Human Action Recognition

DEEP LEARNING MAIN PROJECT

Jumana Haseen P
Data science

INDEX

Human Action Recognition (HAR)

INTRODUCTION

The dataset features 15 different classes of Human Activities. The dataset contains about 12k+ labelled images including the validation images. Each image has only one human activity category and are saved in separate folders of the labelled classes. Human action recognition (HAR) is a dynamic and
demanding field of machine learning with applications in field such as healthcare, sports, and robotics. Traditionally, algorithms based on statistical methods were used to classify human actions. The performance of these algorithm depended upon the engineered and hand crafted features, the
calculation of first and second order statistics, quantization of features to form bag of words, and feature extraction. The classification accuracy of these statistical algorithms was limited due to the separation of feature extraction component from the classification part. The success of deep learning models in image classification, natural languages, and pattern recognition, and their ability to extract features directly from the data make them a valuable tool for HAR. Because of the developments in sensor technology and availability of inexpensive sensors such as wearable sensors and Kinect, HAR has become an active field of interest in machine and deep learning.

PROBLEM STATEMENT:

Human Action Recognition (HAR) aims to understand human behavior and assign a label to each action. It has a wide range of applications, and therefore has been attracting increasing attention in the field of computer vision. Human actions can be represented using various data modalities, such as RGB, skeleton, depth, infrared, point cloud, event stream, audio, acceleration, radar, and WiFi signal, which encode different sources of useful yet distinct information and have various advantages depending on the application scenarios.

Consequently, lots of existing works have attempted to investigate different types of approaches for HAR using various modalities.

Your Task is to build an Image Classification Model using CNN that classifies to which class of activity a human is performing.

DATASET:-

Link to dataset used in CNN model: https://www.kaggle.com/datasets/emirhanai/human-action-detection-artificial-intelligence

Link to dataset used in VGG model: https://www.kaggle.com/datasets/meetnagadia/human-action-recognition-har-dataset

HAR applications using AI

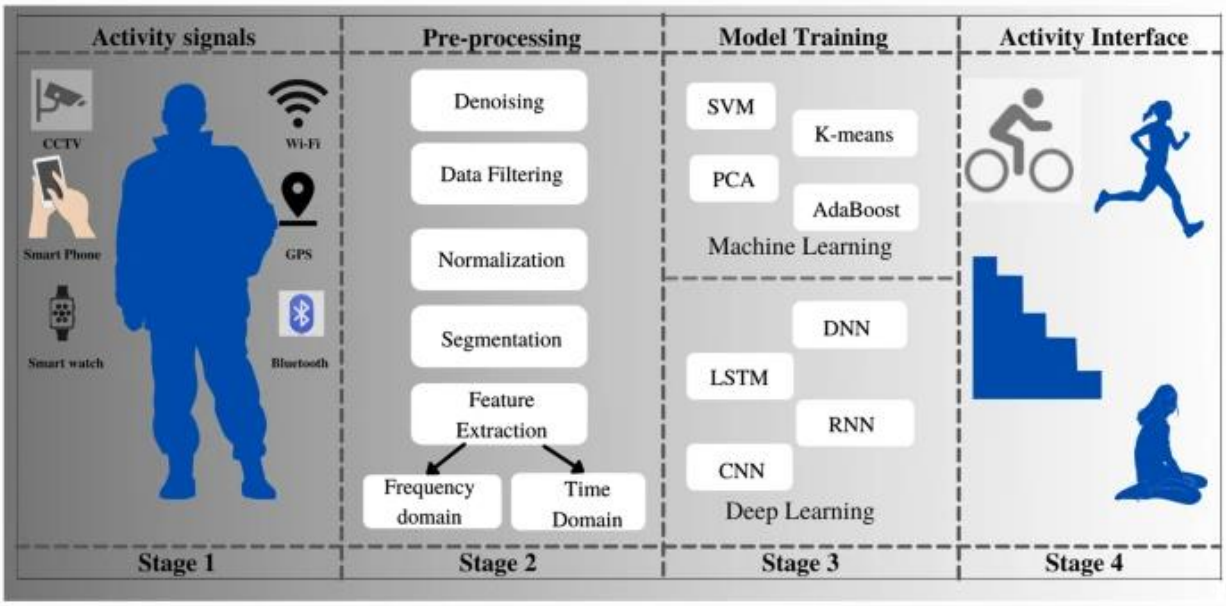
In the last decade, researchers have developed various HAR models for different domains. "What type of HAR device is suitable for which application domain and what is the suitable AI methodology" is the biggest question that pops into the mind, once developing the HAR framework. The description of diverse HAR applications with data sources and AI techniques is illustrated in Table 2. It shows the variation in HAR devices and AI techniques depending on the application domain. The pie chart in Fig. 4d shows the distribution of applications based on existing articles. HAR is used in fields like:

- Crowd surveillance (cSurv): Crowd pattern monitoring and detecting panic situations in the crowd.
- Health care monitoring (mHealthcare): Assistive care to ICU patients, Trauma resuscitation.
- Smart home (sHome): Care to elderly or dementia patients and child activity monitoring.
- Fall detection (fDetect): Detection of abnormality in action which results in a person's fall.
- Exercise monitoring (eMonitor): Pose estimation while doing exercise.
- Gait analysis (gAnalysis): Analyze *gait* patterns to monitor health problems.

STAGES

HAR consists of four stages (Fig. 1) including (1) capturing of signal activity, (2) data pre-processing, (3) AI-based activity recognition, and (4) the user interface for the management of HAR. Each stage can be implemented using several techniques bringing the HAR system to have multiple choices. Thus, the choice of the application domain, the type of data acquisition device, and the processing of artificial intelligence (AI) algorithms for activity detection makes the choices even more challenging.

Fig. 1



Libraries

1.Pandas

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

2.Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

3.OpenCV

Open CV is a library of programming functions mainly aimed at real-time computer vision.

4.TensorFlow

Tensor Flow is a free and open-source software library for machine learning and artificial intelligence.

5. Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

6.Keras

Keras is an open-source software library that provides a Python interface for artificial neural networks

Image Processing

Image processing involves the following basic steps:

- Importing image using Image acquisition tools
- Image Pre-processing / Analysing and manipulating images.
- Output in which either you can alter an image or make some analysis out of it.

We are going to use the OpenCV library for all the image pre-processing tasks. OpenCV reads data from a contiguous memory locationWe will use OpenCV library for resizing the images and creating feature vectors out of it, that can be achieved by converting the image data to numpy arrays. We will use one of the extensions of Deep Neural Nets named CNN(Convolutional Neural Network) for training the model.

Cv2.imread-to read images

Cv2.resize-to resize images

Img/255-convert to pixel

Cv2.cvtColor

CNN

Convolutional Neural Network or CNN is a type of artificial neural network, which is widely used for image/object recognition and classification. Deep Learning thus recognizes objects in an image by using a CNN. There are four types of layers for a convolutional neural network: the convolutional layer, the pooling layer, the ReLU correction layer and the fully-connected layer.

Elements of a Neural Network

Input Layer

This layer accepts input features. It provides information from the outside world to the network, no computation is performed at this layer, nodes here just pass on the information(features) to the hidden layer.

Hidden Layer

Nodes of this layer are not exposed to the outer world, they are the part of the abstraction provided by any neural network. Hidden layer performs all sort of computation on the features entered through the input layer and transfer the result to the output layer

Output Layer

This layer bring up the information learned by the network to the outer world.

The window that moves is called **Kernel.** The distance that the window moves each time is called **stride**. The goal of a convolutional layer is **filtering.** As we move over an image we effectively check for patterns in that section of the image. This works because of **filters,** stacks of weights represented as a vector, which are multiplied by the values output by the convolution.

Pooling works similar to convolution, the difference is the function that is applied to the kernel and the image window isn't linear. The most common pooling functions are **Max pooling** and **Average pooling**. Max pooling takes the max value from the window, while average pooling takes the average of all the values in the window.

RELU is an activation function, that squash the values into a range, typically [0,1] or [-1,1]. **Softmax** is a probabilistic function that allows us to express our inputs as a discrete probability distribution.

The implementation will constitute the following steps:

- Collect training data.
- Train the model using CNN.

Optimizer

An optimizer is a function or an algorithm that modifies the attributes of the neural network, such as weights and learning rate. Thus, it helps in reducing the overall loss and improve the accuracy.The optimizer used here is **Adam.** Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models

Loss function

Loss function is a method of evaluating how well your algorithm is modeling your dataset.Loss function here is **SparseCategoricalcrossentropy**

SparseCategoricalcrossentropy: Used as a loss function for multi-class classification model where the output label is assigned integer value (0, 1, 2, 3...). This loss function is mathematically same as the categorical_crossentropy

Activation function

Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron

1). Linear Function
- Range : -inf to +inf

2). Sigmoid Function :-
- It is a function which is plotted as 'S' shaped graph.
- Value Range : 0 to 1

3). Tanh Function
- Value Range :- -1 to +1

4)Relu
- Value Range :- [0, inf)
- ReLu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations
- RELU learns *much faster* than sigmoid and Tanh function.

5). Softmax Function :- The softmax function is also a type of sigmoid function but is handy when we are trying to handle mult- class classification problems.
- Output:- The softmax function is ideally used in the output layer of the classifier where we are actually trying to attain the probabilities to define the class of each input.
- *RELU* is a general activation function in hidden layers and is used in most cases these days.
- If your output is for binary classification then, *sigmoid function* is very natural choice for output layer.

After creating CNN model and compilation is done with optimizer Adam and loss function Sparse-categorical crossentropy and metrics as accuracy. Model is trained for 15 epochs and batch size 70 , the accuracy is high 99.95 but validation accuracy was too low and validation loss is high and loss were low. There comes overfitting condition

Overfitting

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data**.**
So in order to increase accuracy data augmentation is done and dropout layer is added and early stopping is done at 10th epoch

How can you avoid it?

Overfitting makes the model relevant to its data set only, and irrelevant to any other data sets. Some of the methods used to prevent overfitting include **ensembling, data augmentation, data simplification, and cross-validation**.

Data augmentation
Data augmentation in data analysis are techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. It acts as a regularizer and helps reduce overfitting when training a machine learning model.

Dropout Layer
Another typical characteristic of CNNs is a Dropout layer. The Dropout layer is a mask that nullifies the contribution of some neurons towards the next layer and leaves unmodified all others.

Early stopping
Early stopping is a feature that enables the training to be automatically stopped when a chosen metric has stopped improving. You can see it as a form of regularization used to avoid overfitting.

But after all these steps and model is trained for 50 epochs and batch-size 100,the highest accuracy is 50.18 and validation accuracy is 43.30.So the accuracy is too low.So both are low and underfitting comes

Underfitting

Underfitting refers to a model that can neither performs well on the training data nor generalize to new data. Reasons for Underfitting: High bias and low variance. The size of the training dataset used is not enough.

In order to increase accuracy the Image dataset is trained using VGG model

VGG model

VGG stands for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. The "deep" refers to the number of layers with VGG-16 or VGG-19 consisting of 16 and 19 convolutional layers.

VGG-16

The VGG model, or VGGNet, that supports 16 layers is also referred to as VGG16, which is a convolutional neural network model proposed by A. Zisserman and K. Simonyan from the University of Oxford. These researchers published their model in the research paper titled, "Very Deep Convolutional Networks for Large-Scale Image Recognition." The VGG16 model achieves almost 92.7% top-5 test accuracy in ImageNet. ImageNet is a dataset consisting of more than 14 million images belonging to nearly 1000 classes.

VGG-19

The concept of the VGG19 model (also VGGNet-19) is the same as the VGG16 except that it supports 19 layers. The "16" and "19" stand for the number of weight layers in the model (convolutional layers). This means that VGG19 has three more convolutional layers than VGG16.

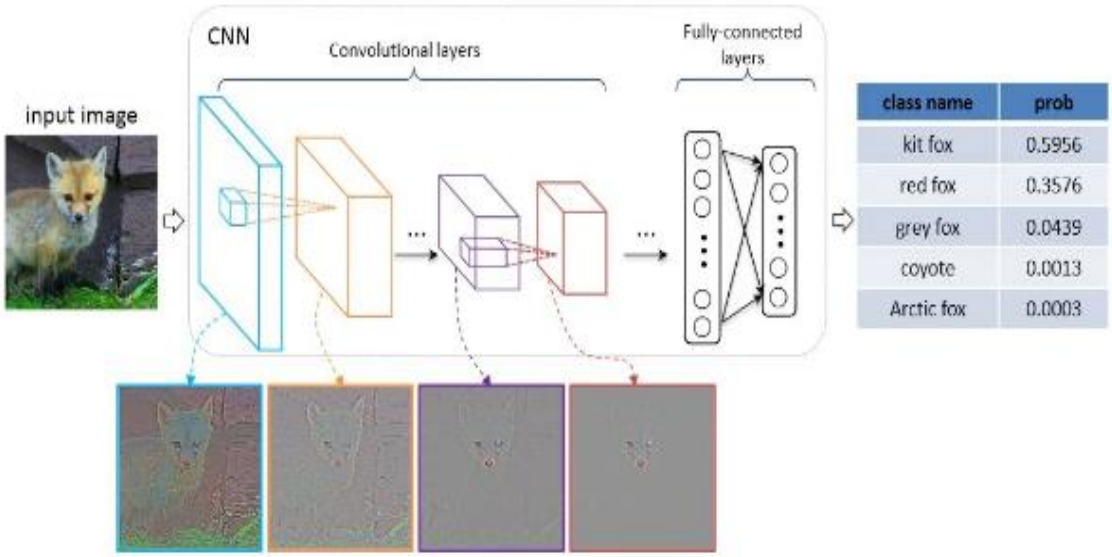
Let's take a brief look at the architecture of VGG:

Input-The VGGNet takes in an image input size of 224×224.

Convolutional Layers- VGG's convolutional layers leverage a minimal receptive field, i.e., 3×3, the smallest possible size that still captures up/down and left/right. Moreover, there are also 1×1 convolution filters acting as a linear transformation of the input. This is followed by a ReLU unit, which is a huge innovation from AlexNet that reduces training time. ReLU stands for rectified linear unit activation function; it is a piecewise linear function that will output the input if positive; otherwise, the output is zero.

Hidden Layers- All the hidden layers in the VGG network use ReLU. VGG does not usually leverage Local Response Normalization (LRN) as it increases memory consumption and training time. Moreover, it makes no improvements to overall accuracy.

Fully-Connected Layers- The VGGNet has three fully connected layers. Out of the three layers, the first two have 4096 channels each, and the third has 1000 channels, 1 for each class



In VGG model creation, we have both csv and Image datasets. Filename and activity (label) are taken using csv file for the image dataset. Preprocessing is done by Image data generator.Model creation x and y are given and converted to arrays using numpy . And the VGG model is created using sequential and pretrained using VGG16 and pretrained model is created.VGG model compilation done using optimizer Adam and loss function Categorical cross entropy and metrics given is accuracy.

VGG model is trained with 10 epochs and the highest accuracy gained by the model is 99.17 and loss is 5. It overcomes the overfitting condition that the model faced last time

Model save

Both accuracy and loss graphs are plotted.and and model is saved using model.save_weights and “main project HAR VGG”.

Performance evaluation

Researchers have adopted different metrics for evaluating the performance of HAR models, and the most popular evaluation metric is accuracy. The evaluation metrics used in existing vision-based HAR models were accuracy . The metrics used in Device-free HAR include F1-score, precision, recall, and accuracy.

VGG Model prediction are done and gained accuracy of 99.17 and loss of 5 ,predicted the output correctly and with probability 87.48

CONCLUSION

Human activity recognition (HAR) can be referred to as the art of identifying and naming activities using Artificial Intelligence (AI) from the gathered activity raw data by utilizing various sources. The foremost goal of HAR is to predict the movement or action of a person based on the action data collected from a data acquisition device. These movements include activities like eating, sleeping, running. It is challenging to predict movements, as it involves huge amounts of unlabelled sensor data, and video data which suffer from conditions like lights, background noise, and scale variation. To overcome these challenges AI framework offers numerous ML, and DL techniques.

Image dataset loaded from kaggle and preprocessing steps are done and model is created using CNN. When model is trained and overfitting condition occurred. To overcome overfitting data augmentation and dropout layer added, But accuracy became low and underfitting occurred .Inorder to overcome that model is created and trained using VGG16 model and acquired 99.17 accuracy and loss 5.and plotted graphs of accuracy and loss

ABBREVIATIONS

ADL:
Activity of daily living
AI:
Artificial intelligence
AUC:
Area under curve
BS:
Batch size
CNN:
Convolution neural network
CONV:
Convolution
CV:
Cross validation
DL:
Deep learning
DO:
Drop out
FC:
Fully connected
GPU:
Graphics processing unit
HAR:
Human activity recognition
LR:
Learning rate
LSTM:
Long short-term memory
MAE:
Mean absolute error
ML:
Machine learning
MP:
Max pooling
MSE:
Mean square error