Deep learning using Human Action Recognition From Depth Maps And Postures

Dr Sathiyaraj R[1], T.Ramya Priya [2], E.Rupa[3], P srimalu[4], J.Anandarao[5], k .Mounica[6]

[1] professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

[2-6]Assistant professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

ABSTRACT

Human Activity Recognition is one of the active research areas in computer vision for various contexts like security surveillance, healthcare and human computer interaction. Over the past years, several methods published for human action recognition using RGB (red, green, and blue), depth, and skeleton datasets. Most of the methods introduced for action classification using skeleton datasets are constrained in some perspectives including features representation, complexity, and performance. However, there is still a challenging problem of providing an effective and efficient method for human action discrimination using a skeleton dataset. The first input is depth images and second input is a proposed moving joints descriptor (MJD) which represents the motion of body joints over time, in order to maximize feature extraction for accurate action classification, CNN channels are trained with different inputs and for Score fusion we are planning to use neural networks. Our proposed method was implementation on public datasets like MSRAction3D

1. INTRODUCTION

Human action recognition has been a popular research topic for a long time, not only because of their widespread application in a variety of applications, such as intelligent surveillance systems, human-robot interaction, and home care systems, but also because it remains a difficult research problem. Convolutional Neural Networks (CNN) have been widely used in many research areas, particularly computer vision and pattern recognition, thanks to the advancement of deep learning over the last few years, and have achieved remarkable performance on classification, detection, segmentation, and tracking tasks. In the subject of action recognition, there are a few things to keep in mind. As human-machine interaction becomes one of the most researched topics in multimedia

processing, traditional communication techniques are being developed in order to address technological advancements and enable disabled people to communicate with machines and understand their activities using computer computing. Many research works have attempted to model and then recognise people's behavior using motion analysis. In this paper, we focus on human behaviour analysis from video scenes, and it is worth noting that many information is hid behind gesture, rapid motion, and walking pace[1],[2].Due to the expressive features provided by the two types of data, recent human action recognition research has been directed toward using depth maps or body postures to represent the action[3]. A strong representation that gives distinct aspects of each action for classification is critical to the success of an action recognition algorithm. For some activities, using depth map data for action recognition is still confusing, resulting in incorrect classification because two actions may appear similar from the front, but they appear differently from the side[4].When substantial occlusions occur, the depth maps collected by the depth cameras are quite noisy, and the 3D positions of the tracked joints may be completely incorrect, increasing intraclass variances in the actions[5].They propose "Skepxels," a

spatio-temporal format for skeletal sequences that uses CNN's 2D convolution kernels to fully leverage "local" connections between joints. Thye use Skepxels to convert skeleton movies into images with adjustable dimensions, and then use the generated images to build a CNN-based framework for successful human action recognition.

2. LITERATURE SURVEY

Zhao-Xuan Yang [7] Two single-view RGB datasets (KTH and TJU), two well-known depth datasets (MSR action 3-D and MSR daily activity 3-D), and one revolutionary multiview multimodal dataset are all used to validate the proposed methodology (MV-TJU). In both RGB and depth modalities, the extensive experimental data show that this method outperform the popular 2-D/3-D component model-based approaches and other competitive techniques for numerous human action recognition. C. Krishna Mohan:[8] They proposed leveraging action bank features to recognise human actions in videos using a deep fully convolutional architecture. Action bank features are linear patterns that describe the resemblance of the video to the action bank films. They are computed against a predefined set of pictures defined as an action ban Miki, Hiroshi:[9] They offer a method for recognising objects that examines the relationship between human behaviour and object functions, with the goal of improving our method by adding human activities into dynamic object segmentation. Ziaeefard, Maryam :[10] This research proposes a revolutionary normalized- polar histogram-based human

action recognition algorithm. The action of accumulating skeletonized pictures was described.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Sometimes might be finding products is easy than waiting in the billing queue because it consumes more time of the customer. So now by taking the motivation of this scenario which was regularly done in all the Shoppe we are designing this system which can be benefited for the customer in all the means and also it was benefited for the Shoppe owner also. So, we design a system by this, the customer can know their bill while adding the items in the cart. The best and most useful example of this Supermarket Basket is that if a customer purchases can easily billed.

3.2 PROPOSED SYSTEM

This system brings new innovation than existing shopping system. The main purpose of this project is to provide centralized and automated billing system using web. Along with the automatic billing some special features incorporated are along .We use new term that is SupermarketBasket

3.3 SYSTEM STUDY

3.3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

♦ ECONOMICAL FEASIBILITY

♦ TECHNICAL FEASIBILITY

♦ SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4. SYSTEM DESIGN

4.1 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-

purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a Qnotation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS :

The Primary goals in the design of the UMLare as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.

3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modeling language.

5. Encourage the growth of OO tools market.

6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

7. Integrate best practices.

USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case

diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.
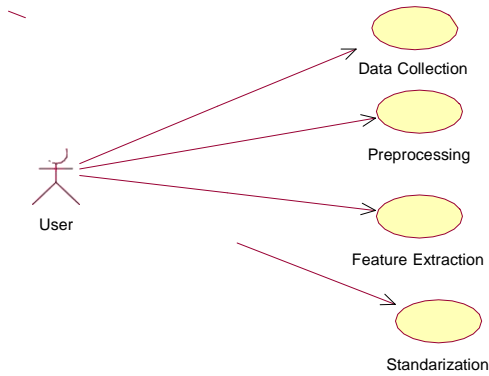

CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.
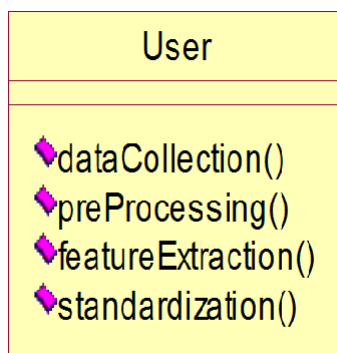
Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.
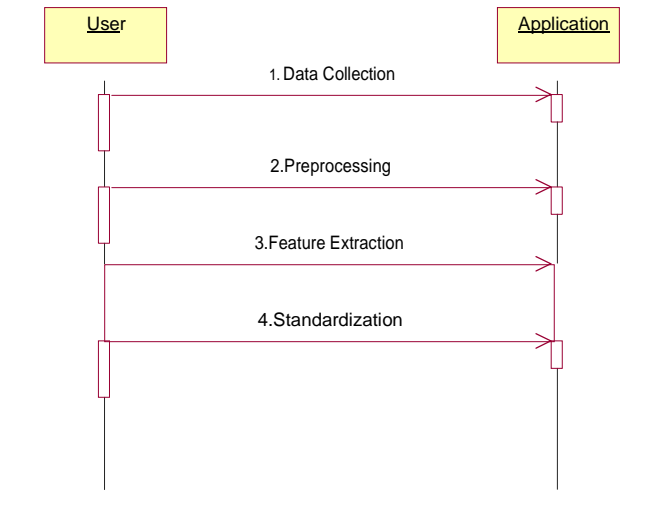
COLLEBORATION DIAGRAM
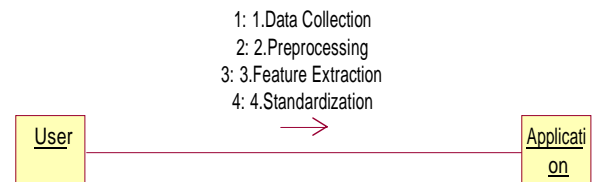
SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message

4.2 IMPLEMENTA

TIONMODULES :

Data Collection:

The first step is to collect multivariatetime series data from the phone's and the watch's sensors. The sensors are sampled with a constant frequency of 30 Hz. After that, the sliding window approach is utilized for segmentation, where the time series is divided into subsequent windows of fixed duration without interwindow gaps (Banos et al., 2014). The sliding window approach does not require preprocessing of the time

series, and is therefore ideally suited to real-

time applications.

Preprocessing:

Filtering is performed afterwards to remove noisy values and outliers from the accelerometer time series data, so that it will be appropriate for the feature extraction stage. There are two basic types of filters that are usually used in this step: average filter (Sharma et al., 2008) or median filter (Thiemjarus, 2010). Since the type of noise dealt with here is similar to the salt and pepper noise found in images, that is, extreme acceleration values that occur in single snapshots scattered throughout the time series. Therefore, a median filter of order 3 (window size) is applied to removethis kind of noise.

Feature Extraction:

Here, each resulting segment will be summarized by a fixed number of features, i.e., one feature vector per segment. The used features are extracted from both time and frequency domains. Since, many activities have a repetitive nature, i.e., they consist of a set of movements that are done periodically like walking and running. Thisfrequency of repetition, also known as dominant frequency, is a descriptive featureand thus, it has been taken into consideration.

Standardization:

Since, the time domain features are measured in (m/s 2), while the frequency ones in (Hz), therefore, all features should

have the same scale for a fair comparison between them, as some classification algorithms use distance metrics. In this step, Z-Score standardization is used, which will transform the attributes to have zero mean and unit variance, and is defined as

$$x_{new} = (x-\mu)/\ \sigma$$

where μ and σ are the attribute's mean and standard deviation respectively (Gyllensten,2010).

5. SOFTWARE DOCUMENTATION

5.1 Wat is Python :-

Below are some facts about Python. Python is currently the most widely used multi-purpose, high-level programminglanguage.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram,

Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard library which can be used for the following

Machine Learning

- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- eb frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv,Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python :-

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensivelibrary and *it* contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. *So,* we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility,

Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language Improved Productivity

The language's simplicity andextensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get morethings done.

4. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

5. Simple and Easy

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

6. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

7. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability,

classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help youwith your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system- dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3.Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Pythonare enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone- based applications. One such application iscalled **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don't need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seemsunnecessary.

This was all about the Advantages and Disadvantages of Python ProgrammingLanguage.

History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC'sinfluence because I'm indebted to everything I learned during that project andto the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration withABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created

a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table(or dictionary, as we call it), a list, strings, and numbers."

What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning. *Supervised learning* involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks

and *regression* tasks: in classification, the

labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such *as* clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are –

Quality of data – Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task – Another challenge faced by ML models is the consumption of time especially for data

acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are

some real-world applications of ML

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?

Arthur Samuel coined the term **"Machine Learning"** in 1959 and defined it as a **"Field of study that gives computers the capability to learn without being explicitly programmed".**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a *344%* growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what

exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&Din Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms fromscratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!! Some of the key concepts in statistics that are important are Statistical Significance,

Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial

Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it's bestif you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

Model – A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.

Feature – A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.

Target (Label) – A target variable or label

is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

Training – The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

Prediction – Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) **Types of Machine Learning**

Supervised Learning – This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

(c) **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.

(d) **Semi-supervised Learning –** This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and isalso more cost-effective than SupervisedLearning.

Reinforcement Learning – This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the

current state and that will maximize thereward in the future.

Advantages of Machine learning :-

1. Easily identifies trends andpatterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysityour project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi- dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while alsotargeting the right customers.

Disadvantages of Machine Learning:

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality.

There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also

needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

Python Development Steps : -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of alist must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.

- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour. Text Vs. Data Instead Of Unicode Vs. 8-bit

Purpose :-

We demonstrated that our approach enables successful segmentation of intra- retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

- Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large

and comprehensive standardlibrary.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly

to write your programs. Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this maybe an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quickto implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project :- Tensorflow

TensorFlow is a free and open- source software library for dataflow and differentiable programming across a range of tasks. It is a

symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy

Numpy is a general-purpose array- processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including theseimportant ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting)functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using

Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, thePython and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. Forexamples, see the sampleplots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge

standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac :

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first releasedin the year 1991

and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does

6. SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTS

6.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be

validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

 Valid Input   : identified classes of valid input must be accepted.

 Invalid Input: identified classes of invalid input must be rejected.

Functions   : identified functions must be exercised.

Output   : identified classes of application outputs must be   exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing

pre- driven process links and integration points.

6.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

6.1.7 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

6.2 Test strategy and approach

Field testing will be performed manuallyand functional tests will be written in detail.

Test objectives

- All field entries must work properly.

- Pages must be activated from the identified link.

- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format

- No duplicate entries should be allowed

- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures

that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7. CONCLUSION

The article presents the use of depth image sequences for recognizing people's activities. The subject of the research is a method using only depth information. The way of determining the features for classification is based on the use of the VFH point cloud descriptors of the 3D point clouds determined from respective depth maps. Three approaches to recognizing activities are considered: using the k-NN classifiers based on DTW measure, using the BiLSTM neural networks, using a fusion of the BiLSTM networks based on the fuzzy integral. Results of the classification experiments obtained for the representative, extensive datasets UTD-MHAD and MSR 3D are comparable or better to those known from the literature.

FUTURE ENHANCEMENT

The efficiency of models can be improved by increasing the amount of data. Also,the disease can be classified on the gender-based such that we can get information about whether male are affected more or females. More feature engineering is needed for better results and deep learning approach can be used in future.

BIBLIOGRAPHY

1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in Proc. IEEE 17th Int. Conf. Pattern Recognit., vol. 3. Cambridge, U.K., Aug. 2004, pp. 32–36.

2] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-

S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Miami, FL, USA, Jun. 2009, pp. 2004–2011.

3] I. Laptev, M. Marszalek, C. Schmid, and

B. Rozenfeld, "Learning realistic human actions from movies," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Anchorage, AK, USA, Jun. 2008, pp. 1–8.

4] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops, San Francisco, CA, USA, Jun. 2010, pp. 9–14. 5] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1290–1297, IEEE, 2012

6] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Skeleton optical spectra based action recognition using convolutional neural networks," arXiv preprint arXiv:1703.03492, 2016.

7] Liu, An-An; Su, Yu-Ting; Jia, Ping-Ping; Gao, Zan; Hao, Tong; Yang, Zhao-Xuan (2015). Multipe/Single-View Human

Action Recognition via Part-Induced Multitask Structural Learning. IEEE Transactions on Cybernetics, 45(6), 1194–

1208. doi:10.1109/tcyb.2014.2347057

8] Ijjina, Earnest Paul; Mohan, C. Krishna (2014). [IEEE 2014 13th International Conference on Machine Learning and Applications (ICMLA) - Detroit, MI, USA (2014.12.3-2014.12.6)] 2014 13th

International Conference on Machine Learning and Applications - Human Action Recognition Based on Recognition of Linear Patterns in Action Bank Features Using Convolutional Neural Networks. , (), 178–182. doi:10.1109/icmla.2014.33

9] Miki, Hiroshi; Kojima, Atsuhiro; Kise, Koichi (2008). [IEEE 2008 Second International Conference on Future Generation Communication and Networking (FGCN) - Hainan, China (2008.12.13-2008.12.15)] 2008 Second

International Conference on Future Generation Communication and Networking - Environment

Recognition Based on Human Actions Using Probability Networks. , (), 441–446. doi:10.1109/fgcn.2008.62

10] Ziaeefard, Maryam; Ebrahimnezhad, Hossein (2010). [IEEE 2010 20th

International Conference on Pattern Recognition (ICPR) - Istanbul, Turkey (2010.08.23- 2010.08.26)] 2010 20th

International Conference on Pattern Recognition - Hierarchical Human Action Recognition by Normalized-Polar Histogram. , (), 3720–3723.

doi:10.1109/ICPR.2010.906 11 ALLEL,

Mejdi; HAVARD, Vincent; BAUDRY, David; SAVATIER, Xavier (2020). [IEEE

2020 IEEE International Conference on Human-Machine Systems (ICHMS) - Rome, Italy (2020.9.7-2020.9.9)] 2020

IEEE International Conference on HumanMachine Systems (ICHMS) - InHARD - Industrial Human Action Recognition Dataset in the Context of

Industrial Collaborative Robotics. , (), 1–6. doi:10.1109/ichms49158.2020.9209531 n

Xu ,HongLin Yuan ,"A RF Fingerprint Recognition Method Based on Deeply Convolutional Neural Network"IEEE 5th Information Technology and Mechatronics Engineering Conference ,2020.

12] Xiao, Qinkun; Si, Yang (2017). [IEEE 2017 3rd IEEE International Conference on Computer and Communications (ICCC) - Chengdu (2017.12.13-2017.12.16)] 2017

3rd IEEE International Conference on Computer and Communications (ICCC) - Human action recognition using autoencoder. , (), 1672–1675. doi:10.1109/CompComm.2017.8322824 13

Chen, Chen; Jafari, Roozbeh; Kehtarnavaz, Nasser (2016). A Real-Time Human Action Recognition System Using Depth and Inertial Sensor Fusion. IEEE Sensors Journal, 16(3), 773–781.   doi:10.1109/JSEN.2015.2487358