International Journal of Computational Methods and Experimental Measurements
Vol. 12, No. 4, December, 2024, pp. 421-428
Journal homepage: http://iieta.org/journals/ijcmem

Accurate Hand Recognition with Neural Architecture Search Technology

Christine Dewi[1*], Yesicca Nataliani[2], Theophilus Wellem[1], Hanna Prillysca Chernovita[2], Ramos Somya[1], Henoch Juli Christanto[3], Lanyta Setyani Gunawan[1], Rio Arya Andika[1], Raynaldo[1]

[1] Department of Information Technology, Satya Wacana Christian University, Salatiga 50711, Indonesia
[2] Department of Information Systems, Satya Wacana Christian University, Salatiga 50711, Indonesia
[3] Department of Informatics Engineering, Soegijapranata Catholic University, Semarang 50234, Indonesia

Corresponding Author Email: christine.dewi@uksw.edu

https://doi.org/10.18280/ijcmem.120411

Received: 11 December 2023
Revised: 18 September 2024
Accepted: 8 October 2024
Available online: 27 December 2024

Keywords:
hand detection, YOLONAS, YOLO, deep learning

ABSTRACT

Hand gesture recognition is a technology that enables computers to interpret and understand hand movements and gestures made by users. It has various applications across various domains, including human-computer interaction, gaming, virtual reality, sign language interpretation, and robotics. Hand recognition faces challenges such as lighting conditions, occlusions, and variations in hand shape and size. Creating reliable and precise recognition systems frequently necessitates tackling these issues. Neural Architecture Search (NAS) is a technique employed in deep learning and artificial intelligence to automate the creation and optimization of neural network topologies. The objective of NAS is to identify neural network designs that are optimally aligned with certain objectives, including image classification, natural language processing, or reinforcement learning while reducing the necessity for manual design and adjustment. YOLONAS model's integration of YOLO's speed and efficiency with NAS-driven optimization results in improved accuracy and performance in gesture recognition tasks, making it a compelling choice for real-time applications requiring accurate and efficient gesture analysis. In this research, we implement YOLO with NAS technology and training with the Oxford Hand Dataset. Performance metrics are employed for monitoring and quantifying important data, such as the number of Giga Floating Point Operations Per Second (GFLOPS), the mean average precision (mAP), and the time taken for detection. The results of our study indicate that the utilization of YOLONAS with a training time of 100 epochs produces a more reliable output when compared to other approaches.

1. INTRODUCTION

Hand identification, also known as hand recognition or hand tracking, holds significant importance in various fields and applications due to its ability to detect and track the position and movement of human hands. 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 [1]. The presence of numerous hand variations depicted in images poses a challenge in identifying hands within uncontrolled scenarios [2]. Hand identification enables natural and intuitive interaction with computers and devices. Users can control and manipulate virtual objects, navigate menus, and perform actions using hand gestures, reducing the need for physical input devices like keyboards and mice [3, 4].

Hand identification can be used in assistive technology applications to assist individuals with disabilities. It enables those with mobility impairments to control computers and communicate using hand gestures, improving their quality of life and independence. Next, hand recognition is fundamental for gesture-based gaming systems. It provides a more immersive and interactive gaming experience by allowing players to control in-game characters and actions with hand movements and gestures. Hand identification is a versatile technology with applications across various industries and domains. It enhances human-computer interaction, accessibility, entertainment, and many other aspects of our daily lives, making it a crucial area of research and development in the fields of computer vision and human-computer interaction [5, 6].

Neural Architecture Search (NAS) is a methodology within the domain of deep learning and artificial intelligence that endeavors to mechanize the procedure of formulating neural network architectures. The objective of Neural Architecture Search (NAS) is to discover neural network structures that are effectively tailored for a given task or dataset while minimizing the need for labor-intensive human design and hyperparameter optimization. NAS technology can streamline and automate the laborious process of developing deep learning models, as well as construct deep neural networks rapidly and efficiently that are tuned to meet certain production requirements [7, 8].

YOLO-NAS is an innovative object detection foundational model created by Deci AI. It is the result of sophisticated Neural Architecture Search technology, which was meticulously designed to overcome the limitations of previous YOLO models. With significant improvements in quantization support and accuracy-latency trade-offs, YOLO-NAS represents a substantial leap in object detection [9, 10].

The following is the most important contribution that can be gained from conducting this research: (1) We implement, analyze, and evaluate the YOLO with NAS technology. (2) Many different types of object detectors are investigated in this study, such as the average mean accuracy (mAP), theintersection over union (IoU), and the number of GFLOPS. (3) Our proposed technique is trained and tested on the Oxford Hand Dataset using the YOLONAS framework.

The technique of gesture recognition has enormous potential in a variety of domains, including human-computer interaction, virtual reality, and other areas as well. Through the use of gesture recognition, users can engage with digital devices or computers in a manner that is more natural and intuitive, thereby imitating how humans communicate with one another. User interfaces that are based on gestures have the potential to simplify jobs by enabling users to carry out operations more expediently and effectively. This is especially useful in situations when manual input is either difficult or impractical.

Gesture recognition systems can monitor drivers' gestures and movements to detect signs of fatigue or distraction, alerting them to take corrective actions or providing assistance as needed. These applications highlight the necessity of research and development in gesture recognition technology to unlock its full potential in improving human-computer interaction, enhancing virtual experiences, and advancing various other fields. As technology continues to evolve, gesture recognition promises to play an increasingly integral role in shaping the way we interact with and experience digital and physical environments.

The subsequent sections of the paper follow a similar organizational structure. Section 2 provides an overview of the existing literature and research that is relevant to the topic at hand. The proposed methodologies, namely YOLONAS, are elaborated upon in Section 3. Section 4 comprises comprehensive elucidations of the conducted experiments, encompassing the intricacies of experimental design and the meticulous analysis of the acquired data. Section 5 provides a summary of the conclusions drawn from the study and outlines potential areas for future investigation.

2. RELATED WORK

2.1 Hand recognition with Convolutional Neural Network (CNN)

Hand recognition using Convolutional Neural Networks (CNNs) is a common computer vision task that involves training a neural network to detect and recognize human hands in images or video frames. This technology has various applications, including gesture recognition, sign language translation, and human-computer interaction. After trying out several different color spaces, Girondel et al. [11] 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 et al. [12], 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 et al. [13] developed a method that makes use of several movable parts. Hand detection is a computer vision technique that involves identifying and locating human hands in images or videos. It has various applications across different domains [14].

Furthermore, Nunez et al. [15] employed a neural network in conjunction with a long short-term memory (LSTM) network to discern three-dimensional hand motions by leveraging the temporal characteristics of a skeletal structure [16]. The computer vision field has witnessed 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 Convolutional Neural Networks (CNN) enables proficient resolution of the challenges associated with multi-scale and diverse rotations, as previously mentioned.

Previously, gesture recognition systems frequently depended on pre-established gesture templates or rule-based methods. These systems had challenges in dealing with the variety of gestures, lighting conditions, and occlusion. To overcome these constraints, more advanced feature extraction approaches were devised, such as handcrafted features like Histogram of Oriented Gradients (HOG) and Haar-like features.

Machine learning has caused a change in gesture identification towards systems that rely on data. Gestures were recognized using techniques such as Hidden Markov Models (HMMs), Support Vector Machines (SVMs), and Dynamic Time Warping (DTW). These approaches frequently necessitated significant training data and had limitations in their capacity to generalize to novel movements or users. Deep learning methods, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), demonstrated potential in acquiring intricate spatiotemporal patterns from unprocessed data.

2.2 YOLO with Neural Architecture Search (NAS)

Neural Architecture Search (NAS) refers to the automated process of designing the architecture of neural networks to attain optimal performance for a given job. The objective is to devise an architectural framework with constrained resources and with minimal human interaction. Figure 1 shows the NAS general framework.

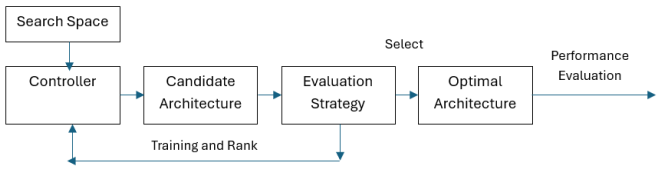

Figure 1. The architecture of NAS general framework

The landscape of NAS algorithms is generally characterized by a considerable degree of complexity and lack of clarity [17]. The prevailing classification scheme classifies NAS systems into three primary components: (1) The space being searched. (2) The search method encompasses the selection of the controller type and the evaluation of potential candidates. (3) The technique of performance evaluation. The term "search strategy" pertains to the systematic approach employed to

locate the most suitable architecture within the given search space. NAS algorithms can be categorized based on their search approach into five primary domains, including random search, reinforcement learning, evolutionary algorithms, sequential model-based optimization, and gradient optimization [18].

NAS is a technique used to automate the design of neural network architectures. Instead of manually designing architectures, NAS employs algorithms to search through a predefined search space of architectures and identifies the optimal architecture for a given task. The search space typically includes various architectural components such as convolutional layers, pooling layers, skip connections and activation functions. NAS algorithms, such as reinforcement learning, evolutionary algorithms, and gradient-based methods, iteratively explore the search space, evaluating different architectures based on performance metrics, and updating the search strategy to find increasingly better architectures.

The object detection basic model, YOLO-NAS, has been developed by Deci AI, representing a significant advancement in this field [19]. The product is a result of employing sophisticated Neural Architecture Search technology, which has been carefully developed to specifically target and overcome the constraints observed in prior iterations of YOLO models. YOLO-NAS demonstrates a substantial advancement in object detection with notable enhancements in quantization support and accuracy-latency trade-offs [20, 21].

The YOLO-NAS framework incorporates quantization-aware blocks and selective quantization techniques to achieve optimal performance [22, 23]. When the model is translated to its INT8 quantized version, there is a negligible decrease in precision, which represents a notable enhancement compared to alternative models. The aforementioned developments result in the development of an enhanced architectural framework that exhibits unparalleled capabilities in object identification and exceptional performance [24, 25].

Advantages of YOLONAS: (1) Customized Architecture: YOLONAS can discover neural network architectures optimized for gesture recognition tasks, potentially outperforming handcrafted architectures. (2) Efficiency: By leveraging the efficiency of YOLO and incorporating NAS, YOLONAS can achieve real-time performance on resource-constrained devices. (3) Adaptability: The automated architecture search process enables YOLONAS to adapt to different gesture recognition scenarios and datasets, improving generalization capabilities.

3. METHODOLOGY

3.1 YOLO-NAS architecture for hand recognition

The YOLO-NAS algorithm encompasses several notable characteristics, which are elucidated as follows: The algorithm's architecture was determined by the utilization of the company's exclusive technology, AutoNAC, which is a neural architecture search (NAS) approach. The AutoNAC algorithm was employed to ascertain the most favorable dimensions and configurations of stages, including block type, quantity of blocks, and number of channels within each stage [26, 27]. During the Neural Architecture Search (NAS) procedure, the model architecture was enhanced by incorporating quantization-aware RepVGG blocks, namely the QSP and QCI blocks depicted in the diagram [28]. This modification was implemented to ensure that the model architecture remains compatible with Post-Training Quantization (PTQ), hence minimizing any potential loss in accuracy. The proposed approach employs a hybrid quantization technique, which selectively quantizes specific components of a model [29, 30]. This method effectively minimizes the loss of information while simultaneously achieving a balance between latency and accuracy. The model underwent training using Objects365, a comprehensive dataset designed for object detection. This dataset encompasses a vast collection of 2 million photos, spanning over 365 distinct categories, and includes a total of 30 million bounding boxes. Additionally, the model underwent training using the RoboFlow100 dataset (RF100), which comprises a compilation of 100 datasets spanning several domains. This training was conducted to showcase the model's proficiency in tackling intricate object detection assignments [31].

The training process of YOLO-NAS is enhanced with the inclusion of Attention Mechanism, Knowledge Distillation, and Distribution Focal Loss. The software exhibits complete compatibility with advanced inference engines such as NVIDIA TensorRT and offers support for INT8 quantization, resulting in unparalleled runtime performance. YOLO-NAS demonstrates exceptional performance in practical contexts, including but not limited to autonomous vehicles, robotics, and video analytics applications, where the ability to minimize latency and optimize processing is of utmost importance. Figure 2 shows the YOLO-NAS architecture [32, 33].

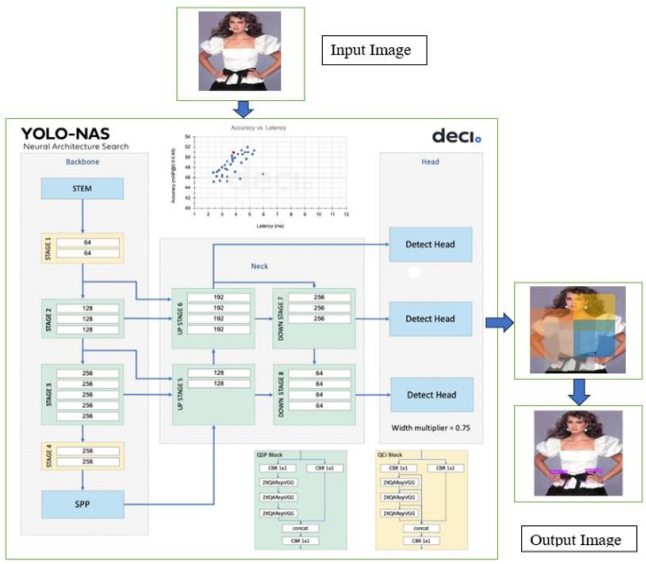

Figure 2. YOLO-NAS architecture

3.2 Oxford hand dataset

The Oxford hand dataset [13] 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. A total of 13,050 instances of hand gestures have been meticulously annotated. Instances that exceed a predetermined area of a bounding box, specifically 1500 square pixels, are deemed sufficiently large for detection purposes and are utilized for evaluation. This yields around 4170 instances of excellent quality hand. During the data collection process, there were no limitations placed on the pose or visibility of individuals, and no restrictions were

imposed on the surrounding environment. In each visual representation, the hands that are readily discernible to the human eye are meticulously labeled. The annotations are comprised of a bounding rectangle that is not necessarily axis-aligned but rather orientated about the wrist. The architectural structure features a series of rectangular prisms, each defined by manual enclosures. The necessary data preparation steps are conducted on the dataset before its export in Yolo format. The training dataset constitutes 70% of the total data, while the testing dataset comprises the remaining 30%. Both sections encompass representations of a wide range of handheld objects. Figure 3 presents an exemplary representation of an image obtained from the Oxford hand dataset.

Figure 3. Oxford hand dataset illustrations
Note: The human features depicted in the figures are derived from publicly available datasets (Oxford Hand Dataset)

4. EXPERIMENT AND RESULT

4.1 Training YOLO-NAS

In this section, we will strive to provide a comprehensive explanation of the training process and its associated results. Figure 4 illustrates the explication of the training process for the labels and predictions of test batch 0. Figure 4 exhibits the training process of YOLO-NAS. The YOLO-NAS architecture's training phase integrates Knowledge Distillation (KD) and Distribution Focal Loss (DFL) techniques to enhance the performance and accuracy of the training model in the context of object detection. The technique known as Knowledge Distillation (KD) is employed in the field of machine learning to decrease the computational resources needed by a model. This is achieved by training a simplified version of the original model, referred to as the student model, to achieve the same level of accuracy as the teacher model. However, the student model accomplishes this with significantly reduced computational requirements and memory usage compared to the teacher model.

Figure 4. Training process of YOLO-NAS

In this case, the model with lower complexity is trained to adjust its predictions to match the predictions made by the more advanced teacher model. The YOLO-NAS student model, which has been refined through the process of information distillation, exhibits enhanced optimization for devices characterized by constrained memory or processing capabilities, such as smartphones and other low-compute devices. The YOLO-NAS architecture has been developed to serve as a flexible neural network framework for various objective detection tasks, particularly in situations that demand fast and efficient prediction with minimal latency. Contemporary object detection methods necessitate robustness and applicability across diverse real-world contexts and application cases. For instance, the breadth and scale of object detection activities can vary considerably. These tasks range from the identification of galaxies or planets in astronomical photographs to the detection of microscopic organisms in biomedical studies using microscopy techniques.

The utilization of the Distribution Focal Loss (DFL) methodology in the YOLO-NAS architectural framework for effective handling of the inherent variability in target object size and position. This capability enhances the model's versatility and suitability for deployment in diverse circumstances. The DFL (Dynamic Focal Loss) is a variant of the focal loss function that was originally developed to tackle the issue of class imbalances in object detection tasks. The functionality of the DFL approach is enhanced through the categorization of a continuous range of likely bounding box values into distinct possibilities. Table 1 shows the YOLO-NAS training efficiency using the Oxford hand dataset. YOLO-NAS_s with 100 epochs shows the highest mAP 79.37% compared with 50 epochs 67.68%.

Table 1. YOLO-NAS training efficiency using the Oxford hand dataset

Model	Epoch	Precision	Recall	mAP	F1	Loss
YOLONAS_s	50	0.0503	0.9668	0. 6768	0.09563	1.8981
YOLONAS_s	100	0.06322	0.8648	0.7937	0.1178	1.87

4.2 Discussions and results

To measure the dataset's performance in these simulations, we use certain measures. The F1 score, Precision, Recall, and Accuracy are the criteria that this study considers. Precision and Recall, two of the metrics, are formally defined in Eqs. (1) and (2), respectively. Consequently, Eqs. (3) and (4), respectively, clearly define Accuracy and F1 [34, 35]. A True Positive (TP) refers to instances where both the model assessment and the actual situation indicate a positive outcome. A True Negative (TN) denotes the instances where both the model's evaluation and the actual situation indicate a negative outcome. The terms "TP" and "TN" are commonly used as abbreviations for true positive and true negative, respectively.

A false positive (FP) is a circumstance in statistical modeling where the observed data does not match the anticipated value produced by the model. A false negative (FN), on the other hand, occurs when the observed data do not match the anticipated value produced from the model [36].

$$Precision\ (P) = \frac{TP}{TP + FP} \tag{1}$$

$$Recall\ (R) = \frac{TP}{TP + FN} \tag{2}$$

$$Accuracy\ (Acc) = \frac{TP + TN}{TP + FN + FP + FN} \tag{3}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

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. (5) and (6) correspondingly.

$$mAP = \int_0^1 p(0)do \tag{5}$$

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*) [37].

$$IoU = \frac{Area_{pred} \cap Area_{gt}}{Area_{pred} \cup Area_{gt}} \tag{6}$$

Moreover, Eq. (7) [38] shows the calculation of the Yolo loss functions.

Let *S* represent the total count of grid cells in the image. *B* represents the count of bounding boxes that are anticipated to be present within each grid cell, and c is the predicted class between each grid cell. In addition, the symbol *pi* (c) represents the confidence probability score. In the context 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 weights λcoord and λnoobj are utilized to determine the relative importance of localization in the context of the task at hand. Table 2 describes the YOLO-NAS performance evaluation using the Oxford Hand Dataset. YOLO-NAS_s achieves the highest mAP 79.37% and Recall 96.68%.

$$Yolo\ Loss\ Function = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] + \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2 + \sum_{i=0}^{s^2} \mathbb{I}_{i}^{obj} \sum_{c \epsilon classes} (p_i(c) - \hat{p}_i(c))^2 \tag{7}$$

Table 2. YOLO-NAS performance evaluation using the Oxford hand dataset

Model	Epoch	Precision	Recall	mAP	F1
YOLO-NAS_s	50	0.05445	0.9624	0.7849	0.103
YOLO-NAS_s	100	0.0581	0.9668	0.7937	0.109

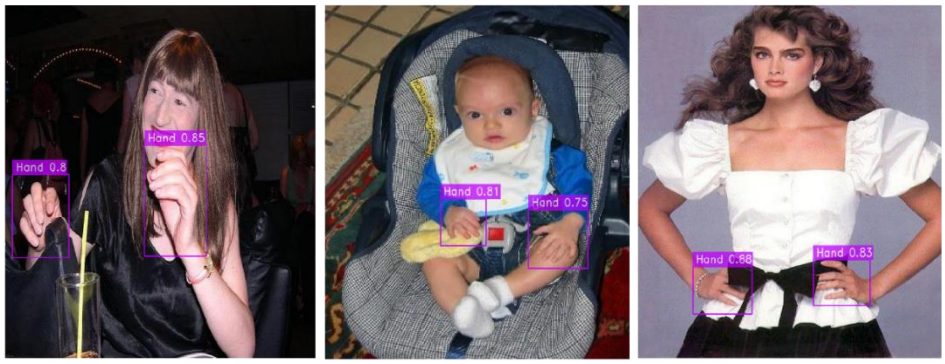

Figure 5. The detection result of Oxford hand dataset with YOLO-NAS

Figure 5 illustrates the recognition outcome of the Oxford Hand Dataset utilizing YOLO-NAS after 100 epochs. The YOLO-NAS model has robust efficacy in identifying all hands in Figure 5, with accuracies between 75% and 88%.

Neural architecture search benefits include: (1) The capacity of NAS to automate the architecture design process reduces the amount of effort necessary to manually create the neural network structure. This is its key advantage. The

development process is greatly accelerated by this automation, which also frees up researchers to concentrate on other areas of the issue. (2) High-level performance: The NAS has proven its capacity to find novel, high-performance architectures that outperform neural networks created by humans in a variety of applications. It has produced important advancements in computer vision, natural language processing, and picture recognition. (3) Scalability for complicated tasks: The manual design of the architecture becomes impractical as jobs get more complex and data intensive. NAS offers a productive method of investigating the potential of a wide range of architectures appropriate for challenging and large-scale activities.

The comparison to the preceding study is described in Table 3.

Table 3. A comparison of the results of previous studies

Author	Method	mAP (%)
Mittal et al. [13]	Classify the framework and two-stage hypothesize	48.2
Roy et al. [39]	R-CNN and skin	49.1
Deng et al. [40]	Joint model	58.10
Le et al. [41]	Multiple Scale Region-based Fully Convolutional Networks (MS RFCN)	75.1
Proposed Method	YOLO-NAS	79.37

Our proposed YOLO-NAS_s method with 100 epochs outperforms prior models on the Oxford Hand datasets in terms of mAP, with an accuracy of 79.37%. Le et al. [41] proposed the MS RFCN and exhibited only 75.1% mAP. Another researcher [42] implement the joint model and only achieve 58.10% mAP.

5. CONCLUSIONS

The YOLO-NAS architecture, developed by Deci's research and engineering team, establishes a new benchmark in object detection performance. It generates diverse models that can be effectively deployed in resource-constrained environments like edge devices. These models offer real-time and highly accurate performance for various object detection tasks. Furthermore, the YOLO-NAS method we propose with 100 epochs outperforms earlier models on the Oxford Hand datasets in terms of mAP, with an accuracy of 79.37%. Our solution performed better than other approaches currently being used for gesture detection and recognition. Extensive testing was used to confirm its usefulness and superiority.

YOLONAS offers several advantages. Firstly, it has a customized architecture that allows it to identify neural network topologies specifically designed for gesture recognition tasks. This customization has the potential to surpass pre-designed structures in terms of performance. (2) YOLONAS achieves real-time performance on devices with limited resources by combining the efficiency of YOLO with the integration of NAS. (3) Adaptability: YOLONAS can adapt to various gesture recognition circumstances and datasets, enhancing its ability to generalize.

In a forthcoming study, we plan to investigate the viability of merging explainable artificial intelligence (XAI) and hand detection. Additionally, we intend to combine hand detection with a logic-based framework in the future so that we may automatically conclude the scene being recognized.

ACKNOWLEDGMENT

This research is supported by the Vice-Rector of Research, Innovation, and Entrepreneurship at Satya Wacana Christian University.

REFERENCES

[1] Gopikha, S., Balamurugan, M. (2023). Regularised layerwise weight norm based skin lesion features extraction and classification. Computer Systems Science & Engineering, 44(3): 2727-2742. https://doi.org/10.32604/csse.2023.028609

[2] Narasimhaswamy, S., Wei, Z., Wang, Y., Zhang, J., Hoai, M. (2019). Contextual attention for hand detection in the wild. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, pp. 9567-9576. https://doi.org/10.1109/ICCV.2019.00966

[3] Kotsidou, D. (2018). Gesture recognition. In Gesture Recognition: Performance, Applications and Features. https://doi.org/10.1007/978-3-319-30973-6_9

[4] Dewi, C., Chen, A.P.S., Christanto, H.J. (2023). Deep learning for highly accurate hand recognition based on yolov7 model. Big Data and Cognitive Computing, 7(1): 53. https://doi.org/10.3390/bdcc7010053

[5] Liu, L., Xie, C., Wang, R., Yang, P., Sudirman, S., Zhang, J., Li, R., Wang, F. (2020). Deep learning based automatic multiclass wild pest monitoring approach using hybrid global and local activated features. IEEE Transactions on Industrial Informatics, 17(11): 7589-7598. https://doi.org/10.1109/TII.2020.2995208

[6] Alvin, A., Shabrina, N.H., Ryo, A., Christian, E. (2021). Hand gesture detection for sign language using neural network with mediapipe. Ultima Computing: Jurnal Sistem Komputer, 13(2): 57-62. https://doi.org/10.31937/sk.v13i2.2109

[7] Liu, Y., Sun, Y., Xue, B., Zhang, M., Yen, G.G., Tan, K. C. (2021). A survey on evolutionary neural architecture search. IEEE Transactions on Neural Networks and Learning Systems, 34(2): 550-570. https://doi.org/10.1109/TNNLS.2021.3100554

[8] Zhang, X., Huang, Z., Wang, N., Xiang, S., Pan, C. (2020). You only search once: Single shot neural architecture search via direct sparse optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(9): 2891-2904. https://doi.org/10.1109/TPAMI.2020.3020300

[9] Kong, G., Li, C., Peng, H., Han, Z., Qiao, H. (2023). EEG-based sleep stage classification via neural architecture search. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 31: 1075-1085. https://doi.org/10.1109/TNSRE.2023.3238764

[10] Tan, H., Cheng, R., Huang, S., He, C., Qiu, C., Yang, F., Luo, P. (2021). RelativeNAS: Relative neural architecture search via slow-fast learning. IEEE Transactions on Neural Networks and Learning Systems, 34(1): 475-489. https://doi.org/10.1109/TNNLS.2021.3096658

[11] Girondel, V., Bonnaud, L., Caplier, A. (2006). A human body analysis system. EURASIP Journal on Advances in Signal Processing, 2006: 61927. https://doi.org/10.1155/ASP/2006/61927
[12] 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. https://doi.org/10.1109/TPAMI.2004.35
[13] Mittal, A., Zisserman, A., Torr, P.H. (2011). Hand detection using multiple proposals. Bmvc, 2(3): 5.
[14] Dewi, C., Chen, R.C. (2022). Automatic medical face mask detection based on cross-stage partial network to combat COVID-19. Big Data and Cognitive Computing, 6(4): 106. https://doi.org/10.3390/bdcc6040106
[15] Nunez, J.C., Cabido, R., Pantrigo, J.J., Montemayor, A.S., Velez, J.F. (2018). Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognition, 76: 80-94. https://doi.org/10.1016/j.patcog.2017.10.033
[16] 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. https://doi.org/10.1109/JSEN.2022.3145844
[17] Dewi, C., Chen, R.C., Liu, Y.T., Jiang, X., Hartomo, K.D. (2021). Yolo V4 for advanced traffic sign recognition with synthetic training data generated by various GAN. IEEE Access, 9: 97228-97242. https://doi.org/10.1109/ACCESS.2021.3094201
[18] Zhang, H., Yao, Q., Kwok, J.T., Bai, X. (2022). Searching a high performance feature extractor for text recognition network. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(5): 6231-6246. https://doi.org/10.1109/TPAMI.2022.3205748
[19] 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: 108129. https://doi.org/10.1016/j.compag.2023.108129
[20] Cao, Y., Wang, H. (2022). Object detection: Algorithms and prospects. In 2022 International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI), Zakopane, Poland, pp. 1-4. https://doi.org/10.1109/ICDACAI57211.2022.00031
[21] Dewi, C., Chen, R.C. (2019). Random forest and support vector machine on features selection for regression analysis. International Journal of Innovative Computing, Information and Control, 15(6): 2027-2037. https://doi.org/10.24507/ijicic.15.06.2027
[22] Dewi, C., Christanto, H.J., Dai, G.W. (2023). Automated identification of insect pests: A deep transfer learning approach using ResNet. Acadlore Transactions on AI and Machine Learning, 2(4): 194-203. https://doi.org/10.56578/ataiml020402
[23] Dai, G., Tian, Z., Fan, J., Sunil, C.K., Dewi, C. (2024). DFN-PSAN: Multi-level deep information feature fusion extraction network for interpretable plant disease classification. Computers and Electronics in Agriculture, 216: 108481. https://doi.org/10.1016/j.compag.2023.108481
[24] Zhao, Y., Rao, Y., Dong, S., Zhang, J. (2020). Survey on deep learning object detection. Journal of Image and Graphics, 25(4): 629-654. https://doi.org/10.11834/jig.190307
[25] Ultralytics. YOLO-NAS, 2023. https://docs.ultralytics.com/models/yolo-nas/.
[26] Zhou, K., Huang, X., Song, Q., Chen, R., Hu, X. (2022). Auto-GNN: Neural architecture search of graph neural networks. Frontiers in Big Data, 5: 1029307. https://doi.org/10.3389/fdata.2022.1029307
[27] Dewi, C., Chen, R.C., Yu, H., Jiang, X. (2023). Robust detection method for improving small traffic sign recognition based on spatial pyramid pooling. Journal of Ambient Intelligence and Humanized Computing, 14(7): 8135-8152. https://doi.org/10.1007/s12652-021-03584-0
[28] Saravanarajan, V.S., Chen, R.C., Dewi, C., Chen, L.S., Ganesan, L. (2024). Car crash detection using ensemble deep learning. Multimedia Tools and Applications, 83(12): 36719-36737. https://doi.org/10.1007/s11042-023-15906-9
[29] Kasim, M.F., Watson-Parris, D., Deaconu, L., Oliver, S., Hatfield, P., Froula, D.H., Gregori, G., Jarvis, M., Khatiwala, S., Korenaga, J., Topp-Mugglestone, J., Viezzer, E., Vinko, S.M. (2021). Building high accuracy emulators for scientific simulations with deep neural architecture search. Machine Learning: Science and Technology, 3(1): 015013. https://doi.org/10.1088/2632-2153/ac3ffa
[30] Ardhianto, P., Santosa, Y.P., Moniaga, C., Utami, M.P., Dewi, C., Christanto, H.J., Chen, A.P.S. (2023). Generative deep learning for visual animation in landscapes design. Scientific Programming, 2023(1): 9443704. https://doi.org/10.1155/2023/9443704
[31] Kang, J.S., Kang, J., Kim, J.J., Jeon, K.W., Chung, H.J., Park, B.H. (2023). Neural architecture search survey: A computer vision perspective. Sensors, 23(3): 1713. https://doi.org/10.3390/s23031713
[32] Kollek, K., Braun, M., Meusener, J.H., Kummert, A. (2022). Real-time traffic counting on resource constrained embedded systems. In 2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS), Fukuoka, Japan, pp. 1-4. https://doi.org/10.1109/MWSCAS54063.2022.9859538
[33] Chitty-Venkata, K.T., Emani, M., Vishwanath, V., Somani, A.K. (2022). Neural architecture search for transformers: A survey. IEEE Access, 10: 108374-108412. https://doi.org/10.1109/ACCESS.2022.3212767
[34] Han, K., Zeng, X. (2021). Deep learning-based workers safety helmet wearing detection on construction sites using multi-scale features. IEEE Access, 10: 718-729. https://doi.org/10.1109/ACCESS.2021.3138407
[35] 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: 03023. https://doi.org/10.1051/matecconf/202235503023
[36] 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. https://doi.org/10.2478/cait-2022-0007
[37] Arcos-García, Á., Alvarez-Garcia, J.A., Soria-Morillo, L.M. (2018). Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing, 316: 332-344. https://doi.org/10.1016/j.neucom.2018.08.009
[38] 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, Las Vegas, NV, USA, pp. 779-788. https://doi.org/10.1109/CVPR.2016.91

[39] Roy, K., Mohanty, A., Sahay, R.R. (2017). Deep learning based hand detection in cluttered environment using skin segmentation. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, pp. 640-649. https://doi.org/10.1109/ICCVW.2017.81

[40] Deng, X., Zhang, Y., Yang, S., Tan, P., Chang, L., Yuan, Y., Wang, H. (2017). Joint hand detection and rotation estimation using CNN. IEEE Transactions on Image Processing, 27(4): 1888-1900. https://doi.org/10.1109/TIP.2017.2779600

[41] Le, T.H.N., Quach, K.G., Zhu, C., Duong, C.N., Luu, K., Savvides, M. (2017). Robust hand detection and classification in vehicles and in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, pp. 39-46. https://doi.org/10.1109/CVPRW.2017.159

[42] 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: 1071-1082. https://doi.org/10.1007/s00138-019-01038-4