IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Human Activity Image Classification Using Deep Learning

[1]Ms.Yeshwin Shaaradha N, [2] Mr.Rohith Kumar M, [3] Mr. Phinehaas Knight

Student, Btech, CSE with AIML

[1]Computer Science Department,

[1]SRM Institute of Science and Technology, Vadapalani, Chennai, India.

Abstract: Recognizing human activities is a crucial yet difficult study area in the field of computer vision. We suggest context features in this work together with a machine learning model to identify the specific subject activity in the image. To enhance the performance of recognition, we use the dataset from various sources. To provide a high-level representation of human activity recognition based on an image collection, we develop a deep neural network structure. Recognizing human activity necessitates forecasting a person's behaviour using image-based information. The photos are divided into recognized activities. The goal is to forecast human activity using machine learning techniques with the highest degree of accuracy. The CNN Algorithm can be used to categorize the photos. To choose the best architecture, more than two architectures were compared. Finally, the model can be deployed in Django framework.

Index Terms **– HAR, computer vision, neural nework, CNN, django**

I. INTRODUCTION:

A well-known research area, human activity recognition (HAR), entails the accurate identification of various activities, sampled in various ways. In the HAR industry, deep and automatic approaches are becoming more popular. The decision to choose significant features from the data is left up to the learning model when data-driven techniques are used to classify images. CNNs are very good at recognizing relationships between images and modelling scaleinvariant features. Convolutional neural networks are used in this paper to address the HAR problem.

Objectives of the Study:

In order to analyses a person's behaviour in a real setting, HAR models are designed to offer information on human actions. The goal of image classification is to recognize and describe the features that appear in a picture in terms of the object or kind of land cover these features actually represent on the ground as a distinct grey level (or colour). The most crucial aspect of digital image analysis is probably picture classification. It additionally, seeks to enhance the current picture classification method by delivering accurate outcomes.

Proposed System:

To categorize human activities and create a deep learning method. Based on a picture collection, it suggested a technique for forecasting human activities. More image samples are gathered and compared with various classes. The shape and texture-focused qualities are what make up the majority of the image's characteristics.

We are initially preparing our dataset and putting more than two CNN architectures into practice. We compare each architecture because each one provides a unique form of precision. following the models .h5 file format saving. because we won't need to continually train the architecture. Once the model has been trained and has achieved accuracy, the model is saved in hierarchical file format. Now, whenever you want, you can check the correctness. The Django Framework supports the deployment of that trained model.

II. LITERATURE SURVEY:

[1]Smartphone Sensors and Deep Convolutional Neural Networks for Human Activity Recognition 2016's Ordóez and Roggen.

The research suggests a deep learning method for smartphone sensors-based human activity identification (HAR). The objective of the study is to use a deep convolutional neural network (CNN) architecture to increase accuracy.30 volunteers who were wearing smartphones and engaged in six different activities—walking, walking upstairs, walking downstairs, sitting, standing, and lying—

were used by the authors to gather data. The accelerometer and gyroscope sensors on the smartphone collected data, which was then processed and used to train and test the deep CNN model. Two convolutional layers, two max-pooling layers, and two fully linked layers make up the proposed deep CNN architecture. Stochastic gradient descent with backpropagation and dropout regularization is used to train the model. The accuracy, precision, and recall measures were used by the authors to assess the model's performance.

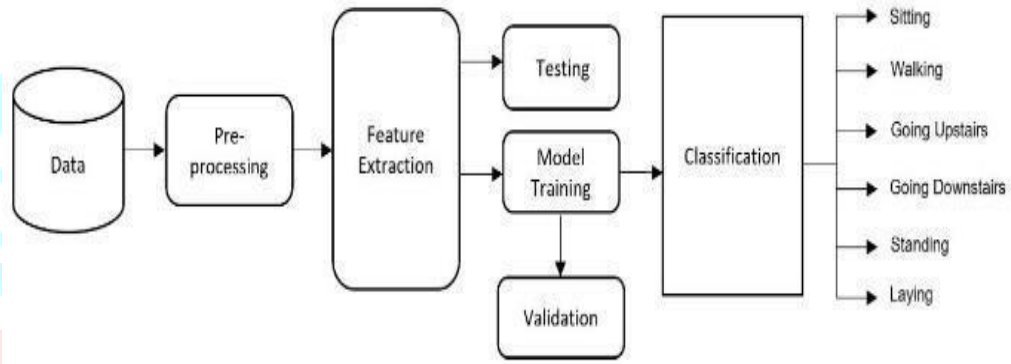
The outcomes demonstrated that the suggested deep CNN methodology outperformed other established machine learning techniques, such as decision trees and support vector machines, and attained excellent accuracy. The accuracy, precision, and recall of the deep CNN model were 91.4%, 91.6%, and 91.3%, respectively.

[2]A Survey on Human Activity Recognition using Wearable Sensors. Bao and Intille – 2014.

The study offers a thorough analysis of the research on wearable sensor-based human activity recognition (HAR). The paper discusses a variety of wearable sensor-related issues, such as sensor placement, feature extraction, classification techniques, and applications. The authors give a thorough description of the many HAR sensor types, including magnetometers, accelerometers, gyroscopes, and physiological sensors. Additionally, the study addresses several feature extraction and selection strategies, including time-domain, frequencydomain, and statistical features. The author compares the advantages and disadvantages of several categorization techniques, such as decision trees, support vector machines, and neural networks

III. SYSTEM ARCHITECTURE:

Fig 3.1



3.1Data preparation: It's crucial to get the labelled data ready for usage in the model before creating the classification model. This could entail normalizing or standardizing the data as well as dividing the data into training, validation, and testing sets.

3.2Feature extraction: The following stage is to extract from the data aspects that can be used to categorize human actions. This may entail taking the raw data and extracting useful features using methods like Fourier transforms, wavelet transforms, or timefrequency analysis.

3.3Collect data: Utilize the established process to get data from the participants. This can entail gathering information outdoors, in a field setting, or in the participants' natural surroundings.

3.4Label the data: Indicate the human activity that corresponds to the data that was obtained. The data may then be manually annotated or labelled using automated techniques in accordance with predetermined standards.

3.5Data Pre-processing: Any machine learning project must start by gathering and preprocessing the data. In the instance of classifying photographs based on human activity, this entails gathering a sizable dataset of pictures that have been marked with the relevant human activity. To make the photos suitable for input into a deep learning model, they may be scaled, normalized, or given further preprocessing.

3.6Model selection: The next step is to choose an appropriate classification model after the features have been extracted. Decision trees, random forests, support vector machines (SVMs), and neural networks are common models for recognizing human action. The dataset's size and complexity, as well as the desired accuracy and computing efficiency, will all influence the model that is selected.

3.7Model training: The model is trained using the labelled data after it has been chosen. The model parameters must be optimized in order to reduce the loss function and raise forecast accuracy.

3.8Model evaluation: To check that the model is successfully generalizing to new data after training, a separate validation set is used for evaluation. If the model is underperforming, the feature extraction methods or model parameters can be changed, and the model can then be retrained.

3.9Model testing: The model is tested using a different testing set once it has been trained and reviewed to determine how well it performs in a real-world scenario.

3.10Deployment: The model can be implemented for real-time human activity recognition in the chosen application once it has undergone testing and validation.

IV. ACCURACY CALCULATION:

A tuned model continually improves its accuracy. The most accurate algorithm out of the four is chosen as the best algorithm based on accuracy. We're using the following formula to do it:

Fig 4.1

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Note: *The most accurate algorithm is chosen after considering the four different approaches. If our model has a high degree of accuracy, which is a great statistic, we can say that it is the best available, but only if the datasets are symmetric and the proportion of false positives and false negatives is roughly equal.*

4.1Model:

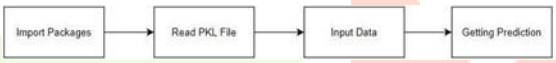
The trained model is then converted to a. pkl file, or Python Pickle File, so that we may move on to the deployment stage after identifying the algorithm with the highest accuracy.

Note: *The Python pickle module is used to build PKL files, which are serialized objects. It comprises of binary strings that stand in for Python project objects. Python's normal syntax, which entails importing the pickle module, can be used to produce PKL files. Basic datasets, machine learning models, and Pandas data frames are just a few of the many objects that may be found in this file.*

4.2Deploying the Model:

The trained model must be incorporated into a real-time system that can recognize human action before it can be used.
The deployment of the model involves the phases listed below.:

Fig 4.2



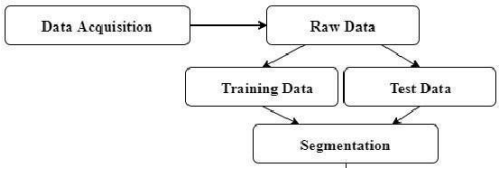
V. MODULES AND IMPLEMENTATION:

It has been demonstrated that Convolutional Neural Networks (CNNs) are efficient in identifying human actions from sensor data.

5.1Dataset: In the human activity picture classification module, a dataset of photos representing the various activities that need to be recognized are gathered. This dataset can be acquired from a variety of sources, including online databases, public repositories, and simply collecting photos or videos of people engaged in the relevant activities. It is crucial to make sure that the dataset is diverse and accurately depicts the kind of actions the model is supposed to be able to identify. The pictures must be of high quality, well-lit, and show the activity from several perspective. To account for changes in how the activity is performed, the dataset should also include pictures of the same activity being carried out by several people. It is crucial to label the dataset in addition to the photographs, which means giving each image a class name that describes the activity it depicts. Depending on the quantity and complexity of the dataset, this labelling can be done manually or with the aid of automated methods. The complexity of the task and the quantity of activities that must be recognized can affect the dataset's size. Larger datasets typically result in higher model performance and generalization since they provide the model more examples to learn from.

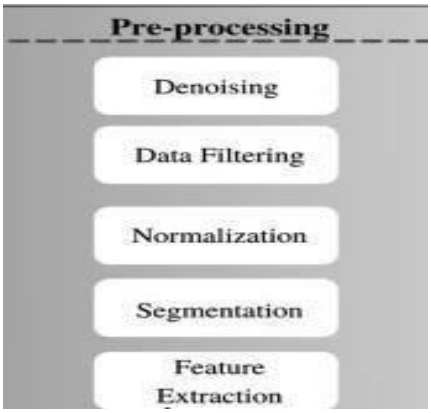
The dataset can be divided into training, validation, and testing sets after it has been gathered and labelled.

Fig 5.1.1



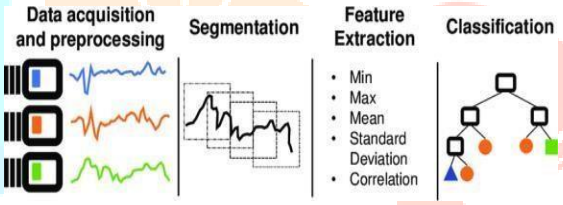
The validation set is used to fine-tune the model hyperparameters and avoid overfitting while the training set is used to train the model. The testing set is employed to gauge how well the model performs when applied to new data.

Fig 5.1.2



5.2Data Preprocessing: Before training the model, the dataset must be prepared using the Human Activity Image Classification's data preprocessing module. This module's objective is to normalize and transform the raw picture data into a form that can be used to feed the image classification model. The steps in the data pre-processing module may comprise the following:

Fig 5.2.1



5.2.1Image Resizing: The pre-processing step of resizing the photographs to a specified size is typical. In order to feed the photos into the model, it is crucial that they all have the same size and format across the dataset.

5.2.2Image Normalization: The model may perform better if the image pixel values are normalized to a common scale. This can be achieved by dividing each pixel by the standard deviation after subtracting the mean pixel value from the dataset.

5.2.3Data Augmentation: By applying several changes to the original photos, such as rotation, flipping, scaling, and translation, data augmentation creates fresh training examples. This aids in expanding the training dataset and enhancing the model's capacity for generalization.

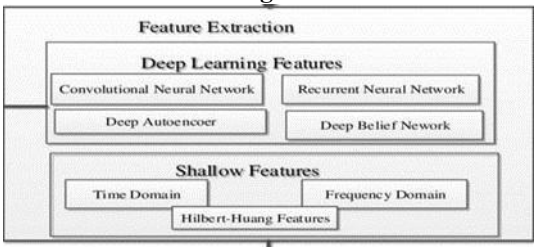
5.2.4Image Cropping: When a picture is cropped, a region of interest is chosen and the rest is removed. By doing so, you may be able to reduce distracting background noise and draw attention to the image's key details.

5.2.5Data Balancing: Data balancing is the process of making sure that each activity class has an equal number of photographs in the dataset. This is crucial to avoid the model favouring one or more classes over other.

The preprocessed dataset can be divided into training, validation, and testing sets when the data preprocessing module is finished so that the model can be trained and its performance assessed.

5.3Feature Extraction:

Fig 5.3.1



In a CNN, the feature extraction procedure is feeding an input picture through the network and retrieving the features at various network layers. The result of the last convolutional layer is often used as the feature map when classifying images of human activities.

The input image is subjected to a collection of learnable filters in the convolutional layer, which generates a series of feature maps, each of which draws attention to a particular pattern or feature in the input image.

The most high-level and abstract elements of the input image that are most important to the classification task are contained in the output of the last convolutional layer.

After being flattened into a one-dimensional vector, the feature map created by the final convolutional layer is input into a fully connected layer for classification. The fully connected layer receives the feature vector and generates an output vector that illustrates the odds that the input image falls into one of the categories of human activity.

The CNN is able to learn a mapping between the input image and the appropriate human activity category by extracting the most crucial elements from the input image using the final convolutional layer and putting them into a fully connected layer. Over several training epochs, this feature extraction and classification procedure is repeated, and the CNN's weights are changed to enhance classification task accuracy.

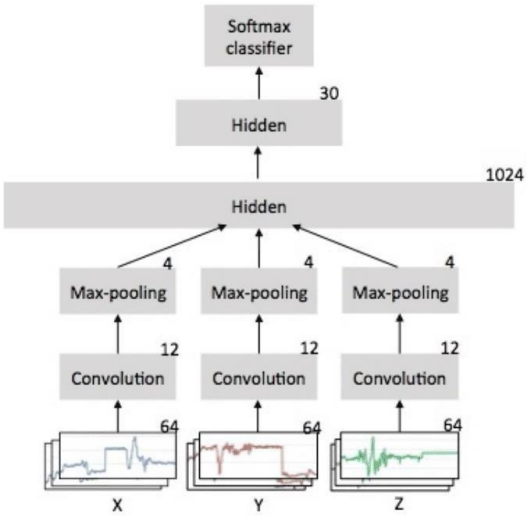
5.4Model Architecture:

Building a machine learning model that can reliably classify human activity photos requires several key steps, one of which is the model architecture module in CNN-based human activity image classification. This module's objective is to choose or create a CNN architecture that can accurately extract pertinent features from photos of human activities and classify those features into the appropriate categories.

A key component of creating a machine learning model that can precisely categorize photos of human activity is the model A key component of creating a machine learning model that can precisely categorize photos of human activity is the model architecture module used in CNN's human activity image classification process. The objective of this module is to choose or create a CNN system that can quickly and accurately identify the key details in images of human activity and classify them into the relevant groups.

For instance, a pre-trained CNN architecture like VGG, ResNet, or Inception that has been trained on a large dataset of images may be a viable option if the classification problem is reasonably straightforward. The feature extraction module can be avoided because these pre-trained architectures have already learnt how to extract features from images, allowing the fully linked layers to be added right away to the pretrained CNN architecture. On the other hand, if the classification task is more complex, a custom CNN architecture may be required. The architecture may consist of multiple convolutional layers with pooling layers in between, followed by fully connected layers for

Fig 5.4.1

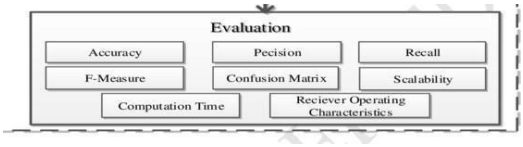


classification. The number of layers and their sizes should be chosen based on the complexity of the classification task and the size of the dataset. A CNN architecture can be fine-tuned using a smaller dataset of images of human behavior specific to the application after being chosen or created. The weights of the pre-trained CNN or the custom CNN are modified during fine-tuning to more accurately classify the images of human activities in the training dataset. The CNN architecture can be regularized using methods like dropout layers, weight decay, and early stopping to reduce overfitting and improve the model's generalization skills.

Overall, a key component of creating a precise and effective machine learning model for classifying human activity from images using CNN is the model architecture module. The model may achieve good accuracy and generalization performance on a variety of human activity picture datasets by carefully choosing or constructing a CNN architecture that can efficiently extract pertinent features from human activity photographs and categorize them into their corresponding categories.

5.5Training:

Fig 5.5.1



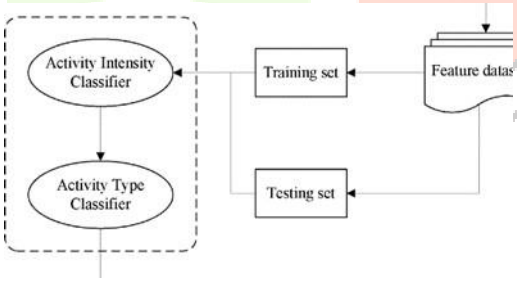
In the training module for CNN-based human activity image categorization, a collection of photographs depicting human activity is used to train the chosen or built CNN architecture. The training module's steps are as follows:

5.5.1Data preprocessing: The dataset of human activity photos needs to be prepared before training the CNN architecture. As part of this, the photos must be uniformly sized, the pixel values must be normalized, and the dataset must be divided into training, validation, and testing sets.

5.5.2Define the loss function: The CNN architecture's performance on the training dataset is gauged by the loss function. Categorical cross-entropy is a widely used loss function in human activity image classification that assesses the discrepancy between the expected probability and the actual labels.

5.5.3Choose an optimizer: During training, the CNN architecture's weights are modified by the optimizer to reduce the loss function. Stochastic gradient descent (SGD), Adam, and RMSprop are common optimizers used in the classification of human activity images.

Fig 5.5.2



5.5.4Train the CNN architecture: The CNN architecture is fed batches of images from the training set during training, and backpropagation is used to update the weights. Based on the size of the dataset and the available computer resources, the number of epochs (i.e., the number of times the full dataset is used to train the model) and the batch size (i.e., the number of images used in each training iteration) should be selected.

5.5.5Evaluate the model: After the CNN architecture has been trained, its accuracy and generalization performance are assessed using the validation and testing sets. Based on the findings of the evaluation, the model may need to be adjusted or changed.

Overall, the CNN training module for classifying human activity photographs is a crucial step in creating a machine learning model that can correctly categorize images of human activity. The model may obtain good accuracy and generalization performance on a variety of human activity image datasets by carefully preparing the dataset, selecting the proper loss functions and optimizers, and training the CNN architecture with the relevant hyperparameters.

5.6CNN Architecture:

Classifying human activity images using a Convolutional Neural Network (CNN) architecture is a common application of deep learning. CNNs are powerful because they can automatically learn features from raw data without the need for hand-crafted features. Here are the general steps you can follow to classify human activity images using a CNN architecture:

5.6.1Collect and preprocess data: Collect a dataset of human activity images and preprocess the data. Preprocessing may involve resizing the images to a standard size, normalizing the pixel values, and splitting the dataset into training, validation, and testing sets.

5.6.2Build a CNN architecture: Create a CNN architecture by stacking convolutional layers, pooling layers, and fully connected layers. The convolutional layers learn feature maps from the input images, the pooling layers reduce the spatial dimensions of the feature maps, and the fully connected layers classify the images based on the learned features.

5.6.3Train the model: Train the model on the training set by minimizing a loss function using an optimizer. The loss function measures the difference between the predicted and actual labels, and the optimizer updates the model parameters to minimize the loss.

5.6.4Evaluate the model: Evaluate the model on the validation set to monitor its performance and prevent overfitting. Overfitting occurs when the model performs well on the training set but poorly on the validation set. You can use techniques such as early stopping and regularization to prevent overfitting.

5.6.5Test the model: Test the model on the testing set to evaluate its performance on unseen data. You can calculate metrics such as accuracy, precision, recall, and F1 score to measure the model's performance.

5.6.6Improve the model: If the model's performance is not satisfactory, you can try to improve it by adjusting the architecture, hyperparameters, or preprocessing techniques.

In this architecture, the input data is an image, which is preprocessed and augmented to increase the diversity of the training data. The processed image is then fed into a convolutional neural network (CNN) which extracts relevant features from the image. The output from the CNN is then fed into a fully connected neural network which further processes the features before outputting a softmax probability distribution over the possible activity classes. Finally, the predicted class is outputted based on the highest probability class in the softmax distribution.

Note: *This is just one example of a system architecture for human activity image classification using deep learning, and there are many variations and modifications that can be made depending on the specific requirements of the task.*

Overall, building a CNN architecture to classify human activity images requires collecting and preprocessing data, building a CNN architecture, training the model, evaluating its performance, testing it, and improving it if necessary.

5.7Model Evaluation:

The CNN pipeline for human activity image classification includes a crucial phase called model evaluation. It entails evaluating the trained model's performance against the test dataset. The steps in the model evaluation module are as follows:

5.7.1Load the trained model: The trained CNN model that has been optimized during training is loaded as the first step.

5.7.2Load the test dataset: The model has never seen the test dataset, which is a distinct dataset from the training dataset. It is used to assess how well the model generalizes to fresh, untested data.

5.7.3Preprocess the test data: Preprocessing of the test data must follow that of the training data. This covers any data augmentation methods, such as scaling, normalization, and others, that were used during training.

5.7.4Evaluate the model on the test dataset: The preprocessed test dataset is then used to evaluate the model. The standard assessment metric makes advantage of the model's correctness. Depending on the issue at hand, additional measures including precision, recall, and F1 score can also be used.

5.7.5Visualize the results: Confusion matrices, which display the proportion of accurate and inaccurate predictions for each class, can be used to visualize the performance of the model. The performance of the model can also be assessed using ROC curves and precision-recall curves.

5.7.6Fine-tune the model: The model can be improved by altering the hyperparameters or changing the model architecture based on the evaluation findings.

Overall, the model evaluation module is a crucial step in evaluating how well the trained model performs on unobserved data and pointing out any flaws or potential areas for improvement. It guarantees that the model is fit for use in practical applications and can generalize successfully to fresh data.

5.8Develpoment :

On the Django framework, deploying a CNN-based human activity image classification model entails setting up a web application that can accept user-provided input photographs, preprocess them, and then use the CNN model to forecast the activity category. The procedures for setting up the model on Django are as follows:

5.8.1Build the Django application:

Making a new Django application that will act as the foundation for the human activity image classification model is the first stage. In order to do this, the required templates, views, and URL configurations must be created. ii.Integrate the model into the Django application:

The trained CNN model must now be incorporated into the Django application. In order to do this, a function that accepts an input image, preprocesses it, and then applies the model to forecast the activity category must be developed.

5.8.2Define the URL routing: The web application's URL routing needs to be defined next. The Django application's views that correlate to the URL patterns must be mapped to achieve this.

5.8.3Create the user interface: The component of the web application that enables user interaction with the model is the user interface. Making a web form that users may utilise to upload photographs and submit them to the model for classification is necessary.

5.8.4Deploy the Django application: Once it has been created, the application can be installed on a web server and used. This entails installing the Django application on the server, establishing the required software dependencies, and setting up a web server.

5.8.5Test and refine the application:

The application needs to be carefully tested after deployment to make sure it is operating as intended. This entails evaluating the model's precision, looking for any faults, and making any necessary adjustments to the application.

A web application that can be used to forecast the activity category of input photographs is integrated with the trained CNN model as part of the deployment of a human activity image classification model on the Django framework.

It demands proficiency in web development and machine learning, as well as knowledge of the Django framework and software deployment. Artificial neural networks, decision trees, and support vector machines are some common approaches.

Tools for putting these algorithms into practice are provided by Python libraries like Scikit-learn, TensorFlow, and Keras.

Additionally, evaluation of the models used for classification. Artificial neural networks, decision trees, and support vector machines are some common approaches. Tools for putting these algorithms into practise are provided by Python libraries like Scikitlearn, TensorFlow, and Keras.

Additionally, human activity recognition can be put into practise in real-time via edge devices, web apps, mobile apps, or cloud services like Amazon Web Services.

IV. CONCLUSION:

A significant area of study, human activity detection has numerous real-world applications in fields like security systems, healthcare, and tracking athletic performance. The objective of human activity recognition is to correctly categorize a person's actions based on information gathered from various sensors, including accelerometers and gyroscopes. Human activity recognition requires the use of several modules and algorithms for data collecting, feature extraction, and classification in order to accomplish correct classification. The specific implementation strategy will be determined by the application's needs, including the kind of sensors used, the level of accuracy sought, and the processing power available. All things considered, human activity recognition has the ability to enhance people's quality of life by offering information about their levels of physical activity and performance. It can also be utilised in security systems to spot suspicious behaviour. The precision and application of human activity detection are likely to improve as sensor technology advances and more data is gathered.

REFERENCES

[1] L. Gutiérrez-Madro nal, L. La Blunda, M. F. Wagner, and I. Medina-Bulo, “Test event generation for a falldetection IoT system,” IEEE Internet Things J., vol. 6, no. 4, pp. 6642–6651, Aug. 2019.

[2] S. Stavrotheodoros, N. Kaklanis, K. Votis, and D. Tzovaras, “A smart-home IoT infrastructure for the support of independent living of older adults,” in Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov., Rhodes, Greece, May 2018, pp. 238–249.

[3] Z. Lin, T. Lv, and P. T. Mathiopoulos, “3-D indoor positioning for millimeter-wave massive MIMO systems,” IEEE Trans. Commun., vol. 66, no. 6, pp. 2472–2486, Jun. 2018.

[4] X. Li, Y. He, and X. Jing, “A survey of deep learningbased humanactivity recognition in radar,” Remote Sens., vol. 11, no. 9, p. 1068,May 2019.

[5] C. Ding et al., “Continuous human motion recognition with a dynamic range-Doppler trajectory method based on FMCW radar,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 9, pp. 6821–6831, Sep. 2019.

[6] X. Bai, Y. Hui, L. Wang, and F. Zhou, “Radar-based human gait recognition using dual-channel deep convolutional neural network,”IEEE Trans. Geosci. Remote Sens., vol. 57, no. 12, pp. 9767–9778, Dec. 2019.

[7] Y. Yang, C. Hou, Y. Lang, T. Sakamoto, Y. He, and W. Xiang,“Omnidirectional motion classification with monostatic radar system using micro-Doppler signatures,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 5, pp. 3574–3587, May 2020.

[8] Y. He, L. Dai, and H. Zhang, "Multi-branch deep residual learning for clustering and beamforming in user-centric network," IEEE Lett., vol. 24, no. 10, pp. 2221–2225, Oct. 2020.

[9] I. Alnujaim, D. Oh, I. Park, and Y. Kim, "Classification of micro-Doppler signatures measured by Doppler radar through transfer learn-ing," in Proc. 13th Eur. Conf. Antennas Propag. (EuCAP), Krakow,Poland, Apr. 2019, pp. 1–3.

[10] A. Shrestha et al., "Cross-frequency classification of indoor activities with DNN transfer learning," in Proc. IEEE RadarConf. (RadarConf),Boston, MA, USA, Apr. 2019, pp. 1–6.

[10] M. S. Seyfioglu, B. Erol, S. Z. Gurbuz, and M. G. Amin, "DNN transfer learning from diversified micro-Doppler for motion classification," IEEE Trans. Aerosp. Electron. Syst., vol. 55, no. 5, pp. 2164–2180, Oct. 2019.

[12] H. Du, T. Jin, Y. Song, Y. Dai, and M. Li, "Efficient human activity classification via sparsity-driven transfer learning," IET Radar, Sonar Navigat., vol. 13, no. 10, pp. 1741–1746, Oct. 2019.

[13] H. Du, T. Jin, Y. Song, and Y. Dai, "Unsupervised adversarial domain adaptation for micro-Doppler based human activity classification," IEEE Geosci. Remote Sens. Lett., vol. 17, no. 1, pp. 62–66, Jan. 2020.