A pilot study on hand posture recognition from wrist-worn camera for human machine interaction

Thanh-Hai Tran*, Hoang-Nhat Tran†, Hong-Quan Nguyen ‡, Trung-Hieu Le †, Van Thang Nguyen*, Trung Kien Tran¶, Cuong Pham‖, Thi-Lan Le*, Hai Vu*, Thanh-Phuong Nguyen**, Huu Thanh Nguyen*

* School of Electronics and Telecommunications, HUST, Hanoi, Vietnam
† MICA, HUST, Hanoi, Vietnam; ‡ Viet-Hung Industrial University, Hanoi, Vietnam; § Dainam University, Hanoi, Vietnam.
¶ Military Institute of Information Technology, Hanoi, Vietnam
‖ Posts and Telecommunications Institute of Technology, Hanoi, Vietnam; **Toulon University, France

***Abstract*—Hand gestures have been shown to be an efficient way for human-machine interaction. Existing approaches usually utilize ambient or head/chest-mounted cameras to capture hand images. This paper presents a new way to capture hand gestures using the wrist-worn camera. The wrist-worn device is designed as a watch with an integrated camera that is much easier and comfortable to wear in daily life context. We then collect a dataset of ten hand postures using the designed prototype by ten subjects. In addition, we deploy state-of-the-art lite CNN models (YOLO family, Single Shot Detector-SSD) as posture detectors and classifiers. Experimental results show that with limited camera angles, the postures are highly distinctive and easily discriminated with the highest performance of 98.85% and 97.40% in terms of precision and recall, which motivates a wide range of applications and new research directions for human-machine interaction, wearables, the Internet of Things (IoT) and so on.**

***Index Terms*—wrist-worn camera, posture recognition, deep learning, human-machine interaction**

I. Introduction

Hand gestures are becoming an intuitive and efficient way for human machine interaction. To this end, machines must be able to understand hand gestures that human performs. A whole hand gesture understanding system generally consists of a sensor that captures human hand gestures, which are recognized by a machine learning algorithm before being converted to commands for controlling devices/machines. Many existing works on hand gestures recognition have been conducted for more than three decades. Most of them utilized visual [1] or physical sensors [2] for capture hands. Visual sensors gain more attention than others because they produce rich information of not only human hand but also objects in interaction and background context.

Visual sensors are usually mounted in the environment (third-person view) or on some body parts of human (first-person view - egocentric vision). Ego-cams (egocentric cameras) is more efficient than ambient cameras as they solely focus on necessary data. Most of the existing egocams are mounted on forehead or chest. Such installation is suitable for specific applications such as rehabilitation evaluation of patients, or performance evaluation of sport players. However, it causes uncomfortability for the wielder. Some commercial products such as Google Glass show its great usability but it is unable to integrate other physical sensors in glass-like devices as all-in-one devices to measure human gestures.

This paper presents a new design of a watch-like wearable device. This device is mounted on the wrist of the wearer that can capture both his/her hand and environment. In addition to the camera, we can easily integrate additional physical sensors (e.g. accelerometer) for further study. By doing so, the designed device can be considered as an ordinary object of human beings, which is more comfortable to wear without disturbing the human by appearance or by wearing. However, we will show in this paper that the camera mounted at the wrist has limited field of view of hand due to its narrow perspective. As a consequence, the gesture/posture acquired by such camera will raise new challenges for machine learning algorithms: the camera looks mostly at the back of the hand than fingers. We tackle this challenge for the first time by deploying powerful state-of-the-art deep models YOLO [3]. Various versions of YOLO will be investigated to detect and classify ten hand postures performed by ten subjects in a context of human-machine interaction [4], [5], [6]. SSD - Single Shot Multibox Detector [7] is also implemented to prove an effectiveness of the proposed study. Experiments show that although the viewing angle of the camera is limited, observation of fingers is still good enough to distinguish the postures.

In summary, the contribution of this paper is three-fold. First, we design a new hand-held device that is able to integrate different sensors (camera, accelerometer, etc.) for capturing images or movement of hand gestures. Second, we collect a new dataset of ten human postures using wrist-camera. This dataset will be released for research purposes. Finally, we evaluate the performance of lastest object detector on our own dataset - YOLO object detector family. In the remainder of this paper, we will present related works in section II. In section III, we describe our designed prototype and data collection. YOLO models will be re-visited in section IV. Experiments and conclusions are presented in section V and VI respectively.

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-20-1-4053.

II. Related works

Hand gesture recognition from visual sensors has been widely studied in the literature. Many methods have been proposed for static hand posture and dynamic hand gestures from ambient or wearable camera [8], [9], [10]. However, the topic of recognition from wrist-worn cameras only emerged recently and a limited number of works initiate the design of prototypes and apply existing machine learning techniques for gestures recognition. In this section, we will only survey the most relevant works on the design of a hand-mounted camera and methods for static hand posture recognition.

Park et al. proposed a prototype that embeds a camera in a wirst-worn device [11]. A set of Korean alphabet has been collected. Captured images are segmented using color space. Then extracted hand shape is utilized for recognition. The recognition rates range from 63% to 91% depending on adaptation techniques.

Chen et al. designed an RGB camera embedded in a wrist-worn device for controlling robot arm [12]. Ten hand postures (from 0 to 9) are collected with 10 subjects, leading to 1000 images in the dataset. The authors utilized hand segmentation and template matching techniques for posture recognition obtained 99.38% of accuracy.

In [13], the authors designed a hand device that embeds a Leap Motion imaging sensor. There are 11 static hand poses consisting of 10 numbers in the American Sign Language, plus a relaxed pose. Besides, they collect also 6 dynamic hand gestures. For recognition of hand postures, the authors employed Inception-v1 model. The highest recognition rate is about 89.4% with real-time individual test.

Wu et al. used a wide-angle RGB camera worn on a watch for collecting and estimating hand poses [14]. Sia subjects participated to perform 10 postures (0-9 of ASL) and 5 dynamic hand gestures as the work in [13]. Different recognition methods have been evaluated: nearest neighbor (44.6%), direct regression (71.2%). The method by [13] obtained 88.6% on this dataset while [14] using DosalNet with an additional MLP obtained 91.4%.

Yamoto et al. [15] proposed a hand gesture interaction method using a low-resolution infrared image sensor worn on the inner wrist. They attach the sensor to the strap of a wrist-worn device, on the palmar side, and apply machine-learning techniques to recognize the gestures made by the opposite hand. Five right-handed male volunteers participated in collecting the data of 6 static postures and 7 dynamic hand gestures in an application of map interface. The accuracy obtained on this dataset ranges from 64.69% to 99.61%.

In summary, existing works on recognition of hand postures using wrist-worn camera remain very limited and countable on the fingers. The collected datasets are not published yet. Most methods utilized conventional machine learning techniques such as K-nearest neighbor or regression, template matching. Some of them employed baseline deep models such as Inception which can be quite complex for a real-time application. In this work, we will investigate different lite versions of real-time object detector YOLO as well as one example of SSD (mobilenetv2-SSD) for real-time application of human-machine interaction. In the following, we will present in detail our designed prototype and experimental framework.

III. Design a prototype of wrist-worn camera and data collection

A. Design the prototype

We design a wrist-worn device which composes of smart-watch-like hand band that helps to mount a camera or other sensors. We use a low-cost conventional RGB camera for the purpose of ordinary usage. The camera model is IMX219-160 which gives the highest resolution of 3280×2464 at 15 fps, side field of view of camera is 160^o. At 1280×720 resolution options, the capture rate may reach 90 fps. The device is worn on the backside of the user's right wrist and the camera will capture images of the hand back. We utilize an embedded computer (i.e. Jetson Nano) to receive images from camera through USB port. Figure 1 illustrates our designed prototype. Comparing to the existing surveyed prototypes in Section II, ours gives a possibility to integrate other multimodal sensors such as accelerometer and gyroscope. We also exploit a low-cost camera then the image quality will be more challenging in the step of hand segmentation.

Fig. 1. Illustration of the designed prototype.

B. Data collection and annotation

We conduct our first pilot with a conventional posture set (numbers from 0 to 9 in ASL) as existing works. On the one hand, that gives us some ideas about the performance for comparison. On the other hand, this posture set can be used to identify appliances to be controlled in human-machine interaction. Templates of the postures are presented in Figure 2.

We invite ten volunteers (five men and five women) to participate in data collection in several environments (home, laboratory, a room in dormitory) Each subject is guided to wear the designed device on his/her wrist and to perform the gestures. We ask them to perform 10 pre-defined postures as they want to control the home appliance in their way as naturally as possible (Fig.3). Each subject performs a posture several times. Each time, he/she changes his/her standing location, the orientation of the hand (camera), and the device to be controlled so that we can capture a dataset with variation in background, orientation, and position.

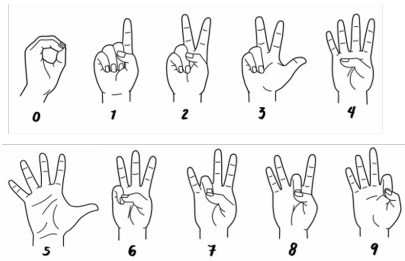


Fig. 2. Set of 10 postures corresponding to 10 numbers (0-9) in American Sign Language.

Fig. 3. llustration of a subject wearing the designed device in home environment for data acquisition.

Images are stored in a wearable embedded device (Jetson Nano). Finally, the collected dataset consists of 762 images of 10 postures with different backgrounds, lighting and orientation of hands. Figure 4 illustrates 10 postures recorded by a subject. Figure 5 shows variation in posture implementation (the number 3) by different subjects under different lighting condition and background.

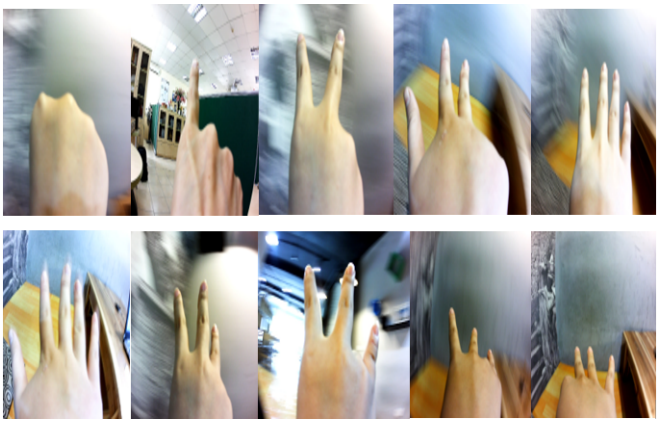


Fig. 4. Examples of 10 hand postures in the collected dataset

Fig. 5. Variation in posture implementation, lighting condition and background.

The dataset is annotated using LabelImg tool. One advantage of a wrist-worn sensor is that the hand is the only object that takes a large portion of the image. On the one side, this mitigates the annotation. On the other side, it could facilitate detection algorithm. However, wrist-worn sensor has drawbacks too. As seen in Figure 5, the camera sees mostly the back of hand palm, hence, it is difficult to observe clearly the fingers, leading to misleading posture recognition. The total time to collect and label all data is approximately 2 weeks. Once labelled, the images are resized to 416416 to fit the grid stride of the architectures (multiples of 32). Dataset can be found online at https://github.com/inspiros/mica_handwrist.

IV. Hand posture detection and classiciation

In this paper, we investigate various versions of YOLO that play the role of detection and classification. Each posture type can be considered as an object class. YOLO has been known to be the best real-time object detector that balances the trade-off between computational time and accuracy. Our objective is to find out the most suitable model in terms of accuracy, computational cost and memory usage. The selected model can be deployed in an embedded computer such as Pi or Jetson Nano for further applications.

A. Proposed framework

As aforementioned, we investigate different lite models of YOLO. Our proposed framework is illustrated in Fig. 6. The framework consists of two phases: training phase and inference phase.

- Training phase: Images of the hand will be captured and transferred to a server for centralized processing. We first annotate image by image using LabelImg tool. Annotated data will be stored posture by posture for training the models.
- Testing phase: Images of the hand are captured and go through the inference phase using the trained models. Posture prediction will be deriveed and IoU (Intersection over Union Index) is computed to determine whether it is a true positive.

B. YOLO re-visiting

YOLO detects objects in images as a single regression problem. The output of YOLO contains bounding box coordinates and class probabilities from image pixels. As one of the cutting-edge detectors, YOLO has many advantages over the others. Introduced in 2016 [3], until now, many modifications of YOLO and its variations have been proposed such as YOLOv2, YOLOv3, YOLOv4, and the latest version YOLOv5 was released in May 2020. To be able to run on non-GPU computers or miniatured devices, YOLOv3 tiny, YOLOv4 tiny, YOLOv5s were subsequently developed. With the same purpose, we will investigate lite versions of YOLO to finally deploy one of the models on a low-powered device.

The basic idea of the original YOLO is to predict the bounding boxes of objects and their class probabilities in one stage. First, the input image is divided into a $S \times S$ grid. Then

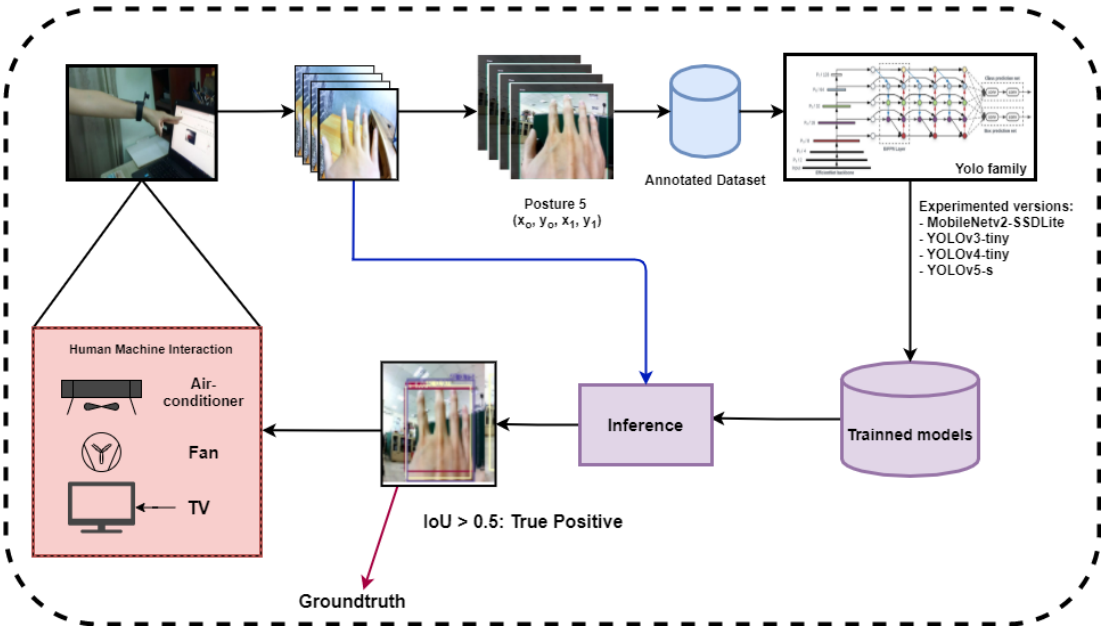


Fig. 6. The framework for hand posture recognition from handwrist camera in our study.

B bounding boxes are defined in every grid cell, each with a confidence score. The score is defined as:

$$\text{Confidence Score} = \Pr(\text{Object }) * IoU_{\text{pred}}^{\text{truth}} \tag{1}$$

where $\Pr(\text{Object })$ is the probability that there is an object inside a grid cell, IoU is the intersection over union, represents a fraction between 0 and 1. While the bounding boxes are determined, each grid cell predicts C conditional class probabilities simultaneously:

$$\begin{aligned} &\Pr\left(\text{Class}_i \mid \text{Object}\right) * \Pr(\text{Object}) * IoU_{\text{pred}}^{\text{truth}} \\ &= \Pr\left(\text{Class}_i\right) * IoU_{\text{pred}}^{\text{truth}} \end{aligned} \tag{2}$$

The loss function is computed according to [3]:

$$\begin{aligned} \mathcal{L} =& \lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{A} \mathbf{1}_{ij}^{\text{obj}} \left[\left(b_{x_i} - b_{\hat{x}_i}\right)^2 + \left(b_{y_i} - b_{\hat{y}_i}\right)^2 \right] \\ &+ \lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{A} \mathbf{1}_{ij}^{\text{obj}} \left[\left(\sqrt{b_{w_i}} - \sqrt{b_{\hat{w}_i}}\right)^2 + \left(\sqrt{b_{h_i}} - \sqrt{b_{\hat{h}_i}}\right)^2 \right] \\ &+ \sum_{i=0}^{s^2} \sum_{j=0}^{A} \mathbf{1}_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 \\ &+ \lambda_{\text{noobj}} \sum_{i=0}^{s^2} \sum_{j=0}^{A} \mathbf{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 \\ &+ \sum_{i=0}^{s^2} \mathbf{1}_{i}^{obj} \sum_{c \in \text{ classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 . \end{aligned} \tag{3}$$

The architecture of YOLO contains 24 convolutional layers and 2 fully connected layers. YOLO is later improved with different versions to minimize localization errors and increase mAP (Mean Average Precision).

1) YOLOv3-tiny: YOLOv3-tiny was introduced in [4] with a better architecture than its previous versions where the feature extractor used was a hybrid of YOLOv2, Darknet-53 (53 convolutional layers), and Residual networks (ResNet). The model used Darknet-53 which originally has the 53-layer network for training feature extractor. After that, 53 more layers were stacked for the detection head for training object detector, making YOLOv3 a total of 106 layers fully convolutional underlying architecture. Thanks to the residual blocks of ResNet, overlaying layers will not degrade network performance. The most notable feature of YOLOv3 is that it makes detections at 3 different scales to be able to detect multi-scale objects.

2) YOLOv4-tiny: The YOLOv4's authors performed a series of experiments with many advanced innovation ideas for each part of the architecture [5]. They chose CSP Darknet53 as being the most optimal model. Before forwarding to feature aggregation architecture in the neck, the output feature maps of the CSPDarknet53 backbone were sent to an additional block (Spatial Pyramid Pooling block) to increase the receptive field and separate out the most important features. The FPN architecture implemented a top-down path to transfer the semantical features and then concatenate them to fine-grained features for predicting small objects in the large-scale detector. YOLOV4 improvements are referred to by the terms "Bag of Freebies" and "Bag of Specials". YOLOv4-tiny is one of the lightweight YOLO series that aims at deployment on embedded devices. Compared with YOLOv3-tiny, the latter uses CSPBlock network to extract features without using the conditional convolution networks and introduces the complete intersection over union to select bounding boxes.

3) YOLOv5-s: A different research team applied various state-of-the-art innovations to create the most recent ver-

sion YOLOv5 [6]. YOLOv5 architecture is very similar to YOLOv4 with micro modifications. However, it possesses engineering advantages. The actively updated codebase is written in Python instead of C language for ease of installation, development, and integration on IoT devices. YOLOv5's authors provided 4 versions with increasing computational complexity and potency, namely YOLOv5-s, YOLOv5-m, YOLOv5-l, and YOLOv5-x. We employ the first one because it is the minimal architecture both in depth and in width. In summary, YOLOv5 consists of CSPDarknet53 as backbone, SPP additional module, PANet path-aggregation neck, and YOLOv3-styled multi-scales anchored heads.

4) Training YOLO models: YOLO models have become prevalent on large-scale datasets (Pascal VOC or COCO) that consists of a wide range of different object classes, but without hand posture classes as in our context. Nevertheless, we can still take advantage of transfer learning. For all considered YOLO models, we continue from pre-trained weights (that were adapted on COCO dataset) and fine-tune them with our training data. We use Adam optimizer, initial learning rate is 0.0001, batch size is 16, number of epochs is 300. We train the models on NVIDIA GeForce GTX 1080 TI GPU.

V. Experiment

A. Experimental setup and Evaluation metrics

The dataset is strategically split into train and test parts with proportions of 80% (609 images) and 20% (153 images) respectively, such that the class-wise demography is best retained in both subsets.

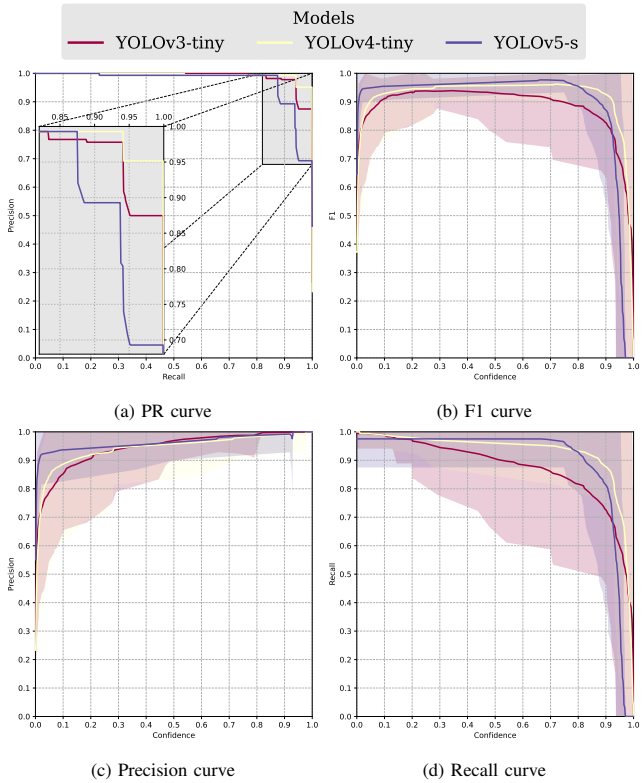
We report the Precision (P), Recall (R), and F1 scores, then compute the Confusion Matrix for each evaluated model. The average Precision/Recall/F1-Confidence, as well as Precision-Recall curves, are plotted for comparison. In addition, we also report the three widely used object detection metrics mAP@0.5, mAP@0.75, and mAP@[0.5:0.05:0.95].

B. Experimental results

Table I reports the outputs of our assessments comparing a series of lite YOLO architectures and MobileNetv2-SSDLite [7]. Throughout the experiments, IoU threshold and NMS threshold are both set to 0.5 while the empirically selected confidence threshold is 0.4. In this setup, both models scored very high with Precision, Recall, and F1 scores generally above 95%. Overall, the finetuned YOLOv4 only slightly outperforms other lightweight models for modest margins, with scores of P=98.85%, R=97.23%, F1=97.92%.

We further plotted the P, R, F1, and PR curves for the lite YOLO family in Figure 7. They both follow the conventional trends and fabricate near-perfect PR curves despite considerably smaller sizes. The colorized areas indicate the class-wise variance, which is synonymous with how consistent the performance of each model is.

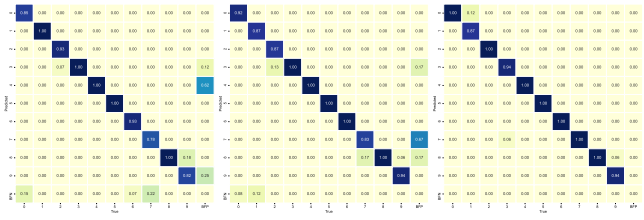
In terms of mAP, both models perform well at lower criteria of IoU between predictions and ground truths (mAP@0.5). However, as we increase the IoU threshold, there is generally a gradual decrease especially beyond 0.75. The trend is less intense in YOLOv5-s as compared to previous versions with its mAP@[0.5:0.05:0.95] remains at 92.62%, keeping large performance gaps to its predecessors (YOLOv3-tiny at 74.02%, YOLOv4-tiny at 71.34%). Comparing to existing performance on the similar datasets [12], [13], [14], YOLO tiny models are able to achieve competitive or even higher results.



(a) PR curve (b) F1 curve

(c) Precision curve (d) Recall curve

Fig. 7. Comparison of P-R curve, and P/R/F1-confidence curves of three evaluated lite models (denoted by color of the line). The transparent surrounding areas indicate the class-wise lower and upper bound statistics of models with corresponding color.



(a) YOLOv3-tiny (b) YOLOv4-tiny (c) YOLOv5-s

Fig. 8. Confusion matrices of three evaluated lite models. BFP and BFN stands for Background False Positive and Background False Negative, the lower the better.

Figure 8 shows the confusion matrices of each model with the aforementioned hyper-parameters. It is noted that in object detection task, the Background FP column indicates false detections over background, whereas Background FN row indicates missed detections. The values of them are only relative and should be zero for a perfect model. We can see that there are not many BFNs because of the ubiquitous presence

TABLE I
COMPARISON OF EXPERIMENTAL RESULTS OF MOBLIENETV2-SSDLITE, YOLOV3-TINY, YOLOV4-TINY AND YOLOV5-S.

Model	P	R	F1	mAP@0.5	mAP@0.75	mAP@[0.5:0.05:0.95]	Size (#params)	GFLOPs
MobileNetv2-SSDLite	98.32	96.65	97.35	98.66	98.66	91.07	4.3M	N/A
YOLOv3-tiny	94.75	93.99	93.90	**98.74**	95.51	74.02	8.7M	5.5
YOLOv4-tiny	97.93	94.93	96.21	99.08	**98.78**	71.34	5.9M	6.8
YOLOv4	**98.85**	97.23	**97.92**	98.61	79.64	68.51	27.6M	52
YOLOv5-s	98.16	**97.40**	97.70	96.92	96.92	**92.62**	1.7M	4.2

of the hand in front of the camera in our specific problem.

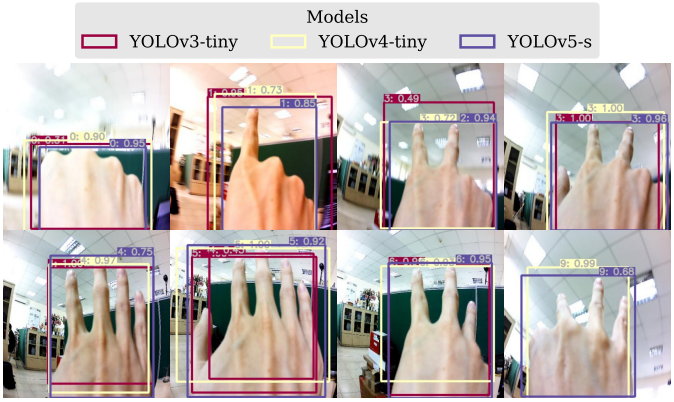


Fig. 9. Some example outputs of YOLOv3-tiny, YOLOv4-tiny, and YOLOv5-s with confidence threshold and NMS threshold set to 0.4 and 0.5 respectively.

Some example output bounding boxes are visualized in Figure 9. Generally, YOLOv5-s outputs more fitting bounding boxes covering the hand, which explains its outstanding mAP@[0.5:0.05:0.95]. The results promote the technical convenience and practicality of wrist-worn cameras for hand posture detection and recognition. As shown in our experiments, even small object detectors, which are dedicated to run on edge devices, can easily achieve sustainable performance with only partial differences in terms of precision of bounding boxes' coordinates regression (filtered by the IoU threshold).

VI. CONCLUSIONS

This paper presented a pilot study on hand posture recognition using a wrist-worn camera. We successfully designed and prototyped a wrist device that is able to capture images of human hand gestures. We collected a set of ten postures by 10 volunteers with the prototyped device in the context of home appliance control. We then evaluated the performance of state-of-the-art deep learning models YOLO and SSD as posture detectors and classifiers over our self-collected dataset. Preliminary results demonstrate that although the limitation of camera angles, fingers' configurations are highly distinctive for nearly perfectly recognizing the postures. This pilot gives the first original dataset of hand posture captured by wrist-worn camera and show the feasibility to recognize them, which are highly promising for human-machine interaction as well as home appliance controlling applications. In the future, we will evaluate the method with continuous video streams in combination with some multimodal sensors like accelerometer or gyroscope, conduct the dynamic hand gesture recognition and deploy the application of home appliance control.

REFERENCES

[1] H.-G. Doan, V.-T. Nguyen, H. Vu, and T.-H. Tran, "A combination of user-guide scheme and kernel descriptor on rgb-d data for robust and realtime hand posture recognition," *Engineering Applications of Artificial Intelligence*, vol. 49, pp. 103–113, 2016.

[2] T.-H. Le, T.-H. Tran, and C. Pham, "The internet-of-things based hand gestures using wearable sensors for human machine interaction," in *2019 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*. IEEE, 2019, pp. 1–6.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[4] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[5] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[6] G. Jocher, A. Stoken, J. Borovec, NanoCode012, A. Chaurasia, TaoXie, L. Changyu, A. V, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, J. Hajek, L. Diaconu, Marc, Y. Kwon, oleg, wanghaoyang0106, Y. Defretin, A. Lohia, ml5ah, B. Milanko, B. Fineran, D. Khromov, D. Yiwei, Doug, Durgesh, and F. Ingham, "ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations," Apr. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4679653

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019.

[8] T.-H. Tran, H.-N. Tran, and H.-G. Doan, "Dynamic hand gesture recognition from multi-modal streams using deep neural network," in *International Conference on Multi-disciplinary Trends in Artificial Intelligence*. Springer, 2019, pp. 156–167.

[9] H.-G. Doang, T.-H. Tran, T.-L. Le, H. Vu, V.-T. Nguyen, S. V. Dinh, T.-O. Nguyen, T.-T. Nguyen, and D.-C. Nguyen, "Multi-view discriminant analysis for dynamic hand gesture recognition," in *Pattern Recognition: ACPR 2019 Workshops, Auckland, New Zealand, November 26, 2019, Proceedings*, vol. 1180. Springer Nature, 2020, p. 196.

[10] H.-N. Tran, H.-Q. Nguyen, H.-G. Doan, T.-H. Tran, T.-L. Le, and H. Vu, "Pairwise-covariance multi-view discriminant analysis for robust cross-view human action recognition," *IEEE Access*, vol. 9, pp. 76 097–76 111, 2021.

[11] H. Park, H.-S. Shi, H.-H. Kim, and K.-H. Park, "A user adaptation method for hand shape recognition using wrist-mounted camera," *The Journal of the Korea institute of electronic communication sciences*, vol. 8, no. 6, pp. 805–814, 2013.

[12] F. Chen, J. Deng, Z. Pang, M. Baghaei Nejad, H. Yang, and G. Yang, "Finger angle-based hand gesture recognition for smart infrastructure using wearable wrist-worn camera," *Applied Sciences*, vol. 8, no. 3, p. 369, 2018.

[13] H.-S. Yeo, E. Wu, J. Lee, A. Quigley, and H. Koike, "Opisthenar: Hand poses and finger tapping recognition by observing back of hand using embedded wrist camera," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019, pp. 963–971.

[14] E. Wu, Y. Yuan, H.-S. Yeo, A. Quigley, H. Koike, and K. M. Kitani, "Back-hand-pose: 3d hand pose estimation for a wrist-worn camera via dorsum deformation network," in *Proceedings of the 33rd Annual ACM*

Symposium on User Interface Software and Technology, 2020, pp. 1147–1160.

[15] Y. Yamato, Y. Suzuki, K. Sekimori, B. Shizuki, and S. Takahashi, "Hand gesture interaction with a low-resolution infrared image sensor on an inner wrist," in *Proceedings of the International Conference on Advanced Visual Interfaces*, 2020, pp. 1–5.