مجلة
كلية التــراث الجامعة

مجلة علمية محكمة
متعددة التخصصات نصف سنوية
العدد الحادي والأربعون
30 نيسان 2025
ISSN 2074-5621

رقم الايداع في دار الكتب والوثائق 719 لسنة 2011

مجلة كلية التراث الجامعة معترف بها من قبل وزارة التعليم العالي والبحث العلمي بكتابها المرقم (ب 3059/4) والمؤرخ في (2014/ 4/7)

Video-Based Human Activity Recognition Using Deep Learning

Zahraa Suhail Ali Alfarttoosi

Nadine Zbib

Abstract

This thesis proposes a human action recognition (HAR) system utilizing Inception V3, a pre-trained convolutional neural network (CNN), combined with transfer learning to automate surveillance tasks. The system aims to mitigate challenges such as human operator fatigue, high monitoring costs, and limited scalability. Fine-tuned on the UCF-11 dataset, which includes 11 action categories, the model achieves a notable accuracy of 98%, demonstrating strong performance in terms of computational efficiency and resilience to variations in lighting and camera angles. However, the approach primarily focuses on spatial feature extraction and does not account for temporal dynamics, such as motion and sequence patterns, which are crucial for understanding complex actions. Additionally, the reliance on a relatively small dataset limits the system's generalizability to real-world applications. Despite these limitations, the model shows promise for deployment in domains like surveillance, healthcare, and robotics. Future enhancements will involve integrating temporal analysis methods, such as optical flow or 3D CNNs, and extending validation to larger, more diverse datasets like Kinetics to improve scalability and real-world applicability.

1- Introduction

Events defined as occurrences anchored in specific times and locations are frequently viewed as manifestations of human actions. Understanding human movement goes beyond merely tracking the physical motion of body parts; it involves interpreting the underlying intentions, emotions, and cognitive states that drive such movements. Consequently, human action recognition has emerged as a vital component in the analysis of human behavior across diverse fields such as surveillance, robotics, healthcare, video retrieval, and human-computer interaction.

In the context of surveillance, for instance, human operators are often responsible for monitoring multiple CCTV feeds to identify abnormal or suspicious activities in public areas. This task is inherently challenging due to the cognitive demands of maintaining sustained attention across numerous screens and processing vast quantities of visual data. Given these limitations, surveillance systems typically require large teams to ensure continuous monitoring, which significantly increases operational costs [1].

To overcome these challenges, deep learning (DL) techniques have gained prominence in computer vision and action recognition. Among these, convolutional neural networks (CNNs) have demonstrated exceptional performance. CNNs are capable of automatically extracting and learning spatial features from individual video frames, making them highly effective for image analysis and human action detection. Unlike traditional fully connected networks, CNNs utilize local connectivity patterns to focus on localized regions within an image, thereby enhancing

computational efficiency. However, CNN-based methods often face difficulties distinguishing between actions involving similar gestures, which can lead to reduced classification accuracy. Additionally, these models typically rely on large volumes of labeled training data—an often scarce resource—limiting their scalability and effectiveness in real-world applications [1].

Deep learning (DL), particularly Convolutional Neural Networks (CNNs), has significantly advanced the fields of computer vision (CV) and human action recognition (HAR) by enabling automated feature extraction from video data. Unlike traditional neural networks, CNNs leverage local connectivity to analyze spatial patterns within video frames, making them highly effective for visual analysis tasks.

Despite their success, CNNs face notable limitations, such as often struggling to differentiate between actions involving similar gestures, which can compromise classification accuracy. As well as they require large amounts of labeled training data—an often scarce resource in real-world applications, limiting their scalability and performance.

To address these challenges, our approach incorporates the following steps:

1. Data Preprocessing: Extracting key spatiotemporal features to enhance model input quality.
2. Model Evaluation: Assessing various deep learning architectures to identify the most effective solution.
3. System Development: Constructing a predictive framework designed to automate the recognition of human behaviors.

Our objective is to develop a robust and efficient HAR system that mitigates existing limitations and improves applicability in practical, real-world scenarios.

2- Related Work

Human Activity Recognition (HAR) in videos is a complex task that involves extracting both spatial and temporal features from video frames to accurately classify various human actions. It has wide-ranging applications, including multimedia indexing, surveillance, and human-computer interaction. Convolutional Neural Networks (CNNs) have emerged as a popular deep learning approach due to their ability to automatically learn spatial features from raw image data. However, CNNs alone are not well-suited for modeling temporal dependencies across frames, which are crucial for understanding motion and sequential patterns in video data. To address this limitation, many researchers have explored hybrid architectures that combine CNNs with Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, to effectively capture both spatial and temporal dynamics for HAR tasks. In the following section, we briefly review three recent studies that utilize CNN-based approaches for HAR in videos, comparing their datasets, model architectures, and performance outcomes.

Method 1 [5]:

This study proposes a hybrid CNN–LSTM architecture for human action recognition. The approach consists of two main components: (1) feature extraction using a pre-trained VGG16

convolutional neural network, followed by (2) classification using a Long Short-Term Memory (LSTM) network to model the temporal dynamics of the video sequences. Additionally, the authors investigate the impact of varying the number of LSTM units on the system's performance. The model is trained and evaluated on three benchmark datasets: KTH, UCF-11, and HMDB-51. Performance is measured using accuracy, considering the relatively balanced class distribution in the datasets. The proposed system achieves classification accuracies of 93% on KTH, 91% on UCF-11, and 47% on HMDB-51. Beyond the reported results, the key contribution of this work lies in its comparative evaluation of different CNN–LSTM configurations for the task of action recognition.

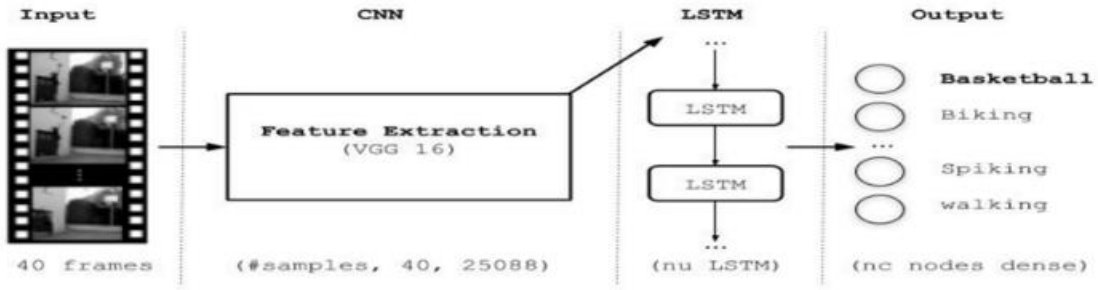

Figure 1. Results Of Video

Results of the proposed video classification (with nu = 320) using the database (UCF-11, HMDB-51, KTH)

Dataset	Acc (%)
KTH	93.86%
UCF-11	91.94%
HMDB-51	47.36%

Method 2 [4]:

This study employs Convolutional Neural Networks (CNNs) for Human Activity Recognition (HAR), demonstrating their effectiveness in processing time-series data. However, manually designing optimal CNN architectures is often time-consuming and prone to human error. To address this, the authors leverage Neural Architecture Search (NAS), an emerging technique that automates the design and optimization of neural networks. NAS enables the discovery of high-performing architectures by exploring a vast design space beyond the limitations of human intuition. In this work, an evolutionary algorithm—Genetic Algorithm (GA)—is used to guide the NAS process. GA is well-suited for black-box optimization problems, where explicit mathematical models or gradient information are unavailable.

The proposed framework, named AUTO-HAR, introduces a novel encoding schema and an expansive search space with a wide variety of operations, enabling efficient exploration of potential network architectures for HAR. Notably, the method does not restrict the maximum depth of the architecture, offering greater flexibility in discovering deeper and more effective models. The approach is evaluated using the UCI-HAR dataset, achieving an impressive average recognition accuracy of 97.5%. These results demonstrate the effectiveness of combining CNNs with evolutionary NAS techniques for automated and accurate human activity recognition.

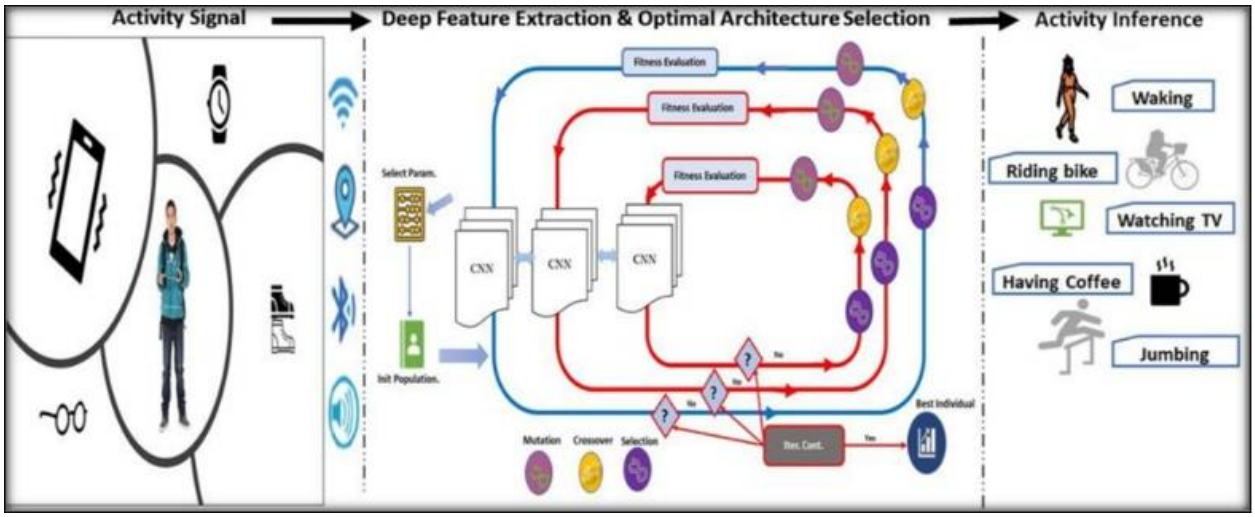

Figure 2. The Proposed AUTO-HAR Deep Learning Architecture.

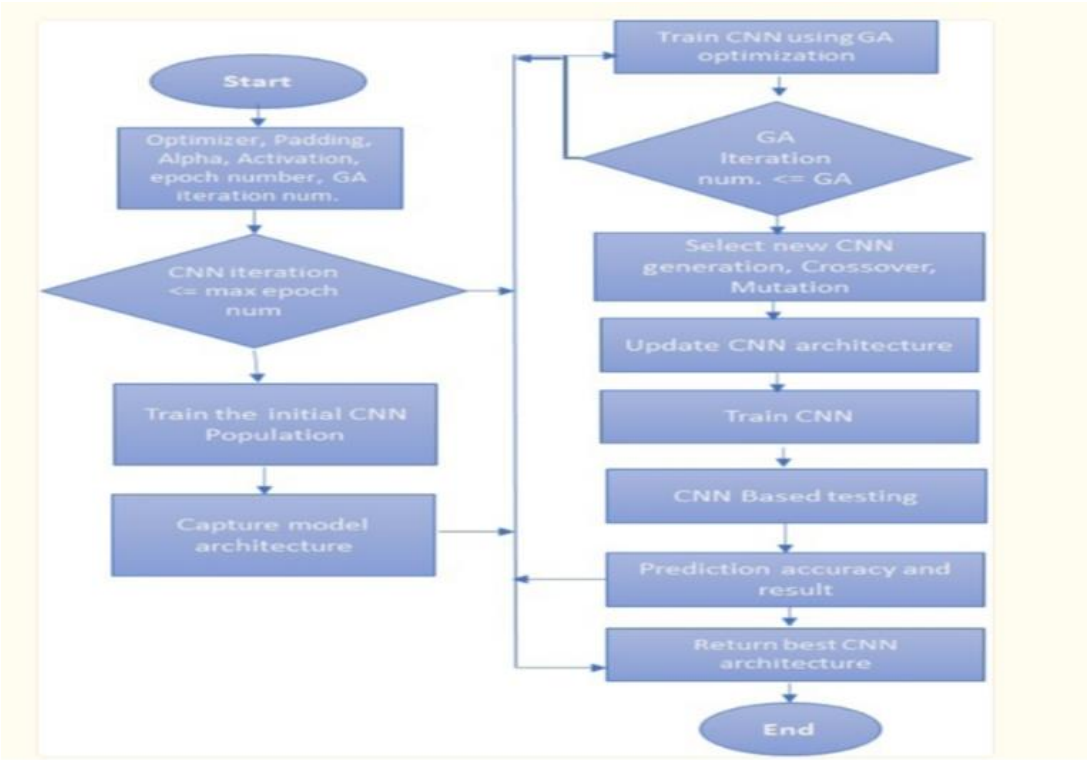

Figure 3. AUTO-CNN architecture design and optimization

Results of the proposed video classification (with nu = 320) using the database (UCI-HAR)

Table 2: Results of the proposed video classification

Dataset	Acc (%)
UCI-HAR	97.5% (∓1.1)

Method 3 [2]:

This study introduces a hybrid 3D convolutional neural network (3D-CNN) and Convolutional LSTM (ConvLSTM) architecture for human activity recognition. While 3D-CNNs are widely used to capture spatio-temporal features in video data, our proposed model enhances this capability by integrating ConvLSTM layers to better model temporal dependencies. We evaluate the architecture on three benchmarks—LoDVP Abnormal Activities, a modified subset of UCF50 (UCF50mini), and MOD20—to provide a comprehensive comparison. The combined 3D-CNN+ConvLSTM model achieves precisions of **89.12%** on LoDVP, **83.89%** on UCF50mini, and **87.76%** on MOD20. These results demonstrate that the addition of ConvLSTM units significantly improves recognition accuracy. Moreover, the model's design supports real-time deployment and could be further enhanced by incorporating multi-sensor inputs, underscoring its potential for practical applications in surveillance and assistive technologies.

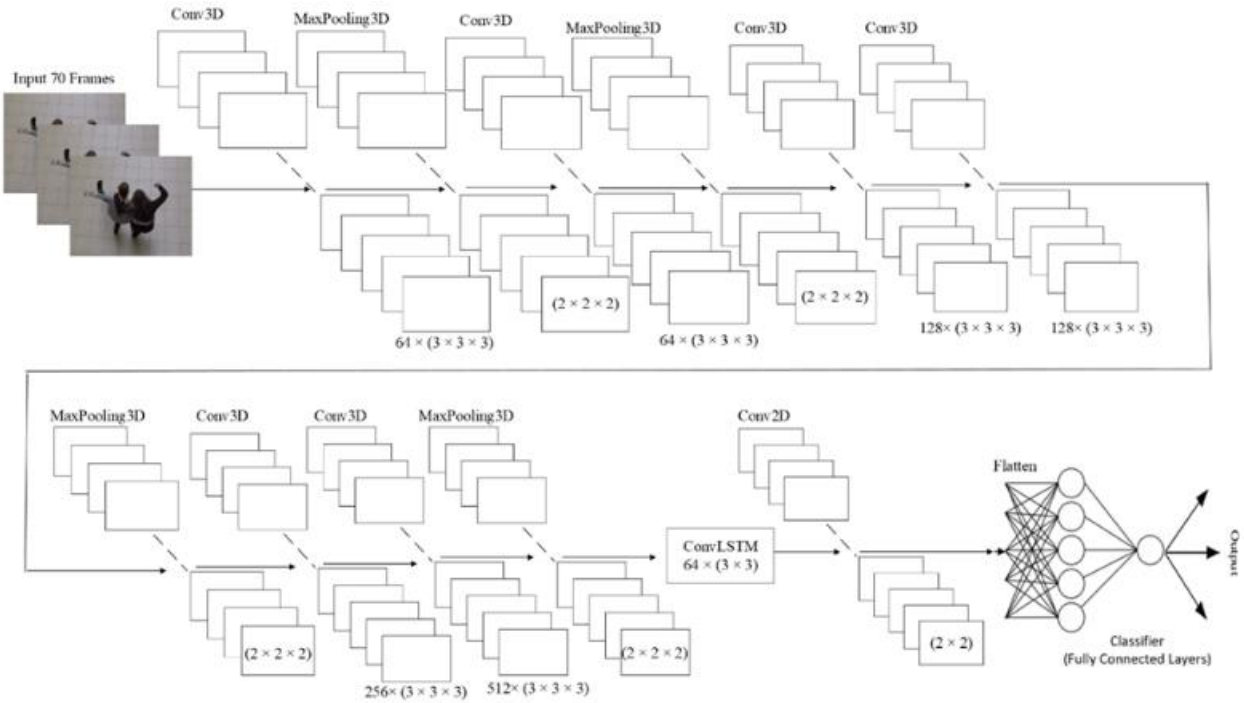

Figure 4. Proposed 3DCNN + ConvLSTM architecture

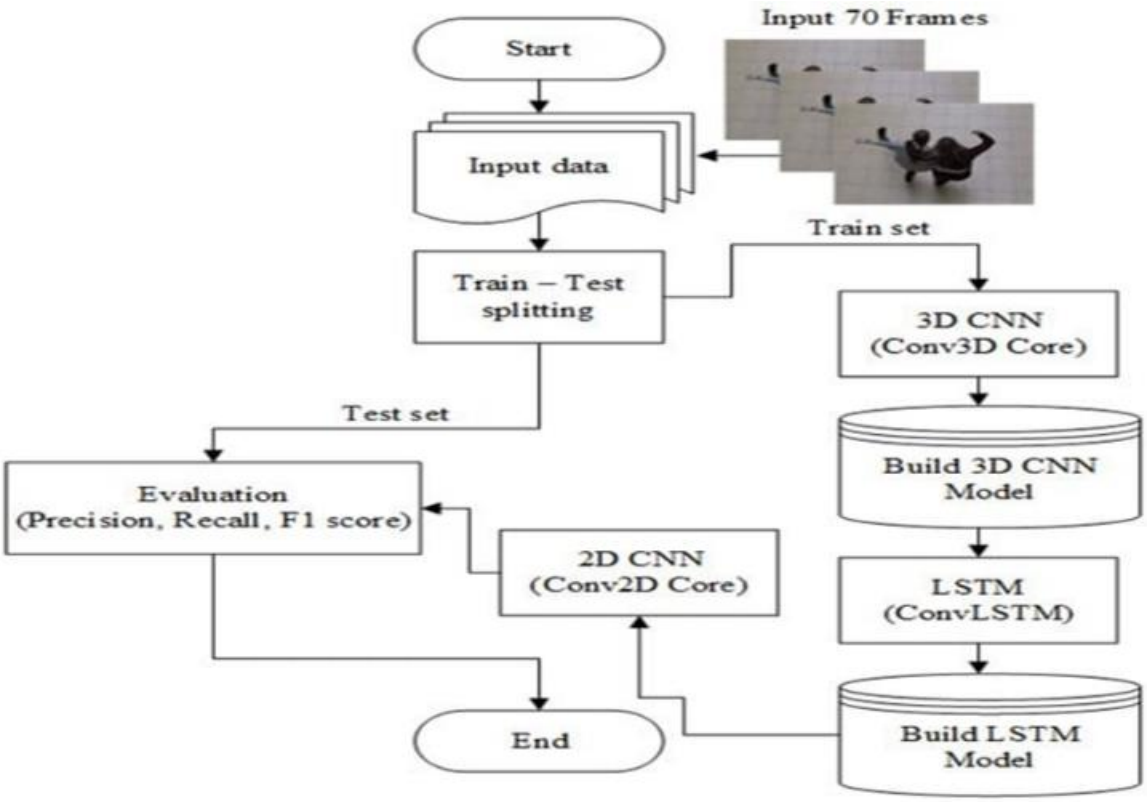

Figure 5. The flowchart of the proposed architecture

3- Proposed HAR System

Here are detailed instructions for carrying out "HAR using Transform Learning":

1- <u>Data gathering and Dataset Description:</u>

The dataset size may change depending on the data's accessibility, but it should ideally be big enough to adequately train a (DL) model. The Data Set should include a variety of activities covering a range of topics and locations. The dataset's features should be balanced, which means that each class should have an equal number of activities. The dataset was derived from YouTube videos as well. The dataset possesses the following characteristics:

1) a combination of stable and unsteady cameras

2) crowded background

3) item scale variation

4) many points of view

5) variable lighting 6) poor resolution.

The 11 categories in this action dataset are (“basketball shooting, volleyball spiking, trampoline jumping, soccer juggling, horseback riding, cycling, diving, swinging, golf swinging, tennis swinging, and walking (with a dog)”). The first four movements are sometimes mistaken for "jumping," the next two actions could have comparable camera motions, and all "swing" movements have a similar set of movements. Some tasks are also accomplished with the aid of a machine, such as a (“horse, bicycle, or dog”) Identification may benefit from both static and regional contextual elements. There are 1168 video sequences in the collection as a whole to avoid the unfair impact of having the same background in recognition.

To get the best results, a portion for training (80% of the research sample) and a portion for testing (20% of the research sample) were used.

- Data split train-test (80_20 %)
- 935 Videos for train and Videos for test
- classification Videos for 11 class

Figure 6. DataSet Sample

a) Image data augmentation

Data augmentation is a technique to increase the diversity and size of a dataset by applying various transformations to the original images. It can improve the performance and generalization of image classification models by reducing overfitting and introducing more variability. Some common data augmentation methods include flipping, rotating, cropping, scaling, shifting, adding noise, changing brightness or contrast, etc.

In our thesis we use the flowing data augmentation methods:

- Rescale: Rescale pixel values to [0, 1]
- shear range: Apply shear transformation.
- horizontal flip: Flip the image horizontally
- rotation range: Apply rotation.
- width shift range: Apply horizontal shift.
- height shift range: Apply vertical shift.

Algorithm (3-1) Image Data Augmentation

Input: dataset
Output: preprocessed images X

Step 1: Define dataset path.
Step2: Define
- rescale=1. /255, # Rescale pixel values to [0, 1]
- shear_range=0.2, # Apply shear transformation
- horizontal_flip=True, # Flip the image horizontally
- rotation_range=10., # Apply rotation
- width_shift_range=0.2, # Apply horizontal shift
- height_shift_range=0.2) # Apply vertical shift

Step3: Define
- target_size= (299, 299), # Resizing the image to 299x299
- batch_size=32, # Size of the batch
- classes=Class_name, # List of class names
- class_mode='categorical') # Mode of classification

Step4: For Each image in dataset X Apply Augmentation
Step5: Return X

b) Proposed HAR Model Architecture:

In our design, the proposed Architecture should contains a max pooling layer (MP), convolutional layer (CL), activation function (AF),and dense layer (DL).

1. Load The Inceptionv3 Model Pre-Trained on The Imagenet Dataset.

2. Add A Global Average Pooling Layer to Reduce the Number of Parameters and Make the Model More Computationally Efficient.
3. Add a fully connected layer (Dense) with 1024 nodes and ReLU activation.
4. Add A Final Logistic Layer with Softmax Activation, Which Outputs Probabilities for Each of The Classes.
5. Create the final model by combining the InceptionV3 base model with the added layers.
6. Compile the model with the Adam optimizer and categorical cross-entropy loss. We also specify accuracy as a metric to monitor during training.
    - optimizer='adam'

Adam optimizer is a popular algorithm for gradient-based optimization of neural networks. It combines the advantages of two other methods: adaptive gradient algorithm (AdaGrad) and root mean square propagation (RMSProp). Adam optimizer can be used for image classification tasks, where the objective is to classify an input picture according to its content. The following are some advantages of employing the Adam optimizer for picture classification:

- Gradient based optimization of neural networks is a popular algorithm and the Adam optimizer is one such algorithm.
- It is a combination of benefits of two other methods namely — adaptive gradient algorithm (AdaGrad) and root mean square propagation (RMSProp). Adam optimizer can be used for image classification for example as an input of the picture is classified by the corresponding ne content. The benefits of using the Adam optimizer for picture classification fall into the following:
- That is, it adjusts the learning rate for each parameter based on history of gradients in order to possibly avoid local minima and improve its convergence speed.
- To improve the stability and accuracy of the optimization process, it uses a momentum term to reduce the oscillations and noise in the gradient updates.
- It has low computational cost and little memory requirement compared to other optimizers and is therefore well suited for solving large scale and high dimensional problems.
- It has a few hyperparameters that need to be tuned, such as the learning rate, the decay rates of the first and second moment estimates, and the epsilon value to prevent division by zero. These hyperparameters can be set to default values or adjusted based on empirical results.
    - **loss='categorical_crossentropy'**
    - **metrics='accuracy'**

c) Model Fitting:

In this Step we define number of steps per epoch(batches) according to our computer Hardware and the size of the dataset (steps_per_epoch=2000), then define number of number of epochs to train the model, define the training and test data, finally define the **callbacks** which is:

- Checkpointer: Criteria for considering the model in specific epoch as the best model and save it as the best model. We use the lowest loos vale and maximum accuracy value to define the best model and save the best model in (best. hdf5)
- Early stopper: It works to stop training when there is no improvement in the accuracy of training in each epoch.
- Tensorboard: log file to save the steps of training and flow the errors.

Algorithm (3-2) Generate the Best Trained Model
Output: final trained model (best. hdf5)
Step 1: Load the InceptionV3 model pre-trained on the ImageNet dataset As Base Model ***Step2:*** Add a Global Average Pooling layer. ***Step3*** Add a fully connected layer with 1024 nodes and ReLU activation. ***Step4:*** Add a final logistic layer with softmax activation. ***Step5:*** Create the final model by combining the InceptionV3 base model with the added layers. ***Step6:*** Compile the model with the Adam optimizer and categorical. cross- entropy loss ***Step7: Fitting the model to the following parameters:*** o # training data generator o train_generator (From Algorithm (3-1)) o # Number of steps (batches) to run in each epoch o steps_per_epoch=2000 o # validation data generator o validation_data=validation_generator (From Algorithm (3-1)) o # Number of epochs to train the model o epochs=5 ***Step8:*** Run the Model and Start Training ***Step9: For Each*** epochs Save the model with the lowest val_loss and maximum val_accuracy to best. hdf5 ***Step10:*** Stop training when there is no improvement in training. ***Step11***: Save the Final Model ***(***best. hdf5***)***

d) Video Classification and Model Evaluation

The final model (best. hdf5) is evaluated in this stage to check the accuracy of the suggested HAR classification by testing resulted final-model on data for test to identify the model's correctness and analyze the effectiveness of the suggested system. We will apply the evaluation metrics mentioned in the previous chapter to the recommended model assessment on the test

suite that we have prepared earlier, these metrics determine the effectiveness of the video recognition model.

Algorithm (3-3) Video Classification and Model Evaluation
Input: best. hdf5, test set ***Output: Predict classes for test data, Model Evaluation***
Step 1: Get the list of classes. ***Step2:*** Load the trained model. ***Step3*** Define batch_size = 32 ***Step4***: Predict classes for test data using the trained model and Print classification. report ***Step5:*** Calculate accuracy score and print it. ***Step6:*** Plot confusion matrix ***Step7:*** Calculate precision, recall, and f1 score, and print them

e) Test the Final model of Real video downloaded from Internet.

In this step, the final model is tested with a real video that is downloaded from the Internet so that the proposed system predicts the Activity in the video.

Algorithm (3-4) Predict Human Activity in Video
Input: Video ***Output: Predict classes***
Step 1: Load the trained model. ***Step2:*** Read the input video file from computer drive or google drive. ***Step3:*** Loop through the frames of the video ***Step4***: Resize the frame and preprocess it for the model. ***Step5:*** Predict the labels for the frame***Step6:*** Plot confusion matrix. ***Step6:*** Store the predictions in a dictionary with the corresponding class names. ***Step7:*** Sort the dictionary by prediction score in descending order. ***Step8:*** Display the top 2 predictions classes. ***Step9:*** ***If*** the score is above 0.7 display prediction class on video.

4- Experimental Results

The proposed system leverages deep learning techniques to distinguish between various types of human behavior. The dataset used for training was collected from YouTube and consists of 11 activity categories: basketball shooting, volleyball spiking, trampoline leaping, soccer juggling, horseback riding, cycling, diving, general swinging, golf swinging, tennis swinging, and walking (with a dog).

These categories were selected to evaluate the model's ability to differentiate between visually similar actions. For instance, the first four activities—basketball shooting, volleyball spiking, trampoline leaping, and soccer juggling—are often visually mistaken for "jumping." Cycling and horseback riding may exhibit similar camera motion patterns, adding complexity to the

recognition task. Additionally, all swinging-related actions (golf, tennis, and general swinging) share highly similar movement characteristics, making them particularly challenging to distinguish.

Some activities also involve interaction with external objects or animals—such as a bicycle, horse, or dog—which can provide useful contextual cues for recognition. To minimize the bias caused by repetitive background scenes and ensure a balanced evaluation, the dataset contains a total of 1,168 video sequences.

To get the best results, a component of the research sample was utilized for training (80% of the research sample) and a piece for testing (20% of the research sample).

- Data split train-test (80_20 %)
- 935 Videos for train and Videos for test
- classification Videos for 11 class

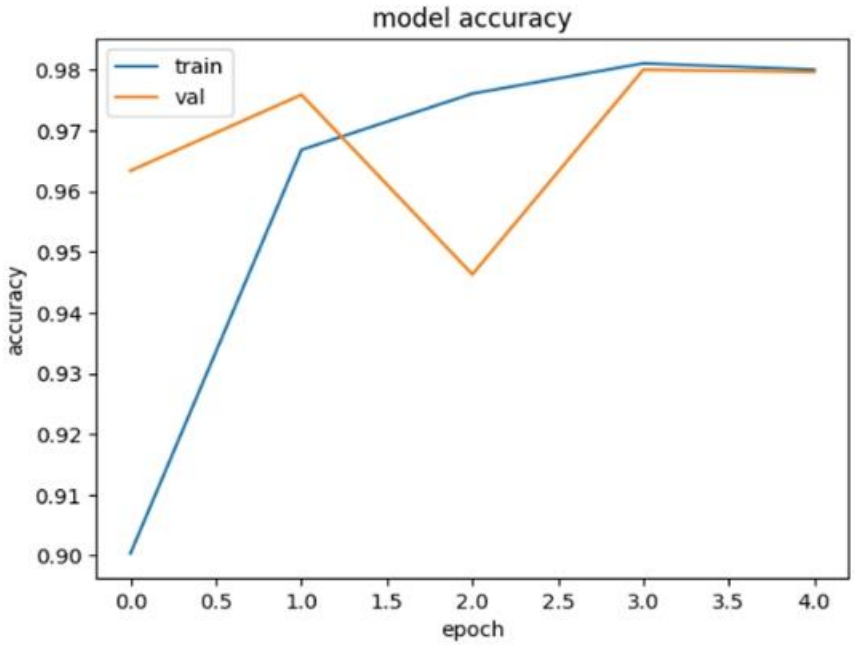

Figure 7. Model Accuracy

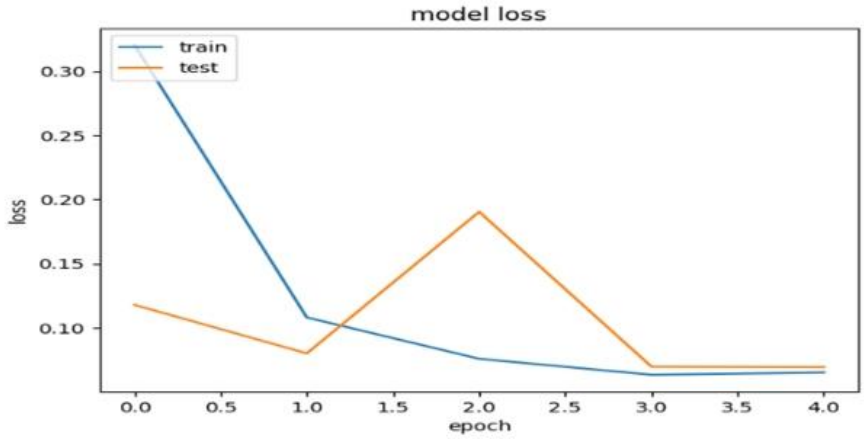

Figure 8. Model Los

	precision	recall	f1-score	support
walking	0.99	0.92	0.95	5546
tennis_swing	1.00	1.00	1.00	5315
basketball	0.95	1.00	0.97	3846
diving	1.00	1.00	1.00	5348
biking	0.95	1.00	0.97	6573
horse_riding	1.00	0.99	0.99	7795
swing	1.00	0.95	0.97	5535
trampoline_jumping	0.99	0.98	0.99	4804
soccer_juggling	1.00	0.98	0.99	9148
volleyball_spiking	1.00	0.96	0.98	2571
golf_swing	0.90	1.00	0.95	4735
accuracy			0.98	61216
macro avg	0.98	0.98	0.98	61216
weighted avg	0.98	0.98	0.98	61216

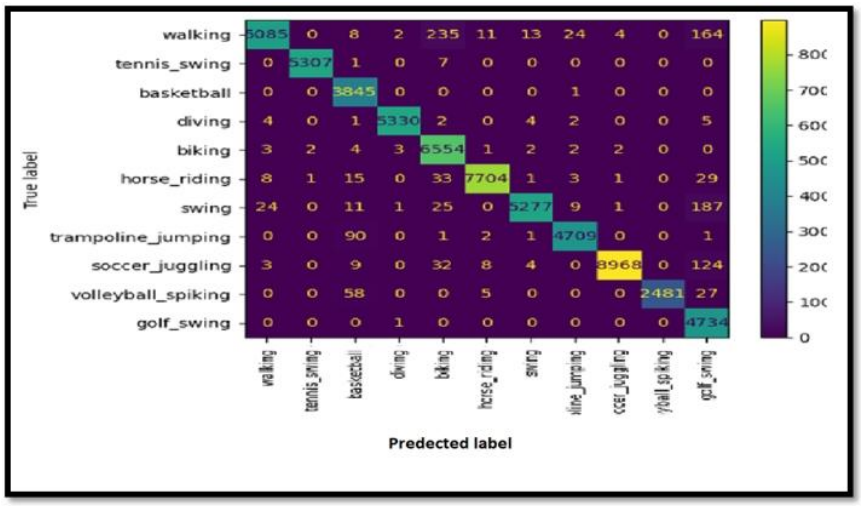

Figure 9. Predecate Label

inceptionv3:0.9800378985886042
Precision = 0.9793596661499986
Recall = 0.9796272930559868

5- Conclusion

Through the implementation of the proposed algorithm as part of this master's thesis project, and by utilizing the UCF-11 dataset along with a supplementary set of videos collected from online sources such as YouTube, the researcher found that the proposed approach achieved a very high classification accuracy of 98%. This result represents an excellent performance level when compared to findings from previous studies reviewed in Chapter Two of this thesis. The high accuracy demonstrates the effectiveness of the proposed method in human action recognition tasks.

In future work, incorporating additional data from diverse sources could further enhance model performance and generalization. As shown in the table below.

6- References

[1] D. Wu, "C02029: Doctor of Philosophy Video-based similar gesture action recognition using deep learning and GAN-based approaches," 2019. [Online]. Available: https://opus.lib.uts.edu.au/bitstream/10453/140177/1/01front.pdf.

[2] R. Vrskova, P. Kamencay, R. Hudec and P. Sykora, "A New Deep-Learning Method for Human Activity Recognition," *Sensors*, vol. 23, p. 2816, 4 2023.

[3] W. N. Ismail, H. A. Alsalamah, M. M. Hassan and E. Mohamed, "AUTO-HAR: An adaptive human activity recognition framework using an automated CNN architecture design," *Heliyon*, vol. 9, p. e13636, 4 2023.

[4] M. Li and Q. Sun, "3D Skeletal Human Action Recognition Using a CNN Fusion Model," 2021. [Online].

[5] R. Yamashita, M. Nishio, R. K. G. Do and K. Togashi, "Convolutional neural networks: an overview and application in radiology," 8 2018. [Online].

[6] C. Orozco, E. Xamena, M. Buemi and J. Berlles, "Human Action Recognition in Videos using a Robust CNN LSTM Approach (Reconocimiento de Acciones Humanas en Videos usando una Red Neuronal CNN LSTM Robusta)," pp. 23-36, 2020.

[7] v. choubey, "Text classification using CNN," 4 2020. [Online]. Available: https://medium.com/voice-tech-podcast/text-classification-using-cnn-9ade8155dfb9.

[8] "Learn K-Nearest Neighbor(KNN) Classification and build a KNN classifier using Python Scikit-learn package.," 2021. [Online]. Available: https://morioh.com/p/620a3a2faf6c.

[9] A. Gupta, "A Comprehensive Guide on Deep Learning Optimizers," 4 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/.

[10] S. Doshi, "Various Optimization Algorithms For Training Neural Network," 4 2020. [Online]. Available: https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6.

[11] R. Sunil, "Understanding Support Vector Machine algorithm from examples (along with code)," 4 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

[12] A. Hirunyawanakul, N. Kerdprasop and K. Kerdprasop, "A novel heuristic method for misclassification cost tuning in imbalanced data," 2018. [Online].

[13] T. . E. Trueman, A. Kumar J., Narayanasamy P. and , Vidya J., "Attention-based C-BiLSTM for fake news detection," *Applied Soft Computing*, vol. 1009.42KB, p. 8, 2021.