IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 December 2023, accepted 22 January 2024, date of publication 1 February 2024, date of current version 28 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3360857

RESEARCH ARTICLE

Single Shot Detector CNN and Deep Dilated Masks for Vision-Based Hand Gesture Recognition From Video Sequences

FAHMID AL FARID[1], (Member, IEEE), NORAMIZA HASHIM[2], (Member, IEEE), JUNAIDI BIN ABDULLAH[2], (Member, IEEE), MD. ROMAN BHUIYAN[2], MAGZHAN KAIRANBAY[3], ZULFADZLI YUSOFF[1], (Senior Member, IEEE), HEZERUL ABDUL KARIM[1], (Senior Member, IEEE), SARINA MANSOR[1], MD. TANJIL SARKER[1], AND GOBBI RAMASAMY[1], (Senior Member, IEEE)

[1]Faculty of Engineering, Multimedia University, Cyberjaya 63100, Malaysia
[2]Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63100, Malaysia
[3]Faculty of Engineering and Natural Sciences, Suleyman Demirel University (SDU), 32260 Almaty, Kazakhstan

Corresponding author: Hezerul Abdul Karim (hezerul@mmu.edu.my)

This work was supported by Multimedia University, Cyberjaya, Selangor, Malaysia, under Grant MMUI/230023.02.

ABSTRACT With an increasing number of people on the planet today, innovative human-computer interaction technologies and approaches may be employed to assist individuals in leading more fulfilling lives. Gesture-based technology has the potential to improve the safety and well-being of impaired people, as well as the general population. Recognizing gestures from video streams is a difficult problem because of the large degree of variation in the characteristics of each motion across individuals. In this article, we propose applying deep learning methods to recognize automated hand gestures using RGB and depth data. To train neural networks to detect hand gestures, any of these forms of data may be utilized. Gesture-based interfaces are more natural, intuitive, and straightforward. Earlier study attempted to characterize hand motions in a number of contexts. Our technique is evaluated using a vision-based gesture recognition system. In our suggested technique, image collection starts with RGB video and depth information captured with the Kinect sensor and is followed by tracking the hand using a single shot detector Convolutional Neural Network (SSD-CNN). When the kernel is applied, it creates an output value at each of the m × n locations. Using a collection of convolutional filters, each new feature layer generates a defined set of gesture detection predictions. After that, we perform deep dilation to make the gesture in the image masks more visible. Finally, hand gestures have been detected using the well-known classification technique SVM. Using deep learning we recognize hand gestures with higher accuracy of 93.68% in RGB passage, 83.45% in the depth passage, and 90.61% in RGB-D conjunction on the SKIG dataset compared to the state-of-the-art. In the context of our own created Different Camera Orientation Gesture (DCOG) dataset we got higher accuracy of 92.78% in RGB passage, 79.55% in the depth passage, and 88.56% in RGB-D conjunction for the gestures collected in 0-degree angle. Moreover, the framework intends to use unique methodologies to construct a superior vision-based hand gesture recognition system.

INDEX TERMS Gesture recognition, video sequences, SVM, SSD-CNN, deep dilated mask.

I. INTRODUCTION

Systems for recognizing hand gestures are at the leading edge of human-computer interaction (HCI). We know that vision-based technology for hand gesture detection is critical in human-computer interaction. Historically,

The associate editor coordinating the review of this manuscript and approving it for publication was Xuewen Chen.

human-computer interaction was accomplished via the use of a mouse and a keyboard. Gesture recognition is also a critical component of human activity recognition, which is concerned with deriving actions from a series of observations. Many applications, including healthcare, human-computer interaction, and video monitoring, rely on vision-based gesture recognition [1]. Researchers in the area of human-computer interaction pay close attention to voice and gesture recognition.

Fahmid Farid et al. blur video frames to eliminate background noise. After that, the images are transformed into the HSV color mode. Through dilation, erosion, filtering, and thresholding, they convert the image to black-and-white. Finally, SVM was used to identify hand motions. Gesture-based technology has the potential to aid both the disabled and the general population in preserving their safety and necessities. Due to the large variation in the parameters of each motion with respect to different people, detecting gestures from video streams is a difficult task [2], [3].

Raimundo F et al. and Roman et al. provide a convolutional neural network-based approach for Hajj applications [4], [5], [7], [10]. Some other different deep learning-based approaches are also considered in the current trend of gesture recognition arena [6], [8], [9], [11], [12].

In this previous work, image processing approaches such as wavelets and empirical mode decomposition were recommended for extracting picture functions in order to recognize manual movements in two dimensions or three dimensions. Additionally, CNN, a classifier of artificial neural networks (ANN), was used for data training and classification (CNN). They quantified three-dimensional gesture discrepancies using left- and right-handed 3D gesture movies [13].

The remaining sections of the paper are organized as follows: Section II examines various works that are linked to the first. Detailed explanations of the suggested approach for hand gesture recognition may be found in Section III. We describe in Section IV the experimental findings of our technique, which are then compared to the results of an existing state-of-the-art method. The following are the most significant contributions made by the paper:

1. To the best of our knowledge, utilizing an SSD convolutional neural network as an option for gesture recognition is a suitable alternative solution.
2. Using SSD-CNN, we suggested a technique for hand tracking that was both efficient and accurate.
3. In terms of accuracy, our suggested methodology outperforms the best available techniques.
4. We have created our own gesture dataset on Different Camera Orientations (DCOG)

II. RELATED WORKS

Hand gesture recognition is now a well-developed subject of study. In this area, a lot of work has been done. Hand segmentation is tough due to the variety of hand shapes and skin colors. Usually appears considerably different depending on the viewpoint; it might be open or closed, partially obscured, or have varying finger placements, for example.

Seniors who are deaf-mute utilize five separate hand gestures to request items such as beverages, food, toilet paper, help, and medicine. Due to the fact that elderly adults are unable to function independently, their requests are sent to their cell phones. The capabilities of the Microsoft Kinect v2 sensor to extract real-time hand motions confines this investigation to a small region [14].

Individuals with exceptional ability may use gestures and voices with a minor loosening of the physical proximity. It has always been critical to investigate effective human-computer interaction (HCI) in order to create novel ideas and techniques. Numerous approaches run into issues such as occlusions, changing illumination, limited resolution, and a high frame rate [15].

A workable prototype for performing gestures based on real-time interactions is constructed, consisting of a wearable gesture detection device equipped with four pressure sensors and the necessary computing framework. To make the system more viable, the hardware design must be streamlined further. Additional study is necessary to determine the optimal mix of system resilience and sensitivity [16].

This paper proposes a lightweight model for gesture identification that is built on the YOLO v3 and DarkNet-53 neural networks that do not need additional preprocessing, image filters, or image enhancement. Even in a complex environment, the proposed model was quite accurate, and movements were efficiently detected even in low-resolution picture mode rapid frame rate. The fundamental issue of this application for real-time gesture recognition is gesture categorization and recognition. Hand recognition is a technique that is employed by a variety of algorithms and concepts for comprehending the movement of a hand, including image and neural networks [17].

In existing work, the purpose is to identify long-run spatial correlations in cloud points by framing gesture recognition as an irregular issue of sequence identification. To transport information from the past to the future while keeping its spatial structure, an innovative and effective PointLSTM is presented. Dot clouds accurately capture the underlying geometric structure and distance information for object surfaces when compared to RGB data, offering additional gesture detection markers [8].

A novel system for dynamic hand gesture recognition is given, employing many architectures to learn how to divide hands, local and global characteristics, globalization, and sequence recognition features. To develop an efficient system for recognition, the following issues must be addressed: hand segmentation, local representation of hand shapes, global corporate configuration, and gesture sequence modeling [18].

The article detects and recognizes human hand gestures using a classification method for neural networks (CNN). This process flow involves segmentation of the hand

region using a mask image, segmentation of the fingers, normalization of the segmented finger images, and CNN classification finger identification. To detect standard gesture strategies, SVM and Naive Bayes classificatory algorithms were utilized, which needed a huge quantity of data for gesture pattern recognition [19].

In these articles, they provide an overview of contemporary convolutional neural networks for action and gesture recognition in visual frames. They present a framework for addressing these challenges that incorporate both deep learning and other handcrafted approaches. These recommended architectures, fusion procedures, primary datasets, and contests are all thoroughly explored. We summarize and analyse the important ideas made so far, with an emphasis on how they address the temporal component of data, identifying potential and challenges for future research using 3D Models [20], [21], [22], [23], [24], [25], [26], [27], [28].

The previous section reviewed the evaluation's findings, accompanying issues, and future research prospects. In the future, it is important to find viable solutions. We believe that the discussions in this area of the work will uncover new research gaps, allowing us to get closer to the much-desired next-generation technologies [29], [30], [31], [32], [33].

Hybrid models that combine classic and new features are predicted to gain traction. Similarly, we anticipate that deep learning solutions for large-scale, real-time action and gesture identification would be of interest to the community. Action and gesture localization in enlarged, uncensored, and realistic videos is also being worked on right now. As a result, we predict that emerging issues including early detection, multi-task learning, captioning, recognition from low-resolution sequences, and lifelogging devices will receive more attention in the next years [34], [35], [36], [37], [38].

SSD partitions the bounding box output space into a collection of default boxes with variable aspect ratios and sizes for each feature map point. It illustrates a strategy for detecting items in photographs using a deep neural network with a single prediction time. The network calculates scores for the presence of each item type inside each default box and modifies the box's shape to better reflect the object's form. Additionally, the network handles objects of changing sizes automatically by mixing predictions from a variety of different feature maps with differing resolutions. SSD outperforms techniques that need object proposals because it omits the proposal generation and subsequent pixel or feature resampling stages and encapsulates all computations in a single network [39].

III. PROPOSED ALGORITHM FOR HAND TRACKING

Image acquisition starts from RGB plus depth videos capture using a Kinect sensor and then tracking the hand using SSD-CNN, then doing the deep dilation to get deep dilated masks, and then the features fed into the SVM classifier so that the hand gestures can be recognized. It dilates the image object, and the objects in the image get thinner after erosion, blurring the image with the kernel size (5*5). We utilized our suggested approach to recognize the hands from a video stream and go through all of the files in order to accomplish hand gesture identification using SSD-CNN.

We track the hand using SSD-CNN and store the centers of the hands as a feature vector. We read the video by each frame and detect the hand. After the hand is detected, we extract the landmark. Then we draw the hand mask and save all the masks in the directory. For a feature layer of size m $\times$ n with p channels, the fundamental component for predicting the parameters of a prospective detection is a 3 $\times$ 3 $\times$ p tiny kernel that generates either a category score or a shape offset relative to the default box coordinates. It generates an output value at each of the m $\times$ n places when the kernel is applied. The overall process and our experimental scenario are depicted in figure 1.

From the figure, we can see very clearly that using the SSD-CNN we started hand tracking from the video frames. After getting the gesture detected frames we got the mask of the detected hand by morphological operation. We merge all the video frames into a single frame. Finally, using the SVM classifier hand gestures are recognized. Figure 2 shows sequentially how the hand tracking is done using SSD-CNN. In this figure, we visualize how the hand tracking was done using python programming. Using the pink dots we detected points on hands and fingers. The accuracy of Hand tracking varies based on the background and illumination conditions of the gesture videos.

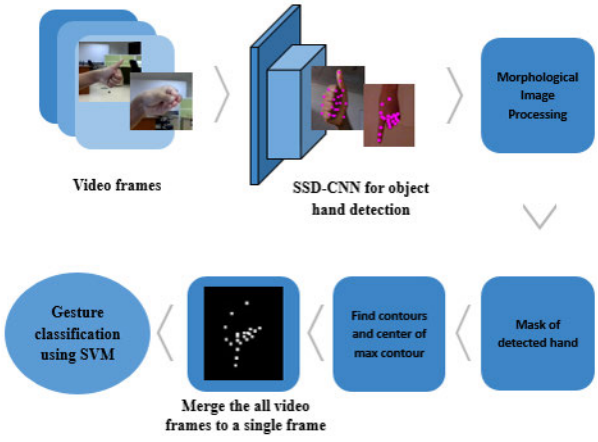

FIGURE 1. Proposed algorithm for gesture recognition using SSD-CNN-based method.

IV. METHODOLOGY

In comparison to our previous paper, we have tried to replace the hand detection part with SSD-CNN since CNN shows significant results on other computer vision-related tasks [2], [3]. We have used the Single Shot Multiple Box Detector approach while detecting the hand [39], where it looks only once at the image and tries to detect the object that is needed. Once the hands are detected in each frame of the video,

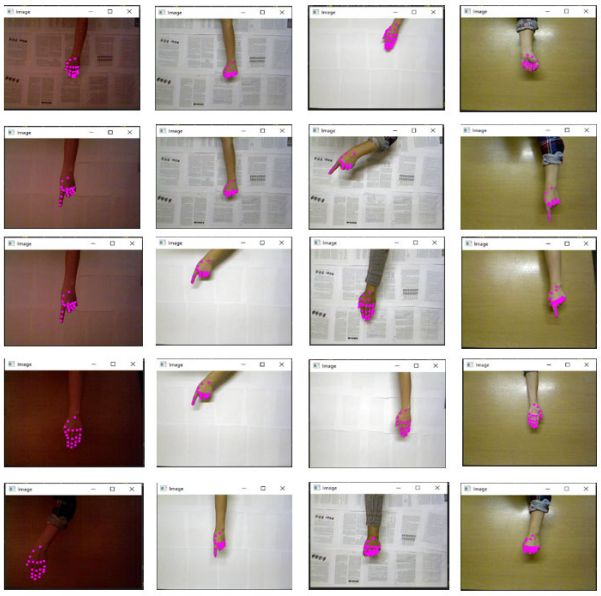

FIGURE 2. Hand tracking using SSD-CNN-based method.

we draw the mask for that frame, where the detected hands are shown in white. However, the rest of the background is drawn in black. In such a way, we can detect the hand gestures and their trajectory from the video. However, for some of the videos, we are not able to detect the hands in the majority of the video frames. Therefore, the masks become more unclear. That is why we concluded using a dilation morphological operator to make the hand more visible in the mask. Once we get clear deep dilated hand masks, we start building the classifier. As a classifier, we have used SVM for flattened images as a one-dimensional feature vector.

A. SVM CLASSIFICATION MATRIX FOR HAND GESTURE RECOGNITION

The SVM classification matrix, also known as the confusion matrix, provides a tabular representation of the performance of a Support Vector Machine (SVM) model for gesture recognition. It compares the predicted labels with the true labels of the testing samples. Here is how we calculate the elements of the classification matrix: Let's assume we have N classes or gestures in our dataset, labeled as G1, G2, G3, …, GN.

1. Calculate the predicted labels: Apply the feature vectors of the testing samples to the trained SVM model and obtain the predicted labels for each sample.

2. Construct the confusion matrix: Create an N x N matrix where each row represents the true labels and each column represents the predicted labels. Initialize all the elements of the matrix to zero.

3. Update the matrix: For each testing sample, increment the corresponding cell in the confusion matrix based on the true and predicted labels. For example, if a sample with the true label G1 is predicted as G2, it would increment the cell (1, 2) in the confusion matrix.

4. Interpretation of the matrix: The confusion matrix provides a clear view of the classification results. Each row of the matrix represents the instances in the true class, while each column represents the instances in the predicted class. The diagonal elements of the matrix represent the number of correctly classified samples for each gesture. The equation for the classification matrix can be represented as:

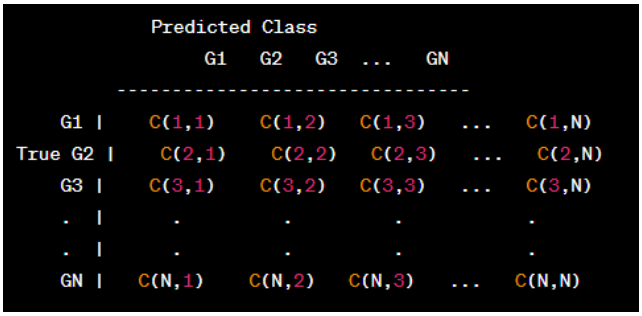

FIGURE 3. Representation of the equation for the classification matrix.

In the equation, C(i, j) represents the cell at the intersection of row i and column j, representing the count of samples with the true label Gi predicted as Gj. By analyzing the elements of the classification matrix, we can calculate various evaluation metrics, which provide a more comprehensive understanding of the performance of our SVM model for gesture recognition.

V. IMPLEMENTATION

On the NVIDIA GEFORCE GTX 1660Ti GPU, deep learning packages are employed. For deep learning, we utilized OpenCV-python 3.4.11.43, NumPy 1.21.2, SciPy 1.21.2, and matplotlib 3.4.3.

A. DATASET

We performed our experiment using the well-known SKIG dataset. A total of 1080 RGB and depth videos are included in the dataset. We grouped the whole dataset into three sections. We classified the data randomly into three of these categories. This dataset has ten unique gesture categories. Circle (clockwise), triangle (anti-clockwise), up-down, right-left, wave, "Z," cross, come here, turn around, and pat are just a few of the options. 10 unique hand postures were used to collect these ten characteristics: fist, index, and flat [40].

We have created our own Different Camera Orientation Gesture (DCOG) dataset where we also have 10 types of gestures such as Circle, Request for Coffee, Request for Doctor, Request for Food, Request for Tea, Request for help, Request Water, Right-Left, Triangle, and Capital letter – Z. utilized two-thirds of the dataset for training and the remainder for testing. Figure 7 shows the representation of our own created dataset.

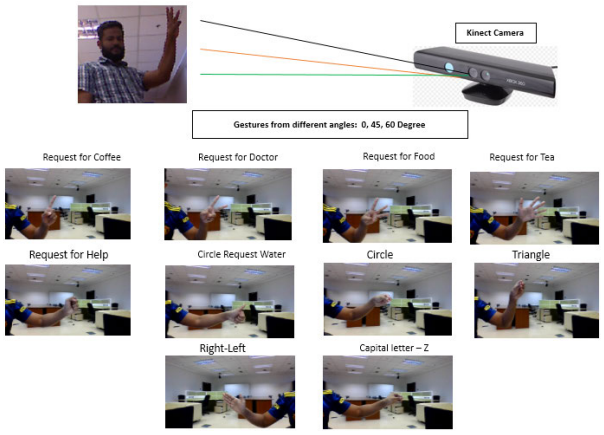

FIGURE 4. Our own created gesture dataset in Different Camera Orientation (DCOG).

B. CONVOLUTIONAL NEURAL NETWORK CONSTRUCTION

Lecun et al. recommended that convolutional neural networks (CNNs) be used as one of the most effective pattern recognition techniques [41]. This system extracts visual information from the input picture using locally learned filters. A convolutional layer, a pooling layer, and a fully connected layer comprise CNN's internal layer structure. The whole CNN technique for gesture recognition, in general, is shown in Figure 5.

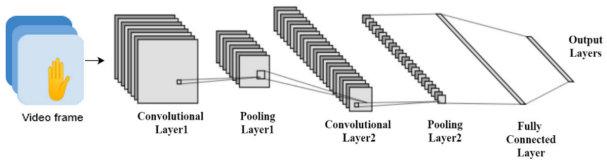

FIGURE 5. The convolutional neural network architecture for gesture recognition.

C. LAYERS OF CONVOLUTION

The more sophisticated feature representation is provided by convolutional operations. The sophisticated functions can be employed in the input picture thanks to a number of fixed-size filters. The weights and biases of each filter are consistent across the image. You may represent a whole picture using the same characteristic by using the weight-sharing approach. The area of a neuron's local receptive field mirrors the previous layer's area. The filter size is used to determine the receptive field size. Eq. 1 and 2 show the mathematical form of the activation function. (1),

$$O_{0,0} = f\left(b + \sum_{t=0}^{c}\sum_{r=0}^{c} w_{t,r} i_{0+t,0+r}\right) \tag{1}$$

$$f(x) = \begin{cases} x & x > 0 \\ 0 & \text{else} \end{cases} \tag{2}$$

D. POOLING LAYER

Convolution and activation functions have been applied to feature maps before they are employed in the pooling technique. Because it reduces complexity while still keeping crucial qualities, feature extraction efficiency is increased by using the local average or maximum value [43].

E. FULLY CONNECTED LAYER

The convolutional and pooling layers alternately transmit the image features, and the fully-connected layer then receives the image feature as an input. It's possible that the topmost layer is just the surface of a much deeper structure. After multiplying the weights by the preceding layer's data, each neuron adds a bias value to the connection weights. The activation function is used to process the measured value before it is passed to the next layer on how to solve a problem (3). This layer displays the computations of neurons. Where f denotes the activation function, w the weight vector, O the qth neuron's input vector, and b the bias value.

$$fc1 = f\left(b + \sum_{q=1}^{M} w_{1,q} * O_q\right) \tag{3}$$

VI. SSD-CNN BASED METHOD

A single-shot detector for multiple categories is quicker than the prior single-shot state-of-the-art (YOLO) and significantly more accurate, equal to slower approaches that conduct explicit region suggestions and pooling (like Faster R-CNN). The heart of SSD employs tiny convolutional filters applied to feature maps to forecast category scores and box offsets for a specific set of default bounding boxes. To achieve high detection accuracy, they explicitly segregate predictions by aspect ratio and construct predictions of various scales from feature maps of various sizes. SSD can do 8732 detections per class, whereas YOLO can detect only 98. In addition, the detection rates are for SSD at 59 frames per second and YOLO at 45 frames per second.

The SSD is a popular convolutional neural network architecture. It combines the strengths of deep CNNs for feature extraction and efficient multi-scale object detection. When configuring an SSD model, several parameters need to be considered. Here are some important parameters for an SSD-CNN model:

Backbone network: The backbone network is typically a pre-trained CNN model that serves as the feature extractor. Common choices include VGGNet, ResNet, or MobileNet. The choice of backbone network affects the model's capacity, speed, and accuracy.

Input size: The input size refers to the dimensions (width and height) of the input image. It is typically defined as a square image, such as 300 × 300 or 512 × 512 pixels. Larger input sizes generally allow the model to detect smaller objects but may require more computational resources.

Feature map scales: SSD uses multiple feature maps of different sizes to detect objects at various scales. The scale of a feature map determines the size of objects it can detect. Common scales include 19 × 19, 10 × 10, 5 × 5, 3 × 3,

and 1 × 1. These scales are defined based on the size of the input image and the architecture of the backbone network.

Aspect ratios: SSD uses default anchor boxes, or priors, to predict object locations and sizes. Aspect ratios determine the width-to-height ratios of these anchor boxes. By using different aspect ratios, the model can handle objects of various shapes. Common aspect ratios include 1:1, 1:2, 2:1, 1:3, 3:1, etc.

Number of anchor boxes: The number of anchor boxes per feature map location affects the model's ability to capture objects of different scales and aspect ratios. Typically, multiple anchor boxes are defined at each location. The total number of anchor boxes depends on the number of feature maps and aspect ratios used.

Confidence threshold: During inference, the model assigns confidence scores to each detected object. The confidence threshold determines the minimum score required for an object to be considered a valid detection. Adjusting this threshold affects the trade-off between precision and recall.

Non-maximum suppression (NMS) threshold: To eliminate duplicate detections, NMS is applied. It removes highly overlapping bounding boxes by considering their confidence scores. The NMS threshold defines the overlap threshold at which boxes are considered duplicates and only the one with the highest score is retained.

Loss functions: SSD uses several loss functions to train the model, including localization loss (e.g., smooth L1 loss) and classification loss (e.g., cross-entropy loss). The weights assigned to these losses can be adjusted to balance the impact of each loss during training.

The SSD is built on a feed-forward convolutional network that generates a fixed-size collection of bounding boxes and scores for the existence of object class instances inside those boxes. A non-maximum suppression step is then used to produce the final detections. The early layers of the network, which we will refer to as the base network, are based on a common design for high-quality image categorization (cut off before any classification layers) [45] Then, we augment the network with additional structure to produce detections with the following critical characteristics:

A. MULTI-SCALE FEATURE MAPS FOR DETECTION

We add convolutional feature layers to the underlying network's final layer. These layers gradually diminish in size and provide predictions of detections at numerous sizes. Unlike Overfeat [46] and YOLO [47], which work on a single-scale feature map, the convolutional model for predicting detections is distinct for each feature layer in this algorithm.

B. CONVOLUTIONAL PREDICTORS FOR DETECTION

Using a collection of convolutional filters, each additional feature layer (or alternatively an existing feature layer from the base network) may provide a defined set of detection predictions. These are shown above the SSD network architecture seen in Fig. 2. For a feature layer of size m *times* n with p channels, the fundamental ingredient for predicting the parameters of a prospective detection is a 3 × 3 × p tiny kernel that generates either a category score or a shape offset relative to the default box coordinates. At each of the m × n sites where the kernel is applied, an output value is generated. The output values for the bounding box offset are measured relative to the default box position for each feature map point (cf the architecture of YOLO [46] that uses an intermediate fully connected layer instead of a convolutional filter for this step).

C. DEFAULT BOXES AND ASPECT RATIOS

We connect a set of default bounding boxes with each feature map cell for several top-level feature maps. The default boxes tile the feature map in a convolutional fashion, such that the location of each box relative to its associated cell remains constant. At each feature map cell, we anticipate the offsets relative to the default box shapes and the per-class scores that signal the existence of a class instance in each of those boxes. Specifically, we calculate c class scores and the four offsets relative to the original default box shape for each box out of k at a given position. This leads in a total of (c + 4)k filters being applied around each feature map position, producing (c + 4) kmn outputs for a m × n feature map. Please refer to Figure 1 for a representation of default boxes. Our default boxes resemble the anchor boxes used by Faster R-CNN [44]; however, we apply them to several feature maps with varying resolutions. Allowing distinct default box shapes in multiple feature maps allows us to discretize the space of potential output box shapes in an effective manner.

VII. TRAINING

The primary difference between training SSD and training a conventional detector that use region suggestions is that ground truth information must be given to particular outputs in the predefined set of detector outputs. This is also necessary for training in YOLO [46] and the region proposal phase of Faster R-CNN [44] and MultiBox [48]. After determining this assignment, the loss function and back propagation are applied end-to-end. Training also entails selecting the collection of default detection boxes and scales, as well as the hard negative mining and data augmentation techniques.

A. MATCHING STRATEGY

During training, we must identify which default boxes correlate to a detection of ground truth and train the network appropriately. We choose default boxes for each ground truth box that differ in location, aspect ratio, and scale. We begin by matching each ground truth box to the default box with the greatest jaccard overlap (similar to MultiBox [48]). In contrast to MultiBox, we match default boxes to any ground truth with jaccard overlap above a threshold (0.5). This simplifies the learning issue by letting the network to predict high scores for numerous overlapping default boxes,

as opposed to having it to choose the one with the greatest overlap.

B. TRAINING OBJECTIVE

The SSD training goal is based on the MultiBox objective [48], [49], however, it is expanded to cover several object types. Let $x_{ij}^{p} = \{1, 0\}$ be the indication for matching the i-th default box to the j-th ground truth box of the category p. We can have $\sum_i x_{ij}^{p} \geq 1$ in the above matching technique. The total objective loss function is a weighted average of the localization and confidence losses (loc, conf):

$$L(x, c, l, g) = \frac{1}{N}\left(L_{\text{conf}}(x, c) + \alpha L_{loc}(x, l, g)\right) \quad (4)$$

in where N is the total number of selects. Whenever N is zero, the loss should be zero regardless of whether or not it is raining. The localization error is defined as the discrepancy [47] between the expected (l) and observed (r) box parameters (g). The default bounding box (d) has been customized by setting offsets for its center (cx; cy), width (w), and height (h), much as Faster R-CNN [44] (h).

$$\begin{aligned} L_{loc}(x, l, g) &= \sum_{i \in \text{Pos}}^{N} \sum_{m \in \{cx, c\ y, w, h\}} x_{ij}^{k}\, \text{smooth}_{\text{L1}}\left(l_i^m - \hat{g}_j^m\right) \\ \hat{g}_j^{cx} &= \left(g_j^{cx} - d_i^{cx}\right)/d_i^w \hat{g}_j^{cy} = \left(g_j^{cy} - d_i^{cy}\right)/d_i^h \\ \hat{g}_j^w &= \log\left(\frac{g_j^w}{d_i^w}\right) \quad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right) \end{aligned} \quad (5)$$

VIII. RESULTS AND DISCUSSION

In our experiment, we used 3 folds cross-validation. In folds 3 and 2, we both achieved the average accuracy of 94.78% and 94.% respectively. Data from fold 3 is used for testing, whereas data from folds 1 and 2 is used for training. Data from fold 2 is used for testing, whereas data from fold 1 and fold 3 is used for training. In fold 1, the average accuracy is substantially lower, at 91.62%. The total average accuracy, however, was 93.68. Table 1 contains all of this information which is done in RGB passage.

TABLE 1. shows the results of accuracies in different folds using SSD-CNN in RGB passage.

Training	Testing	Accuracy
Fold1 , Fold 2	Fold 3	94.78
Fold 1, Fold 3	Fold 2	94.65
Fold 2, Fold 3	Fold 1	91.62
	Average accuracy	93.68%

The upgraded version of our suggested gesture recognition system now has higher average accuracy, and, more significantly, this technique is substantially different from previous hand-crafted methods. On the SKIG dataset, a comparison of the RGB Channel's categorization accuracy (percentage). Figure 5 and 8 depict the feature representation as well as graphically presenting training, testing, and classification. Figure 6 shows a graphical representation of

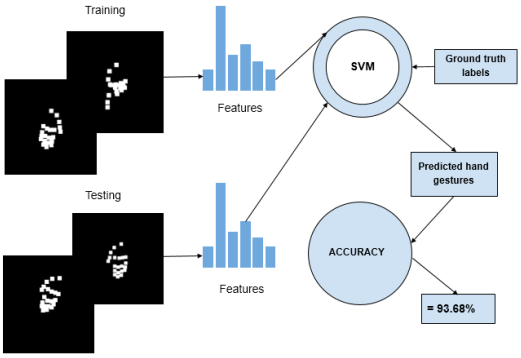

FIGURE 6. Training, testing, and classification.

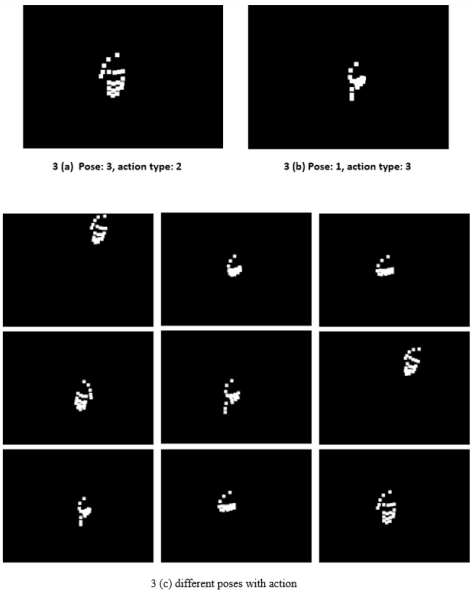

FIGURE 7. Feature representations with SSD-CNN (a) Pose: 3, action type: 2, 3 (b) Pose: 1, action type: 3 and 3 (c) different poses with action.

classification accuracies by comparison to the SKIG dataset. Figure 9, 10, and 11 shows graphical representation of classification accuracies by comparison on DCOG dataset in 0 degree, 45 degree and 60 degree respectively. In table 3 we have compared our proposed method with the other state-of-the-art deep learning and machine learning methods such as YOLO, RGGP, HOG, HOF, HMHI, and DBN in RGB passage, Depth passage, and RGB – D conjunction. We recognize hand gestures with higher accuracy of 93.68% in RGB passage, 83.45% at depth passage, and 90.61% in RGB-D conjunction on the SKIG dataset compared to the state-of-the-art. In the context of our own created Different Camera Orientation Gesture (DCOG) dataset we got a higher accuracy of 92.78% in RGB passage, 79.55% in the depth passage, and 88.56% in RGB-D conjunction for the gestures collected in 0-degree. These are presented in Table 2.

Furthermore, In the context of our own created Different Camera Orientation Gesture (DCOG) dataset we got an accuracy of 89.62% in RGB passage, 80.55% in the depth

TABLE 2. Shows the results of a comparison of classification accuracies (%) on the DCOG in the 0-degree angle dataset.

Method	SSD-CNN	YOLO	RGGP	HOG	HOF	HMHI	DBN
RGB Passage	92.78	91.63	83.71	80.61	80.42	78.31	82.23
Depth Passage	79.55	78.51	75.15	71.15	66.91	65.31	72.81
RGB - D conjunction	88.56	86.53	85.21	82.41	81.05	79.51	83.71

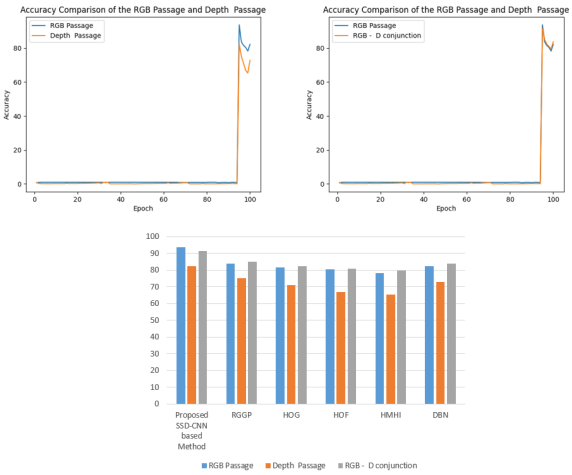

FIGURE 8. Graphical representation by comparison of classification accuracy on DCOG dataset (0-degree angle).

TABLE 3. Shows the results of a comparison of classification accuracies (%) on the SKIG dataset.

Method	SSD-CNN	RGGP	HOG	HOF	HMHI	DBN
RGB Passage	93.68	85.63	82.73	82.56	79.41	84.14
Depth Passage	83.45	77.14	73.14	68.95	66.32	74.83
RGB - D conjunction	90.61	87.26	84.45	83.04	81.53	85.71

passage, and 88.62% in RGB-D conjunction for the gestures collected in a 45-degree angle. These are presented in Table 4.

In addition, with our DCOG dataset, we got an accuracy of 87.56% in RGB passage, 79.65% in the depth passage, and 85.67% in RGB-D conjunction for the gestures collected at a 45-degree angle. These are presented in Table 5.

All of these algorithms were applied to RGB dataset alone, depth dataset only, and RGB-D feature concatenation to produce the final findings in Tables 2,3,4, and 5. Figure 4 shows the Graphical representation by comparison of classification accuracy on SKIG dataset. However, figure 7,8, and 9 shows Graphical representation by comparison of classification accuracy on DCOG dataset in 0-degree, 45-degree, and 60-degree respectively.

All these comparisons based on the data collected in different camera orientations or at different viewpoints still

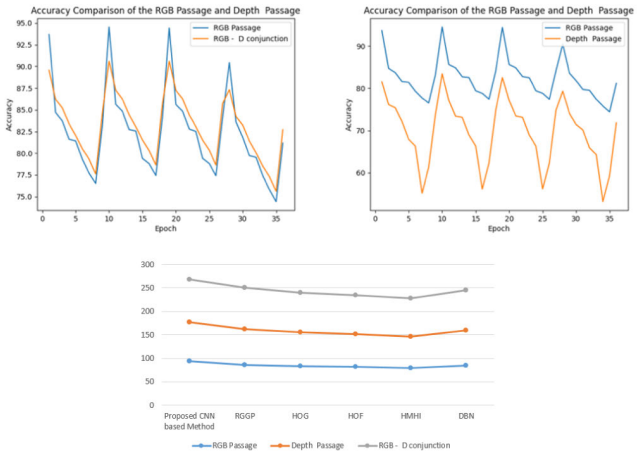

FIGURE 9. Graphical representation by comparison of classification accuracy on SKIG dataset.

TABLE 4. With SSD-CNN in DCOG dataset in 45-degree angle: Comparison with other algorithms.

Method	SSD-CNN	YOLO	RGGP	HOG	HOF	HMHI	DBN
RGB Passage	89.62	87.41	81.61	77.78	78.54	77.42	81.15
Depth Passage	80.55	78.45	74.13	70.13	64.91	63.34	71.81
RGB - D conjunction	88.61	87.67	83.24	80.41	80.25	78.53	81.73

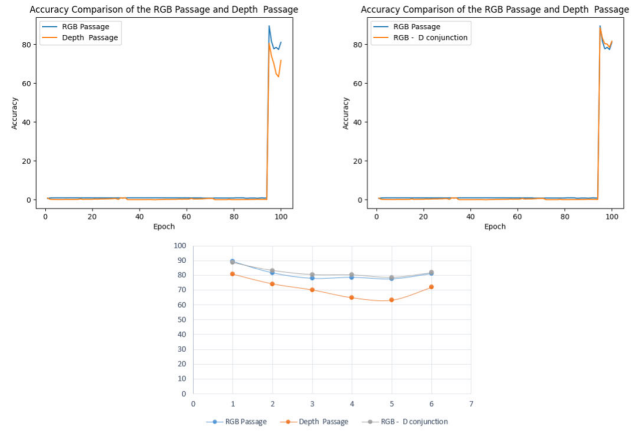

FIGURE 10. Graphical representation by comparison of classification accuracy on DCOG dataset (45-degree angle).

show that the accuracy never decreases much in our proposed SSD-CNN-based algorithm.

On SKIG dataset the combined RGB and depth data may also be used to train features for RGB-D fusion in DBN. We can see from these findings that our SSD-CNN approach greatly surpasses the most current handcrafted and deep learning algorithms on both datasets, due to the higher performance of our method's simultaneous description and fusion of RGB and depth channels. As a result of the feature learning mechanism's implicit supervised nature, the

TABLE 5. With SSD-CNN in DCOG dataset in 60-degree angle: Comparison with other algorithms.

Method	SSD-CNN	YOLO	RGGP	HOG	HOF	HMHI	DBN
RGB Passage	87.56	85.51	80.33	75.58	76.60	73.55	78.45
Depth Passage	79.65	78.20	72.77	69.44	62.67	61.65	70.65
RGB - D conjunc-tion	85.67	84.35	81.34	78.48	78.66	76.66	80.53

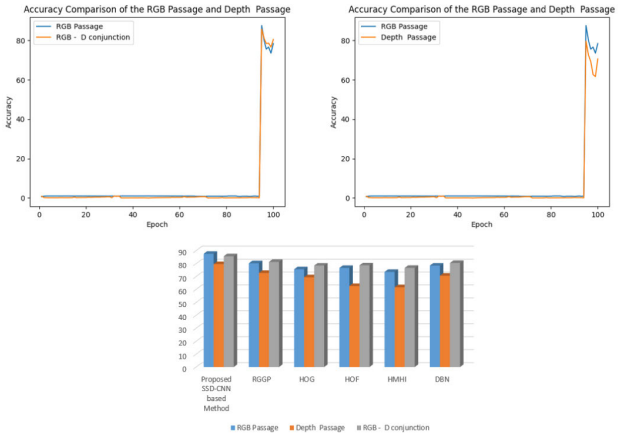

FIGURE 11. Graphical representation by comparison of classification accuracy on DCOG dataset (60-degree angle).

features are better able to discriminate. To further highlight our method's superiority, we may purposefully compare it against the whole system presented using the SKIG dataset. This dataset is fed into our SSD-CNN, which is then used to train our SVM classifier. Given that their approach is far more complicated and uses advanced body joints and skeleton models, our method's performance increase on raw video data is substantial.

IX. CONCLUSION

Our proposed approach begins with RGB videos and depth sequences from the Kinect sensor and then tracks the hand using SSD-CNN. The dilatation is used to improve the quality of the gestures in the image masks. For each video, we only received one trajectory. These are the features that we fed into SVM. Hand movements were eventually detected using the SVM classification method. On the SKIG dataset, we recognise hand gestures with 93.68, 83.45, and 90.61 percent accuracy on RGB passage, Depth passage, and RGB-D conjunction, respectively, compared to the state-of-the-art. Our method attempts to develop a better vision-based hand gesture recognition system capable of providing a noble solution to the issue. We enhanced the overall accuracy of gesture recognition in our proposed technique by fusing RGB and depth information. We built our own dataset, which we also evaluated using our suggested technique. We also want to utilise online gesture videos to benefit the computer vision arena, where the majority of videos are still in RGB. Our future goal is to use robot vision in combination with gesture recognition to enhance our ability to monitor crowds. Additionally, we want to create gesture-based monitoring systems for constrained and densely populated spaces.

REFERENCES

[1] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li, "A review on human activity recognition using vision-based method," *J. Healthcare Eng.*, vol. 2017, pp. 1–31, Jul. 2017.

[2] F. A. Farid, N. Hashim, and J. Abdullah, "Vision-based hand gesture recognition from RGB video data using SVM," in *Proc. Int. Workshop Adv. Image Technol. (IWAIT)*, Mar. 2019, pp. 265–268.

[3] F. A. Farid, N. Hashim, and J. Abdullah, "Vision based gesture recognition from RGB video frames using morphological image processing techniques," *Int. J. Adv. Sci. Technol.*, vol. 28, no. 13, pp. 321–332, 2019.

[4] F. A. Farid, N. Hashim, J. Abdullah, M. R. Bhuiyan, W. N. S. M. Isa, J. Uddin, M. A. Haque, and M. N. Husen, "A structured and methodological review on vision-based hand gesture recognition system," *J. Imag.*, vol. 8, no. 6, p. 153, May 2022.

[5] Z. Lu, S. Qin, X. Li, L. Li, and D. Zhang, "One-shot learning hand gesture recognition based on modified 3D convolutional neural networks," *Mach. Vis. Appl.*, vol. 30, pp. 1157–1180, 2019.

[6] Y. Zhang, L. Shi, Y. Wu, K. Cheng, J. Cheng, and H. Lu, "Gesture recognition based on deep deformable 3D convolutional neural networks," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107416.

[7] M. R. Bhuiyan, D. J. Abdullah, D. N. Hashim, F. A. Farid, D. J. Uddin, N. Abdullah, and D. M. A. Samsudin, "Crowd density estimation using deep learning for Hajj pilgrimage video analytics," *FResearch*, vol. 10, p. 1190, Nov. 2021.

[8] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed, R. Damaševičius, R. Maskeličnas, and K. H. Abdulkareem, "Real-time hand gesture recognition based on deep learning YOLOV3 model," *Appl. Sci.*, vol. 11, no. 9, p. 4164, May 2021.

[9] H. Wang, "Two stage continuous gesture recognition based on deep learning," *Electronics*, vol. 10, no. 5, p. 534, Feb. 2021.

[10] M. S. Islam, S. Sultana, F. A. Farid, M. N. Islam, M. Rashid, B. S. Bari, N. Hashim, and M. N. Husen, "Multimodal hybrid deep learning approach to detect tomato leaf disease using attention based dilated convolution feature extractor with logistic regression classification," *Sensors*, vol. 22, no. 16, p. 6079, Aug. 2022.

[11] N. Ageishi, F. Tomohide, and A. B. Abdallah, "Real-time hand-gesture recognition based on deep neural network," in *Proc. InSHS Web Conf.*, vol. 102, 2022, p. 04009.

[12] J.-H. Sun, T.-T. Ji, S.-B. Zhang, J.-K. Yang, and G.-R. Ji, "Research on the hand gesture recognition based on deep learning," in *Proc. 12th Int. Symp. Antennas, Propag. EM Theory (ISAPE)*, Dec. 2018, pp. 1–4.

[13] M. R. Bhuiyan, J. Abdullah, N. Hashim, F. A. Farid, M. A. Samsudin, N. Abdullah, and J. Uddin, "Hajj pilgrimage video analytics using CNN," *Bull. Electr. Eng. Informat.*, vol. 10, no. 5, pp. 2598–2606, Oct. 2021.

[14] N. Alnaim, "Hand gesture recognition using deep learning neural networks," Ph.D thesis, Brunel University London, 2020.

[15] M. Oudah, A. Al-Naji, and J. Chahl, "Elderly care based on hand gestures using kinect sensor," *Computers*, vol. 10, no. 5, 2021. [Online]. Available: https://doi.org/10.3390/computers10010005

[16] P. Joseph, "Recent trends and technologies in hand gesture recognition," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, May 2017.

[17] Y. Zhang, B. Liu, and Z. Liu, "Recognizing hand gestures with pressure-sensor-based motion sensing," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 6, pp. 1425–1436, Dec. 2019.

[18] Y. Min, Y. Zhang, X. Chai, and X. Chen, "An efficient PointLSTM for point clouds based gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5760–5769.

[19] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, T. S. Alrayes, H. Mathkour, and M. A. Mekhtiche, "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," *IEEE Access*, vol. 8, pp. 192527–192542, 2020.

[20] P. S. Neethu, R. Suguna, and D. Sathish, "An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks," *Soft Comput.*, vol. 24, no. 20, pp. 15239–15248, Oct. 2020.

[21] Z. Liu, C. Zhang, and Y. Tian, "3D-based deep convolutional neural network for action recognition with depth sequences," *Image Vis. Comput.*, vol. 55, pp. 93–100, Nov. 2016.

[22] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4597–4605.

[23] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Daps: Deep action proposals for action understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 768–784.

[24] E. Mansimov, N. Srivastava, and R. Salakhutdinov, "Initialization strategies of spatio-temporal convolutional neural networks," 2015, *arXiv:1503.07274*.

[25] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Proc. Int. Workshop Hum. Behav. Understand.*, 2011, pp. 29–39.

[26] Y. Ye and Y. Tian, "Embedding sequential information into spatiotemporal features for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 1110–1118.

[27] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2016, pp. 1933–1941.

[28] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1049–1058.

[29] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.

[30] N. Neverova, C. Wolf, G. W. Taylo, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 474–490.

[31] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4305–4314.

[32] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.

[33] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2718–2726.

[34] Z. Xu, L. Zhu, Y. Yang, and A. G. Hauptmann, "UTS-CMU at THUMOS 2015. THUMOS challenge," in *Proc. CVPR'15 Int. Workshop Competition Action Recognit. Large Number Classes*, 2015, p. 3.

[35] G. Gkioxari and J. Malik, "Finding action tubes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 759–768.

[36] H. J. Escalante, E. F. Morales, and H. J. Sucar, "A naive Bayes baseline for early gesture recognition," *Pattern Recognit. Lett.*, vol. 73, pp. 91–99, Apr. 2016.

[37] X. Xu, T. M. Hospedales, and S. Gong, "Multi-task zero-shot action recognition with prioritised data augmentation," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 343–359.

[38] A. Montes, A. Salvador, S. Pascual, and X. Giro-i-Nieto, "Temporal activity detection in untrimmed videos with recurrent neural networks," 2016, *arXiv:1608.08128*.

[39] K. Nasrollahi, S. Escalera, P. Rasti, G. Anbarjafari, X. Baro, H. J. Escalante, and T. B. Moeslund, "Deep learning based super-resolution for improved action recognition," in *Proc. Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Nov. 2015, pp. 67–72.

[40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[41] E. Akleman, "Deep learning," *Computer*, vol. 53, no. 9, p. 17, Sep. 2020.

[42] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G. Z. Yang, "Deep learning for health informatics," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 4–21, Dec. 2016.

[43] M. A. Nielsen, *Neural Networks and Deep Learning*, vol. 25. San Francisco, CA, USA: Determination press, 2015.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–9.

[45] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*.

[46] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[47] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[48] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2155–2162.

[49] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, high-quality object detection," 2014, *arXiv:1412.1441*.

FAHMID AL FARID (Member, IEEE) received the B.S. degree in computer science and engineering from the University of Chittagong, Bangladesh, in 2010, the M.S. degree from the Faculty of Computer Science and Electrical Engineering, University of Ulsan (UOU), South Korea, in 2015, and the Ph.D. (by Research) degree in information technology from Multimedia University (MMU), Cyberjaya, Malaysia. From 2013 to 2014, he was a Research Assistant with the Embedded System Laboratory, UOU. In 2015, he was a Research Assistant with the Ubiquitous Computing Technology Research Institute (UTRI), Sungkyunkwan University, South Korea. He is currently a Postdoctoral Scientist with the Faculty of Engineering, MMU. His current research interests include artificial intelligence, algorithm design, computer vision, human–computer interaction, image and video analysis, power generation, and green technology. He received the Korean BK21 PLUS Scholarship supported by the Korean Government for the M.S. degree, from 2012 to 2014. He also received an ICT Fellowship from Bangladesh Government, in 2014.

NORAMIZA HASHIM (Member, IEEE) received the Diplôme d'Ingénieur (M.Sc.) degree in engineering and the D.E.A. (master's) degree in research from the Higher Institute for Advanced Technologies of Saint-Etienne (ISTASE), Université Jean Monnet, France, in 2002, and the joint Ph.D. degree in information technology from Multimedia University (MMU), Malaysia, and Université de La Rochelle, France, in 2008. She is currently a Lecturer with the Faculty of Computing and Informatics, MMU. Her research interests include digital image processing and object recognition.

JUNAIDI BIN ABDULLAH (Member, IEEE) received the B.Eng. degree (Hons.) in engineering from the University of Bristol, U.K., and the Ph.D. degree in computer science (computer vision and augmented reality) from the University of Southampton, U.K., in 2005. He is currently an Associate Professor with the Faculty of Computing and Informatics (FCI), Multimedia University (MMU), Malaysia, where he is also the Dean. He is the Chairperson of the Assistive Technology Special Interest Group. He has published more than 60 internationally multi-disciplinary refereed conference papers, journal articles, and books. His research interests include augmented reality, image and video processing, and computer vision.

MD. ROMAN BHUIYAN received the master's degree in software engineering from the Faculty of Computer Science and Engineering, FTMS College, Malaysia, in collaboration with Leeds Beckett University, U.K., in 2017, and the Ph.D. (by Research) degree in information technology from Multimedia University, Cyberjaya, Malaysia, in 2022. He is currently a Postdoctoral Fellow with Fraunhofer IGD, Rostock, Germany. His current research interests include artificial intelligence, algorithm design, computer vision, deep learning, machine learning, image analysis, and video analysis.

MAGZHAN KAIRANBAY received the bachelor's degree in software engineering and the master's degree in computer science from International IT University, Kazakhstan, in 2013, and the Ph.D. degree from Multimedia University, Malaysia, in 2020, with a focus on large-scale aesthetic evaluation of photographs using deep learning. He was a Researcher with ABY Applied Science Company, where he did research on number plate recognition systems. He is currently a Senior Lecturer with Suleiman Demirel University and a Senior Machine Learning Engineer with Socar, a Korean Car Sharing Company.

ZULFADZLI YUSOFF (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electronics and communications from the University of York, U.K., in 1999, and the Ph.D. degree in optical communications from the University of Southampton, U.K., in 2004. From 2001 to 2004, he was a Researcher with the Optoelectronics Research Centre (ORC), University of Southampton, where he studied applications of nonlinear effects in photonic crystal fibers in optical communications. Since 1999, he has been a member of the academic staff at the Faculty of Engineering, Multimedia University (MMU). He is currently an Associate Professor and the Deputy Director of the Research Management Centre (RMC). He specializes in photonics engineering. He had taught numerous bachelor's and master's subjects in the field of photonics and communications. He has also conducted training and short courses for various companies in the area of telecommunications, specifically related to photonics technologies. To date, he has published more than 100 research papers in international journals and conferences with more than 1000 citations. He was awarded the Endeavour Postdoctoral Research Fellowship, in 2009, to spend six months at The University of Sydney, Australia, to study nonlinear effects in photonic nanowires. He was a recipient of a number of research grants from various sources worth more than RM 10 million.

HEZERUL ABDUL KARIM (Senior Member, IEEE) received the B.Eng. degree in electronics with communications from the University of Wales Swansea, U.K., in 1998, the M.Eng. degree in science from Multimedia University (MMU), Malaysia, in 2003, and the Ph.D. degree from the Center for Communication Systems Research (CCSR), University of Surrey, U.K., in 2008. He is currently a Professor with MMU, where he is also the Deputy Dean of Student Affairs and Alumni with the Faculty of Engineering. He has been teaching multimedia and computing engineering subjects. His research interests include telemetry, 2D/3D image/video coding and transmission, error resilience, algorithms, bioinformatics, deep learning, machine learning, and neural networks.

SARINA MANSOR received the B.Eng. degree (Hons.) in electronics and electrical engineering, majoring in computer science, from University College London, in 1998, the M.Eng.Sc. degree from Multimedia University (MMU), in 2002, and the D.Phil. degree in engineering science from the University of Oxford, U.K., in 2009. She is currently a Senior Lecturer with the Faculty of Engineering, Multimedia University (MMU), Malaysia. She is also a Programme Coordinator for B.Eng. studies. Her research interests include signal and image analysis, medical imaging, computer vision, machine learning, and the Internet of Things.

MD. TANJIL SARKER received the B.Sc. degree in electrical and electronics engineering (EEE) and the Master of Business Administration (M.B.A.) degree in human resource management (HRM) from Bangladesh University, Dhaka, Bangladesh, in 2013 and 2015, respectively, the M.Sc. degree in computer science and engineering (CSE) from Jagannath University, Dhaka, in 2018, and the Ph.D. degree in engineering from the Faculty of Engineering, Multimedia University (MMU), Malaysia, in 2022. He is currently a Postdoctoral Research Fellow with the Faculty of Engineering, MMU. He has conducted many research works in relevant fields. His research interests include system identification, signal processing and control, renewable energy, power system analysis, and high-voltage engineering. He was a Graduate Student Member of the IEEE Student Branch of the Malaysia Section, in 2021.

GOBBI RAMASAMY (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the University of Technology, Malaysia, the master's degree in technology management from the National University of Malaysia, and the Ph.D. degree in torque control of switched reluctance motors from Multimedia University (MMU), Malaysia. He has been associated with technical education for more than ten years. He was a Research and Development Engineer in an electronics company before becoming a Lecturer in electrical and electronics engineering. He is currently an Associate Professor with the Faculty of Engineering, MMU. He is the Project Leader and a member of various government research projects related to electric motors and drive systems. He is a Consultant providing solutions for many problems associated with electric motors and drive systems for various industries. He has supervised many research projects on power electronics, variable-speed drives, automation, and domestic electrical installations.