Hand Pose Detection Using YOLOv8-pose

1st Pushkar Kadam
School of Engineering, Design and Built Environment
Western Sydney University
Locked Bag 1797, Penrith, NSW, 2751, Australia
18745753@student.westernsydney.edu.au

2nd Gu Fang
School of Engineering, Design and Built Environment
Western Sydney University
Locked Bag 1797, Penrith, NSW, 2751, Australia
g.fang@westernsydney.edu.au

3rd Farshid Amirabdollahian
Robotics Research Group
University of Hertfordshire
Hatfield, United Kingdom
f.amirabdollahian2@herts.ac.uk

4th Ju Jia Zou
School of Engineering, Design and Built Environment
Western Sydney University
Locked Bag 1797, Penrith, NSW, 2751, Australia
j.zou@westernsydney.edu.au

5th Patrick Holthaus
Robotics Research Group
University of Hertfordshire
Hatfield, United Kingdom
p.holthaus@herts.ac.uk

***Abstract*—Hand detection and pose estimation are prominent problems in computer vision. They have applications in augmented and virtual reality, human-robot interaction, and gesture recognition which can be incorporated into controlling various interfaces, such as those used in assistive technology. The hand detection problem involves three sub-problems, i.e. hand localisation, hand classification, and pose estimation. Different hand detection methods approach this problem in multiple stages. However, there is a scope to train an end-to-end network that addresses these three problems at once. In this paper, we contribute to hand detection, classification, and pose estimation by first modifying the FreiHAND dataset to ensure both left and right hand images, along with their annotations, are present for training. Then, we train the YOLOv8-pose networks from nano to extra-large sizes to perform a comparative study of the performance of each network. Further, we perform quantitative and qualitative analysis on three public hand datasets that shows the strengths and limitations of YOLOv8-pose networks. Our experiments on training YOLOv8-pose networks from nano to extra-large sizes showed that the mean average precision score increases with the network size. We also conclude that the ratio of hand size to the image size in training affects the confidence score and classification during inference detection.**

***Index Terms*—YOLO, hand detection, hand pose estimation, deep learning**

I. Introduction

Human hand detection is finding its way into several mainstream applications such as virtual/augmented reality (VR/AR) [1]–[4] and robotics [5], [6]. The hand detection problem can be further classified into palm detection, hand classification, and hand pose estimation. Palm detection involves detecting the palm with a bounding box. Hand classification involves detecting whether the hand is the left or right hand of the person. Pose detection involves identifying the location of the keypoints on different locations of the fingers. The pose estimation process can be further expanded to detect the keypoints in 3D space using depth-based cameras. In this paper, we provide methods for addressing the hand pose detection problem by using a public hand dataset and YOLO (You Only Look Once) [7] Convolutional Neural Network (CNN) approach. Using the YOLO approach has a significant impact on real-time applications in fields of sign language interpretation [8], AR and VR, and robot learning from demonstration [9].

Hand detection is one of the problems in the area of object detection. Considering hand detection, hand classification, and pose estimation problems, most current research approaches treat it as a two-stage process [10]–[12]. Furthermore, due to the symmetry of the left and right hand, first, a human pose estimation is used to locate and classify hands, followed by flipping the left hand to treat it as a right hand for pose estimation [11]. Since the advent of ImageNet [13], deep learning using CNN has shown prominence in object classification. Furthermore, methods such as YOLO [7], SSD (Single Shot Detection) [14], and R-CNN (Regions with CNN features) [15] allow the localisation and classification of the objects. YOLO-pose [16] showed that given the bounding box and keypoint information, a single-stage end-to-end network can be trained that performs both human detection and pose estimation. In this paper, we build on the hypothesis that training an end-to-end network can be useful in the hand pose detection goal by providing classification, bounding box, and keypoint annotations for hands.

YOLO has been known for its end-to-end network with real-time performance [17]. With YOLOv8 [18], hand detection, hand classification, and pose estimation can be performed in a single stage with an end-to-end network. It is convenient to train a deep network with the availability of FreiHAND [19], PASCAL VOC [20], and Panoptic [21] datasets. To approach this problem, we extract and augment the annotations and images from FreiHAND [19] data to YOLO format. Then, we set up training on the different sizes of YOLOv8-pose networks and performed a comparative study of the trained network on different hand datasets. Therefore, building upon the hypothesis of solving hand pose detection problem by employing an end-to-end deep network, the following are our main contributions:

1) FreiHAND dataset [19] modifications to include bounding box location and augment the data to include left

hand images and annotations.
2) Training different sizes of YOLOv8-pose networks and performing a comparative study on the network performance.
3) Identifying the strengths and limitations of the end-to-end approach using quantitative and qualitative analysis by providing hand box ratio as an evaluation metric.

II. Related Works

The classical approaches use different image processing techniques for hand detection. A template image is scanned over the target image to localize the object in the target image frame in the template matching method [22]. The template matching process can be used to detect the fingertips using an infrared camera that detects the hand temperature [1]. A skeleton model of the hand was developed using polygonal approximation of binary images by Mestetkiy et al. [23]. Gil et al. [5] used RGB-D, a depth-based camera, to detect a human hand by identifying human skin region and by using a descriptor classifier that detects the hand from the point cloud constructed by RGB-D image. Shin et al. [24] also used an RGB-D camera to detect the human hand by first detecting the human pose and then isolating the hand regions. Tran et al. [25] also used an RGB-D camera and segmented the hand region by using contouring operation and identified fingertips using K-cosine corner detection [26]. Jirak et al. [27] also used a similar approach to detecting fingertips by corner detection for gesture recognition. Golash et al. [28] used skin segmentation by colour space transformation from RGB to CIE and template matching using SIFT [29] algorithm for hand detection. The classical approach of hand detection relies on transforming colour space, contouring operations, blob detection, and corner detection. However, the classical approaches do not address the problems of hand pose estimation and use various assumptions or human body pose detection for hand classification.

The deep learning approaches for detecting hands and pose estimation involve using a large annotated dataset and training a classifier. Since hand detection can be treated as an object detection problem, deep learning methods such as SSD [14], R-CNN [15], and YOLO [7] can be used. Furthermore, hand gesture recognition involves tracking the movement of the hand and therefore, different tracking methods [30] can be used. While a hand detection can be treated as an object detection problem, a deep network can be trained to localise the hand in the image.

Different research methods [11], [12] perform the pose detection process in two stages, where they first detect the palm using a deep network and then perform pose estimation with another network. Yadav et al. [8] treated hand detection as a segmentation problem. Recent developments in pose estimation using YOLO-pose [16] demonstrated that detection and pose estimation can be performed using end-to-end training in one stage. A deep network for detection can be trained using public datasets such as PASCAL [20], FreiHAND [19], and Panoptic [21]. However, the annotations have problems, such as the missing hand classification and only right hand annotations.

III. Dataset

FreiHAND [19] dataset contains 32560 unique training images of the right hand that are captured in front of a green screen and 3960 images for evaluation. Fig. 1 shows the same hand image, where Fig. 1a is the original image of the hand captured with a green screen background, and Fig.1b and Fig. 1c contain augmented backgrounds with different colour intensities. Since the hand image remains the same, the annotations can be repeated four times over each set of four background-augmented images. There are 130240 images in the FreiHAND dataset. Since this set of images only contains right hands, mirroring these images to create left hands would double both the training and evaluation dataset to 260480 and 7920, respectively.

(a) 00000000.jpg (b) 00032560.jpg (c) 00065120.jpg

Fig. 1: Image of the hand behind different background and illumination conditions in the FreiHAND dataset (Sub-figure captions indicate the image name in the dataset)

IV. Methodology

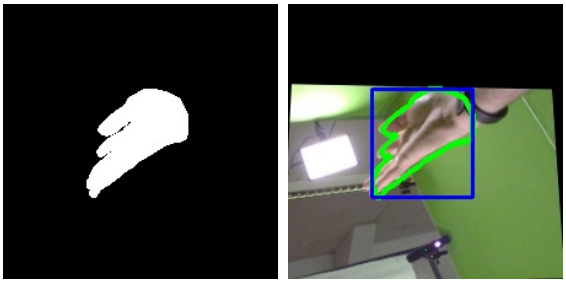
We use the YOLO [7] approach to solve hand detection, classification, and pose estimation problems. We modify the FreiHAND [19] dataset to include left hand images and annotations because the hand classification and pose estimation problem requires both the left and right hand datasets. The data augmentation helps to generate additional data for the left hand, which is essential for classification and pose estimation. Then, we train the YOLOv8-pose network of different sizes, that evaluates the model performance and provide strengths and limitations of the end-to-end learning approach. The following sub-sections outline the implementation details.

A. Bounding box generation

FreiHAND [19] dataset needs to be converted into YOLO-pose [16] format. For the YOLOv8-pose [18], every hand annotation must be as mentioned in (1).

$$L = \{c_l, x, y, w, h, kx_1, ky_1, \ldots, kx_{21}, ky_{21}\} \quad (1)$$

where, c_l denotes the hand class (right hand: 0, left hand: 1), (x, y, w, h) provide the bounding box dimensions with box center coordinates (x, y) and the box width and height (w, h). (kx_i, ky_i) are the i-th hand keypoint coordinates. The 21 keypoints for hand detection were developed for pose estimation and incorporated in different datasets [11], [19].



(a) Segmentation mask (b) Contour around hand

Fig. 2: Bounding box generation

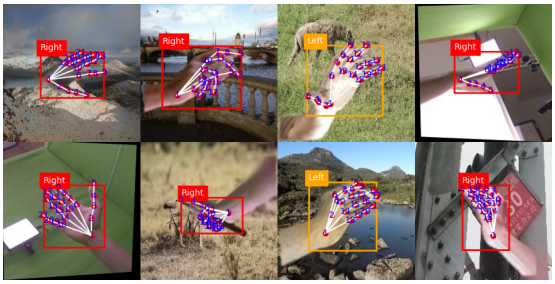


Fig. 3: Rendered images with class, bounding box, and keypoint annotations

FreiHAND dataset does not have bounding box information; however, the dataset comes with a hand segmentation mask, as shown in Fig. 2a. Therefore, we use an external contouring image processing operation to generate a contour that provides the information of maximum and minimum coordinates for the hand bounding box location. The contour over the hand and the bounding box obtained using the minimum and maximum coordinates are shown in Fig. 2b.

B. Hand keypoints extraction

The camera coordinates for the 21 keypoints, along with the camera intrinsic matrix, are provided in the FreiHAND dataset [19]. The homogenous image plane coordinates are derived using (2).

$$\tilde{\boldsymbol{u}} = M_{int}\tilde{\boldsymbol{x_c}} \tag{2}$$

where M_{int} is the camera intrinsic matrix, $\tilde{\boldsymbol{x_c}}(\tilde{x_c}, \tilde{y_c}, \tilde{z_c})$ are the camera coordinates, and $\tilde{\boldsymbol{u}}(\tilde{u}, \tilde{v}, \tilde{w})$ are the homogenous image plane coordinates. The image plane coordinates are obtained by $u = \tilde{u}/\tilde{w}$ and $v = \tilde{v}/\tilde{w}$.

C. Mirroring the coordinates

Since the dataset only consists of right hand images, a mirroring technique can flip the image and annotations along the vertical axis and translate them along the horizontal axis. The mirroring annotation operation is as shown in (3).

$$\begin{bmatrix} x_m \\ y_m \\ w_m \\ h_m \\ kx_{1m} \\ ky_{1m} \\ \vdots \\ kx_{21m} \\ ky_{21m} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & \dots & 0 & p \\ 0 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & p \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & p \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \\ h \\ kx_1 \\ ky_1 \\ \vdots \\ kx_{21} \\ ky_{21} \\ 1 \end{bmatrix} \tag{3}$$

where the subscript m represents the mirrored values of the coordinates. $p = 1$ helps in translation after mirroring along the vertical axis for normalised annotations. Fig. 3 shows the rendered images of different classes of hands along with bounding box and keypoints.

D. Training using YOLOv8-pose

We select the end-to-end learning approach using YOLOv8-pose and the software development kit (SDK) provided by Ultralytics [18]. Terven et al. [17] provided a detailed architecture of YOLOv8. The weights used for training pose are pre-trained on MS COCO dataset [31]. The training was performed by using the default parameters provided by Ultralytics [18] for 100 epochs on nano (YOLOv8n-pose), small (YOLOv8s-pose), medium (YOLOv8m-pose), large (YOLOv8l-pose), and extra-large (YOLOv8x-pose). The letters {n,s,m,l,x} in the YOLOv8 naming convention represent nano, small, medium, large, and extra-large, respectively. In this paper, we use set notation to collectively represent all the models as YOLOv8{n,s,m,l,x}-pose. The batch size of 16 was used for YOLOv8{n,s,m,l}-pose network model. The batch size is 9 for the YOLOv8x-pose network, which was auto-selected using the Ultralytics SDK since the computational resources could not handle a large batch size on the extra-large network. For training, the default loss functions, CIoU [32] and DFL [33], Binary cross entropy (BCE), and object keypoint similarity (OKS) [16] were used for bounding box, classification, and keypoints respectively.

The training setup was run on a computer with Ubuntu 22.04 OS, 13th Gen Intel i9 CPU, 24 GB NVIDIA GPU, and 64 GB of RAM.

V. Results

A. Training results

The training is run on nano, small, medium, large, and extra-large models. Fig. 4 shows the YOLOv8{n,s,m,l,x}-pose model training and validation loss graphs with box, pose, class, and DFL loss for 100 epochs.

B. Comparative study

A comparative study is performed by comparing the precision, recall, F_1, mAP50, and mAP50-90 (Mean Average Precision) scores. The evaluation metric is similar to that of MS COCO [31]. Table I shows the results of the evaluation metrics for bounding box detection and pose estimation. The results, evaluated on different sizes of YOLOv8-pose models, show that, given the same dataset as the input, the evaluation metrics do not significantly change. The only major difference

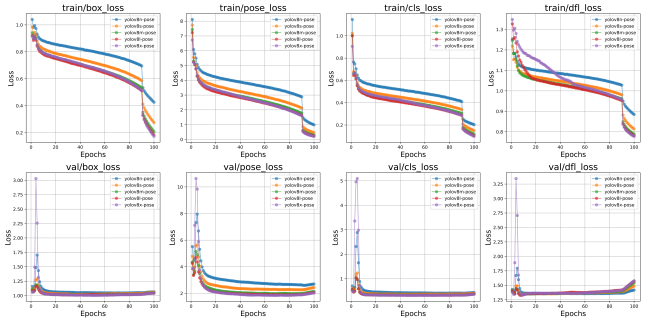


Fig. 4: Box, pose, class, and DFL loss graphs for training and validation of the modified FreiHAND [19] dataset for YOLOv8{n,s,m,l,x}-pose for 100 epochs

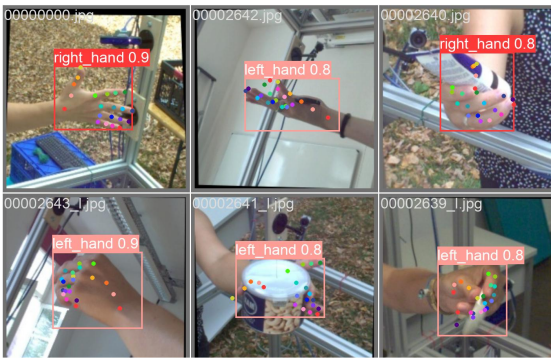


Fig. 5: Sample prediction from YOLOv8n-pose

observed was in the precision and recall values for the pose from YOLOv8n-pose to other models.

Table II outlines different aspects of the trained model based on the training time, network size, layers, number of parameters, and GFLOPs. Comparing Table I and Table II, it is observed that as the network size gets larger, the precision and recall are improved.

C. Qualitative Results

1) FreiHAND dataset evaluation: Fig. 5 shows the rendering of the samples from the evaluation dataset. The network is able to localise, classify and estimate the pose of the hand. Since the training and evaluation were performed on the FreiHAND [19] dataset, it is essential to test on the Panoptic [21] and PASCAL VOC [20] datasets.

2) Panoptic and PASCAL VOC evaluation: The ground truth for the Panoptic [21] dataset is not available in the YOLO format for hand pose detection. Therefore, a qualitative evaluation is performed. Upon evaluating the trained network on an arbitrarily selected Panoptic data image in Fig. 6a (Image name: 00000000.jpg), there was no successful detection. To understand the ratio of the hand size to that of the image size, the hand box ratio r is calculated as shown in (4).

$$r = \frac{a}{A} = \frac{w \times h}{W \times H} \tag{4}$$

where a and A are the area of the box and image, respectively, and (w, h) and (W, H) are the width and height of the box and image, respectively. The mean hand box ratio (r) for training and evaluation FreiHAND dataset [19] is 0.1647 and 0.1681, respectively. This means that the hand size should be up to approximately 16% of the image for the detection. The hand box ratio, $r \approx 4.74 \times 10^{-3}$, is less than 1% of the image size for Fig. 6a.

The training image consists of a single hand and occupies, on average, 16% of the image space as shown in Fig. 3. Detection can be accomplished by cropping the image such that the hand box ratio is $0.01 < r < 0.2$. Fig. 6 shows the result of detection over different networks. The size of the image (I) in Fig. 6a is 1080×1920. Fig. 6b to Fig. 6f are manually cropped as $I_{crop} = I[250 : 700, 450 : 1200]$. The cropping process results in the image size of 450×750, and it is further reduced by 30% to the size of 315×525 before the detection process in order to approximately match the training image size.

Fig. 6a shows that there was no detection in the uncropped original image. Detections are observed with the cropped images where the hand box ratio is $r \approx 0.05$. Except for YOLOv8l-pose detection in Fig. 6e where a false positive of right hand appears along with the left hand detection, all other YOLOv8{n,s,m,x}-pose models were able to successfully detect and classify the hands with varying confidence scores.

The detection confidence is higher in individual hands when the image is further cropped to only include individual hands. Fig. 6a (I) is cropped such that $I_a = I[300 : 600, 500 : 900]$ to obtain Fig. 7a and $I_b = I[300 : 600, 800 : 1200]$ to obtain Fig. 7b, each of size 300×400. These cropped images (I_a, I_b) are reduced by 30% to the size of 210×280 for detection to approximately match the training image size. The hand box ratio for these cropped images is $r \approx 0.12$. A significant difference is observed by comparing single and multiple hand detection with the detection confidence score in Fig. 7b, which is 0.90, with 0.68 in Fig. 6d.

A similar assessment can be made on the PASCAL VOC [20] dataset. Fig. 8 shows the detection implemented on two arbitrarily selected sample images from the VOC dataset. Fig. 8a is of the size 500×375 with hand box ratio $r \approx 0.04$ and Fig. 8b is of the size 375×500 with the hand box ratio $r \approx 0.05$.

VI. Discussion

To train the YOLOv8-pose for hand detection, we used the FreiHAND [19] dataset. While the FreiHAND dataset provides images of hands in different illumination settings, it does not provide annotation data on different scales. We developed a method for generating the left hand images from the given dataset by augmenting the images and annotations. We provide a transformation matrix in (3) that helps to generate the left hand annotation from the given right hand annotation by mirroring along the vertical axis. These augmented images and annotations fit the YOLO format as shown in (1) for training

TABLE I: Performance comparison of the YOLOv8-pose models on Box and Pose metrics

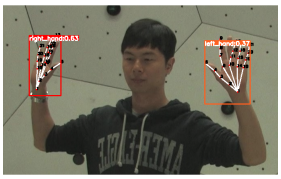
Model	Box					Pose				
	Precision	***Recall***	***mAP-50***	***mAP50-95***	F_1 ***score***	***Precision***	***Recall***	***mAP-50***	***mAP50-95***	F_1 ***score***
YOLOv8n-pose	0.954	0.954	0.985	0.726	0.954	0.88	0.831	0.872	0.481	0.854
YOLOv8s-pose	0.967	0.967	0.989	0.736	0.967	0.91	0.876	0.909	0.537	0.8926
YOLOv8m-pose	0.98	0.979	0.993	0.744	0.9795	0.929	0.908	0.935	0.583	0.9183
YOLOv8l-pose	0.983	0.983	0.993	0.745	0.983	0.937	0.92	0.944	0.594	0.9284
YOLOv8x-pose	0.984	0.983	0.992	0.745	0.9835	0.937	0.923	0.945	0.597	0.9299

TABLE II: Training comparison of YOLOv8-pose models

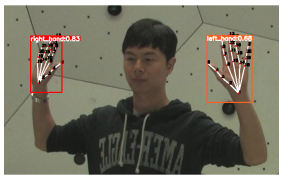
Model	Training time (Hours)	Network size (MB)	Layers	Parameters	GFLOPs
YOLOv8n-pose	14.646	6.7	187	3,228,680	8.9
YOLOv8s-pose	23.188	23.3	187	11,518,344	29.8
YOLOv8m-pose	44.339	53.2	237	26,407,252	80.9
YOLOv8l-pose	67.087	89.4	287	44,464,596	168.6
YOLOv8x-pose	107.809	139.4	287	69,460,980	263.2

(a) Uncropped original image: 00000000.jpg

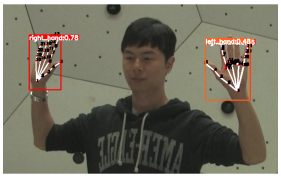
(b) YOLOv8n-pose



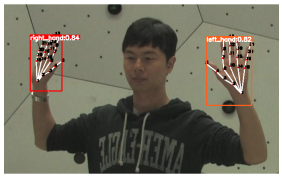
(c) YOLOv8s-pose



(d) YOLOv8m-pose



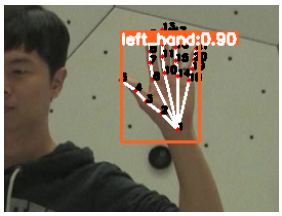
(e) YOLOv8l-pose



(f) YOLOv8x-pose

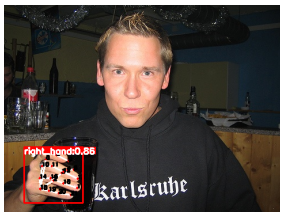
Fig. 6: Detection on Panoptic image

(a) Cropped image focusing on right hand



(b) Cropped image focusing on left hand

Fig. 7: Detection on Panoptic image with individual hands using YOLOv8m-pose



(a) Image: 2007_000170.jpg

(b) Image: 2007_000999.jpg

Fig. 8: Detection on VOC images using YOLOv8m-pose

a network using Ultralytics SDK [18] to detect, classify and estimate hand pose.

Table II shows that the mAP score for all the trained end-to-end networks at 0.5 IOU threshold is above 98%. The best-recorded score using a two-stage approach [10] is the average precision of 95.7% evaluated on their in-house dataset. Training all the available models on the modified dataset shows that the mAP score for the bounding box is almost equivalent in all the networks, except for the mAP score for Pose, which varies from the size of the network as shown in Table II.

We performed tests by arbitrarily selecting images from the Panoptic [21] and PASCAL VOC [20] datasets. The common failure detection cases consist of images where the hand size is too small relative to the image size. To address this error in detection, we measure the appropriate hand size by providing a hand box ratio (r) in (4). Since the average hand box ratio for the FreiHAND [19] dataset was 0.16, we found the value of the hand box ratio as $0.01 < r < 0.2$. The hand box ratio is crucial as it indicates the limitation in the training dataset, thus providing future scopes to develop a dataset that incorporates a wide range of hand box ratios. The detection confidence score is observed to be higher when a single hand is present with a higher hand box ratio as opposed to multiple hands present, as shown by comparing Fig. 6 and Fig. 7.

VII. Conclusion

In this paper, we showed that hand detection, classification of left and right hand, and 21 hand keypoints localisation can be performed in a single stage using trained YOLOv8-

pose models. Our contributions include providing methods to modify the FreiHAND dataset by augmenting images and annotations in YOLO format, training YOLOv8{n,s,m,l,x}-pose networks and providing network comparative analysis. We also provide hand box ratio during qualitative evaluation as a significant measure that will be important to consider while developing new hand detection datasets.

Hand pose detection has applications in human-robot or human-computer interaction, gesture recognition, virtual and augmented reality, and smartphone applications. The broad scope of applications requires further research in this area. The future scope for hand pose detection could focus on developing a hand pose dataset that includes a wide range of hand box ratios and 3D keypoint annotations using depth camera images. Further evaluation can be performed on the trained networks on different computing platforms, such as smartphone devices and virtual reality headsets.

REFERENCES

[1] Y. Sato, Y. Kobayashi, and H. Koike, "Fast tracking of hands and fingertips in infrared images for augmented desk interface," *Proceedings - 4th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2000*, pp. 462–467, 2000.

[2] K. Oka, Y. Sato, and H. Koike, "Real-time fingertip tracking and gesture recognition," *IEEE computer graphics and applications*, vol. 22, no. 6, pp. 64–71, 2002.

[3] M. S. Alam, K.-C. Kwon, M. A. Alam, M. Y. Abbass, S. M. Imtiaz, and N. Kim, "Trajectory-based air-writing recognition using deep neural network and depth sensor," *Sensors (Basel, Switzerland)*, vol. 20, no. 2, pp. 376–, 2020.

[4] J. Shin and C. M. Kim, "Non-touch character input system based on hand tapping gestures using kinect sensor," *IEEE Access*, vol. 5, pp. 10 496–10 505, 2017.

[5] P. Gil, C. Mateo, and F. Torres, "3d visual sensing of the human hand for the remote operation of a robotic hand," *International journal of advanced robotic systems*, vol. 11, no. 2, pp. 26–, 2014.

[6] S. Sreenath, D. I. Daniels, A. S. D. Ganesh, Y. S. Kuruganti, and R. G. Chittawadigi, "Monocular tracking of human hand on a smart phone camera using mediapipe and its application in robotics," in *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)*. IEEE, 2021, pp. 1–6.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[8] K. S. Yadav, A. M. Kirupakaran, and R. H. Laskar, "Gcr-net: A deep learning-based bare hand detection and gesticulated character recognition system for human-computer interaction," *Concurrency and Computation: Practice and Experience*, vol. 35, pp. 1–1, 3 2023.

[9] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, 5 2009.

[10] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," 6 2020.

[11] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," *arXiv.org*, 2017.

[12] S. An, X. Zhang, D. Wei, H. Zhu, J. Yang, and K. A. Tsintotas, "Fasthand: Fast monocular hand pose estimation on embedded systems," *Journal of Systems Architecture*, vol. 122, p. 102361, 1 2022.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84–90, 5 2017.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[16] D. Maji, S. Nagori, M. Mathew, and D. Poddar, "Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2022-June, pp. 2636–2645, 4 2022.

[17] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction 2023, Vol. 5, Pages 1680-1716*, vol. 5, pp. 1680–1716, 11 2023.

[18] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[19] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. J. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 813–822, 10 2019.

[20] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 6 2010.

[21] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social motion capture," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3334–3342.

[22] R. Brunelli and T. Poggio, "Template matching: matched spatial filters and beyond," *Pattern Recognition*, vol. 30, pp. 751–768, 5 1997.

[23] L. Mestetskiy, I. Bakina, and A. Kurakin, "Hand geometry analysis by continuous skeletons," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 130–139.

[24] J. Shin and C. M. Kim, "Non-touch character input system based on hand tapping gestures using kinect sensor," *IEEE Access*, vol. 5, pp. 10 496–10 505, 2017.

[25] D.-S. Tran, N.-H. Ho, H.-J. Yang, S.-H. Kim, and G. S. Lee, "Real-time virtual mouse system using rgb-d images and fingertip detection," *Multimedia tools and applications*, vol. 80, no. 7, pp. 10 473–10 490, 2021.

[26] M. K. Khan, S. Sarker, S. M. Ahmed, and M. H. Khan, "K-cosine-means clustering algorithm," *Proceedings of International Conference on Electronics, Communications and Information Technology, ICECIT 2021*, 2021.

[27] D. Jirak, D. Biertimpel, M. Kerzel, and S. Wermter, "Solving visual object ambiguities when pointing: an unsupervised learning approach," *Neural Computing and Applications*, vol. 33, pp. 2297–2319, 4 2021.

[28] R. Golash and Y. K. Jain, "Real-time robust and cost-efficient hand tracking in colored video using simple camera," in *Social Networking and Computational Intelligence*, ser. Lecture Notes in Networks and Systems. Singapore: Springer Singapore, 2020, pp. 565–573.

[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 11 2004.

[30] P. Kadam, G. Fang, and J. J. Zou, "Object tracking using computer vision: A review," *Computers 2024, Vol. 13, Page 136*, vol. 13, p. 136, 5 2024. [Online]. Available: https://doi.org/10.3390/COMPUTERS13060136

[31] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, pp. 740–755, 2014.

[32] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12 993–13 000, 4 2020.

[33] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.