**MINI PROJECT-1 *1***

**(2020-21)**

**DROWNING RISK ANALYSIS USING MACHINE LEARNING OPEN CV**

**PROJECT SYNOPSIS**



**G.L.A. UNIVERSITY, MATHURA (UP)**

**Institute of Engineering & Technology**

**TEAM MEMBERS**

MUSKAN GARG

(181500403)

REEMA AGRAWAL

(181500559)

RITIKA DAS

(181500583)

HIMANSHI AGRAWAL

(181500267)

**Supervised By**

**Mr. Mandeep Singh**

**(Technical Trainer)**

**Department of Computer Engineering & Applications**

***2***

**CONTENT PAGE:**

**TOPICS PAGE NO.**

* **About the Project 3**
* **Importance 3**
* **Objectives 4**
* **Technologies used**
* **Machine Learning 4-5**
* **Open CV 5**
* **Algorithm Used**
* **RESNET 6-7**
* **YOLOv3 8-9**
* **LSTM 10-11**
* **Hardware Specifications 11**
* **Software Specifications 12**

**ABOUT THE PROJECT**  ***3***

Using YOLO object detection, this program will detect whether a person is drowning or not. This software can be used with a Raspberry Pi Camera, which can then be placed underwater with an appropriate case.

The algorithm consists of three core steps: pre-processing, objects-detection and drowning detection.

**IMPORTANCE**

When we started looking at the statistics about drowning deaths, we noticed that drowning is the second leading cause of death especially for children under 15. Even if they are under the supervision of guardian but a slightest ignorance can result into disaster.1.2 million people around the world die or get injured by drowning every year, i.e more than 2 people per minute.

Detection and fast response time are most important factor to save lives from drowning and avoid permanent damage.

Thus our project would be very useful in the school swimming pools for the kids who are not so used to of swimming. It would be very useful in saving the lives of kids. It also helps in places where even adult drowning risk is high.

Unsupervised places for swimming are death traps since there is no one to educate and manage the crowd or to assist and alert in case of an emergency. So, it will help in preventing the risk factor. As a whole nothing can be more important than saving one’s life.

**OBJECTIVES** ***4***

Drowning most often occurs without noise or splashing. It also can be very hard to differentiate between drowning and water play.

No matter how watchful and dedicated lifesavers are, they are also only human. It’s impossible for them to monitor every swimmer in a pool, at every minute. But it’s important to reach a drowning victim before it’s too late as every second counts.

Our main objective is to combine a underwater camera with a computer vision detection system which wraps up as a drowning detection system. This will provide pool lifeguards with “extra eyes” under water.

By placing a modified lifeguard routines it will incorporate regular visits to perform the Lifeguarding activities.

**TECHNOLOGIES USED**

**Machine learning with OpenCV (Open source Computer Vision library)**

**TECHNOLOGIES DESCRIPTION**

1. **Machine Learning:**

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer’s part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather then having human programmers specify every needed step. The discipline of machine learning

***5***

employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, One approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers.

1. **Open CV:**

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to understand and automate tasks that the human visual system can do.

Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g. in the forms of decisions. Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that make sense to thought processes and can elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

The scientific discipline of computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, multi-dimensional data from a 3D scanner or medical scanning device. The technological discipline of computer vision seeks to apply its theories and models to the construction of computer vision systems.

Sub-domains of computer vision include scene reconstruction, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, visual serving, 3D scene modeling, and image restoration.

***6***

**Algorithms used**

* **Single Pose Estimation (RESNET)**
* **Real – time Object Detection. (YOLOv3)**
* **Human Activity Recognition (LSTM)**
* **RESNET**

ResNet is a short name for a residual network, but what’s residual learning?

Deep convolutional neural networks have achieved the human level image classification result. Deep networks extract low, middle and high-level features and classifiers in an end-to-end multi-layer fashion, and the number of stacked layers can enrich the “levels” of features. The stacked layer is of crucial importance.

Human Pose Estimation is defined as the problem of localization of human joints (also known as key-points like elbows, wrists, etc) in images or videos. It is also defined as the search for a specific pose.

Human pose estimation refers to the process of inferring poses in an image. Essentially, it entails predicting the positions of a person’s joints in an image or video. This problem is also sometimes referred to as the localization of human joints. It’s also important to note that pose estimation has various sub-tasks such as single pose estimation, estimating poses in an image with many people, estimating poses in crowded places, and estimating poses in videos. Pose estimation can be performed in either 3D or 2D.

Some of the approaches used in the papers we’ll highlight are bottom-up and top-down. Essentially, in a bottom-up approach, the processing is done from high to low resolutions, while in top-down processing is done from low to high resolutions.

***7***

The top-down approach starts by identifying and localizing individual person instances using a bounding box object detector. This is then followed by estimating the pose of a single person. The bottom-up approach starts by localizing identity free semantic entities, then grouping them into person instances.





***8***

* **YOLOv3**

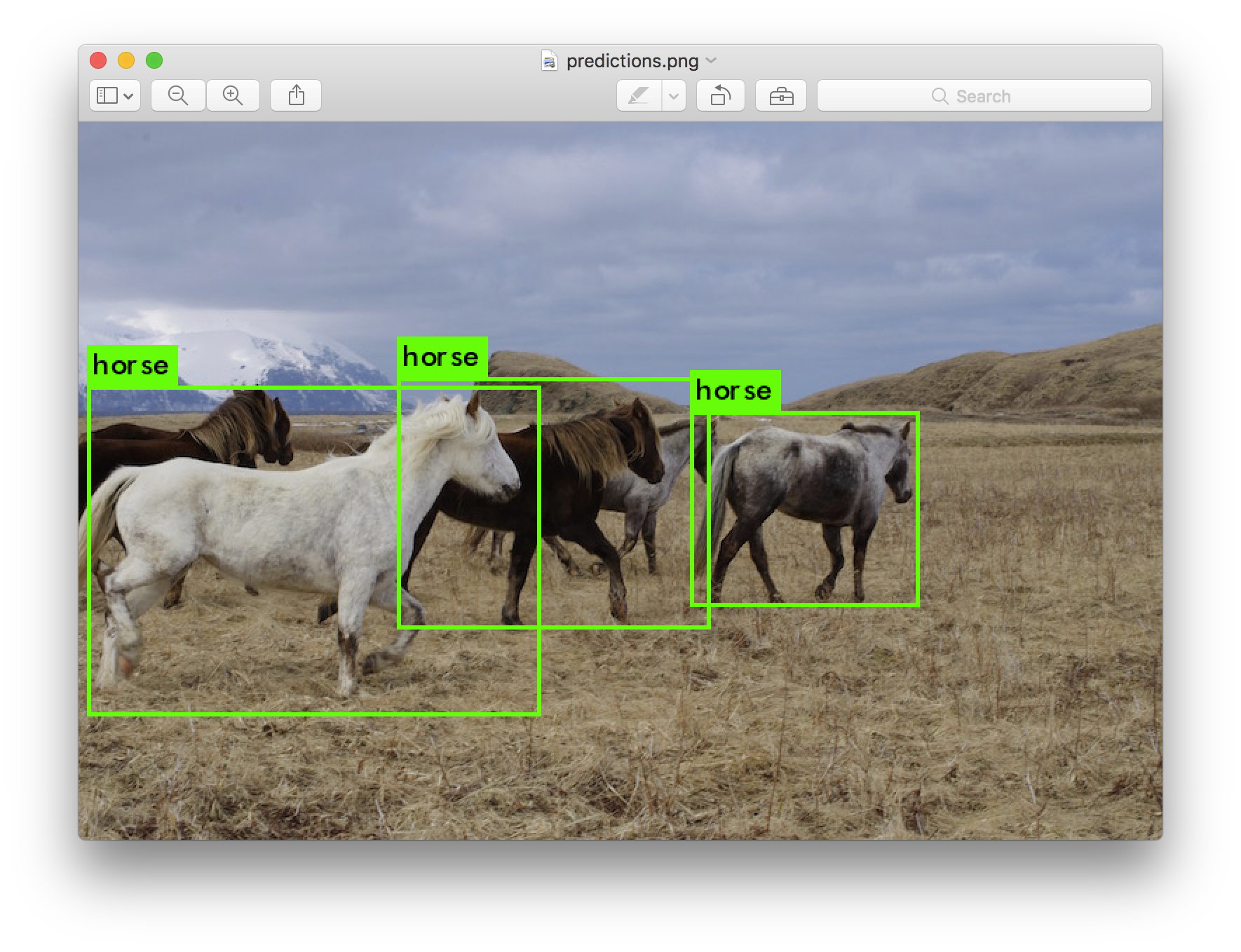
You only look once, or YOLO, is one of the faster object detection algorithms out there, it is a very good choice when you need real-time detection, without loss of too much accuracy.

YOLO is a fully convolutional network and its eventual output is generated by applying a 1 x 1 kernel on a feature map. In YOLO v3, the detection is done by applying 1 x 1 detection kernels on feature maps of three different sizes at three different places in the network.

Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.

It uses a totally different approach by appling a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

***9***

 To understand the YOLO algorithm, it is necessary to establish what is actually being predicted. Ultimately, we aim to predict a class of an object and the bounding box specifying object location. Each bounding box can be described using four descriptors:

1. Center of a bounding box (bxby)

2. Width (bw)

3. Height (bh)

4. Value cis corresponding to a class of an object (such as: car, traffic lights, etc.).

In addition, we will predict the pc value, which is the probability that there is an object in the bounding box.

***10***

* **LSTM**

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN.LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

The Long Short Term Memory architecture was motivated by an analysis of error flow in existing RNNs which found that long time lags were inaccessible to existing architectures, because back-propagated error either blows up or decays exponentially.

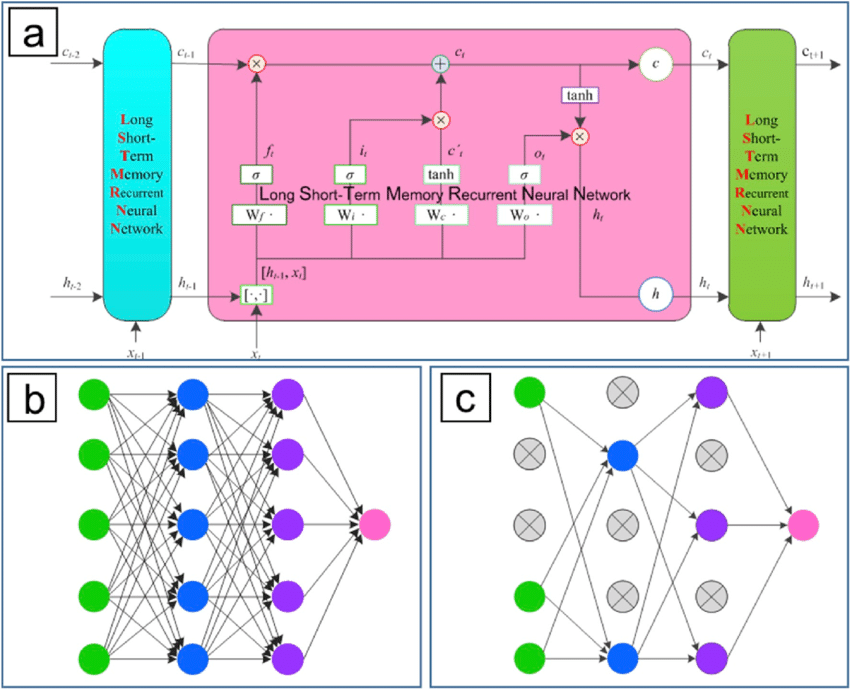
Human Activity Recognition is the process of identifying, analyzing and interpreting what kind of actions and goals one or more agents or persons will be performing. The decisions will be taken based on their previous actions performed with their behavior.

Here a CNN LSTM architecture was used which the CNN layers are used for feature extraction on input data and LSTM to support sequence prediction. The basic steps of constructing the CNN LSTM neural network is as follows.

1. Load Data

2. Fit and Evaluate Model

***11***



**HARDWARE SPECIFICATIONS**

• Processor : Intel i5

• Operating System : Windows 10

• RAM : 8GB

• Hardware Devices : Computer System

• Hard disk : 256 GB

***12***

**SOFTWARE SPECIFICATIONS**

We would be using the concept of MEAN STACK for defining its front end, back end and data base.

• Front End: Python TKinter for GUI

• Back End: Open CV using Python

• Database: MySQL

• Coding to be done in python