**ROAD ACCIDENT DETECTON**

**ABSTRACT**

In today’s era, need of efficient accident detection has drawn much attention as number of accidents are increasing day by day. One of the widely employed method is to use accelerometer to detect a crash. In this method, acceleration (g) value measured from the accelerometer is calibrated to detect an accident. This method, however is limited by the accuracy of the accelerometer. To make an efficient accident detection system, convolutional neural network (CNN) methodology can be incorporated in the system. CNN is the state-of-the-art method for image classification. In the recent work, image classification has been used to detect accident. However, CNN takes large time, data and computing power to be trained. To mitigate these issues, transfer learning technique has been innovatively incorporated for the accident detection application, which involves retraining the already trained network. Inception-v3 is an image classifier developed by google, which is incorporated for this purpose. In this work, accident detection system is designed using advanced and efficient Transfer Learning algorithm, which gives 84.5% of accuracy. Also, an effective comparison between this advanced method and the traditional accelerometer based technique have been made.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT**  **LIST OF FIGURES**  **LIST OF SYMBOLS**  **LIST OF ABBREVIATIONS**  **LIST OF TABLES** | i  v  vii  xi  xii |
| 1. | **CHAPTER 1 : INTRODUCTION**   * 1. GENERAL   2. OBJECTIVE   3. EXISTING SYSTEM   1.3.1EXISTINGSYSTEM DISADVANTAGES  1.3.2 LITERATURE SURVEY  1.4 PROPOSED SYSTEM  1.4.1 PROPOSED SYSTEM ADVANTAGES |  |
| 2. | **CHAPTER 2 :PROJECT DESCRIPTION**  2.1 GENERAL  2.2 METHODOLOGIES  2.2.1 MODULES NAME  2.2.2 MODULES EXPLANATION  2.2.3 MODULE DIAGRAM  2.2.4GIVEN INPUTAND EXPECTED OUTPUT  2.3 TECHNIQUE OR ALGORITHM |  |
| 3. | **CHAPTER 3 : REQUIREMENTS**  3.1 General  3.2 Hardware REQUIREMENTS  3.3 Software REQUIREMENTS |  |
| 4. | **CHAPTER 4 : SYSTEM DESIGN**  **4.1 general**  4.1.1 activity diagram  4.1.2 USE CASE DIAGRAM  4.1.3 DATA FLOW DIAGRAM  4.1.4SEQUENCE DIAGRAM  4.1.5 COLLABORATION DIAGRAM  4.1.6CLASS DIAGRAM  4.1.7 SYSTEM ARCHITECTURE  4.1.8 OBJECT DIAGRAM  4.1.9 STATE DIAGRAM  4.1.10 COMPONENT DIAGRAM  4.1.11 E-R DIAGRAM  4.2 DATABASE DESIGN (ALL LEVEL |  |
| 5. | **CHAPTER 5 :SOFTWARE SPECIFICATION**  5.1 Python  5.2 History of python  5.3 impartance of python  5.4 features of python  5.5 libraries used |  |
| 6. | **CHAPTER 6 : IMPLEMENTATION**  6.1 GENERAL  6.2 IMPLEMENTATION  6.3 DATA BASE TABLE STRUCTURE |  |
| 7. | **CHAPTER 7 : SNAPSHOTS**  7.1 GENERAL  7.2 VARIOUS SNAPSHOTS |  |
| 8. | **CHAPTER 8 : SOFTWARE TESTING**  8.1 GENERAL  8.2 DEVELOPING METHODOLOGIES  8.3 TYPES OF TESTING |  |
| 9. | **CHAPTER 9 :**  **APPLICATIONS AND FUTURE ENHANCEMENT**  9.1 GENERAL  9.2 APPLICATIONS  9.3 FUTURE ENHANCEMENTS |  |
| 10. | **CHAPTER 10 :**  10.1CONCLUSION  10.2 REFERENCES |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO.** |
| 2.3.2 | Module Diagram |  |
| 4.1 | Use case Diagram |  |
| 4.2 | Class Diagram |  |
| 4.3 | Object Diagram |  |
| 4.4 | Component Diagram |  |
| 4.5 | Deployment Diagram |  |
| 4.6 | Sequence diagram |  |
| 4.7 | Collaboration diagram |  |
| 4.8 | State Diagram |  |
| 4.9 | Activity Diagram |  |
| 4.10 | Data flow diagram |  |
| 4.11 | E-R Diagram |  |
| 4.12 | Architecture Diagram |  |

**LIST OF SYSMBOLS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NOTATION**  **NAME** | **NOTATION** | **DESCRIPTION** |
| 1. | Class | *Class Name*  *-attribute*  *-attribute*  *+operation*  *+operation*  *+operation*  *+ public*  *-private*  *# protected* | Represents a collection of similar entities grouped together. |
| 2. | Association | name  Class B  Class A  Class A  Class B | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
| 3. | Actor | Class A  Class A  Class B  Class B | It aggregates several classes into a single classes. |
| 4. | Aggregation | Interaction between the system and external environment |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | Relation  (uses) | uses | Used for additional process communication. |
| 6. | Relation  (extends) | extends | Extends relationship is used when one use case is similar to another use case but does a bit more. |
| 7. | Communication |  | Communication between various use cases. |
| 8. | State | State | State of the processs. |
| 9. | Initial State |  | Initial state of the object |
| 10. | Final state |  | F inal state of the object |
| 11. | Control flow |  | Represents various control flow between the states. |
| 12. | Decision box |  | Represents decision making process from a constraint |
| 13. | Usecase |  | Interact ion between the system and external environment. |

|  |  |  |  |
| --- | --- | --- | --- |
| 14. | Component |  | Represents physical modules which are a collection of components. |
| 15. | Node |  | Represents physical modules which are a collection of components. |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or acion. |
| 17. | External entity |  | Represents external entities such as keyboard,sensors,etc. |
| 18. | Transition |  | Represents communication that occurs between processes. |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object communications. |
| 20. | Message | Message | Represents the message exchanged. |

**LIST OF ABBREVATION**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ABBREVATION** | **EXPANSION** |
| 1**.** | ML | Machine Learning |
| 2. | SVM | Support Vector Machine |
| 3. | COMPUTER VISION & IMAGE PROCESSING TECHNIQUES | Convolutional Neural Networks |
| 4. | ANN | Artificial Neural Networks |
| 5. | AI | Artificial Intelligence |
| 6. | DNN | Deep Neural Networks |

**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL**

With increase in number of individuals owning vehicle, road accidents are also increasing correspondingly. Vehicles have become inevitable part of everyone’s life nowadays. With increasing congestion, limited sense of traffic rules among citizens, and lack of enforcement of traffic rules, causalities and death rates due to road accidents are enormous and among the highest of all deaths. Most of the road casualties happen on highways where most of the vehicles are heavy and moving with higher speeds. According to World Health Organization (WHO), it is reported that there were 1.25 million road traffic deaths in 2013 globally. It is also predicted that this number is likely to increase in future. Low and middle income countries have higher road fatalities than higher income countries due to lack of law enforcement and poor availability of emergency medication facilities.

Various social activists, researchers and concerned agencies have made several attempts and proposed solutions in the direction to lessen road accidents and provide viable solutions or systems to mitigate road accidents. For the fast relief operation initialization, some automated system deployment in the vehicle itself shall be very beneficial. In the present work, crash detection system in vehicle that senses accident and also raises emergency help is devised. Many times victim of road accidents die due to delay in getting medical treatment. So, in case of accident, the proposed system can save many lives. There have been a few methods proposed and implemented for accident detection system till date. Different methods are briefly discussed here. In all these cases different methods are used to detect accident, however GPS and GSM are common techniques used to locate the victim and inform emergency services respectively, Vehicular AdHoc Networks (VANETs) based approach has been used for accident detection system.

In this system, accident is detected by accelerometer and all the drivers within a certain range receives an alert to inform them about the crash in nearby area and to stop further casualties. Message is transmitted using 802.11g standards. In, GPS based method is used to detect accident. GPS measures speed of vehicle every second and compares it with the previous readings. The algorithm is so designed that, it detects accident whenever the speed of the vehicle is below a certain level. Android app based accident detection system using tilt of accelerometer is described in. But it detects accident only when the vehicle is tilted with reference to the road, which is not the case every time.

In, accident is detected based on sensors system. The authors have gathered data from 400 sensors and 72 accident incidents from recent past in Turkish highways. These are then preprocessed and analyzed. Based on some features like velocity difference, occupancy difference, capacity usage difference, weekend/weekday, rush hour - accident is detected. But the system applicability is limited as malfunctioning of sensors raise false alarms which is a bottleneck in performance accuracy.

An accident detection unit with accelerometer using ARM7, which is equipped with gas and alcohol sensors is described in. Accident detection using some machine learning techniques such as Neural network, SVM, decision trees are described in. Machine learning is progressively making its way in almost every area nowadays. Innovatively, the method of accident detection using image classification with the use of machine learning is proposed in this paper. Transfer learning can be applied on the already trained network of Google’s Inception V3, which is Google’s image classifier on the top of tensor flow, which uses convolutional neural networks to classify images among given classes.

* 1. **OBJECTIVE**

In this section, we have tried to compare our work with other accident detection techniques. Most of the studies in this field revolve around the enhancement of tangible infrastructure rather than on Intelligent Transportation Systems (ITS) which include traffic congestion detection, accident detection, detecting the occurrence of an event etc. Even the few existing studies in the domain lack implementation details and are terrain specific i.e. there are constraints both in the geographical as well as demographic aspects.

**Existing System:**

In this section, we have tried to compare our work with other accident detection techniques. Most of the studies in this field revolve around the enhancement of tangible infrastructure rather than on Intelligent Transportation Systems (ITS) which include traffic congestion detection, accident detection, detecting the occurrence of an event etc. Even the few existing studies in the domain lack implementation details and are terrain specific i.e. there are constraints both in the geographical as well as demographic aspects. These techniques have been discussed below: Lexus vehicles introduced in 2014 came with a feature called the “Lexus Enform” wherein an impact sensor was placed at the rear end of the vehicle. In the occurrence of an accident, the sensors would react and thus notify the user via the application. However, the disadvantages of this system were plenty. Sensors were to be placed in every individual vehicle rendering the concept expensive. Also, it requires physical entities like smartphones. An ancillary company of General Motors called OnStar Corporation introduced an accident notification application called Chevy star. It offered options like on-field assistance to victims as well as a self-regulated crash response. However, this service was based on a subscription model rendering the service expensive. Also, reviews suggested that the service lacked quality because of which the system itself was ineffective. SoSmart SpA came up with a smartphone application called SOSmart which provided free assistance to the victim of the accident at the time of occurrence. This facility was easy to use and you could avail help at the click of a button. But the obvious flaw is that it is a manual reporting system.

**Disadvantages:**

* This system were plenty.
* Sensors were to be placed in every individual vehicle rendering the concept expensive.

**LITERATURE SURVEY:**

**Title:** A Critical Issue Requiring Government Attention.

**Author:** P. M. Vitousek, H. A. Mooney, J. Lubchenco, and J. M. Melillo.

**Year:** 2015.

**Description:**

Human alteration of Earth is substantial and growing. Between one-third and one-half of the land surface has been transformed by human action; the carbon dioxide concentration in the atmosphere has increased by nearly 30 percent since the beginning of the Industrial Revolution; more atmospheric nitrogen is fixed by humanity than by all natural terrestrial sources combined; more than half of all accessible surface fresh water is put to use by humanity; and about one-quarter of the bird species on Earth have been driven to extinction. By these and other standards, it is clear that we live on a human-dominated planet.

**Title:** Accelerometer Based Real-Time Remote Detection and Monitoring of Hand Motion.

**Author:** An analysis of a large scale habitat monitoring application.

**Year:** 2011.

**Description:**

Habitat and environmental monitoring is a driving application for wireless sensor networks. We present an analysis of data from a second generation sensor networks deployed during the summer and autumn of 2003. During a 4 month deployment, these networks, consisting of 150 devices, produced unique datasets for both systems and biological analysis. This paper focuses on nodal and network performance, with an emphasis on lifetime, reliability, and the the static and dynamic aspects of single and multi-hop networks. We compare the results collected to expectations set during the design phase: we were able to accurately predict lifetime of the single-hop network, but we underestimated the impact of multi-hop traffic overhearing and the nuances of power source selection. While initial packet loss data was commensurate with lab experiments, over the duration of the deployment, reliability of the backend infrastructure and the transit network had a dominant impact on overall network performance. Finally, we evaluate the physical design of the sensor node based on deployment experience and a <i>post mortem</i> analysis. The results shed light on a number of design issues from network deployment, through selection of power sources to optimizations of routing decisions.

**Title:** Automatic road accident detection techniques.

**Author:** Chen, X.; Zhou, G.; Chen, A.; Pu, L.; Chen, W.

**Year:** 2001.

**Description:**

Decision support tools used for vegetation management require accurate information on the spatial array of different plant communities and an herbivore's grazing location. We tested the accuracy and precision of locations derived using the satellite navigation global positioning system (GPS). Before May 2000, the accuracy and precision of GPS-derived locations were degraded by a process known as selective availability (SA); after May 2000, SA was disabled. In this study we investigated how to handle and improve the quality of data generated both when SA was enabled and when SA was disabled using relative GPS (rGPS). rGPS entails the post-processed correction of the roving GPS module with simultaneously acquired positional errors recorded at a known stationary reference location. With SA enabled, GPS data were obtained at a fixed known location to obtain baseline information, and from a roving module that essentially mimicked surveying techniques or the movement of a free-ranging animal. The mean accuracy of GPS with SA enabled was 21 m for the fixed module and 25 m for the roving module. Use of rGPS and further manipulation of the data improved the mean accuracy of the data to 7 m for the fixed module and 10 m for the roving module. With SA disabled, data were similarly recorded from the fixed known location and resulted in a mean location accuracy of 5 m. The use of rGPS resulted in a significant improvement of this value to 3Ã¯Â¿Â½6 m and precision measured by the 95\% quantile was \< 10 m. For mapping and wildlife tracking, such quality in terms of location accuracy and precision is unprecedented and demonstrates that rGPS may still be useful in many applications. GPS enables the world-wide collection of accurate and precise location information at 1-second intervals. Furthermore, by programming the GPS receiver to overdetermined location by using information from all visible satellites, many of the limitations that arise in habitats or environments with a limited view of the sky may be overcome. With SA now disabled, the potential use of GPS will increase. With further miniaturization, surveying of remote featureless landscapes or the tracking of crepuscular or far-ranging animals will become more accurate and more quantifiable than ever before.

**Title:** Smart vehicle accident detection and alarming system using a smartphone.

**Author:** R. Kays, S. Tilak, B. Kranstauber, P. A. Jansen, C. Carbone, M. J. Rowcliffe, T. Fountain, J. Eggert, and Z. He,

**Year:** 2010.

**Description:**

Studying animal movement and distribution is of critical importance to addressing environmental challenges including invasive species, infectious diseases, climate and land-use change. Motion sensitive camera traps offer a visual sensor to record the presence of a broad range of species providing location -specific information on movement and behavior. Modern digital camera traps that record video present new analytical opportunities, but also new data management challenges. This paper describes our experience with a terrestrial animal monitoring system at Barro Colorado Island, Panama. Our camera network captured the spatio-temporal dynamics of terrestrial bird and mammal activity at the site - data relevant to immediate science questions, and long-term conservation issues. We believe that the experience gained and lessons learned during our yearlong deployment and testing of the camera traps as well as the developed solutions are applicable to broader sensor network applications and are valuable for the advancement of the sensor network research. We suggest that the continued development of these hardware, software, and analytical tools, in concert, offer an exciting sensor-network solution to monitoring of animal populations which could realistically scale over larger areas and time spans.

**Title:** Wireless System for Vehicle Accident Detection and Reporting using Accelerometer and GPS.

**Author:** Behera, S.K.; Rath, A.K.; Mahapatra, A.; Sethy, P.K.

**Year:** 2015.

**Description:**

How long does it take for the human visual system to process a complex natural image? Subjectively, recognition of familiar objects and scenes appears to be virtually instantaneous, but measuring this processing time experimentally has proved difficult. Behavioral measures such as reaction times can be used1, but these include not only visual processing but also the time required for response execution. However, event-related potentials (ERPs) can sometimes reveal signs of neural processing well before the motor output2. Here we use a go/no-go categorization task in which subjects have to decide whether a previously unseen photograph, flashed on for just 20 ms, and contains an animal. ERP analysis revealed a frontal negativity specific to no-go trials that develops roughly 150 ms after stimulus onset. We conclude that the visual processing needed to perform this highly demanding task can be achieved in under 150 ms.

**Proposed System**

Convolutional neural network is the state-of-the-art tech-nique in image classification and recognition. It has a stack of convolutional layers, ReLu layers and pooling layers. This is discussed in detail in [10]. Convolutional layer is feature extraction layer which detects from general features like edges to more specific features like objects and colors from the images. Pooling layer reduces the dimensionality. It takes much time and computing power to train such a network and very large image dataset is also required.

For example, Google’s inception-v3 image classifier is trained on the ImageNet dataset of 100000 images to classify among 1000 classes. It has 22 hidden layers and it took weeks to train it. Transfer learning technique comes in handy to avoid such limitations. Transfer learning refers to using learning from previous training session to a new training session.

**ADVANTAGES**

* It is very efficient to train data.
* It makes no assumptions about distributions of classes in feature space.
* It is very fast at classifying unknown records.

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 METHODOLOGIES**

**2.1.1** **MODULES NAME:**

1. Uniform Aspect Ratio

2. Image-Scaling

3. Normalizing Image Inputs

**1. Uniform Aspect Ratio**

It is the first step that is followed in the pre-processing of the image data. By applying this technique of unfirming all the images in the dataset, the model can be built seamlessly without any problems. Here each image in the dataset is brought to an equal size and shape for better execution of the model. All the images are brought into the same dimensions.

**2. Image-Scaling**

After unfirming all the image sizes in the dataset, by using the function image data generator from Keras package, the image scaling technique is used. Images are either upscaled or downscaled accordingly with the size fixed in the model.

**3. Normalizing Image Inputs**

This technique ensures the distribution of the data similarly according to the input parameters that are set previously. It helps in faster training of the data. The normalization is performed by deducting the average from each one of the pixels and dividing the obtained result with standard deviation**.**

**CHAPTER 3**

**REQUIREMENTS ENGINEERING**

**3.1 GENERAL**

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

* PROCESSOR : DUAL CORE 2 DUOS.
* RAM : 4GB DD RAM
* HARD DISK : 250 GB

**3.3 SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

* Operating System : Windows 7/8/10
* Platform : Spyder3
* Programming Language : Python
* Front End : Spyder3

**3.4 FUNCTIONAL REQUIREMENTS**

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

**3.5 NON-FUNCTIONAL REQUIREMENTS**

**EFFICIENCY**

Our multi-modal event tracking and evolution framework is suitable for multimedia documents from various social media platforms, which can not only effectively capture their multi-modal topics, but also obtain the evolutionary trends of social events and generate effective event summary details over time. Our proposed mmETM model can exploit the multi-modal property of social event, which can effectively model social media documents including long text with related images and learn the correlations between textual and visual modalities to separate the visual-representative topics and non-visual-representative topics.

**CHAPTER 4**

**DESIGN ENGINEERING**

**4.1 GENERAL**

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

**4.2 UML DIAGRAMS**

**4.2.1 USE CASE DIAGRAM**



**EXPLANATION:**

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. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

**4.2.2 CLASS DIAGRAM**

****

**EXPLANATION**

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

**4.2.3 OBJECT DIAGRAM**



**EXPLANATION:**

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

**4.2.8 STATE DIAGRAM**



**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

**4.2.9 ACTIVITY DIAGRAM**

****

**EXPLANATION:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

**4.2.6 SEQUENCE DIAGRAM**

****

**EXPLANATION:**

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 Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

**4.2.7 COLLABORATION DIAGRAM**

****

**EXPLANATION:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

**4.2.4 COMPONENT DIAGRAM**



**EXPLANATION**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

**4.2.5 DEPLOYMENT DIAGRAM**



**EXPLANATION:**

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

**Data Flow Diagram**

**Level-0:**

User

Dataset input

Read Road Accident dataset

Pre-Processing

**Level-1**

User Input

System Prediction

Road Accident Detection Results

Splitting

Transfer Learning

**System Architecture**

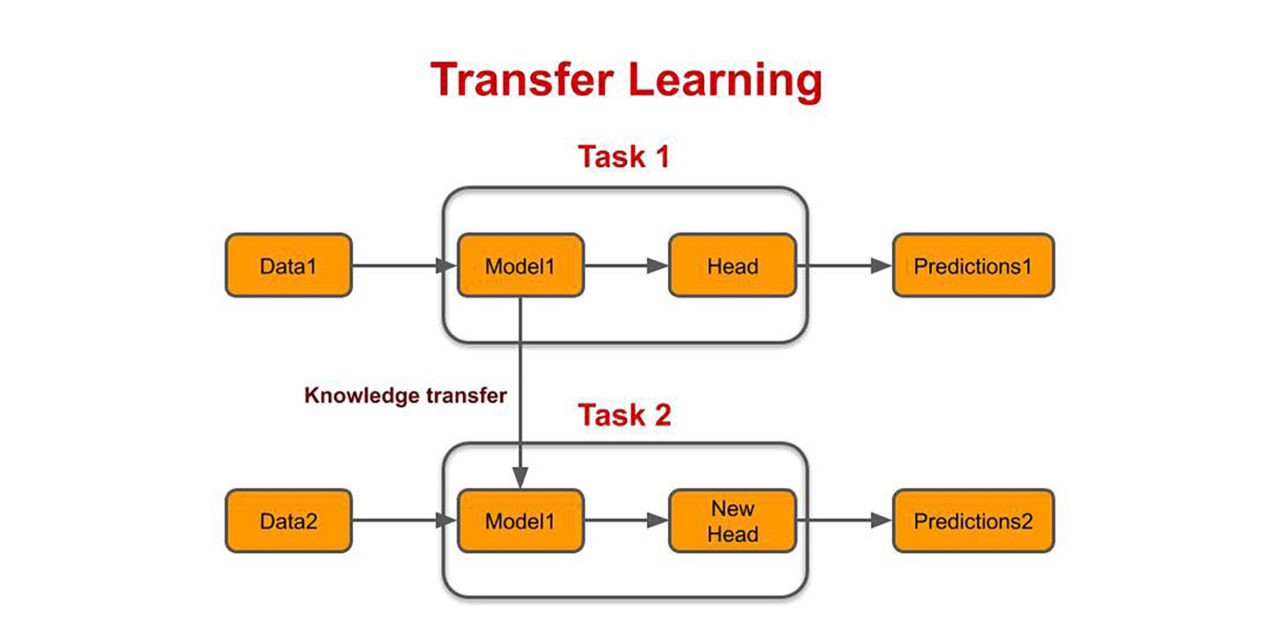
****

Fig 4.12: System Architecture

**CHAPTER 5**

**DEVELOPMENT TOOLS**

**5.1 Python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

## 5.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

#### 5.3 Importance of Python

* **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 is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

#### 5.4 Features of Python

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**5.5 Libraries used in python**

* numpy - mainly useful for its N-dimensional array objects.
* pandas - Python data analysis library, including structures such as dataframes.
* matplotlib - 2D plotting library producing publication quality figures.
* scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.



Figure : NumPy, Pandas, Matplotlib, Scikit-learn

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 GENERAL**

**Coding:**

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.model\_selection import GridSearchCV

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score, roc\_auc\_score, roc\_LANE

# Importing the dataset

dataset = pd.read\_csv('../Dataset/diabetes.csv')

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 8].values

# Splitting the dataset into the Training set and Test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 42)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Parameter evaluation

treeclf = DecisionTreeClassifier(random\_state=42)

parameters = {'max\_depth': [6, 7, 8, 9],

'min\_samples\_split': [2, 3, 4, 5],

'max\_features': [1, 2, 3, 4]

}

gridsearch=GridSearchCV(treeclf, parameters, cv=100, scoring='roc\_auc')

gridsearch.fit(X,y)

print(gridsearch.best\_params\_)

print(gridsearch.best\_score\_)

# Adjusting development threshold

tree = DecisionTreeClassifier(max\_depth = 6, max\_features = 4,

min\_samples\_split = 5,

random\_state=42)

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X, y, random\_state=42)

tree.fit(X\_train, y\_train)

print("Accuracy on training set: {:.3f}".format(tree.score(X\_train, y\_train)))

print("Accuracy on test set: {:.3f}".format(tree.score(X\_test, y\_test)))

# Predicting the Test set results

y\_pred = tree.predict(X\_test)

# Making the Confusion Matrix

from sklearn.metrics import classification\_report, confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

print('TP - True Negative {}'.format(cm[0,0]))

print('FP - False Positive {}'.format(cm[0,1]))

print('FN - False Negative {}'.format(cm[1,0]))

print('TP - True Positive {}'.format(cm[1,1]))

print('Accuracy Rate: {}'.format(np.divide(np.sum([cm[0,0],cm[1,1]]),np.sum(cm))))

print('Misclassification Rate: {}'.format(np.divide(np.sum([cm[0,1],cm[1,0]]),np.sum(cm))))

round(roc\_auc\_score(y\_test,y\_pred),5)

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import accuracy\_score, roc\_auc\_score, roc\_LANE

from sklearn.neighbors import KNeighborsClassifier

# Importing the dataset

dataset = pd.read\_csv('../Dataset/diabetes.csv')

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 8].values

# Splitting the dataset into the Training set and Test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 42)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Parameter evaluation

knnclf = KNeighborsClassifier()

parameters={'n\_neighbors': range(1, 20)}

gridsearch=GridSearchCV(knnclf, parameters, cv=100, scoring='roc\_auc')

gridsearch.fit(X, y)

print(gridsearch.best\_params\_)

print(gridsearch.best\_score\_)

# Fitting K-NN to the Training set

knnClassifier = KNeighborsClassifier(n\_neighbors = 18)

knnClassifier.fit(X\_train, y\_train)

print('Accuracy of K-NN classifier on training set: {:.2f}'.format(knnClassifier.score(X\_train, y\_train)))

print('Accuracy of K-NN classifier on test set: {:.2f}'.format(knnClassifier.score(X\_test, y\_test)))

# Predicting the Test set results

y\_pred = knnClassifier.predict(X\_test)

# Making the Confusion Matrix

from sklearn.metrics import classification\_report, confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

print('TP - True Negative {}'.format(cm[0,0]))

print('FP - False Positive {}'.format(cm[0,1]))

print('FN - False Negative {}'.format(cm[1,0]))

print('TP - True Positive {}'.format(cm[1,1]))

print('Accuracy Rate: {}'.format(np.divide(np.sum([cm[0,0],cm[1,1]]),np.sum(cm))))

print('Misclassification Rate: {}'.format(np.divide(np.sum([cm[0,1],cm[1,0]]),np.sum(cm))))

round(roc\_auc\_score(y\_test,y\_pred),5)

# Importing the libraries

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import accuracy\_score, roc\_auc\_score, roc\_LANE

from sklearn.svm import SVC

import matplotlib.pyplot as plt

from sklearn.metrics import classification\_report

# Importing the dataset

dataset = pd.read\_csv('../Dataset/diabetes.csv')

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 8].values

# Splitting the dataset into the Training set and Test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 42)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

#svm with grid search

svm = SVC(random\_state = 42)

parameters = {'kernel':('linear', 'rbf'), 'C':(1,0.25,0.5,0.75),

'gamma': (1,2,3,'auto'),'decision\_function\_shape':('ovo','ovr'),

'shrinking':(True,False)}

scores = ['precision', 'recall']

for score in scores:

print("# Tuning hyper-parameters for %s" % score)

print()

svm = GridSearchCV(SVC(), parameters, cv=5,

scoring='%s\_macro' % score)

svm.fit(X\_train, y\_train)

print("Best parameters set found on development set:")

print()

print(svm.best\_params\_)

print()

print("Grid scores on development set:")

print()

means = svm.cv\_results\_['mean\_test\_score']

stds = svm.cv\_results\_['std\_test\_score']

for mean, std, params in zip(means, stds, svm.cv\_results\_['params']):

print("%0.3f (+/-%0.03f) for %r"

% (mean, std \* 2, params))

print()

print("Detailed classification report:")

print()

print("The model is trained on the full development set.")

print("The scores are computed on the full evaluation set.")

print()

y\_true, y\_pred = y\_test, svm.predict(X\_test)

print(classification\_report(y\_true, y\_pred))

print()

svm\_model = SVC(kernel='rbf', C=100, gamma = 0.0001, random\_state=42)

svm\_model.fit(X\_train, y\_train)

spred = svm\_model.predict(X\_test)

print ('Accuracy with SVM {0}'.format(accuracy\_score(spred, y\_test) \* 100))

# Making the Confusion Matrix

from sklearn.metrics import classification\_report, confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

print('TP - True Negative {}'.format(cm[0,0]))

print('FP - False Positive {}'.format(cm[0,1]))

print('FN - False Negative {}'.format(cm[1,0]))

print('TP - True Positive {}'.format(cm[1,1]))

print('Accuracy Rate: {}'.format(np.divide(np.sum([cm[0,0],cm[1,1]]),np.sum(cm))))

print('Misclassification Rate: {}'.format(np.divide(np.sum([cm[0,1],cm[1,0]]),np.sum(cm))))

svm.fit(X\_train, y\_train)

round(roc\_auc\_score(y\_test,y\_pred),5)

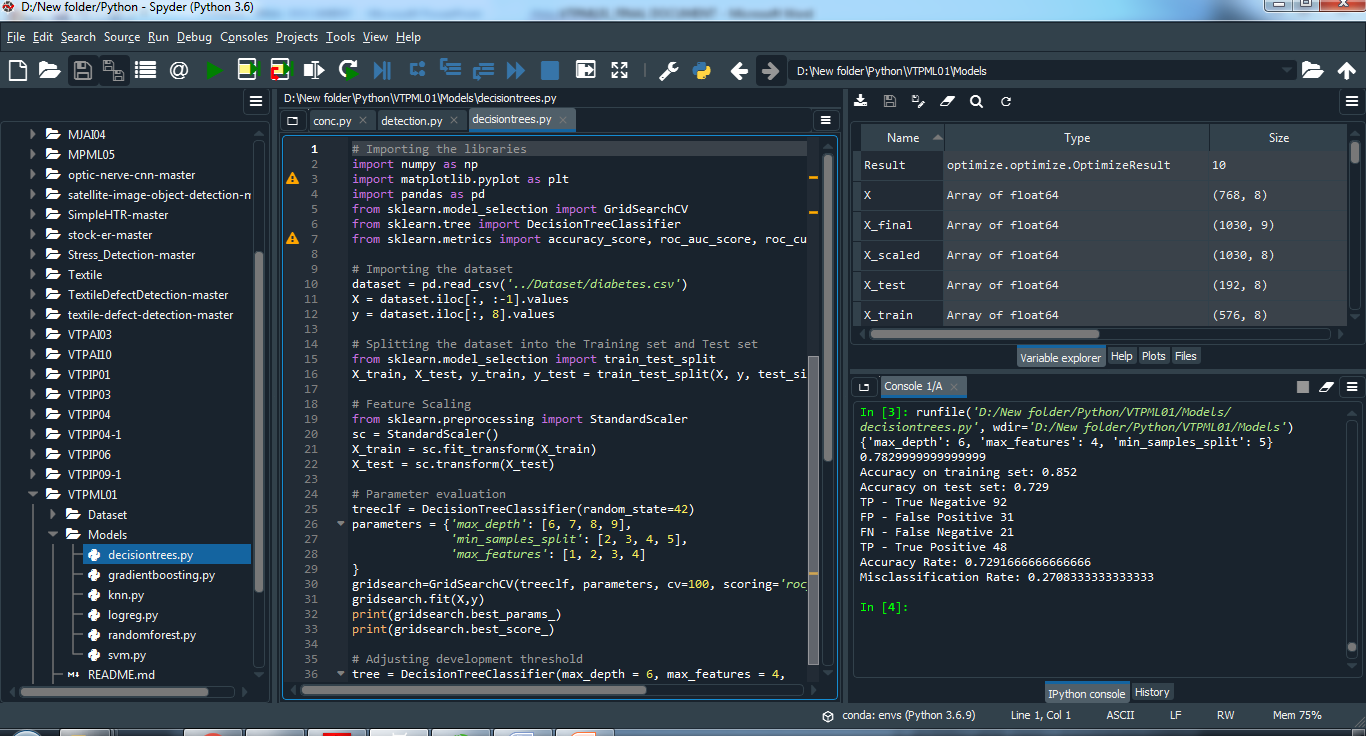
**CHAPTER 7**

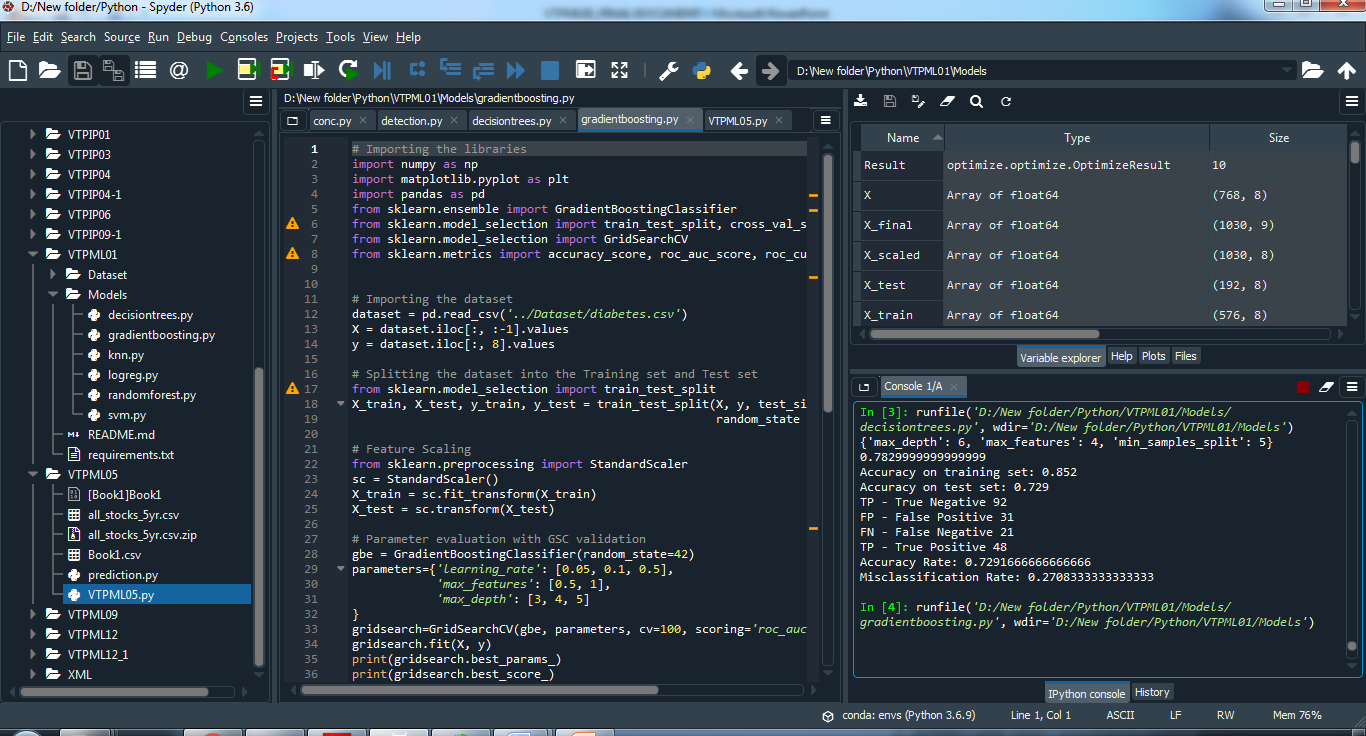
**SNAPSHOTS**

**General:**

This project is implements like application using python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

**SNAPSHOTS**

****

****

**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

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.

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3Types of Tests**

**8.3.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input 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.

**8.3.2 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.

**8.3.3 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.

**8.3.4 Performance Test**

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 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.

**8.3.6 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.

**Acceptance testing for Data Synchronization:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node.
* The Route add operation is done only when there is a Route request in need.
* The Status of Nodes information is done automatically in the Cache Updating process.

**8.2.7 Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**CHAPTER 9**

**FUTURE ENHANCEMENT**

In the near future, we focus on developing a “hybrid” wild animal classification framework whose automated module working as a recommendation system for the existing citizen science-based Wildlife Spotter project.

**CHAPTER 10**

**CONCLUSION & REFERENCE**

**10.1 CONCLUSION**

Efficient and reliable monitoring of wild animals in their natural habitats is essential to inform conservation and management decisions. In this paper, using the Wildlife Spotter dataset, which contains a large number of images taken by trap cameras in South-central Victoria, Australia, we proposed and demonstrated the feasibility of a deep learning approach towards constructing scalable automated wildlife monitoring system. Our models achieved more than 96% in recognizing images with animals and close to 90% in identifying three most common animals (bird, rat and bandicoot). Furthermore, with different experimental settings for balanced and imbalanced, the system has shown to be robust, stable and suitable for dealing with images captured from the wild. We are working on alternative ways to improve the system’s performance by enhancing the dataset, applying deeper CNN models and exploiting specific properties of camera trap images. Towards a fully automated wild animal recognition system, we would investigate transfer learning to deal with problem of highly imbalanced data. In the near future, we focus on developing a “hybrid” wild animal classification framework whose automated module working as a recommendation system for the existing citizen science-based Wildlife Spotter project.

**REFERENCES**

1] WHO road traffic deaths [http://www.who.int/gho/road safety/mortality/en/](http://www.who.int/gho/road%20safety/mortality/en/)

[2] J.L. DSouza,Road Safety: A Critical Issue Requiring Government Attention. International Journal of Education and Science Research review, Volume-2, Issue-2, April, 2015.

[3] J. Pang, I. Singh, Accelerometer Based Real-Time Remote Detection and Monitoring of Hand Motion. Proceedings of the World Congress on Engineering and Computer Science 2011 Vol II WCECS 2011, October 19-21, 2011, San Francisco, USA.

[4] Author: Scott J. Weiner, Feasibility of a 802.11 VANET Based Car Accident Alert System.

[5] halil, U., Javid, T., & Nasir, A. (2017). Automatic road accident detection techniques: A brief survey. 2017 International Symposium on Wireless Systems and Networks (ISWSN).

[6] Faiz, A. B., Imteaj, A., & Chowdhury, M. (2015). Smart vehicle accident detection and alarming system using a smartphone. 2015 International Conference on Computer and Information Engineering (ICCIE).

[7] Ozbayoglu, M., Kucukayan, G., & Dogdu, E. (2016). A real-time autonomous highway accident detection model based on big data processing and computational intelligence. 2016 IEEE International Conference on Big Data (Big Data).

[8] Shailesh Bhavthankar, Prof. H.G.Sayyed, Wireless System for Vehicle Accident Detection and Reporting using Accelerometer and GPS. International Journal of Scientific & Engineering Research, Volume 6, Issue 8, August-2015.

[9] Miao Chong, Ajith Abraham and Marcin Paprzycki, Traffic Accident Analysis Using Machine Learning Paradigms. Informatica 29 (2005) 8998.

[10] Keiron OShea, Ryan Nash, An Introduction to Convolutional Neural Networks. arXiv:1511.08458v2 [cs.NE] 2 Dec 2015.

[11] Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 13451359.

[12] Dogru, N., & Subasi, A. (2018). Traffic accident detection using random forest classifier. 2018 15th Learning and Technology Conference (L&T).

[13] Jayadeva, Khemchandani, R., & Chandra, S. (2007). Twin Support Vector Machines for Pattern Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(5), 905910.

[14] Ding, S., Yu, J., Qi, B., & Huang, H. (2012). An overview on twin support vector machines. Artificial Intelligence Review, 42(2), 245252.

[15] Tanveer2015application, title=Application of smoothing techniques for linear programming twin support vector machines, author=Tanveer, M, journal=Knowledge and Information Systems, volume=45, number=1, pages=191–214, year=2015, publisher=Springer.

[16] Richhariya, B., & Tanveer, M. (2018). A robust fuzzy least squares twin support vector machine for class imbalance learning. Applied Soft Computing, 71, 418432.

[17] Li, D., & Tian, Y. (2014). Twin Support Vector Machine in Linear Programs. Procedia Computer Science, 29, 17701778.

[18] P.Kaliuga Lakshmi, C.Thangamani, AN EFFICIENT VEHICLE ACCIDENT DETECTION USING SENSOR TECHNOLOGY. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5 Issue 3, March 2016.

[19] Peddi Anudeep, K. Hari Babu, Wireless Reporting System for Accident Detection at Higher Speeds in Peddi Anudeep Int. Journal of Engineering Research and Applications.

[20] Marek Dbrowski, Tomasz Michalik, How effective is Transfer Learning method for image classification. Position papers of the Federated Conference on Computer Science and Information Systems, pp. 39.

[21] Xiaoling Xia, Cui Xu and Bing Nan, Inception-v3 for Flower Classification. 2017 2nd International Conference on Image, Vision and Computing.

[22] Prajapati, S. A., Nagaraj, R., & Mitra, S. (2017). Classification of dental diseases using CNN and transfer learning. 2017 5th International Symposium on Computational and Business Intelligence (ISCBI).

[23] Vaishnavi Ravindran, Lavanya Viswanathan and Dr. Shanta Rangaswamy, A Novel Approach to Automatic Road-Accident Detection using Machine Vision Techniques. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 11, 2016.

[24] Nejdet Dogru, Abdulhamit Subasi, Traffic Accident Detection By Using Machine Learning Methods. Conference: International Symposium on Sustainable Development, At Bosnia and Herzegovina, Volume: 2

[25] Namrata H. Sane, Damini S. Patil, Snehal D. Thakare and Aditi V. Rokade, Real Time Vehicle Accident Detection and Tracking Using GPS and GSM. International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 4 Issue: 4. [26] N.Upendra Yadav, Smart Vehicle Monitoring System using IOT. IJDCST@March-April-2017,Issue-V-5,I-3,SW-31.

[27] P. Shruthi,V.T. Venkatesh, B. Viswakanth, C. Ramesh, P.L. Sujatha, I. R. Dominic, Analysis of Fatal Road Traffic Accidents in a Metropolitan City of South India J Indian Acad Forensic Med. October-December 2013, Vol. 35, No. 4 ISSN 0971-0973.