IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 27, Issue 5, Ser. 1 (Sept. – Oct. 2025), PP 56-67
www.iosrjournals.org

Real Time Vision Hand Gesture Recognition Using C.N.N

Tarek Salah Elhabian
The Higher Institute Of Computer And Information Technology, El Shorouk Academy, Cairo, Egypt

Abstract
Hand gestures are a powerful way for people to communicate without speaking, bridging cultural and language gaps. They are especially important for helping people who are deaf or hard of hearing. As technology advances, using hand gestures to control and interact with digital devices has become a key focus. The proposed progress is driven by new developments in computer vision and machine learning, making interactions with technology feel more natural and intuitive. As proposed work, presenting a new hand gesture recognition system that uses the You Only Look Once (YOLO)v5 algorithm, a cutting-edge tool that employs Convolutional Neural Networks (CNNs) for quick and accurate gesture recognition. The proposed approach includes thorough costumed processing to reduce noise and clearly separate the background from key features, which helps to improve the system's ability to recognize gestures accurately. By using advanced algorithms and carefully chosen settings, proposed system is designed to enhance how users interact with technology, making it easier and more accessible for everyone which achieved recognition accuracies of 80.8%. Therefore, the proposed approach has great potential for use in different real-world applications. The results of proposed system indicate that deep neural networks can provide a robust and effective solution to hand gesture recognition tasks

Keywords: *Hand gesture recognition, Computer Vision, Human- computer interaction, gesture recognition, Convolutional Neural Networks, YOLO, Interactive technology*

Date of Submission: 09-09-2025 Date of Acceptance: 19-09-2025

I. Introduction

Hand gestures serve as a pivotal mode of non-verbal communication that significantly enriches our interaction with digital devices, making such interactions more intuitive and natural. This integration plays a crucial role in the field of human-computer interaction (HCI) [36], where it ushers in innovative and user-friendly ways to engage with technology daily. Importantly, these advancements are especially beneficial for individuals who encounter challenges with traditional methods of interaction, such as those who are deaf or hard of hearing. By enhancing accessibility, hand gesture recognition technologies promise to make technology universally accessible and easy to use. The study of hand gestures as a communication tool has seen substantial growth as computational technologies have become more sophisticated. With the significant strides made in computer vision and machine learning [29], it has become feasible to train computers to recognize and understand human gestures with impressive accuracy.

This advancement is driving forward exciting innovations in fields like virtual reality and smart home systems, where users can control devices simply by moving their hands, without any physical contact. In the pursuit of exploring this cutting-edge area, the current research employs some of the most advanced machine learning technologies available. At the heart of the proposed system is the "You Only Look Once" (YOLOv5) algorithm [37], which is celebrated for its speed and efficiency. This algorithm enables computers to recognize objects in images swiftly, which is crucial for applications that demand immediate responses, such as gesture-controlled systems. Alongside YOLOv5, the setup includes a Convolutional Neural Network (CNN), a type of deep learning model that is particularly adept at processing and analyzing visual information.

This combination not only enhances the system's capability to detect and interpret gestures but also ensures that the system can operate in real-time with high reliability. To further refine the system's ability to accurately interpret gestures, several image preprocessing techniques are implemented. These techniques are essential for enhancing the quality of input images, thereby allowing the model to focus more effectively on the essential features of gestures. Techniques such as noise reduction help to clarify the images, making it easier for the model to detect subtle nuances in the gesture data. Background segmentation is another crucial technique used to isolate gestures from unimportant background details, ensuring that the system's focus remains on the gestures themselves.

The proposed paper is divided into several key sections to facilitate a detailed discussion of the research. The initial section lays the foundation: Section 2.0 provides an extensive review of related works, offering a thorough background and situating this study within the existing landscape of gesture recognition research.

Section 3.0 describes the proposed system in detail, including the corpus used for experiments, the processes of feature extraction and gesture recognition, the metrics used for evaluation, and the various settings employed in the experiments. Section 4.0 presents the results obtained from these experiments, providing a clear view of how the proposed system performs under different conditions. Section 5.0 discusses the implications of the findings, considering their impact on current technologies and their potential to influence future developments in the field. Section 6.0 concludes the document, summarizing the research and suggesting potential areas for further exploration.

II. Related Work

The journey into advancing hand gesture recognition began with a foundational study that utilized Discrete Wavelet Transforms and F-ratio based feature descriptors to enhance the accuracy of gesture detection systems significantly [26]. This early work set a robust foundation for integrating classical methodologies with modern computational techniques, providing a critical framework for future advancements (Sahoo, Ari et al. 2018)[1]. Building on this foundational research, the potential of neural networks for real-time communication was demonstrated, particularly improving educational accessibility for the hearing impaired. This advance showcased the capability of deep learning to facilitate real-time translation of sign language [28] [30], opening new avenues for inclusive technology (Gauni, Bastia et al. 2021)[2]. The subsequent study employed advanced image processing and feature extraction techniques to refine the accuracy of interpreting American Sign Language. By integrating edge detection and ORB features with machine learning classifiers, the research enhanced the model's precision and reliability, pushing the boundaries of traditional computational approaches (Sharma, Mittal et al. 2020)[3].

Continuing with the theme of accessibility, the development of a system that translated static sign language gestures into text and speech with near-perfect accuracy was introduced. Utilizing convolutional neural networks, this innovation bridged significant communication gaps, highlighting the profound impact of deep learning on enhancing daily communication for the hearing impaired (Ismail, Aziz et al. 2021)[4]. The field also saw the introduction of novel, data-free class-incremental learning systems utilizing the BOAT-MI algorithm combined with SVM [35]. This innovative approach effectively managed the challenges of incremental learning without the need for retaining old data, achieving high accuracy, and demonstrating the adaptability of gesture recognition systems(Saiful, Isam et al. 2022)[5]. Deep convolutional neural networks were then applied to enhance the recognition of local sign languages, creating specialized datasets tailored to specific communities [25] [27]. This focus on localization significantly improved the practical applicability and accuracy of gesture recognition systems [34], highlighting the importance of targeted technological solutions (Kumar 2021)[6].

The integration of real-time machine learning algorithms, including SVM and CNNs, marked a significant leap in operational capability [33]. This study ensured the practical deployment of gesture recognition systems in everyday technology interactions, enhancing user experience and real-time responsiveness (Triyono, Pratisto et al. 2018)[7]. A landmark achievement in 2023 was the integration of YOLOv4 with SVM and Mediapipe, which achieved remarkable accuracy in continuous word-level sign language recognition. This hybrid model showcased the effectiveness of combining multiple advanced technologies to ensure precision and real-time responsiveness, setting a new standard in the field (Alsharif, Altaher et al. 2023)[8]. Further innovations included the development of specialized sensors and advanced deep learning algorithms to handle nuanced gestures. This technology enhanced the sensitivity and responsiveness of gesture recognition systems, enabling them to detect subtle variations in gestures with unprecedented precision, making them ideal for complex human-computer interactions (Thanasekhar, Kumar et al. 2019)[9].

State-of-the-art machine learning techniques were also developed to reduce latency in gesture processing and improve system adaptability to different user gestures. This ensured more reliable and user-friendly communication aids, further pushing the boundaries of what gesture recognition technology could achieve (Areyur Shanthakumar, Peng et al. 2020)[10]. The exploration of continuous word-level sign language recognition using an expert system combining sophisticated models like YOLOv5 with deep learning algorithms [37]. This integration provided seamless real-time translation of sign language, significantly enhancing communicative accessibility for sign language users (Popov and Laganiere 2022)[11]. The potential of LSTM networks to capture temporal dependencies in gesture sequences was explored, enhancing the model's ability to understand and predict movements accurately in real-time scenarios. This approach added a new layer of complexity to gesture recognition technologies, opening further possibilities for their application in more dynamic settings (Yu, Qin et al. 2022)[12]. The focus on static hand gesture recognition introduced advanced image processing and CNNs, aimed at increasing recognition speed and accuracy. This study highlighted the ongoing refinement of gesture recognition technologies, emphasizing the adaptability of these systems to complex backgrounds and dynamic gesture recognition (Cruz, Vásconez Hurtado et al. 2023)[13].

A novel LSTM-based recurrent neural network approach was introduced to improve hand gesture recognition. This study highlighted the benefits of recurrent neural networks in capturing dynamic and complex

gesture sequences, showcasing a significant advancement in the technology's ability to handle more intricate interactions (Al-Saedi and Al-Asadi 2019)[14]. Later, deep learning was applied to further enhance gesture recognition systems, focusing on optimizing algorithms to better interpret a wide array of human gestures. This research demonstrated the ongoing potential of deep learning to broaden the applicability of gesture recognition systems across various platforms, making them more versatile and robust (Ma, Zhang et al. 2021)[15].

Sophisticated algorithms were developed to effectively handle the challenges of gesture recognition in dynamic environments. This study focused on ensuring that systems could adapt to changes in gesture intensity and speed, illustrating the continuous efforts to improve the technology's performance in varied settings (Sharma, Mittal et al. 2020)[16]. Further innovations included the integration of geometric shape analysis with machine learning, providing new insights into improving gesture recognition accuracy. This approach was particularly effective in dynamic settings where gesture shapes and movements vary greatly, showing the potential for these technologies to adapt to more complex scenarios (Aich, Ruiz-Santaquiteria et al. 2023)[17]. A comprehensive survey of hand gesture recognition systems provided a detailed overview of historical and contemporary approaches to technology.

This survey highlighted the evolution of gesture recognition methods, emphasizing the diverse techniques and methodologies that have been explored over the years(Sharma 2021) [18]. The use of edge detection and ORB features integrated with machine learning classifiers explored the efficient interpretation of American Sign Language. Combining multiple technologies led to improved outcomes in gesture recognition, showcasing the effectiveness of integrating various computational methods (Yu, Qin et al. 2022)[19]. Continuing the exploration of word recognition, the application of machine learning techniques in 2019 demonstrated the adaptability of gesture recognition systems to understand and process complex linguistic patterns. This study not only improved the accuracy of word recognition but also enhanced the system's ability to interact in more linguistically diverse environments (Sreemathy, Turuk et al. 2023)[20]. The development of human motion gesture recognition systems in 2021 focused on using advanced computer vision techniques.

This approach helped in accurately recognizing and interpreting human movements, offering new opportunities for gesture recognition technologies to be applied in fields such as sports and physical therapy (Toro-Ossaba, Jaramillo-Tigreros et al. 2022)[21]. Exploring the use of color filtering combined with the Haar-cascade classifier in 2021 provided a fresh perspective on refining feature extraction for gesture recognition. This method proved effective in enhancing the clarity and precision of the captured gesture data, allowing for more accurate system responses (Popov and Laganiere 2022)[22]. Another significant contribution in 2021 was made with the introduction of a sophisticated image processing approach that included threshold creation and finding contours. This technique was particularly effective in sign language recognition, offering a more detailed and nuanced understanding of hand gestures (Ma, Zhang et al. 2021)[15].

The meticulous study on using Discrete Wavelet Transforms and F-ratio based feature descriptors provided a thorough analysis of how specific image processing techniques could be optimized to improve the performance of gesture recognition systems. This research offered a deep dive into the mathematical frameworks that support effective gesture analysis, underscoring the technical sophistication required to enhance detection and interpretation accuracy (Tchantchane, Zhou et al. 2023)[23]. Then it introduced a robust method involving the use of static hand gesture recognition paired with deep learning to improve the precision and efficiency of gesture classification. The study was significant for its innovative use of a combination of convolutional neural networks (CNNs) and principal component analysis (PCA) [32], aimed at reducing the computational load while maintaining high accuracy. This method allowed for faster processing times and lower resource consumption, making the technology more accessible and practical for real-time applications. The approach highlighted in this research provides a promising direction for future developments in gesture recognition technology, particularly in applications requiring immediate system responsiveness without sacrificing accuracy (Areyur Shanthakumar, Peng et al. 2020)[24].

III. Proposed Work

The system starts with high-quality image acquisition for the customized dataset, followed by detailed image preprocessing to enhance the quality and clarity of the input data. Advanced feature extraction techniques are then employed to capture the essential characteristics of hand gestures, which are crucial for the subsequent recognition phase. In-depth training and validation procedures ensure that the YOLOv5s model is well-adjusted to the nuances of hand gesture dynamics, enabling it to perform with high precision in diverse settings. This section will detail each component of the proposed system, outlining the technical strategies and algorithms employed to achieve a seamless and efficient gesture recognition workflow. The aim is to provide a comprehensive overview of the methods and technologies that underpin the research, highlighting how they collectively contribute to pushing the boundaries of what is achievable in real-time gesture-based interaction systems. Fig. 1 shows the block diagram of the work.

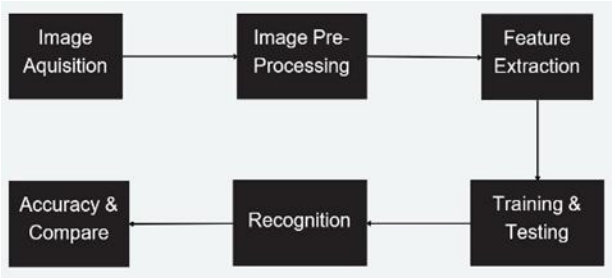

Fig.1 Block Diagram of proposed approach

Image acquisition (dataset)

The system initiates image capture through a webcam interface controlled by OpenCV, a widely used library for real-time computer vision. This setup allows for the dynamic acquisition of high-resolution images to the customized dataset, essential for accurate gesture recognition. The dataset consists of 20 classes. A total of 11,000 images. The words created are shown in Fig.2. Some of the words created are "Never Mind"," I'm Busy", "I'm Cold"," Don't Worry"," I'm Fine"," Nice to meet you". These words are used as they are some of the most used words in day-to-day conversations. The dataset is created with the help of some of our friends.

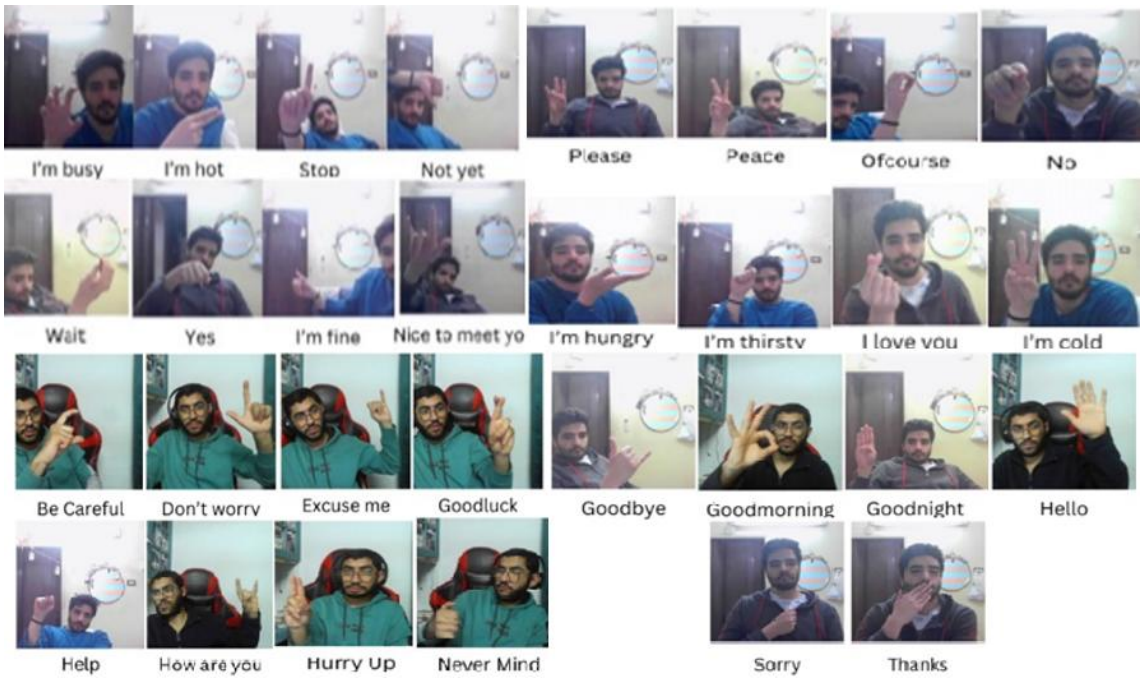

Fig. 2 Custom dataset used.

Image Pre-processing

Advanced preprocessing techniques are done to enhance the quality and consistency of the input images and to facilitate more effective feature extraction, several preprocessing techniques are employed:

Median Blurring: Applied to reduce noise from the images. Using the median blur approach, the value of each pixel is replaced by the median value of the intensity levels about that pixel. The process is mathematically represented as in Eq. (1).

$I_{xy} = \text{median}\left\{I_{(x+a)(y+b)}\right\}$ for $a, b \in [-N, N]$ (1)

where Ixy is the pixel value at position (x, y) and N is the size of the neighborhood around each pixel (defined by blur limit).

Greyscale Conversion: Converting images to greyscale simplifies the process by reducing the dimensionality of the data. This conversion can be represented by the equation as in Eq. (2).

$I' = 0.299 \times R + 0.587 \times G + 0.114 \times B$ (2)

where RR, GG, and BB are the red, green, and blue channel values of the pixels, and I' is the resulting greyscale image.

Contrast Limited Adaptive Histogram Equalization (CLAHE): Used to improve the contrast of the images. CLAHE operates by applying histogram equalization in localized regions of an image. The number of contextual regions is defined by tile_grid_size, and the contrast enhancement limit is controlled by clip limit. The method is beneficial for enhancing the visibility of features in regions that are darker or lighter than most parts of the image in Eq. (3).

$$\text{CLAHE}(I) = \bigcup_{k=1}^{K} \text{HE}_k(I_k) \quad (3)$$

where Ik is the k[th] tile of the image, HE is the histogram equalization applied to that tile, and K is the total number of tiles.

Feature Extraction

Feature extraction is a crucial stage in the proposed gesture recognition system, where significant characteristics from the pre-processed images are identified, extracted, and used for further analysis and classification. The YOLOv5 model, which includes advanced convolutional neural network (CNN) architectures, plays a central role in this process. CNNs are specialized kinds of neural networks that are particularly effective for image processing and analysis due to their ability to learn spatial hierarchies of features automatically and adaptively from image data. Fig. 2 shows the CNN typically consists of several types of layers:

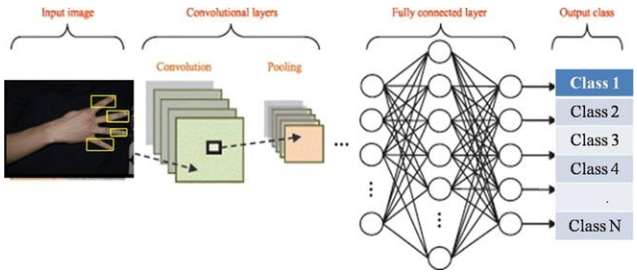

Fig. 3 Convolutional Neural Network (CNN) Architecture

Convolutional Layers: Perform the convolution operation that filters the input image with kernels (or weights) that are learned during training. The convolution operation captures local dependencies in the image and extracts low-level features such as edges and textures in the initial layers. As the data progresses through subsequent layers, the features become increasingly abstract and complex. The mathematical representation of the convolution operation in a layer is given by in Eq. (4).

$$g(x,y) = \sum_{a=-A}^{A} \sum_{b=-B}^{B} f(x-a, y-b) \cdot h(a,b) \quad (4)$$

where f is the image, h is the filter kernel, and g is the output feature map.

Pooling Layers: Reduce the spatial dimensions (width and height) of the input volume for the next convolutional layer. They perform down-sampling operations such as max pooling or average pooling to reduce the number of parameters and computation in the network. Max pooling, for example, selects the maximum value from each cluster of neurons at the prior layer, thereby emphasizing the most present features, reducing overfitting, and improving model generalization.

Activation Functions: Non-linear activation functions like Rectified Linear Unit (ReLU) are used after each convolution operation to introduce non-linear properties to the system, enabling the model to learn more complex patterns. ReLU is defined as in Eq. (5) Fig.4.

$$f(x) = max(0, x) \quad (5)$$

which sets all negative pixel values in the feature map to zero, simplifying the network and reducing training time.

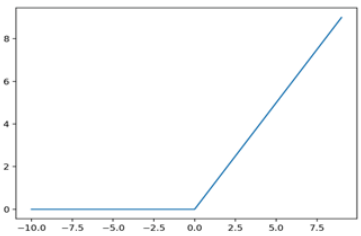

Fig. 4 ReLU Activation Function

The sigmoid activation function Fig.5 is used to map predictions to probability. It is especially useful in the final layer of a classification network, as it helps convert logits (raw prediction values) into probabilities that sum to one. The sigmoid function is defined as in Eq. (6).

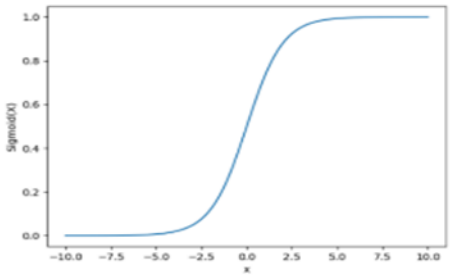

Fig. 5 Sigmoid Activation Function

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad (6)$$

where x is the input to the function. Outputs are in the range (0, 1), making them interpretable as probabilities, which is particularly useful for binary classifications such as object presence in a bounding box. Fig. 4 shows Sigmoid Activation Function graph.

Training & Testing

The effectiveness and robustness of any machine learning model, particularly in the context of hand gesture recognition, are largely dependent on comprehensive training, testing, and validation phases. These stages are crucial for turning the model to achieve high accuracy and ensuring that it generalizes well to new, unseen data. This subsection will explore each of these phases in the context of the YOLOv5s model employed in the proposed gesture recognition system.

Training: The training phase involves feeding the YOLOv5s model a large dataset of pre-processed images, where each image is labeled with the correct output, such as the type of gesture it represents. This dataset is typically divided into batches, and the model iteratively learns to recognize and predict gestures by adjusting its internal parameters (weights and biases).

Loss Function: The performance of the model during training is quantified using a loss function, which measures the difference between the predicted output and the true output. For YOLOv5s, a combination of binary cross-entropy loss for class predictions and mean squared error for bounding box predictions is commonly used. The loss function for YOLOv5s can be expressed as in Eq. (7).

$$\text{Loss} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj } j}[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{\text{obj}} \sum_{i=0}^{S^2} 1_i^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 \qquad (7)$$

xi, yi are the predicted coordinates, Ci is the confidence score, $x\hat{}i$, $y\hat{}i$, $C\hat{}i$ are the corresponding ground truth values, $1^{\text{obj}}_{\text{ij}}$ indicates if object is present in the cell, and λ_{coord}, λ_{obj} are coefficients to balance the contribution of each part of the loss.

Optimizer: An optimizer such as Adam or SGD (Stochastic Gradient Descent) is used to minimize the loss function by updating the model's weights. The optimizer adjusts the weight based on the computed gradients of the loss function with respect to the weights.

Testing: Once the model has been trained, it undergoes a testing phase where it is evaluated using a separate set of images that were not included in the training set. This phase is critical for assessing how well the model performs on new data, simulating real-world conditions where the model will encounter gestures it has not seen during training. The performance of the model during the testing phase is evaluated using metrics such as precision, recall, and the mean Average Precision (mAP). These metrics provide insight into the accuracy and reliability of the model in detecting gestures.

Validation: Validation typically occurs alongside or after the testing phase and involves running another set of data through the model to verify its performance. This can include cross-validation techniques where the dataset is split into several smaller sets; the model is trained on each set and tested on the remaining parts. Based on the outcomes of the testing and validation phases, the model may be fine-tuned. Adjustments can be made to the

model's architecture, hyperparameters, or training process to improve its accuracy and efficiency based on the specific needs identified during these evaluations.

Together, the three phases (training, testing, and validation) form a rigorous framework that ensures the YOLOv5s model is not only accurate in recognizing hand gestures but also robust and reliable across various environments and conditions. This comprehensive approach is essential for developing a high-performing gesture recognition system that can be effectively deployed in real-world applications.

Recognition

Recognition refers to the process of identifying and detecting the result of the training and see if the model after training can detect and identify what he trained about from the process of training.

YOLO algorithm.

YOLO, or "You Only Look Once," is a state-of-the-art, real-time object detection system that applies a single neural network to the full image. This allows it to predict bounding boxes and class probabilities for these boxes in one evaluation. Unlike systems which make predictions with a sliding window or patch, YOLO frames object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. This approach makes YOLO incredibly fast and accurate.

In its essence, YOLO divides the image into a grid and predicts bounding boxes and probabilities for each grid cell. The model applies a single neural network to the entire image, making predictions directly from full images and full feature maps. This unified model is extraordinarily fast and capable of being trained directly on full images.

The specific model used in this proposed work is YOLOv5s, the smallest variant ("s" for small) of the YOLOv5 models. It is designed for operational efficiency and is suitable for environments where computational resources are limited yet requires high-speed processing with relatively high accuracy. YOLOv5s maintains a delicate balance between speed and accuracy, making it ideal for real-time applications like gesture recognition.

YOLOv5 extends these basic CNN components by integrating them into a cohesive framework specifically optimized for speed and accuracy in object detection tasks. It employs a more complex structure with the following enhancements of its architecture Fig.6:

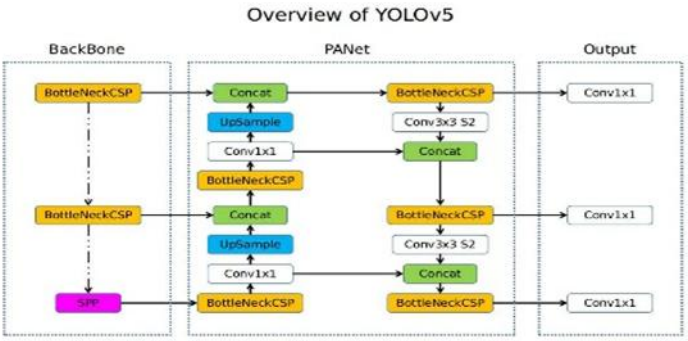

Fig.6 YOLOv5 Architecture

Backbone: The backbone of YOLOv5 is responsible for feature extraction. It usually consists of a pre-trained model like CSPDarknet53, adapted for efficient feature learning at various scales. The backbone processes the input image to create a rich set of feature maps that encapsulate different aspects of the image. CSPDarknet53: It consists of a series of convolutional layers with Mish activation functions and Cross-Stage Partial (CSP) connections. CSP helps to manage the gradient flow and reduces the redundancy of feature maps. The mathematical operation for a convolution layer in the backbone can be described as Eq. (8).

$$Y = \sigma(BN(\text{Conv}\ (X))) \quad (8)$$

where X is the input to a convolution layer, Conv represents the convolution operation, BN stands for Batch Normalization, and σ is the Mish activation function, defined as Eq. (9).

$$\sigma(x) = x \cdot \tanh\ (\ln(1 + e^{x})) \quad (9)$$

Neck: The neck of the YOLOv5 architecture, typically consisting of additional convolutional layers, is used to refine and reshape these features for better prediction by the head. This component often uses mechanisms like the Path Aggregation Network (PANet) to enhance the feature information flowing from the backbone to the detection heads.

PANet (Path Aggregation Network): PANet improves the information flow between different layers of the network. It aids in feature fusion by aggregating different levels of features from the backbone, enhancing feature utilization efficiency, and boosting detection performance, especially for small objects. It reuses the lower-level features by a bottom-up pathway and an additional top-down pathway to enhance feature hierarchy, which is crucial for detecting objects at different scales. The process of aggregating features can be mathematically summarized by the following operations in Eq. (10).

$$P_i = \text{Conv}\,(U_{i+1}) + U_i \quad (10)$$

where Pi is the output feature map, Ui are the up sampled features from a lower layer, and Conv represents a convolutional operation to refine features. Up sampling is often done using nearest neighbor or bilinear interpolation.

Head: The detection head is the final component that makes the actual predictions. It processes the refined features from the neck and outputs the bounding boxes, abjectness scores, and class probabilities. The head applies a 1x1 convolution to predict bounding boxes for each grid cell, including their dimensions, confidence scores, and class predictions. Detection Layer: It applies a 1x1 convolution to the feature maps to predict several attributes per bounding box: the center coordinates (bx, by), the width and height (bw, bh), the abjectness score, and class probabilities. The bounding box attributes are predicted relative to grid cells in Eq. (11):

$$\left(b_x, b_y, b_w, b_k\right) = \left(\sigma(t_x) + c_x, \sigma\left(t_y\right) + c_y, p_v e^{t_y}, p_k e^{t_s}\right) \quad (11)$$ e

Here, σ is the sigmoid function ensuring the outputs are between 0 and 1, (tx, ty, tw, th) are the network outputs, and (cx, cy) are the top-left coordinates of the grid cell. (pw, ph) are the anchor dimensions for the box.

This detailed breakdown of the YOLOv5 architecture, particularly the YOLOv5s variant, highlights the advanced engineering and thought that has gone into optimizing this model for tasks that require both precision and speed, such as hand gesture recognition in real-time applications. Each component, from the backbone through to the head, is meticulously designed to contribute efficiently towards the rapid and accurate detection of objects, underscoring the robustness and technological sophistication of the YOLOv5s model within the domain of computer vision.

IV. Experimental & Results

The proposed custom word recognition has been developed out by dividing the dataset into 70% for training, 15% for testing and 15% for validation. The data is fed into the network with a batch size of 64 samples in each training step and a total of 30 epochs have been conducted. This model for hand gesture recognition is trained using Adam optimizer, considering its ability to adapt to learning rates based on moving window of gradient updates. The initial learning rate and decay factor of Adam optimizer are set to 1 and 0.95 respectively. Table.1 shows the most important parameters and values for YOLOv5.

Table 1: Parameter and Values for YOLOv5.

Parameter	*Value*
Learning Rate	*0.001*
Number of Classes	*20*
Activation	*Mish*
Optimizer	*Adam*
Decay	*0.9*
Batch Size	*64*

Precision: The proportion of true positive predictions out of all positive predictions made by the model Eq (12). **Recall (Sensitivity or True Positive Rate):** The proportion of true positive predictions out of all actual positive instances Eq (13). **F1-Score:** The harmonic means of precision and recall, providing a balance between the two metrics Eq (14).

$$\text{Precision} = \frac{Tp}{TP+FP} \quad (12)$$

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

$$\text{F1Score} = \frac{2\times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (14)$$

Fig. 7 shows the result of the training. Fig. 8 shows the confusion matrix of the training. Fig. 9 shows the F1 score for each class.

Class	Images	Instances	P	R	mAP50	mAP50-95:
all	3014	2480	0.648	0.649	0.655	0.438
Goodbye	3014	90	0.638	0.627	0.689	0.494
Goodmorning	3014	89	0.778	0.854	0.916	0.658
Goodnight	3014	80	0.421	0.0125	0.0548	0.013
Hello	3014	81	0.714	0.924	0.909	0.656
How_are_You	3014	49	0.316	0.857	0.365	0.237
I_am_fine	3014	65	0.298	0.613	0.328	0.168
I_love_You	3014	44	0.384	0.409	0.318	0.233
Nice_to_meet_you	3014	86	0.831	0.698	0.879	0.549
No	3014	79	0.357	0.759	0.664	0.422
Peace	3014	87	0.799	0.655	0.804	0.547
Please	3014	73	0.813	0.954	0.964	0.654
Sorry	3014	35	0.675	0.971	0.85	0.528
Thanks	3014	70	0.84	0.827	0.904	0.559
Wait	3014	61	0.358	0.492	0.295	0.197
Yes	3014	128	0.457	0.32	0.349	0.216
Be_Careful	3014	98	0.922	0.605	0.841	0.57
Don't_Worry	3014	98	0.959	0.945	0.986	0.761
Excuse_Me	3014	95	0.646	0.726	0.71	0.471
Good_Luck	3014	94	0.562	0.84	0.733	0.489
Help	3014	143	0.576	0.466	0.464	0.361
Hurry_up	3014	89	0.516	0.865	0.672	0.487
I_am_Hungry	3014	98	0.981	0.529	0.678	0.361
I_am_thirsty	3014	87	0.281	0.368	0.326	0.234
I'm_Busy	3014	99	0.967	0.88	0.919	0.67
I'm_Cold	3014	83	0.672	0.788	0.726	0.494
I'm_Hot	3014	81	0.81	0.421	0.611	0.378
Never_Mind	3014	71	0.73	0.718	0.773	0.465
Not_Yet	3014	91	0.814	0.144	0.378	0.151
Of_Course	3014	42	0.43	0.905	0.747	0.548
Stop	3014	94	0.9	0.286	0.795	0.553

Fig. 7 Results of Training

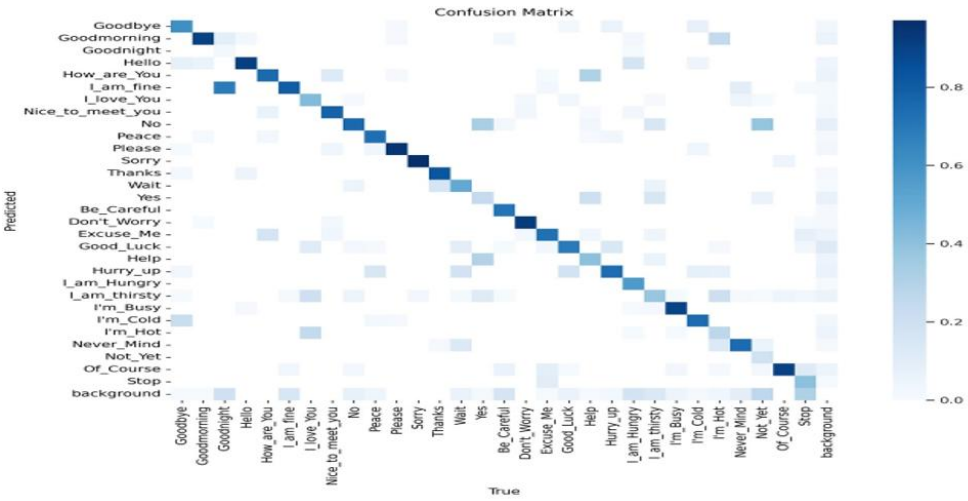

Fig. 8 Confusion Matrix

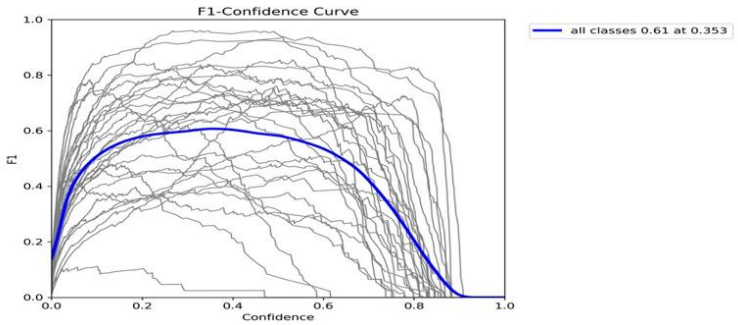

Fig.9 F1 Confidence Curve

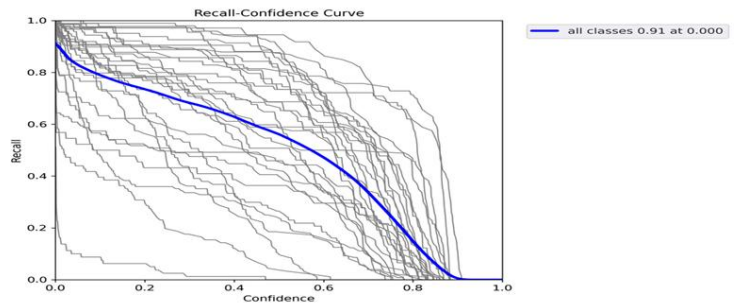

Fig.10 Recall Confidence Curve

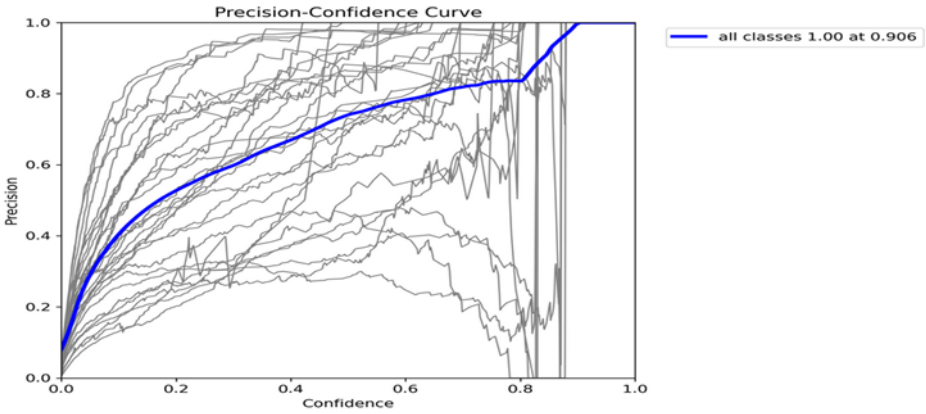

Fig.11 Precision Confidence Curve

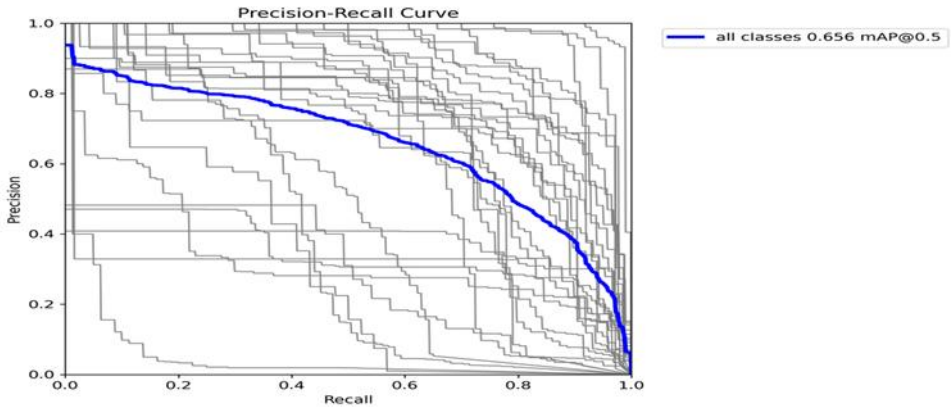

Fig.12 Precision-Recall Confidence Curve

Online test

We obtained a weight matrix from Experiment #4. Tested the matrix using external images not included in the dataset for online testing. Evaluation involved diverse conditions including various backgrounds, lighting, and angles. Testing included 30 classes with 900 samples each, assessing the robustness of the system.

Fig.13 images used for testing

Online test Results

Classes/Percentage: Be careful 86%, Don't Worry 82%, Excuse me 83%, Goodluck 65%, Goodbye 85%, Good morning 90%, Goodnight 80%, Hello, 95% Help 75%, How are you 95%, Hurry up 85%, I'm fine 70, I'm hungry 75%, I'm thirsty 77%, I love you 78%, I'm busy 78%, I'm cold 80%, I'm hot 75%, Never mind 88%, Nice to meet you 89%, No 82%, Not yet 82%, Of course 78%, Peace 65%, Please 80%, Sorry 96%, Stop 73%, Thanks 76%, Wait 77, Yes 85%

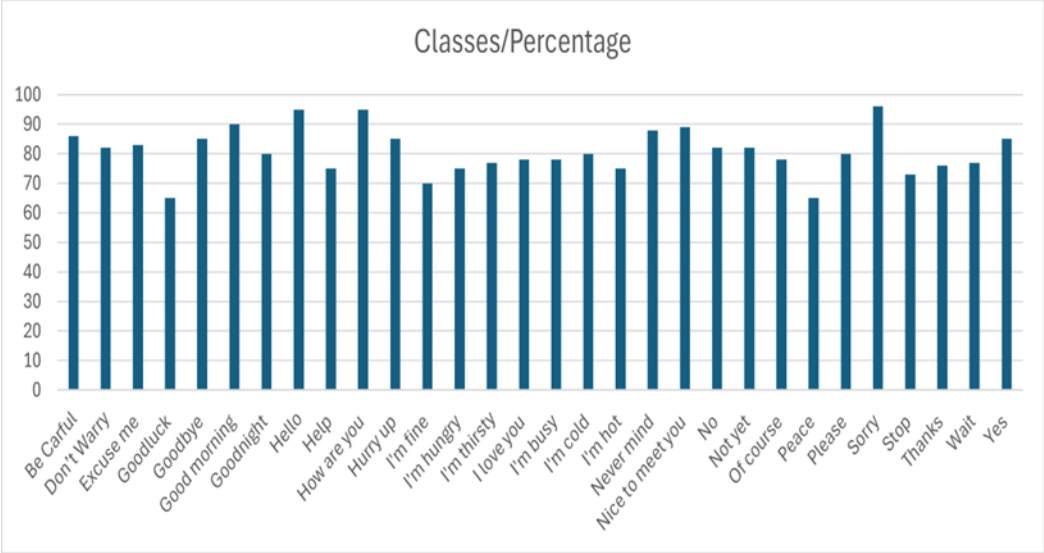

Fig.14 graph for the results

V. Conclusion

In conclusion, our research successfully demonstrated the effectiveness of a deep learning-based approach based on high-quality images acquired for personalized hand gestures. Specifically, we used the YOLO (You Only Look Once) model after applying advanced preprocessing techniques to improve the quality and consistency of the input images and facilitate more efficient feature extraction and then trained it and optimized the results. The utilization of the YOLO model has yielded promising recognition results, achieving a recall rate of 80.8% in our experimental setup.

References

[1]. Jaya Prakash Sahoo, Samit Ari, Dipak Kumar Ghosh. "Hand Gesture Recognition Using DWT And F-Ratio Based Feature Descriptor.", IET Image Processing, Vol. 12, Iss. 10, 2018.
[2]. Gauni, S., Et Al.. "Translation Of Gesture-Based Static Sign Language To Text And Speech." Journal Of Physics: Conference Series 1964(6): 062074, 2021.
[3]. Sharma, A., Et Al. "Hand Gesture Recognition Using Image Processing And Feature Extraction Techniques." Procedia Computer Science 173: 181-190, 2020.
[4]. Ismail, A. P., Et Al. "Hand Gesture Recognition On Python And Opencv" IOP Conference Series: Materials Science And Engineering 1045: 012043, 2021.
[5]. Md. Nafis Saifu, Abdulla Al Isam, Et Al. "Real-Time Sign Language Detection Using CNN", INTERNATIONAL CONFERENCE ON DATA ANALYTICS FOR BUSINESS AND INDUSTRY, October 2022, BAHRAIN
[6]. Kumar, R.. "An Improved Hand Gesture Recognition Algorithm Based On Image Contours To Identify The American Sign Language." IOP Conference Series: Materials Science And Engineering 1116: 012115, 2021.
[7]. Triyono, L., Et Al., "Sign Language Translator Application Using Opencv." IOP Conference Series: Materials Science And Engineering 333: 012109, 20218.
[8]. Alsharif, B., Et Al., "Deep Learning Technology To Recognize American Sign Language Alphabet." Sensors 23, 2023.
[9]. Thanasekhar, B., V Akshay, Abdul Majeed Ashfaaq, " Real Time Conversion Of Sign Language Using Deep Learning For Programming Basics", 11th International Conference On Advanced Computing (Icoac), IEE, December 2019, India.
[10]. Areyur Shanthakumar, V., Et Al., "Design And Evaluation Of A Hand Gesture Recognition Approach For Real-Time Interactions." Multimedia Tools And Applications 79, 2020.
[11]. Pavel A. Popov, Robert Laganière, "Long Hands Gesture Recognition System: 2 Step Gesture Recognition With Machine Learning And Geometric Shape Analysis.", Multimedia Tools And Applications 81, 2022.
[12]. Yu, J., Et Al., "Dynamic Gesture Recognition Based On 2D Convolutional Neural Network And Feature Fusion." Scientific Reports 12: 4345, 2022.
[13]. Cruz, P., Et Al., "A Deep Q-Network Based Hand Gesture Recognition System For Control Of Robotic Platforms." Scientific Reports 13, 2023.
[14]. Al-Saedi, A. And A. Al-Asadi, "Survey Of Hand Gesture Recognition Systems." Journal Of Physics: Conference Series 1294: 042003, 2019.
[15]. Rui Ma, Zhendong Zhang, Enqing Chen, "Human Motion Gesture Recognition Based On Computer Vision." Complexity 2021:
[16]. Sakshi Sharma, Sukhwinder Singh, "Vision-Based Hand Gesture Recognition Using Deep Learning For The Interpretation Of Sign Language" Expert Systems With Applications, Vol. 182: 115657, 2021.
[17]. Shubhra Aich; Jesus Ruiz-Santaquiteria; Zhenyu Lu; Et Al., "Data-Free Class-Incremental Hand Gesture Recognition", IEEE/CVF International Conference On Computer Vision (ICCV), 2023.
[18]. Jaya Prakash Sahoo , Samit Ari, Dipak Kumar Ghosh., "Hand Gesture Recognition Using DWT And F-Ratio Based Feature Descriptor." IET Image Processing, Vol. 12, Issue 10, 2018.
[19]. Toro-Ossaba, A., Et Al., "LSTM Recurrent Neural Network For Hand Gesture Recognition Using EMG Signals." Applied Sciences 12, 2022.

[20]. Sreemathy, R., Et Al., . "Continuous Word Level Sign Language Recognition Using An Expert System Based On Machine Learning." International Journal Of Cognitive Computing In Engineering 4, 2023.
[21]. Toro-Ossaba, A., Et Al., "LSTM Recurrent Neural Network For Hand Gesture Recognition Using EMG Signals." Applied Sciences, Vol.12, 2022.
[22]. Popov, P. And R. Laganiere (2022). "Long Hands Gesture Recognition System: 2 Step Gesture Recognition With Machine Learning And Geometric Shape Analysis." Multimedia Tools And Applications, Vol. 81, 2022.
[23]. Tchantchane, R., Et Al., "A Review Of Hand Gesture Recognition Systems Based On Noninvasive Wearable Sensors." Advanced Intelligent Systems 5(10): 2300207, 2023.
[24]. Areyur Shanthakumar, V., Et Al., "Design And Evaluation Of A Hand Gesture Recognition Approach For Real-Time Interactions." Multimedia Tools And Applications 79, 2020.
[25]. Stuart Russell Peter Norvig, " Artificial Intelligence: A Modern Approach (Third Edition)", Copyright 2010, 2003, 1995 By Pearson Education, Inc.
[26]. A. M. TURING, "Computing Machineray And Intelligence", Mind 49: 433-460, "MIND,Https: // Academic.Oup.Com/ Mind/ Article / LIX/236/433/986238.
[27]. Nils J. Nilsson, " Logic Of Artificial Intelligence", Artificial Intelligence, Vol. 47, Issues 1–3, January 1991, Pages 31-56.
[28]. Yann Lecun, Yoshua Bengio, Geoffrey Hinton, "Deep Learning", Nature Volume 521, Pages436–444 , 2015.
[29]. J. Bell, "What Is Machine Learning? In Machine Learning And The City: Applications In Architecture And Urban Design," Pp. 207-2016, 2022.
[30]. M. Najafabadi, F. Villanustre, T. Khoshgoftaar, N. Seliya, R. Wald And E. Muharemagic, "Deep Learning Applications And Challenges In Big Data Analytics," 2015. .
[31]. S. Fouladi, A. Safaei, N. Arshad, M. Ebadi And A. Ahmadian, "The Use Of Artificial Neural Networks To Diagnose Alzheimer's Disease From Brain Images.," 2022.
[32]. R. Saravanan And P. Sujatha, " A Perspective Of Supervised Learning Approaches In Data Classification.," Pp. 945-949, 2018.
[33]. Batta Mahesh, "Machine Learning Algorithms- A Review", International Journal Of Science And Research (IJSR), ISSN: 2319-7064, Pp. 381-386, 2020.
[34]. BASMH ALKANJR1, IMAD MAHGOUB2, "A Novel Deception-Based Scheme To Secure Location Information For Iobt Entities", IEEE Access, DOI 10.1109/ACCESS.2023.3244138.
[35]. Sindhu Padakandla, "A Survey Of Reinforcement Learning Algorithms For Dynamically Varying Environments", ACM Computing Surveys (CSUR), Volume 54, Issue 6, Pp. 1-25, 2021.
[36]. A. Energy, "Medium," Https://Medium.Com/Unpackai/The-One-Two-Threes-Of-Data-Labeling-For-Computer-Vision-4c0b022cef4.
[37]. Świeżewski, J. (2020). YOLO Algorithm And YOLO Object Detection. Appsilon. Retrieved 20 May 2022, From Https://Appsilon.Com/Object-Detection-Yolo-Algorithm/.
[38]. Geeksforgeeks, .Https://Www.Geeksforgeeks.Org/What-Are-Different-Types-Of-Denoising-Filters-In-Matlab/.
[39]. Peter Bankhead, "Analyzing Fluorescence Microscopy Images With Imagej", Queen's University Belfast, 2014.
[40]. L. G. Arie, "Medium," Https://Towardsdatascience.Com/The-Practical-Guide-For-Object-Detection-With-Yolov5-Algorithm-74c04aac4843.
[41]. C. Perauer, "Bachelor Thesis "Development And Deployment Of A Perception Stack For The Formula Student Driverless Competition"," Researchgate, 2021.