings presented in Table 3, an evaluation was conducted on YOLOv8n and YOLOv8s models using 50 and 100 epochs. The results indicate that both models showed comparable performance. The ultimate prediction is a collective representation

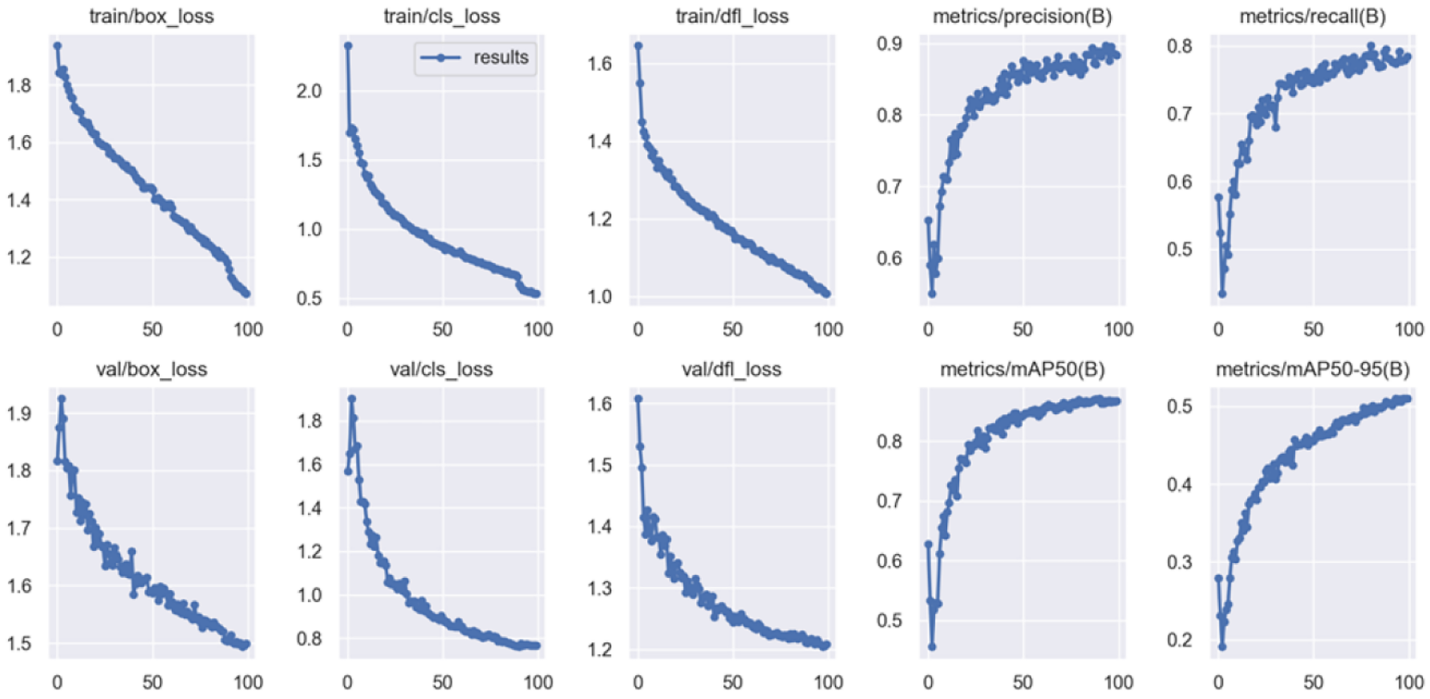


Figure 7 **YOLOv8n training graph with 100 epochs.** Full-size DOI: 10.7717/peerj-cs.3244/fig-7

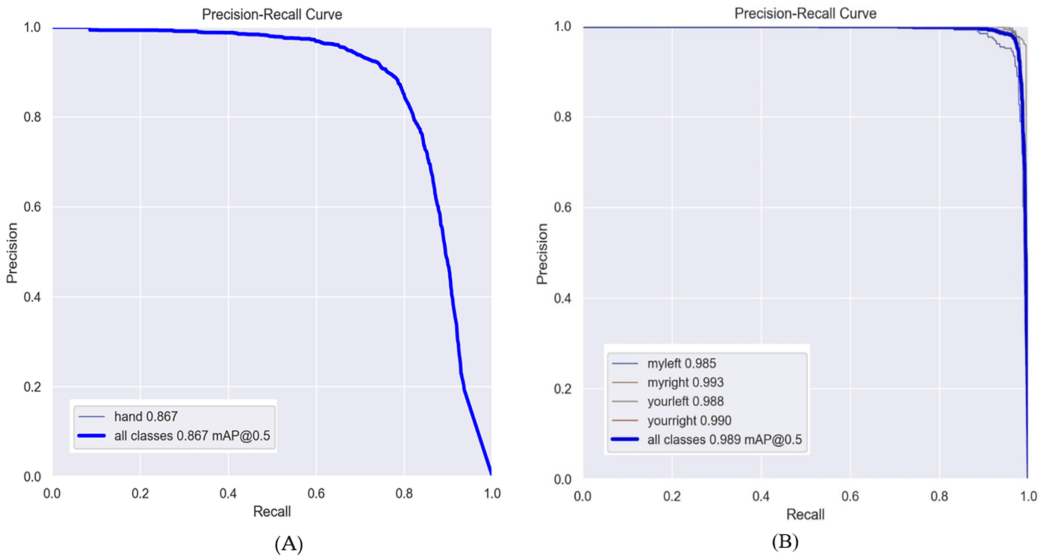


Figure 8 **Precision and Recall curve (A) Oxford Hand Dataset and (B) EgoHand Dataset used YOLOv8n with 100 epochs.** Full-size DOI: 10.7717/peerj-cs.3244/fig-8

comprising all the enhanced iterations of the photos. Test-time augmentations, also known as Test-Time Augmentation (TTA), can be used to enhance the accuracy of predictions by applying them after the inference process.

Table 3 Testing YOLOv8's performance with the Oxford hand dataset.

Epoch	Model	Images	Class	Labels	P	R	mAP@.5
50	Yolov8n	1,223	All	2,898	0.776	0.687	0.758
100	Yolov8n	1,223	All	2,898	0.882	0.784	0.867
50	Yolov8s	1,223	All	2,898	0.762	0.706	0.747
100	Yolov8s	1,223	All	2,898	0.768	0.696	0.75

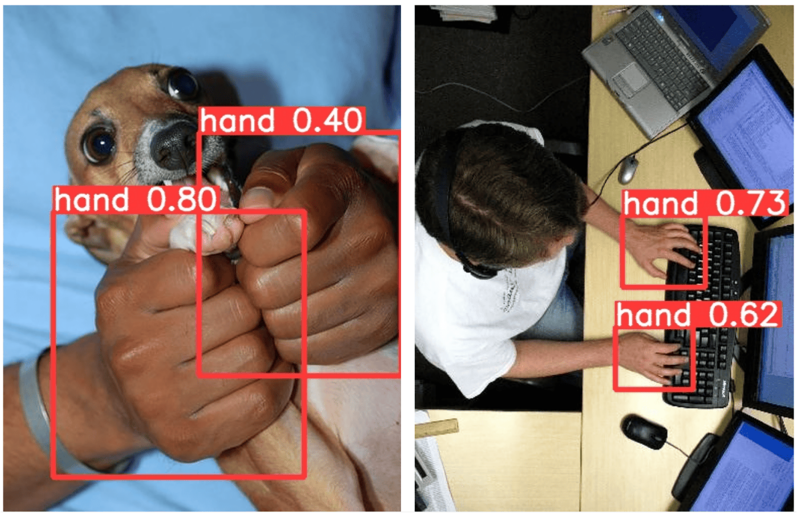


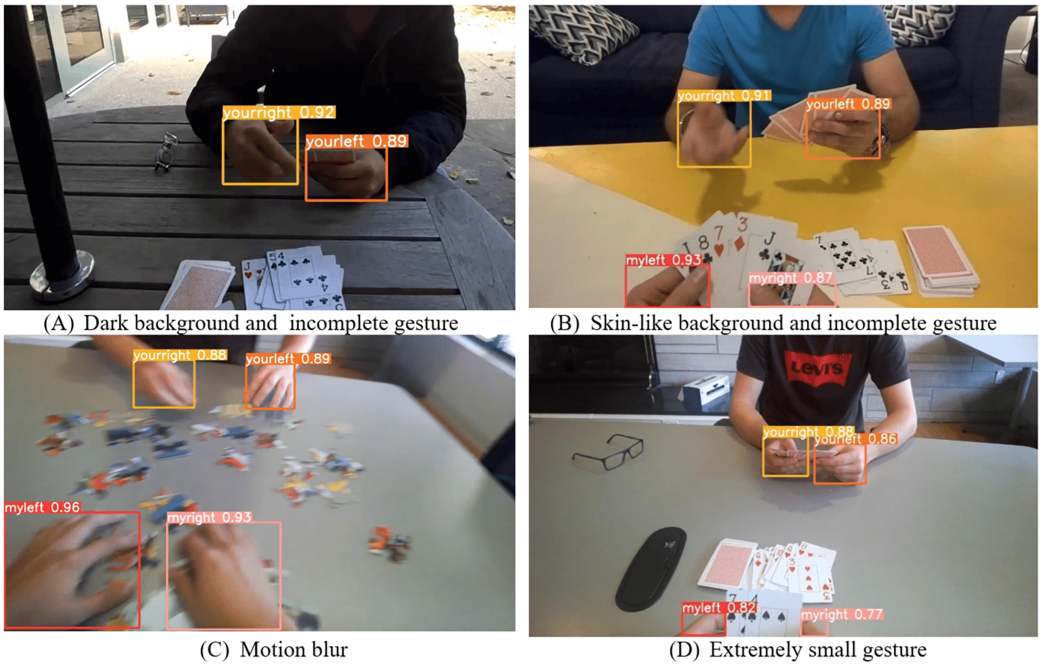
Figure 9 Oxford hand dataset recognition result with YOLOv8n.
Full-size DOI: 10.7717/peerj-cs.3244/fig-9

The performance evaluation of YOLOv8 models involved the use of sample sets of images from each category. Based on the results obtained from our experimental study, it can be concluded that YOLOv8n and YOLOv8s exhibit superior performance when subjected to a training regimen consisting of a cumulative 100 epochs. The YOLOv8n model proves a precision of 88.2%, a recall of 78.4%, and a mean average precision (mAP) of 86.7%. Subsequently, YOLOv8s exhibited a precision of 76.8%, a recall of 69.6%, and a mean average precision (mAP) of 75%. In the field of deep learning, a set of parameters known as hyperparameters is established before the commencement of formal training. This stays true, even in situations when the rise in model complexity is accompanied by a proportionate growth in validation loss. Even though the model's ability to detect outliers showed only a moderate improvement. Two indices of model complexity are the substantial size of its weight and the abundance of its parameters. The indices show an upward trend in tandem with the increase in model complexity. Consequently, the GPU needs a greater amount of memory (RAM) to accommodate the model during the training process.

The recognition result of the Oxford Hand Dataset with YOLOv8n after 100 epochs is depicted in Fig. 9. The YOLOv8n model successfully detects all hands in Fig. 9, achieving accuracy ranging from 40% to 80%. Table 4 shows the testing YOLOv8 performance with

Table 4 Testing YOLOv8's performance with the EgoHand dataset.

Epoch	Model	Images	Class	Labels	P	R	mAP@.5
100	Yolov8n	480	All	1,515	0.975	0.968	0.989
100	Yolov8n	480	myleft	262	0.949	0.962	0.986
100	Yolov8n	480	myright	384	0.984	0.972	0.994
100	Yolov8n	480	yourleft	455	0.984	0.971	0.989
100	Yolov8n	480	yourright	450	0.985	0.967	0.991



(A) Dark background and incomplete gesture

(B) Skin-like background and incomplete gesture

(C) Motion blur

(D) Extremely small gesture

Figure 10 Recognition result of EgoHand Dataset with YOLOv8n in complex background.
Full-size DOI: 10.7717/peerj-cs.3244/fig-10

the EgoHand Dataset. During the testing stage, YOLOv8n with 100 epochs achieves the average mAP of 98.9% for all classes.

When examining Fig. 10A, it is seen that the backdrops exhibit a darker shade and there are instances of incomplete gestures. However, it is noteworthy that the YOLOv8n algorithms successfully found all gestures with an accurate rate of 91%. In Fig. 10B, where the pictures have skin-like backgrounds and incomplete gestures, the YOLOv8n model detects all classes correctly with an accuracy of 91%, 89%, 93%, and 87%. From looking at Fig. 10C, some movements are obscured by motion and our proposed model can detect them properly. Despite this, our proposed methods accurately recognized the blurred gesture, which had a slightly lower recognition accuracy for 'myleft' class only achieving 67%.

Table 5 presents an overview of YOLOv8 models using the Oxford Hand Dataset and EgoHand Dataset. The YOLOv8n model is included of 168 layers and has a total of

Table 5 A summary of YOLOv8 models.

Epoch	Dataset	Model	Layers	Parameters	Gradient	GFLOPS
50	Oxford hand dataset	YOLOv8n	168	3,005,843	0	8.1
100	Oxford hand dataset	YOLOv8n	168	3,005,843	0	8.1
50	Oxford hand dataset	YOLOv8s	168	11,125,971	0	28.4
100	Oxford hand dataset	YOLOv8s	168	11,125,971	0	28.4
100	EgoHand dataset	YOLOv8n	168	3,006,428	0	8.1

3,005,483 parameters. During training, it achieves a performance of 8.1 GFLOPS. The model is trained for 50 and 100 epochs. On the other hand, YOLOv8s is composed of 168 layers, has 11,125,971 parameters, a gradient of 0, and a computational power of 28.4 GFLOPS. The YOLOv8 model serves as the basis for efficient general GPU processing, prioritizing maximum efficiency.

Hand recognition in real-world situations faces obstacles such as data transfer interruptions, lighting instability, noisy background, and various hand orientations. YOLOv8 provides real-time and high-precision object detection, which ideal for gesture-based recognition as well as in security, healthcare, robotics, and augmented/virtual reality applications. It allows us to take advantages such as (1) accommodates diverse hand positions and sizes, (2) functions effectively across different skin tones and cultural contexts, and (3) manages hand motion blur in real-time video streams. Furthermore, sign language recognition enhances multilingual accessibility.

The benefits of YOLOv8 are abundant, such like: (1) YOLOv8 demonstrates incredible speed, which is one of its main advantages compared to alternative deep learning architectures. According to Ultralytics, the YOLOv8 model shows significant improvements in the area of image segmentation, most notably an impressive throughput of 81 frames per second. This model shows superior performance compared to other models, such as Mask region-based convolutional neural network (R-CNN), which is limited to processing about 6 frames per second. Information processing speed is becoming a critical factor in real-time applications, including autonomous vehicle domains, surveillance systems, and video analytics (*Iriani Sapitri et al., 2023*). (2) Accuracy: YOLOv8 demonstrates excellent precision in object and segment identification within images while maintaining fast processing speed. The updated loss function and advanced model architecture improve accuracy by reducing false positive and false negative occurrences. (3) The cohesive architecture for model training offered by YOLOv8 allows the execution of multiple image segmentation tasks within a single model. The tasks encompass object detection, instance segmentation, and image classification. This adaptability is crucial for applications requiring the execution of several tasks, such as video surveillance and image retrieval systems. Additional instances encompass autonomous vehicles.

The procedure for integrating YOLOv8 hand recognition into practical systems is as follows: (1) Data Acquisition and Preprocessing: collect varied hand gesture datasets in real-world conditions. Implement data augmentation techniques, including alterations in

Table 6 A comparison of the results of previous studies.

Dataset	Author	Method	mAP (%)
Oxford hand dataset	*Mittal, Zisserman & Torr (2011)*	Classify framework and two-stage hypothesize	48.2
Oxford hand dataset	*Roy, Mohanty & Sahay (2017)*	R-CNN and skin	49.1
Oxford hand dataset	*Deng et al. (2018)*	Joint model	58.10
Oxford hand dataset	*Le et al. (2017)*	Multiple scale region-based fully convolutional networks (MS RFCN)	75.1
Oxford hand dataset	*Yang et al. (2019)*	Convolutional neural network (CNN) and MobileNet	83.2
Oxford hand dataset	*Dewi, Chen & Christanto (2023)*	YOLOv7x	86.3
Oxford hand dataset	*Mohammed, Lv & Islam (2019)*	Lightweight CNN hand gesture recognition	72
Oxford hand dataset	Our method	YOLOv8n with 100 epochs	86.7
EgoHand dataset	*Chen & Tian (2023)*	YOLOv5l + ELAN + CBAM	75.6
EgoHand dataset	*Mohammed, Lv & Islam (2019)*	Lightweight CNN based hand gesture recognition	93
EgoHand dataset	*Deng et al. (2018)*	Joint model	77.10
EgoHand dataset	*Roy, Mohanty & Sahay (2017)*	Faster R-CNN	50
EgoHand dataset	*Roy, Mohanty & Sahay (2017)*	R-CNN and skin	92.96
EgoHand dataset	*Roy, Mohanty & Sahay (2017)*	Faster R-CNN and skin	96
EgoHand dataset	Our method	YOLOv8n with 100 epochs	98.9

lighting, simulation of occlusion, and application of motion blur. (2) Model Training and Optimization: refine YOLOv8 for domain-specific gestures in rehabilitation, gaming, and security. Employ quantization and pruning to enhance real-time performance on edge devices. (3) System Deployment and Empirical Testing: integrate the model with software application programming interfaces (API) (Python, TensorFlow, OpenCV).

The comparison to the preceding study is described in Table 6. Our proposed YOLOv8n method with 100 epochs outperforms prior models on the Oxford Hand datasets in terms of mAP, with an accuracy of 86.7%. *Le et al. (2017)* proposed the Multiple Scale Region-based Fully Convolutional Networks (MS RFCN) and showed only 75.1% mAP. Another researcher (*Yang et al., 2019*) implements CNN and MobileNet and achieves 83.2% mAP. Furthermore, *Dewi, Chen & Christanto (2023)*, implement YOLOv7x and achieves 86.3% mAP. Moreover, with the EgoHand dataset, our proposed method achieves the highest performance, 98.9% compared to previous research results. *Roy, Mohanty & Sahay (2017)* with Faster R-CNN and skin only achieved 96%. We were able to boost the overall performance of a recent study on hand detections in research.

CONCLUSIONS

This research contains a synopsis of the YOLOv8 family of object identification algorithms, which includes YOLOv8n and YOLOv8s with 50, 100, and epochs, as well as a brief discussion of each of these variants. Furthermore, the YOLOv8n and YOLOv8s algorithms, each with 50 and 100 epochs, will serve as the primary centers of examination

for the duration of this study. During our exploratory research, we put a wide variety of today's object detectors to the test and analyzed them. Detectors that are created specifically to recognize the Oxford Hand Dataset and EgoHand Dataset are only one example of the kind of detectors that we investigate. After compiling all the findings of our inquiry into a single body of data, the following is the overall conclusion that we have come to begin, the YOLOv8n model, when trained with 100 epochs, consists of 168 layers, and has a total of 3,005,483 parameters. Next, according to the results of the experiment, out of all the models that were evaluated, the one with 100 iterations had the best performance, and the number of iterations influenced the training result. The longer the period, the better the performance; however, the longer it takes to process, the longer the epoch needs to be.

In addition, the YOLOv8n technique that we have suggested with 100 epochs beats previous models that have been applied to the Oxford Hand datasets in terms of mAP, with an accuracy of 86.7% and with EgoHand Dataset with 98.9% mAP. In comparison to the existing approaches, our method demonstrated superior performance. Notwithstanding these gains, numerous limits persist. The experiment focuses solely on YOLOv8, employing two publicly accessible datasets: the Oxford Hand Dataset and the Ego Hand Dataset. The advancement of YOLO models is rapid, exemplified by YOLOv9, YOLOv10, and YOLOv11. In our upcoming research, we are going to investigate the possibility of combining hand detection with real-time video datasets and explainable artificial intelligence (XAI). Further, we plan to couple hand detection in a logic-based framework to make automatic inferences based on the detected scene (*Calimeri et al., 2019*).

ACKNOWLEDGEMENTS

We acknowledge the use of ChatGPT (https://chat.openai.com/, (accessed on 1 September 2023)) and QuillBot (https://quillbot.com/, 1 November 2023) for English correction across all manuscript sections.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This research is supported by the Vice-Rector of Research, Innovation, and Entrepreneurship at Satya Wacana Christian University number 035/RIK-RPM/07/2024. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Vice-Rector of Research, Innovation, and Entrepreneurship at Satya Wacana Christian University: 035/RIK-RPM/07/2024.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Budhi Kristianto conceived and designed the experiments, performed the experiments, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Christine Dewi conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Hindriyanto Dwi Purnomo conceived and designed the experiments, performed the experiments, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.
- Kristoko Dwi Hartomo analyzed the data, performed the computation work, prepared figures and/or tables, and approved the final draft.
- Siti Zaiton Mohd Hashim analyzed the data, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The data is available at GitHub and Zenodo:

- https://github.com/ChristineDewi/YOLOv8-Hand-Detection, https://zenodo.org/records/15653510.

The Oxford Hand Dataset is available at:

- https://www.robots.ox.ac.uk/~vgg/data/hands.

- Dewi, C. (2025). Oxford Hand Dataset [Data set]. Zenodo. https://doi.org/10.5281/zenodo.15653510.

The EgoHand Dataset is available at: http://vision.soic.indiana.edu/projects/egohands.

REFERENCES

Adhikari S, Gangopadhayay TK, Pal S, Akila D, Humayun M, Alfayad M, Jhanjhi NZ. 2023. A novel machine learning-based hand gesture recognition using HCI on IoT assisted cloud platform. *Computer Systems Science and Engineering* **46(2)**:2123–2140 DOI 10.32604/csse.2023.034431.

Alam MM, Islam MT, Rahman SMM. 2022. Unified learning approach for egocentric hand gesture recognition and fingertip detection. *Pattern Recognition* **121(6)**:108200 DOI 10.1016/j.patcog.2021.108200.

Arcos-García Á, Álvarez-García JA, Soria-Morillo LM. 2018. Evaluation of deep neural networks for traffic sign detection systems. *Neurocomputing* **316(6)**:332–344 DOI 10.1016/j.neucom.2018.08.009.

Ashiquzzaman A, Lee H, Kim K, Kim HY, Park J, Kim J. 2020. Compact spatial pyramid pooling deep convolutional neural network based hand gestures decoder. *Applied Sciences* **10(21)**:1–22 DOI 10.3390/app10217898.

Azhar M, Ullah S, Ullah K, Shah H, Namoun A, Rahman KU. 2023. A three-dimensional real-time gait-based age detection system using machine learning. *Computers, Materials & Continua* **75(1)**:165–182 DOI 10.32604/cmc.2023.034605.

Bambach S, Lee S, Crandall DJ, Yu C. 2015. Lending a hand: detecting hands and recognizing activities in complex egocentric interactions. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Piscataway: IEEE, 1949–1957 DOI 10.1109/ICCV.2015.226.

Calimeri F, Cauteruccio F, Cinelli LUCA, Marzullo A, Stamile C, Terracina G, Durand-Dubief F, Sappey-Marinier D. 2019. A logic-based framework leveraging neural networks for studying the evolution of neurological disorders. *Theory and Practice of Logic Programming* **21(1)**:80–124 DOI 10.1017/S1471068419000449.

Chen RC, Dewi C, Huang SW, Caraka RE. 2020. Selecting critical features for data classification based on machine learning methods. *Journal of Big Data* **7(1)**:1–26 DOI 10.1186/s40537-020-00327-4.

Chen R, Tian X. 2023. Gesture detection and recognition based on object detection in complex background. *Applied Sciences* **13(7)**:4480 DOI 10.3390/app13074480.

Dai G, Fan J, Dewi C. 2023. ITF-WPI: image and text-based cross-modal feature fusion model for wolfberry pest recognition. *Computers and Electronics in Agriculture* **212(2)**:108129 DOI 10.1016/j.compag.2023.108129.

Deng X, Zhang Y, Yang S, Tan P, Chang L, Yuan Y, Wang H. 2018. Joint hand detection and rotation estimation using CNN. *IEEE Transactions on Image Processing* **27(4)**:1888–1900 DOI 10.1109/TIP.2017.2779600.

De Souza Vieira A, Ribeiro Filho M, De Salles Soares Neto C. 2021. Production and evaluation of an educational process for human-computer interaction (HCI) courses. *IEEE Transactions on Education* **64(2)**:172–179 DOI 10.1109/TE.2020.3024936.

Dewi C, Chen R-C. 2022. Combination of Resnet and spatial pyramid pooling for musical instrument identification. *Cybernetics and Information Technologies* **22(1)**:104–116 DOI 10.2478/cait-2022-0007.

Dewi C, Chen APS, Christanto HJ. 2023. Deep learning for highly accurate hand recognition based on Yolov7 model. *Big Data and Cognitive Computing* **7(1)**:53 DOI 10.3390/bdcc7010053.

Dewi C, Chen RC, Liu YT, Jiang X, Hartomo KD. 2021. Yolo V4 for advanced traffic sign recognition with synthetic training data generated by various GAN. *IEEE Access* **9**:97228–97242 DOI 10.1109/ACCESS.2021.3094201.

Dewi C, Chen R-C, Yu H, Jiang X. 2023. XAI for image captioning using SHAP. *Journal of Information Science and Engineering* **39(4)**:711–724 DOI 10.6688/JISE.202307_39(4).0001.

Dewi C, Chen R-C, Zhuang Y-C, Christanto HJ. 2022. Yolov5 series algorithm for road marking sign identification. *Big Data and Cognitive Computing* **6(4)**:149 DOI 10.3390/bdcc6040149.

Dewi C, Juli Christanto H. 2022. Combination of deep cross-stage partial network and spatial pyramid pooling for automatic hand detection. *Big Data and Cognitive Computing* **6(3)**:85 DOI 10.3390/bdcc6030085.

Dillon R, Jordan K, Hong J, Ahmad D. 2023. Real-time flying object detection with YOLOv8. ArXiv DOI 10.48550/arXiv.2305.09972.

Dong X, Zhao Z, Wang Y, Zeng T, Wang J, Sui Y. 2022. FMCW radar-based hand gesture recognition using spatiotemporal deformable and context-aware convolutional 5-D feature representation. *IEEE Transactions on Geoscience and Remote Sensing* **60(1)**:1–11 DOI 10.1109/TGRS.2021.3122332.

Farooq U, Mohd Rahim MS, Abid A. 2023. A multi-stack RNN-based neural machine translation model for English to Pakistan sign language translation. *Neural Computing and Applications* **35(18)**:13225–13238 DOI 10.1007/s00521-023-08424-0.

Girondel V, Bonnaud L, Caplier A. 2006. A human body analysis system. *EURASIP Journal on Advances in Signal Processing* **2006(1)**:222 DOI 10.1155/ASP/2006/61927.

Gopikha S, Balamurugan M. 2023. Regularised layerwise weight norm based skin lesion features extraction and classification. *Computer Systems Science and Engineering* **44(3)**:2727–2742 DOI 10.32604/csse.2023.028609.

Guan Y, Aamir M, Hu Z, Abro WA, Rahman Z, Dayo ZA, Akram S. 2021. A region-based efficient network for accurate object detection. *Traitement du Signal* **38(2)**:481–494 DOI 10.18280/ts.380228.

Han K, Zeng X. 2022. Deep learning-based workers safety helmet wearing detection on construction sites using multi-scale features. *IEEE Access* **10(1)**:718–729 DOI 10.1109/ACCESS.2021.3138407.

Iriani Sapitri A, Nurmaini S, Naufal Rachmatullah M, Tutuko B, Darmawahyuni A, Firdaus F, Rini DP, Islami A. 2023. Deep learning-based real-time detection for cardiac objects with fetal ultrasound video. *Informatics in Medicine Unlocked* **36(16)**:101150 DOI 10.1016/j.imu.2022.101150.

Jiang L, Liu H, Zhu H, Zhang G. 2022. Improved YOLO v5 with balanced feature pyramid and attention module for traffic sign detection. *MATEC Web of Conferences* **355(6)**:3023 DOI 10.1051/matecconf/202235503023.

Knights E, Mansfield C, Tonin D, Saada J, Smith FW, Rossit S. 2021. Hand-selective visual regions represent how to grasp 3D tools: brain decoding during real actions. *The Journal of Neuroscience* **41(24)**:5263–5273 DOI 10.1523/JNEUROSCI.0083-21.2021.

Le THN, Quach KG, Zhu C, Duong CN, Luu K, Savvides M. 2017. Robust hand detection and classification in vehicles and in the wild. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Piscataway: IEEE DOI 10.1109/CVPRW.2017.159.

Long JW, Yan ZR, Peng L, Li T. 2021. The geometric attention-aware network for lane detection in complex road scenes. *PLOS ONE* **16(7)**:e0254521 DOI 10.1371/journal.pone.0254521.

Lou H, Duan X, Guo J, Liu H, Gu J, Bi L, Chen H. 2023. DC-YOLOv8: small-size object detection algorithm based on camera sensor. *Electronics* **12(10)**:2323 DOI 10.3390/electronics12102323.

Marrahi Gomez V, Belda-Medina J. 2023. The integration of augmented reality (AR) in education. *Advances in Social Sciences Research Journal* **9(12)**:475–487 DOI 10.14738/assrj.912.13689.

Mengash HA, Alharbi LA, Alotaibi SS, AlMuhaideb S, Nemri N, Alnfiai MM, Marzouk R, Salama AS, Duhayyim MAl. 2023. Deep learning enabled intelligent healthcare management system in smart cities environment. *Computers, Materials & Continua* **74(2)**:4483–4500 DOI 10.32604/cmc.2023.032588.

Meriläinen M. 2023. Young people's engagement with digital gaming cultures—validating and developing the digital gaming relationship theory. *Entertainment Computing* **44**:100538 DOI 10.1016/j.entcom.2022.100538.

Mittal A, Zisserman A, Torr P. 2011. Hand detection using multiple proposals. *Available at https://www.robots.ox.ac.uk/~vgg/publications/2011/Mittal11/*.

Mohammed AAQ, Lv J, Islam MDS. 2019. A deep learning-based end-to-end composite system for hand detection and gesture recognition. *Sensors* **19(23)**:5282 DOI 10.3390/s19235282.

Narasimhaswamy S, Wei Z, Wang Y, Zhang J, Nguyen MH. 2019. Contextual attention for hand detection in the wild. In: *Proceedings of the IEEE International Conference on Computer Vision*. Piscataway: IEEE DOI 10.1109/ICCV.2019.00966.

Núñez JC, Cabido R, Pantrigo JJ, Montemayor AS, Vélez JF. 2018. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition* **76(Supplement C)**:80–94 DOI 10.1016/j.patcog.2017.10.033.

Rapp A, Curti L, Boldi A. 2021. The human side of human-chatbot interaction: a systematic literature review of ten years of research on text-based chatbots. *International Journal of Human-Computer Studies* **151(3)**:102630 DOI 10.1016/j.ijhcs.2021.102630.

Redmon J, Divvala S, Girshick R, Farhadi A. 2016. You only look once: unified, real-time object detection. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Piscataway: IEEE, 779–788 DOI 10.1109/CVPR.2016.91.

Rossoshansky G. 2023. What is SEO: the ultimate guide, roboflow. *Available at https://www.researchgate.net/publication/371138226_What_is_SEO_The_Ultimate_Guide*.

Roy K, Mohanty A, Sahay RR. 2017. Deep learning based hand detection in cluttered environment using skin segmentation. In: *Proceedings—2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*. Piscataway: IEEE DOI 10.1109/ICCVW.2017.81.

Shin J, Matsuoka A, Hasan MAM, Srizon AY. 2021. American sign language alphabet recognition by extracting feature from hand pose estimation. *Sensors* **21(17)**:5856 DOI 10.3390/s21175856.

Sigal L, Sclaroff S, Athitsos V. 2004. Skin color-based video segmentation under time-varying illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26(7)**:862–877 DOI 10.1109/TPAMI.2004.35.

Tran NC, Wang JH, Vu TH, Tai TC, Wang JC. 2023. Anti-aliasing convolution neural network of finger vein recognition for virtual reality (VR) human-robot equipment of metaverse. *The Journal of Supercomputing* **79(3)**:2767–2782 DOI 10.1007/s11227-022-04680-4.

Ultralytics. 2022. YOLOv8 (WWW Document). *Available at https://github.com/ultralytics/ultralytics?ref=blog.roboflow.com* (accessed 5 December 2023).

Xia Z, Xu F. 2022. Time-space dimension reduction of millimeter-wave radar point-clouds for smart-home hand-gesture recognition. *IEEE Sensors Journal* **22(5)**:4425–4437 DOI 10.1109/JSEN.2022.3145844.

Xu C, Cai W, Li Y, Zhou J, Wei L. 2020. Accurate hand detection from single-color images by reconstructing hand appearances. *Sensors* **20(1)**:192 DOI 10.3390/s20010192.

Yang L, Qi Z, Liu Z, Liu H, Ling M, Shi L, Liu X. 2019. An embedded implementation of CNN-based hand detection and orientation estimation algorithm. *Machine Vision and Applications* **30(6)**:1071–1082 DOI 10.1007/s00138-019-01038-4.

Zhao S, Zhang K. 2021. Online predictive connected and automated eco-driving on signalized arterials considering traffic control devices and road geometry constraints under uncertain traffic conditions. *Transportation Research Part B: Methodological* **145(15)**:80–117 DOI 10.1016/j.trb.2020.12.009.