Advancements in Real Time Human Activity Recognition via Innovative Fusion of 3DCNN and Convlstm Models

[1]Roopa R and [2]Humera Khanam M
[1,2] Department of CSE, S V University College of Engineering, S V University, Tirupati, Andhra Pradesh, India.
[1]roopa509@gmail.com, [2] humera.svec@gmail.com

Correspondence should be addressed to Roopa R: roopa509@gmail.com

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404071
Received 10 March 2023; Revised from 02 April 2024; Accepted 20 June 2024
Available online 05 July 2024.

Abstract – Object detection (OD) is a computer vision procedure for locating objects in digital images. Our study examines the crucial need for robust OD algorithms in human activity recognition, a vital domain spanning human-computer interaction, sports analysis, and surveillance. Nowadays, three-dimensional convolutional neural networks (3DCNNs) are a standard method for recognizing human activity. Utilizing recent advances in Deep Learning (DL), we present a novel framework designed to create a fusion model that enhances conventional methods at integrates three-dimensional convolutional neural networks (3DCNNs) with Convolutional Long-Short-Term Memory (ConvLSTM) layers. Our proposed model focuses on utilizing the spatiotemporal features innately present in video streams. An important aspect often missed in existing OD methods. We assess the efficacy of our proposed architecture employing the UCF-50 dataset, which is well-known for its different range of human activities. In addition to designing a novel deep-learning architecture, we used data augmentation techniques that expand the dataset, improve model robustness, reduce overfitting, extend dataset size, and enhance performance on imbalanced data. The proposed model demonstrated outstanding performance through comprehensive experimentation, achieving an impressive accuracy of 98.11% in classifying human activity. Furthermore, when benchmarked against state-of-the-art methods, our system provides adequate accuracy and class average for 50 activity categories.

Keywords – Object Detection, Human Activity Recognization, Deep Learning, 3DCNN, ConvLSTM.

I. INTRODUCTION

Object detection (OD) is a computer vision strategy that allows machines to detect and identify objects of interest in images or videos. The system will also return a confidence level that indicates its level of assurance about the accuracy of a forecast. The task involves identifying and classifying object positions and boundaries, which is crucial for vision recognition, image classification, and retrieval. OD [1] also benefits video surveillance or image recovery systems applications. The primary goal of OD in computer vision is to locate instances of visual objects such as people, cars, houses, and animals [2] in digital images and determine their locations within the image. OD plays a significant role in computer vision by identifying objects in images in particular classes [3]. OD in images is complicated due to objects' extensive potential locations and sizes and each detection provides important positional information through careful exploration is required. The position, scale, bounding box, or segmentation mask of the object are all crucial information that could be included with any detection. In alternative situations, the more precise posture information contains the parameters of a linear or non-linear transformation. For instance, a face detector can determine the bounding box of the face and the positions of the mouth, nose, and eyes.

An alternative way to characterize the pose would be to use a three-dimensional transformation to indicate the object's location on the camera. Nevertheless, it might be challenging to develop models that capture a large amount of diversity in images. OD is necessary for further computer vision tasks such as object tracking [4], image captioning [5], and instance segmentation [6]. OD advancements have been made in the past few years due to the quick development of Artificial Intelligence (AI) systems. Multiple real-world applications, such as autonomous driving [7], robotics [8], medical image analysis, and video surveillance [9], heavily depend on OD. Images of specific classes of objects are inconsistent. The actual imaging method is one cause of variation. Even in a static image, fluctuations in illumination, camera position, and digitization artifacts can cause noticeable differences in the appearance of an image. Additionally assuming no change in the imaging technique, the second source of variance arises from the intrinsic appearance variety of objects within a class. People

differ in terms of their shapes and clothing choices, for instance, and the handwritten number 7 can be written in various ways, such as with varied slants, stroke widths, and a combination of these characteristics. Creating computationally efficient invariant identification techniques for these alterations is the problematic part [1]. The primary tasks in this field include Referring Expression Comprehension (REC) [10], [11], [12], [13] and Open Vocabulary Object Detection (OVD) [14], [15], [16], [17]. OD, a vital aspect of computer vision, has multiple practical applications.

Consequently, there's a significant drive to improve detection models, specifically in dealing with a broad range of objects. OD techniques are often classified into two primary categories: generative and discriminative [18]. Generative methods [19] involve constructing a probability model containing an object's various and a model for how the object appears in an image that provides a particular pose. These methods also contain a model for background images that don't include the detected object. The parameters of these models are specified from training data, and decisions are made based on the probabilities computed. poses In contrast, discriminative methods concentrate on building classifiers that differentiate between images containing instances of target objects and those that do not. These classifiers are developed to reduce errors in the training data, often with adjustments to control overfitting. Additional distinctions among detection algorithms include the instruments utilized for image scanning, the kind of image representation incorporated into the models, and the volume of training data required. Acknowledging the actions of humans in images or videos is a complex task involving looking at different factors, like what objects are present, how they're positioned, how people are moving, their postures, and even when they're resting. Understanding what people are doing requires the ability to accurately recognize human actions in images or videos, independent of the involvement of objects.

The system has essential stages, including identifying human movements such as cycling, jogging, walking, sleeping, standing, playing, sitting, running, handshaking, and more, and identifying objects [20]. Identifying human activity involves more than just individuals because it involves the things around them. For example, knowing if someone is wearing specific gear while jogging or carrying something like a water bottle is helpful. It helps us understand the situation. Scientists and researchers have spent a lot of effort studying how to identify these actions and objects in videos. Most researchers have delved into human activity and OD using video recognition techniques, focusing on the UCF-50 dataset [21]. This dataset incorporates a diverse set of 50 action categories, representing YouTube videos captured under realistic conditions. We, too, utilized the UCF-50 dataset to assess and precisely classify the actions in our study and to evaluate and accurately categorize the human activity in our work.

The following is the study's fundamental contribution:

- We used rigorous preprocessing approaches and augmentation strategies to diversify and enrich our video dataset to overcome data restrictions and improve model performance and generalization.
- We combined the 3DCNN and LSTM architectures in a novel way to create a new model that is better at capturing temporal and spatial characteristics, which improves the model's ability to analyze complicated video data.
- Our study includes a thorough analysis of every class in the dataset, where our suggested model consistently showed excellent accuracy, confirming its effectiveness in accurately classifying various activity types.
- Compared with previous studies, our proposed approach greatly improves the accuracy of classifying human activity from video data, representing a noteworthy development in the activity recognition field.

This paper is structured into multiple sections. In Section II, we will first review the current approaches to real-time OD and classifying human activities. We will also pinpoint some fundamental limitations of OD and the UCF-50 dataset. Then, in Section III, we will summarize our methodological statement and used materials for this purpose. Section IV briefly discusses the experimental outcomes for our proposed model implementation. Finally, Section VI will present the results and findings of our research.

II. LITERATURE REVIEW

Over the years, several scholars have conducted numerous studies to improve efficacy in OD and classification. Here, we present a few noteworthy and current studies in this field.

Yun et al. [22] introduced a method for real-time estimation of occupant metabolic rates (MET) to enhance indoor thermal comfort by combining a pose-based activity classification model with an OD model, and different METs can be accurately evaluated. The custom OD model performed a real-time classification accuracy of 89%, with a 100% accuracy when evaluating over 15-second intervals. The MET estimation algorithm demonstrated an 83% real-time accuracy for six METs and 99% accuracy over 15-second intervals. Hu et al. [23] presented a novel approach for detecting hidden human targets by utilizing physiological characteristics and their spatiotemporal interdependencies. Experimental outcomes on a homemade hidden human object dataset indicated notable improvements over existing methods, performing detection accuracies of 64%, 44%, and 54% for indoor, outdoor, and overall scenarios, respectively. These accuracies surpassed YOLO v4 and traditional feature-based models (HOG, LBP, Haar) by at least 22%. Every module in the suggested strategy was shown to be effective through ablation experiments. Promising outcomes indicate potential applications in public security inquiries, military rescue, and other fields. The handcrafted dataset will be made accessible to the public upon acceptance, thereby advancing this field's study.

Su et al. [24] introduced a novel framework for Human Activity Recognition (HAR) using Graph Neural Networks (GNN) to examine human-object interactions and enhance the identification of Activities of Daily Living (ADL). Compared to prior methods, which frequently depended on object classification and posture estimates from camera frames, authors considered the relationships between ADL and human-object interactions. The framework deduced various actions and their

relevant environmental objects by automatically encoding these relationships. The results demonstrated better performance than traditional feed-forward neural networks, with an ADL classification accuracy of 0.88, using the Toyota Smart Home dataset for evaluation. Moreover, object-inference performance was improved by adding encoded information from relational data, resulting in an accuracy increase from 0.71 to 0.77. This work showed that GNNs can be beneficial for identifying everyday activities and showed how explicit examination of human-object interactions can lead to more reliable HAR systems.

Nabiei et al. [25] studied real-time human activity recognition employing hidden Markov models (HMMs) and sensorized objects while focusing on making tea for stroke patients who have apraxia or action disorganization syndrome (AADS). The proposed method used parallel detectors with inputs from sensors on objects and hand coordinates, each in charge of identifying a sub-goal in the tea-making process. Experiments revealed different error rates: sensorized item detectors had less than 5% inaccuracy, whereas hand coordination depended up to 30% on detectors; however, the system operated in real time.

Suriani et al. [26] presented a method using Histograms of Oriented Gradient (HOG) and Histograms of Oriented Optical Flow (HOOF) to understand human actions in monitored areas. The HOG technique allowed identify critical object features, while HOOF defined object states, mainly during human-object interactions. Their method is acceptable for distinguishing between passive and active objects in the observed space. HOG determined objects based on their special traits, involving a robust foundation for additional analysis. Meanwhile, HOOF captured how object properties differ over time. These features were categorized into activity classes using SVM techniques, allowing the system to learn patterns from the input video data. Through stringent testing across different scenarios, they achieved an 89% accuracy rate with an 11.3% error rate, presenting real-world applications demanding authentic human action recognition.

Hashim et al. [27] proposed a hardware-based activity recognition system to support the independent living of older adults by using a Jetson Nano 2GB, a monitor, and a web camera. The system was developed to detect and recognize the activities of older adults in real time. Two datasets were built for training the YOLO network, with training employing Google Colab. After that, the trained weights were deployed on the Jetson Nano 2GB to be tested in a real-world environment. The hardware implementation and simulation accuracy exceeded 80%, indicating the system's efficacy. The study emphasized how crucial hardware-based activity recognition systems were, especially for helping older people.

After going over the body of research, we identified some constraints connected to OD and the UCF-50 dataset – One fundamental limitation of working with the UCF-50 dataset is the challenge of performing high accuracy. Most researchers who worked on this dataset have needed help to achieve no table levels of accuracy in tOur study investigates researchers concentrate just on a subset of classes within the dataset, which frequently leads to this issue. Moreover, using 3DCNN with ConvLSTM for training deepens this limitation because these advanced models require substantial volumes of high-quality data to generate satisfactory outcomes. Unfortunately, obtaining such data is a hurdle in many real-world applications where such resources are limited or unavailable. This constraint restricts the models' ability to perform excellently in various tasks and situations.

III. MATERIALS AND METHODS

In our study, we're investigating the combination of 3DCNN and ConvLSTM networks for video classification. ConvLSTM networks utilize their capability to retain temporal information, allowing them to grasp spatiotemporal patterns within videos. Meanwhile, 3DCNN networks employ the third dimension to discern features for classification. These networks are extensively used across industries and medical fields for video and image categorization tasks.

Dataset Description

The UCF50 dataset comprised 50 action categories, including YouTube videos that are realistic. The dataset showed many differences in lighting, disorderly backdrops, camera motion, etc. Video content in the same group may share commonalities like the same subject, backdrop, point of view, etc. There were 50 subcategories in the UCF50 dataset, including Baseball Pitch, Basketball Shooting, Bench Press, Biking, Billiards Shot, Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, Playing Guitar, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Lunges, Military Parade, Mixing Batter, Nun chucks, Playing Piano, Pizza Tossing, Pole Vault, Pommel Horse, Pull-Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, Playing Tabla, TaiChi, Tennis Swing, Trampoline Jumping, Playing Violin, Volleyball Spiking, Walking with a Dog, and Yo-Yo [28]. **Fig 1** illustrates the UCF50 dataset sample.

Data Preprocessing and Augmentation

Multiple phases are involved in preparing data for DL algorithms so that it is appropriate for training. The sequential nature of the frames and the requirement for size and scale uniformity make this process especially difficult for video data. Firstly, we extract the frames from the videos and systematically arrange them. Every video is made up of a sequence of frames that are shown over time. We select a certain number of frames from each video and resize them to a standard size, usually (58, 224, 224, 3), corresponding to 58 frames with 224×224 pixel dimensions and 3 RGB channels each. Confirming that all pixel values lower within the range of 0 to 1 is crucial for consistency across the dataset. This normalization step permits the model to interpret the data accurately and prevents any biases introduced by variations in pixel intensity. Furthermore,

data augmentation becomes essential when meeting with limited datasets to improve the model's performance and avoid overfitting. Data augmentation concerns creating variations of existing data by using transformations like RandomCrop, HorizontalClip, VerticalClip, RandomFlip, GaussianBlur, and RandomRotate to each frame. These modifications artificially improve the dataset's diversity, strengthening the model and reducing overfitting. However, despite these efforts, overfitting can still appear, mainly with complicated models like 3D CNNs. Techniques like regularization and dropout layers are typically utilized to address this issue. To keep the model from becoming excessively complicated, dropout layers randomly deactivate some neurons during training, encouraging the network to acquire more resilient features. Regularization works by placing penalties on large weight values. Data preprocessing and augmentation are crucial to preparing video data for DL studies. Standardizing, normalizing, and augmenting the dataset can enhance the model's performance and generalization capabilities, showing more accurate outcomes on unseen data.

Fig 1. Samples of the UCF50 Dataset.

Proposed Method

3DCNN Architecture:

In particular, in medical imaging, the 3DCNN neural network can recognize and classify various moving 2D objects inside and 3D images. As illustrated in **Fig 2**, 3DCNN entails the dataset's 3D convolution operation along three axes (x, y, and z) using a three-dimensional filter. The values in the layer of the three-dimensional filter must be exactly non-negative. The calculation for each place in the layer's 3D convolution map of features is displayed in the equation below:

$$z_{mn}^{abc} = tanh\,(k_{mn}) + \sum_x \sum_{k=0}^{K_{m-1}} \sum_{r=0}^{R_{m-1}} \sum_{s=0}^{S_{m-1}} w_{mnxz}^{krs}\; z(f+k) + (g+r) + (e+s) + (m-1)x \qquad (1)$$

Where w_{mnxz}^{krs} defines the value of the kernel connected to the convolutional feature map in the prior layer,S^i represents the size of the 3D kernel [29]. 3D convolution is made by stacking layers around the center of a cube, with connected convolution maps capturing motion data. However, individually, convolutional kernels can only extract one type of feature.

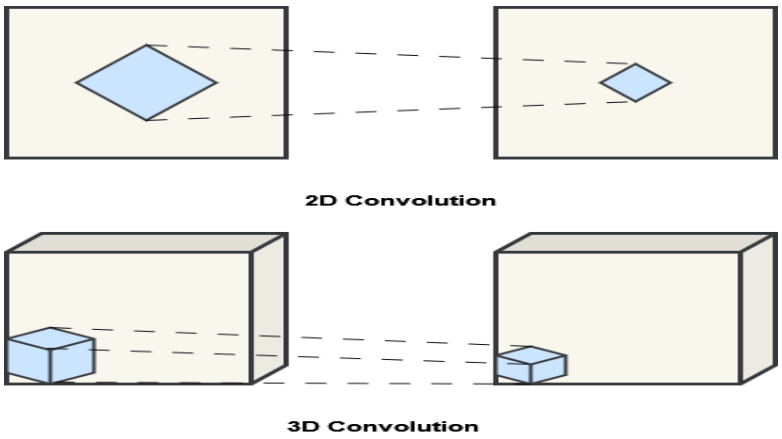

Fig 2. Analysis Of The Differences Between The Mathematical Operations Of 2D And 3D Convolution [30].

In essence, 3DCNN compares Conv2D, the 2D Convolutional Neural Network. Similar to 2D convolution, incorporating numerous convolutional layers can improve 3DCNN performance. Correctly specifying the number of layers, filters for each layer, and filter size is essential in 3DCNN construction. For the pooling size to fit the 3D data, three

dimensions must be included in the network design if pooling is used. A 3D volume space is the shape that a 3DCNN network outputs [30], [31].

ConvLSTM Architecture

The ConvLSTM neural network was designed by incorporating a Convolutional Neural Network (CNN) with an LSTM network. Like the LSTM network, the ConvLSTM network operates as a memory network, performing convolution operations on the relations between layers. The internal structure of a ConvLSTM network is demonstrated in **Fig 3** [32].

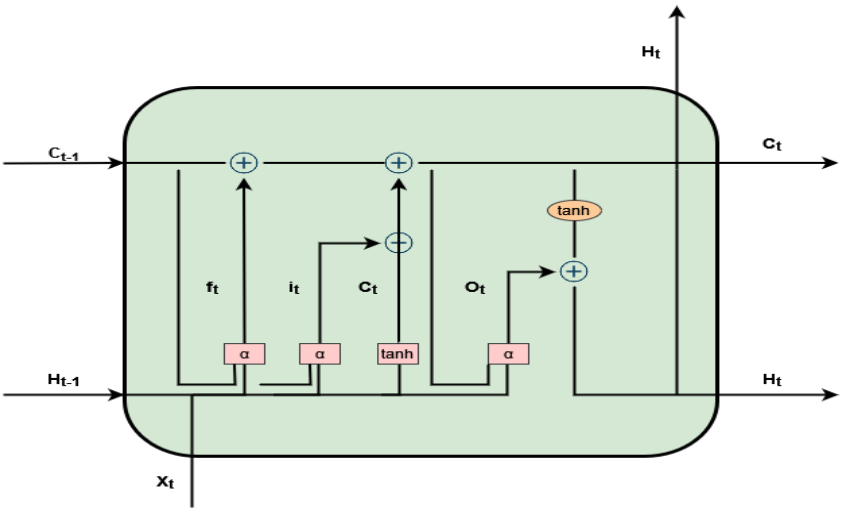

Fig 3. Internal Architecture of ConvLSTM [32].

The ConvLSTM neural network is extensively used for recognizing time-dependent patterns in images and videos due to its capability to capture spatial and temporal relationships. ConvLSTM involves a convolutional operation to the transitions between states and inputs. This architecture allows the network to perceive changes over time, comparable to how it recognizes spatial features in traditional CNNs. A critical factor in ConvLSTM is the size of the transition kernel when we consider the states as representations of moving objects. A smaller transition kernel better captures slower motions, but the network can capture faster motions with a bigger transition kernel. Due to its flexibility in kernel size, ConvLSTM can be applied to various dynamic data problems. Here is a condensed form of the crucial equation:

$$\text{Pq} = \sigma\,(Z_{xi} * Y_t + Z_{hi} * S_{t-1} + Z_{ci} o T_{t-1} + b_i) \tag{2}$$

$$\text{Ft} = \sigma\,(Z_{xf} * Y_t + Z_{hf} * S_{t-1} + Z_{cf} o T_{t-1} + b_f) \tag{3}$$

$$t_t \;=\; \text{ft}\; o\; T_{t-1} + pq\; o\; tanh\;(Z_{xc} * Y_t + Z_{hc} * S_{t-1} + b_f) \tag{4}$$

$$ot \;= \sigma\,(Z_{xo} * Y_t + Z_{ho} * S_t + Z_{co} o T_{t-1} + b_o) \tag{5}$$

$$s_t \;= ot\; tanh(T_t) \tag{6}$$

In this equation:
- pq defines the input gate.
- σ represents the sigmoidal function.
- ∗ indicates the convolution operator.
- ◦ represents the Hadamard (element-wise) product.
- $Z_{xi},\ Z_{hi},\ and\ Z_{ci}$ the input, hidden state, and cell state, respectively, are represented by convolutional kernels.
- Z_t denotes the cell inputs.
- S_{t-1} means the hidden states from the prior time step.
- T_{t-1} defines the cell states from the earlier time step.
- b_i indicates the bias term associated with the input gate.

Proposed 3DCNN + ConvLSTM Architecture

Our presented neural network architecture incorporates Conv3D layers with a ConvLSTM layer and a Conv2D layer. This architecture, known as 3DCNN + ConvLSTM, comprises a few Conv3D layers observed by a single ConvLSTM layer and a single Conv2D layer. These layers are illustrated in **Fig 4**. The following layers in this proposed architecture:

- *Conv3D layers:* These layers process input video data to extract spatiotemporal features. The number of Conv3D layers can be adjusted according to the task complexity. They employ a three-dimensional filter by convolving in three directions (x, y, and z).

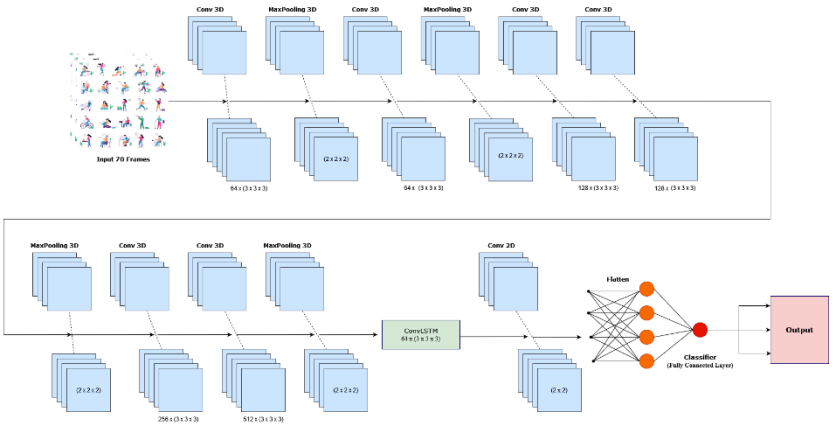

Fig 4. The Architecture of Proposed 3DCNN + ConvLSTM.

- *MaxPooling3D layer***:** This operation decreases 3D data.
- *ConvLSTM layer:* Through processing features from the Conv3D layers, temporal dependencies between frames are captured.
- *Conv2D layer: T*his layer uses the output from the previous layers to conduct the final classification of 2D data convolving.
- *Flatten layer:* It transforms the output matrix into a vector.

This architecture combines Conv3D and ConvLSTM networks using their respective strengths. It incorporates multiple 3D convolutional layers, a single ConvLSTM layer, a single 2D convolutional layer, batch normalization, a flattened layer, and a dense layer. The 3D convolutional component is adjusted from a previous study [29], while the ConvLSTM part is based on another study [33]. The output of Conv3D layers is ensured to be non-negative integers by mathematically determining hyperparameters such as the number of filters and kernel size for the 3D convolutional layers and MaxPooling.

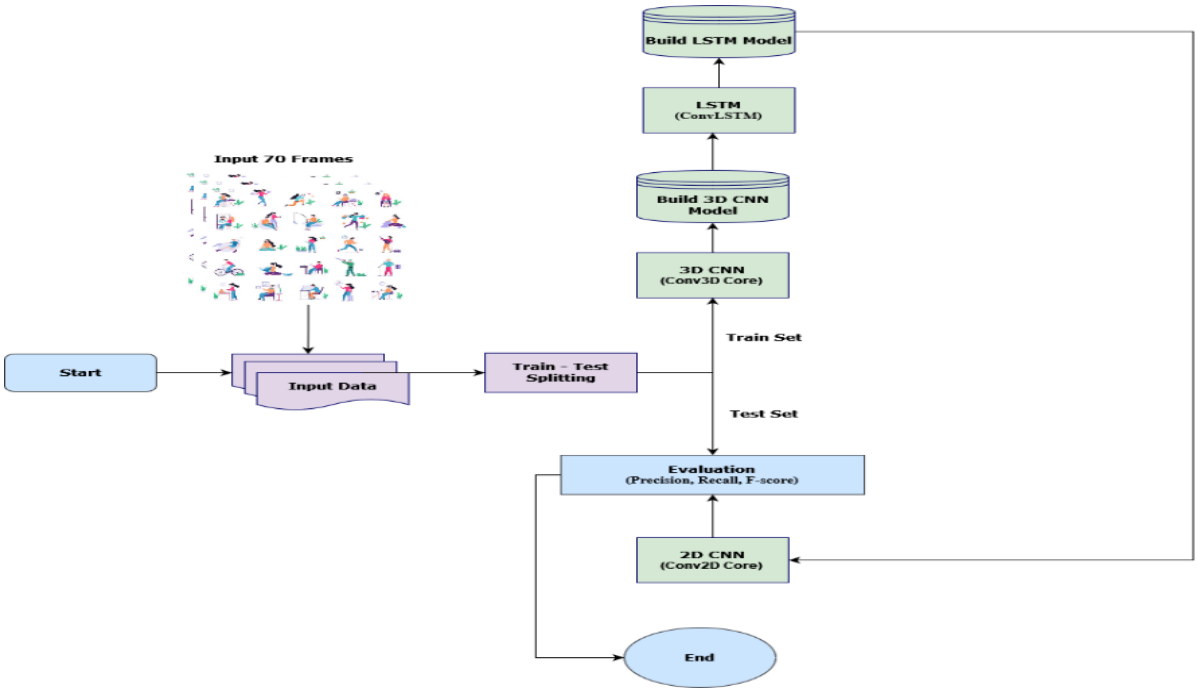

Fig 5. An Illustration Of The Proposed Architecture's Flow.

Fig 5 displays the flowchart for this proposed architecture. The presented architecture in this study incorporates 3DCNN and ConvLSTM models. It combines six 3D convolution layers, four MaxPooling3D layers, one ConvLSTM layer, and a subsequent Conv2D layer.

Algorithm 1 Proposed 3DCNN + ConvLSTM Architecture

1: Input: 3D Volumes (X), Labels (Y)
2: Output: Predicted Labels (ˆY)
3: Initialization: Initialize network parameters
4: Data Preprocessing: Normalize, augment, and split data into training and testing sets
5: Define 3DCNN Backbone: Construct a 3DCNN architecture
6: Define ConvLSTM Layer: Construct a ConvLSTM layer
7: Connect 3DCNN and ConvLSTM: Incorporate the 3DCNN backbone and ConvLSTM layer
8: Compile Model: Describe the loss function and optimizer
9: Training: Train the incorporated model using training data
10: Evaluation: Evaluate model performance on testing data
11: Hyperparameter Tuning: Optimize hyperparameters employing cross-validation
12: Prediction: Utilize the trained model to predict labels for new data
13: Output Results: Show evaluation metrics and visualization of outcomes

The input dimensions (width, height, and channels) are 100 × 100 × 3. The first 3D convolution layer contains 64 filters with a 3 × 3 × 3 kernel size. Every 3D convolution layer is followed by a MaxPooling3D layer with a stride of 2 and a length of 2 × 2 × 2. Two 3 convolution layers with 128 filters are the successive layers, and then there is another MaxPooling3D layer. The last two 3D convolution layers use 256 and 512 filters, respectively. After every MaxPooling3D layer, batch normalization layers are added to aid training. The ConvLSTM network comprises a single 3 × 3 ConvLSTM layer with 64 filters. After this layer, there is a Conv2D layer with 16 filters of size 2 × 2 and a batch normalization layer. Next, the Conv2D layer's output is flattened to turn it into a vector. A final dense layer of a single neuron predicts the input class. The "Adamax" optimization algorithm is employed, and its learning rate is 0.001. Algorithm 1 presents the Proposed 3DCNN + ConvLSTM Architecture algorithm, reducing understanding and implementation complexity.
Table 1 displays the total number of parameters and the number of trainable and non-trainable parameters. DL models were constructed employing Python frameworks like Keras and TensorFlow, while experimental outcomes were obtained using Nvidia CUDA libraries. The input dataset contained images of 100 × 100 pixels with 3 color channels. In each dataset, 70% of the samples were assigned for training, 20% for testing, and the remaining 10% for validation.

Table 1. The Entire Number Of Parameters In Our Proposed Architecture

Parameters of the Presented Architecture	Value
Total number of parameters	2,326,914
Trainable number of parameters	2,325,250
Non-trainable number of parameters	1664

IV. RESULTS AND DISCUSSION

We used a powerful GPU, such as the NVIDIA GeForce RTX or NVIDIA Tesla, and a high-performance workstation with a multi-core CPU to implement human activity recognition using DL on the UCF-50 dataset. Deep neural network training requires a lot of computation, which the GPU helped to speed up. On the other hand, enough RAM was needed to handle the massive amounts of video data and model parameters effectively during training. We mainly used Python, a flexible programming language popular in artificial intelligence and machine learning, for our software needs. We used two of the most popular DL frameworks, TensorFlow and PyTorch, renowned for their adaptability and broad support for creating intricate neural network architectures.

Evaluation Criteria

The presented model is evaluated using several evaluation criteria. The evaluation metrics are listed below:

Precision:

- Precision evaluates the accuracy of the positive forecasts.
- It is calculated as the ratio of total positive predictions (TP + FP) to true positive (TP) predictions.
- Precision = $\frac{TP}{TP + FP}$

Recall:

- Recall, sometimes referred to as sensitivity or true positive rate (TPR), evaluates the ability of the classifier to find all the positive samples.

- It can be defined as the ratio of real positive samples(TP + false negatives (FN)) to true positive (TP) expectations.
- Recall = $\frac{TP}{TP + FN}$

F-score:

- The F-score is the mean of precision and recall, giving a single score that balances precision and recall.
- F-score = $\frac{2\ x\ Precision\ x\ Recall}{Precision + Recall}$

Accuracy:

- Accuracy evaluates the true results (both true positives and true negatives) among the whole number of cases examined.
- It is the ratio of correct predictions to the whole number of predictions.
- Accuracy = $\frac{TP + TN}{TP + TN + FP + FN}$

Terms:

- TP (True Positive): Accurately predicted positive instances.
- TN (True Negative): Accurately predicted negative instances.
- FP (False Positive): Erroneously predicted positive instances (predicted positive but negative).
- FN (False Negative): Erroneously predicted negative instances (predicted negative but positive)

Result

The results of our model on the UCF-50 dataset will be briefly discussed in this section, along with a comparison with the results of other models that used the same dataset. The performance of 3DCNN, ConvLSTM, and Proposed 3DCNN + ConvLSTM on the UCF-50 dataset presented variations in testing, training, and validation accuracy.

Table 2. The Performance Of The Suggested Model Testing, Training, And Validation Accuracy

Model	Testing Accuracy	Training Accuracy	Validation Accuracy
3DCNN	85.63%	80%	75%
ConvLSTM	90.55%	95%	84%
Proposed 3DCNN + ConvLSTM	98.11%	99%	94%

Table 2 shows the proposed model's performance in testing, training, and validation accuracy. The 3DCNN, ConvLSTM, and the Proposed 3DCNN + ConvLSTM have presented different levels of accuracy across training, testing, and validation datasets. The 3DCNN performed a testing accuracy of 85.63%, indicating its ability to generalize well to unseen data. However, its training accuracy is barely lower at 80%, suggesting some degree of underfitting. The validation accuracy, even low at 75%, pointed out that the model may generalize poorly to new data outside the testing set. This difference between training, testing, and validation accuracies presented that the model may perform less well on samples it has yet to see because it has yet to comprehend all the data patterns during training. In contrast, the ConvLSTM model reached a higher testing accuracy of 90.55%, outperforming the 3DCNN. Its training accuracy increased to 95%, suggesting a better fit to the training data. However, the validation accuracy was lower than the testing accuracy at 84%, indicating some overfitting.

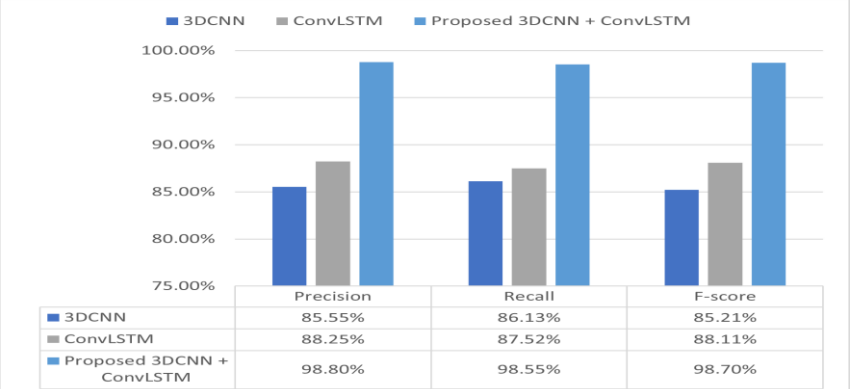

Fig 6. Performance Of The Proposed Model.

Overfitting appears when a model learns to perform well on the training data but fails to generalize to new data. The Proposed 3DCNN + ConvLSTM model surpassed both individual models, performing an outstanding testing accuracy of

98.11%. Its training accuracy is the highest at 99%, demonstrating a perfect fit to the training data. The validation accuracy of 94% is also extensively high, indicating that the model generalizes well to unseen data. This combined model integrates the strengths of both 3DCNN and ConvLSTM architectures, effectively capturing spatial and temporal features in the data. While all models demonstrated promising results, the Proposed 3DCNN + ConvLSTM model showed the best accuracy across all metrics. However, the differences between training, testing, and validation accuracies in each model underscored the importance of careful evaluation to generalize unseen data and mitigate issues such as overfitting or underfitting. The three models, 3DCNN, ConvLSTM, and the proposed 3DCNN+ConvLSTM, favorably compared precision, recall, and F-score performance metrics.

Fig 6 graphically represents the performance of evaluation metrics of the models. Starting with 3DCNN, it reached a precision of 85.55%, a recall of 86.13%, and an F-score of 85.21%. These metrics indicated its proficiency in correctly identifying relevant instances while keeping a balanced trade-off between precision and recall. On the other hand, ConvLSTM exhibited more increased precision at 88.25%, barely lower recall at 87.52%, and an admirable F-score of 88.11%, which suggested ConvLSTM'sability classify positive instances more accurately, even though it had a slightly lower recall rate than 3DCNN. The proposed model, 3DCNN+ConvLSTM, showcased remarkable performance across all metrics, with a precision of 98.80%, recall of 98.55%, and an outstanding F-score of 98.70%.

The proposed approach capitalized on the strengths of 3DCNN and ConvLSTM, generating better use of temporal and spatial information to obtain higher classification accuracy. The proposed model is superior to 3DCNN and ConvLSTM, even if each performs well in certain respects. Its substantially higher precision, recall, and F-score suggested a synergistic advancement over individual models. Combining convolutional and recurrent neural network architectures, the proposed model harnesses spatial and temporal dependencies more exhaustively, significantly enhancing performance. Therefore, the proposed 3DCNN+ConvLSTM model is the most suitable choice due to its exceptional overall accuracy and balanced performance across precision and recall. Several hyperparameters are essential in determining how the model behaves and performs. First, by choosing 'cate precision at 88.25%, barely lower recall at 87.52%, and an admirable F-score of 88.11%, which suggested ConvLSTM's ability to classify positive instances more accurately, even though it had a slightly lower recall rate than 3DCNN. The proposed model, 3DCNN+ConvLSTM, showcased remarkable performance across all metrics, with a precision of 98.80%, recall of 98.55%, and an outstanding F-score of 98.70%. This proposed approach capitalized on the strengths of 3DCNN and ConvLSTM, generating better use of temporal and spatial information to obtain higher classification accuracy.

Table 3. Hyperparameters of the Model

Hyperparameter	Value
Optimizer	Adamax
Loss of function	'categorical cross-entropy'
Rate of learning	0.001
Patience	15
Restore best weights	True
Epoch	50

The proposed model is superior to 3DCNN and ConvLSTM, even if each performs well in certain respects. Its substantially higher precision, recall, and F-score suggested a synergistic advancement over individual models. Combining convolutional and recurrent neural network architectures, the proposed model harnesses spatial and temporal dependencies more exhaustively, significantly enhancing performance. Therefore, the proposed 3DCNN+ConvLSTM model is the most suitable choice due to its exceptional overall accuracy and balanced performance across precision and recall. Several hyperparameters are essential in determining how the model behaves and performs. First, by choosing 'categorical_crossentropy' as the loss function, the model strives to minimize the discrepancy between the actual class labels and predicted probabilities for different activity classes. The optimizer 'Adamax' is used in gradient descent optimization.

It controls the step size in updating the model's parameters during training with a learning rate of 0.001. This learning rate is a vital factor that influences the training process's stability and rate of convergence. The amount of total passes through the dataset during training is specified by the number of epochs set to 50. Individually epoch defines one forward pass and one backward pass of all the training examples.

The dataset is divided into batches, as demonstrated by the batch size of 4, and the model's parameters are upgraded employing the average gradient estimated across these batches. For validation, 10% of the training data is reserved when the validation split is 0.1. This may prevent overfitting by enabling the model to observe its performance on untested data during training. We assessed the model using several metrics once trained, including accuracy, f-score, precision, and recall. **Table 3** depicts the Hyperparameters of the model. **Fig 7** shows every epoch's testing accuracy and loss. Apart from the conventional evaluation metrics like accuracy, F-score, precision, and recall, we also used Cohen's Kappa and Matthews Correlation Coefficient (MCC) to evaluate the performance of our model. Cohen's Kappa measures the degree of agreement between predicted and observed classifications by considering the possibility that agreement could have happened by accident alone. MCC, on the other hand, accounts for both true and false positives as well as negatives when calculating the correlation between predicted and observed classifications. By adding these extra metrics, the model's predictive power is comprehensively assessed, expanding our comprehension of its effectiveness beyond traditional

metrics. **Table 4** shows the kappa and MCC values. With an MCC of 82.65% and a Kappa score of 82.87%, the DCNN model performs admirably. These metrics show that there is a good degree of agreement between the model's forecasts and the actual dataset observations. However, it performs marginally worse in MCC and Kappa compared to the other evaluated models. The ConvLSTM model performs better than the DCNN model, as shown by its higher MCC of 88.54% and Kappa score of 88.03%. This model captures temporal and spatial dependencies in the data by utilizingLSTM and convolutional layers. More excellent agreement with the ground truth labels and a more robust predictive capability is indicated by the higher values of MCC and Kappa. The suggested hybrid model performs significantly better than the DCNN and ConvLSTM models, combining 3DCNN with ConvLSTM layers. With an MCC of 97.51% and a Kappa score of 97.96%, it exhibits remarkable agreement and predictive accuracy with the dataset's true labels. This notable performance gain indicates that the hybrid architecture's integration of spatial and temporal information greatly improves the model's capacity to represent intricate patterns and dynamics in the data. Overall, the comparative analysis demonstrates the efficacy of the suggested 3DCNN + ConvL-STM model in our study, outperforming the individual DCNN and ConvLSTM models in terms of both Kappa and MCC. Our suggested model effectively met every assessment criterion after thoroughly examining the research findings.

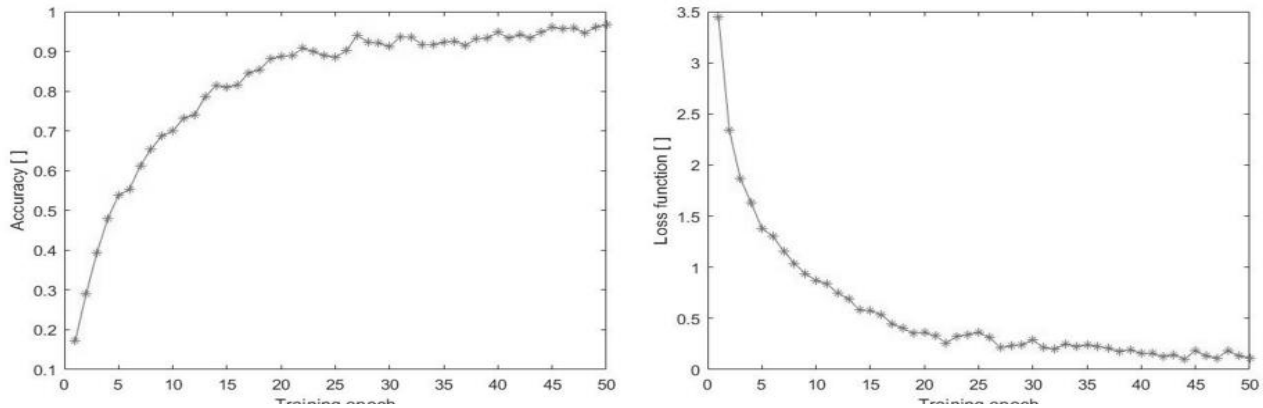

Fig 7. Visualization of Testing Accuracy and Loss for Each Epoch.

Table 4. Kappa and MCC Analysis

Model	Kappa	MCC
3DCNN	82.87%	82.65%
ConvLSTM	88.03%	88.54%
Proposed 3DCNN +ConvLSTM	97.96%	97.51%

First, we examined how well it could generalize data. We found that the proposed model had a well-fitted profile with no cases of overfitting or underfitting, suggesting that it was robust when dealing with unknown data. Our proposed model demonstrated outstanding results after examining the true positive, true negative, false positive, and false negative rates. The evaluation was carried out using precision, recall, and F-score metrics, all of which showed the exceptional effectiveness of the model in precisely identifying instances in various classes. The suggested model also showed a respectable degree of agreement in this instance, confirming its dependability and accuracy in identifying the underlying patterns in the data. The results of these analyses are summarized in **Fig 8**, where the pie chart captures the overall achievement of our research projects.

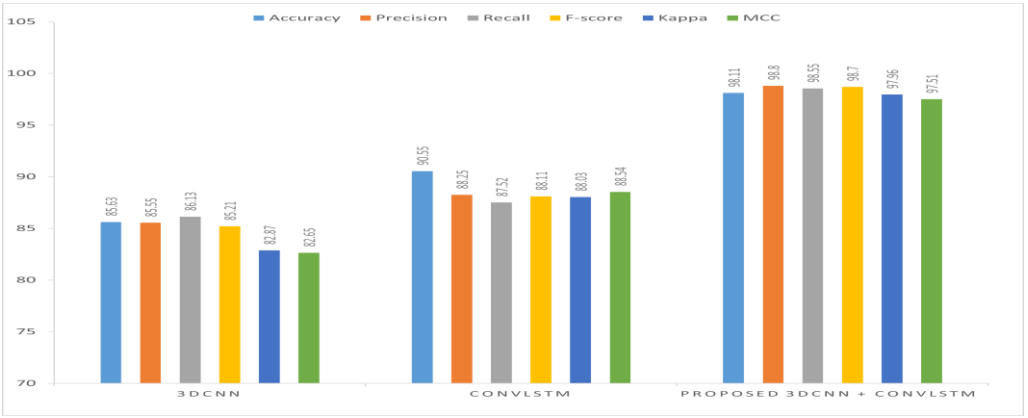

Fig 8. Graphical Representation Of The Model Evaluation Metrics.

Discussion

Our study thoroughly assesses the suggested paradigm on several fronts. Our results show a remarkable ability to recognize objects and activities in real time from video data. **Table 5** compares different model performances and summarizes the

Table 5. Comparing Existing Models That Applied To The UCF-50 Dataset With Our Proposed Model

Reference	Model	Accuracy
Shi et al. [34]	CNN + Transform	83.41%
Ramya et al. [35]	Distance transform + Entropy features	80%
Vaghela et al. [36]	I3D model with VGG19	98.06%
Aldahoul et al. [37]	EfficientNetB7-LSTM	80%
Proposed Model	3DCNN + ConvLSTM	98.11%

research achievements in the field. The comparison of our model with other existing work that trained on the same dataset is shown in **Table 5**. When we compared the models on the UCF-50 dataset, the suggested model—which combined the architectures of 3DCNN and ConvLSTM—achieved a remarkable accuracy rate of 98.11%. With an accuracy of 98.06%, the top reference model [36] that uses an I3D model in conjunction with VGG19 architecture is closely matched by our model. The accuracy of 83.41%, 80%, and 80% correspondingly was attained by using CNN with transformations [34], distance transforms with entropy features [35], and the EfficientNetB7- LSTM architecture [37], which is less accurate than our model. These outcomes highlight our suggested model's outstanding performance and competitiveness in object and activity detection in video datasets, establishing it as a notable competitor among cutting-edge techniques. In our analysis of the UCF-50 dataset, which has 50 different classes, we found that earlier research used a more detailed approach, with researchers focusing on particular courses for their findings. Similarly, we found that various classes had different performance outcomes from our model evaluation. We compared our class-wise results to those of Kumar et al. [38], who took into account all 50 classes in their research to give a thorough comparison. **Table 6** provides a concise summary of this comprehensive class-wise comparison, illuminating the subtle differences in performance across various activity categories and facilitating a better understanding of the model's efficacy over the whole range of activities. Vital accuracy

Table 6. Our Suggested Model's Accuracy Was Compared Class-Wise To The Work Of Kumar Et Al. [38]

Class	Proposed Model	Kumar et al. [38]
Baseball Pitch	96%	98%
Breaststroke	90%	86%
Golf Swing	97%	95%
Hula Hoop	80%	86%
Kayaking	92%	96%
Playing Piano	96%	95%
Pull Ups	87%	88%
Rowing	91%	86%
Soccer Juggling	97%	95%
Trampoline Jumping	93%	95%
Basketball shooting	89%	86%
Clean and Jerk	94%	88%
Playing Guitar	98%	95%
Playing Violin	89%	86%
Javelin Trow	90%	88%
Lunges	96%	95%
Playing Tabla	88%	86%
Punch	93%	98%
Salsa Spins	83%	88%
Swing	98%	95%

is demonstrated by our suggested model in various tasks, with remarkable success rates of 96% for baseball pitching, 97% for golf swinging, and 96% for piano playing. These findings frequently match or even exceed the accuracy reported by Kumar et al. For example, our model's 96% accuracy for Baseball Pitch is only slightly less accurate than Kumar et al.'s 98%. Our model beats the benchmark set by Kumar et al. in certain instances. For example, our model obtains greater accuracies (97% and 98%, respectively) in golf swing and guitar playing than the 95% reported by Kumar et al. This indicates that correctly identifying these actions from video data is an area in which our model shines. Nonetheless, there are several situations in which the accuracy of our model is marginally less than that stated by Kumar et al. For instance,

our model achieves 90% and 87% accuracy in tasks like pull-ups and breaststrokes, respectively, although Kumar et al. reported somewhat higher values of 86% and 88%.

V. CONCLUSION

Human activity recognition has several applications, including video surveillance, healthcare, and human-computer interaction, which have been the focus of the present research. Recognizing human activities through video data presents a challenge wherein temporal features hold marked importance. mCNNs are widely employed for image classification; however, adapting CNNs to Human Activity Recognition (HAR) presents challenges because of the temporal dynamics involved. Our contribution lies in delivering a novel approach that merges ConvLSTM and 3DCNN models, explicitly tailored for improving human activity recognition by employing the UCF-50 dataset. After training and testing, our model performed exceptionally well in every class on the UCF-50 dataset. Notably, our presented architecture, combining 3DCNN with ConvLSTM, surpassed standalone ConvLSTM and 3DCNN models, highlighting the strength of DL methodologies in human activity recognition. Our outcomes underscore the efficacy of the merged 3DCNN + ConvLSTM in accurately classifying video data featuring different human activities. The importance of our study expands beyond academic realms, as the insights gleaned pave the way for improved OD systems in diverse applications, mainly in human care, and our research advances more robust and trustworthy solutions in this domain in OD within human activities. Moreover, our study provides a basis for future direction and motivation for evaluations on the UCF50 dataset and is expanded upon with the UCF101 dataset, therefore increasing the range and applicability of these cutting-edge models.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests.

References

[1]. Y. Amit, P. Felzenszwalb, and R. Girshick, “Object Detection,” Computer Vision, pp. 875–883, 2021, doi: 10.1007/978-3-030-63416-2_660.
[2]. T. J. Palmeri and I. Gauthier, “Visual object understanding,” Nature Reviews Neuroscience, vol. 5, no. 4, pp. 291–303, Apr. 2004, doi: 10.1038/nrn1364.
[3]. X. Wu, D. Sahoo, and S. C. H. Hoi, “Recent advances in deep learning for object detection,” Neurocomputing, vol. 396, pp. 39–64, Jul. 2020, doi: 10.1016/j.neucom.2020.01.085.
[4]. A. Yilmaz, O. Javed, and M. Shah, “Object tracking,” ACM Computing Surveys, vol. 38, no. 4, p. 13, Dec. 2006, doi: 10.1145/1177352.1177355.
[5]. H. Sharma, M. Agrahari, S. K. Singh, M. Firoj, and R. K. Mishra, “Image Captioning: A Comprehensive Survey,” 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Feb. 2020, doi: 10.1109/parc49193.2020.236619.
[6]. L. Yang, Y. Fan, and N. Xu, “Video Instance Segmentation,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2019, doi: 10.1109/iccv.2019.00529.
[7]. E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A Survey of Autonomous Driving: Common Practices and Emerging Technologies,” IEEE Access, vol. 8, pp. 58443–58469, 2020, doi: 10.1109/access.2020.2983149.
[8]. A. Sophokleous, P. Christodoulou, L. Doitsidis, and S. A. Chatzichristofis, “Computer Vision Meets Educational Robotics,” Electronics, vol. 10, no. 6, p. 730, Mar. 2021, doi: 10.3390/electronics10060730.
[9]. S. Jha, C. Seo, E. Yang, and G. P. Joshi, “Real time object detection and trackingsystem for video surveillance system,” Multimedia Tools and Applications, vol. 80, no. 3, pp. 3981–3996, Sep. 2020, doi: 10.1007/s11042-020-09749-x.
[10]. M. Cao, J. Jiang, L. Chen, and Y. Zou, “Correspondence Matters for Video Referring Expression Comprehension,” Proceedings of the 30th ACM International Conference on Multimedia, Oct. 2022, doi: 10.1145/3503161.3547756.
[11]. J. Liu et al., “PolyFormer: Referring Image Segmentation as Sequential Polygon Generation,” 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2023, doi: 10.1109/cvpr52729.2023.01789.
[12]. M. Li and L. Sigal, "Referring transformer: A one-step approach to multi-task visual grounding," Advances in Neural Information Processing Systems, vol. 34, pp. 19652-19664, 2021.
[13]. Y Y. Zhou et al., “A Real-Time Global Inference Network for One-Stage Referring Expression Comprehension,” IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 1, pp. 134–143, Jan. 2023, doi: 10.1109/tnnls.2021.3090426.
[14]. A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang, “Open-Vocabulary Object Detection Using Captions,” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2021, doi: 10.1109/cvpr46437.2021.01416.
[15]. S. Wu, W. Zhang, S. Jin, W. Liu, and C. C. Loy, “Aligning Bag of Regions for Open-Vocabulary Object Detection,” 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2023, doi: 10.1109/cvpr52729.2023.01464.
[16]. J. Wang et al., “Open-Vocabulary Object Detection With an Open Corpus,” 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2023, doi: 10.1109/iccv51070.2023.00622.
[17]. M. A. Bravo, S. Mittal, S. Ging, and T. Brox, “Open-vocabulary Attribute Detection,” 2023 IEEE/CVF Conference on Computer Vision and

Pattern Recognition (CVPR), Jun. 2023, doi: 10.1109/cvpr52729.2023.00680.
[18]. I. Ulusoy and C. M. Bishop, “Generative versus Discriminative Methods for Object Recognition,” 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), doi: 10.1109/cvpr.2005.167.
[19]. K. Compton, A. Smith, and M. Mateas, “Anza Island,” Proceedings of the The third workshop on Procedural Content Generation in Games, May 2012, doi: 10.1145/2538528.2538539.
[20]. A. Joshi, H. Parmar, K. Jain, C. Shah, and Patel Prof. Vaishali R., “Human Activity Recognition Based on Object Detection,” IOSR Journal of Computer Engineering, vol. 19, no. 02, pp. 26–32, Mar. 2017, doi: 10.9790/0661-1902012632.
[21]. M. Safaei, P. Balouchian, and H. Foroosh, “UCF-STAR: A Large Scale Still Image Dataset for Understanding Human Actions,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 03, pp. 2677–2684, Apr. 2020, doi: 10.1609/aaai.v34i03.5653.
[22]. J. Y. Yun, E. J. Choi, M. H. Chung, K. W. Bae, and J. W. Moon, “Performance evaluation of an occupant metabolic rate estimation algorithm using activity classification and object detection models,” Building and Environment, vol. 252, p. 111299, Mar. 2024, doi: 10.1016/j.buildenv.2024.111299.
[23]. M. Hu et al., “Physiological characteristics inspired hidden human object detection model,” Displays, vol. 81, p. 102613, Jan. 2024, doi: 10.1016/j.displa.2023.102613.
[24]. P. Su and D. Chen, “Adopting Graph Neural Networks to Analyze Human–Object Interactions for Inferring Activities of Daily Living,” Sensors, vol. 24, no. 8, p. 2567, Apr. 2024, doi: 10.3390/s24082567.
[25]. R. Nabiei, M. Parekh, E. Jean-Baptiste, P. Jancovic, and M. Russell, “Object-Centred Recognition of Human Activity,” 2015 International Conference on Healthcare Informatics, Oct. 2015, doi: 10.1109/ichi.2015.14.
[26]. N. S. Suriani, F. N. Rashid, and M. H. Badrul, "Semantic object detection for human activity monitoring system," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 10, pp. 115-118, 2018.
[27]. B. A. Mohammed Hashim and R. Amutha, “Elderly People Activity Recognition Based on Object Detection Technique Using Jetson Nano,” Wireless Personal Communications, vol. 134, no. 4, pp. 2041–2057, Feb. 2024, doi: 10.1007/s11277-024-10982-y.
[28]. K. K. Reddy and M. Shah, “Recognizing 50 human action categories of web videos,” Machine Vision and Applications, vol. 24, no. 5, pp. 971–981, Nov. 2012, doi: 10.1007/s00138-012-0450-4.
[29]. R. Vrskova, R. Hudec, P. Kamencay, and P. Sykora, “Human Activity Classification Using the 3DCNN Architecture,” Applied Sciences, vol. 12, no. 2, p. 931, Jan. 2022, doi: 10.3390/app12020931.
[30]. S. Ji, W. Xu, M. Yang, and K. Yu, “3D Convolutional Neural Networks for Human Action Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 221–231, Jan. 2013, doi: 10.1109/tpami.2012.59.
[31]. P. Partila, J. Tovarek, G. H. Ilk, J. Rozhon, and M. Voznak, “Deep Learning Serves Voice Cloning: How Vulnerable Are Automatic Speaker Verification Systems to Spoofing Trials?,” IEEE Communications Magazine, vol. 58, no. 2, pp. 100–105, Feb. 2020, doi: 10.1109/mcom.001.1900396.
[32]. Z. Yuan, X. Zhou, and T. Yang, “Hetero-ConvLSTM,” Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Jul. 2018, doi: 10.1145/3219819.3219922.
[33]. K. Ashok, M. Ashraf, J. Thimmia Raja, M. Z. Hussain, D. K. Singh, and A. Haldorai, “Collaborative analysis of audio-visual speech synthesis with sensor measurements for regulating human–robot interaction,” International Journal of System Assurance Engineering and Management, Aug. 2022, doi: 10.1007/s13198-022-01709-y.
[34]. C. Shi and S. Liu, “Human action recognition with transformer based on convolutional features,” Intelligent Decision Technologies, vol. 18, no. 2, pp. 881–896, Jun. 2024, doi: 10.3233/idt-240159.
[35]. P. Ramya and R. Rajeswari, "Human action recognition using distance transform and entropy based features," Multimedia Tools and Applications, vol. 80, pp. 8147-8173, 2021.
[36]. R. Vaghela, D. Labana, and K. Modi, "Efficient I3D-VGG19-based architecture for human activity recognition," The Scientific Temper, vol. 14, pp. 1185-1191, 2023.
[37]. N. Aldahoul, H. A. Karim, A. Q. Md. Sabri, M. J. T. Tan, Mhd. A. Momo, and J. L. Fermin, “A Comparison Between Various Human Detectors and CNN-Based Feature Extractors for Human Activity Recognition via Aerial Captured Video Sequences,” IEEE Access, vol. 10, pp. 63532–63553, 2022, doi: 10.1109/access.2022.3182315.
[38]. M. Kumar, A. K. Patel, M. Biswas, and S. Shitharth, “Attention-based bidirectional-long short-term memory for abnormal human activity detection,” Scientific Reports, vol. 13, no. 1, Sep. 2023, doi: 10.1038/s41598-023-41231-0.