**ACTIVITY RECOGNITION USING KERAS**

**A Mini-Project Report submitted in partial fulfilment of the requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**Akash Dabral, 1215316903**

**Kati Silas Swarnakanth, 1215316926**

**Muddurthi Deepak Siddardha, 1215316935**

**Nimmagadda Sai Ujwal, 1215316937**

**Under the esteemed guidance of**

**Ms P. Mansa Devi**

**Assistant Professor, GITAM.**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM**

**(Deemed to be University)**

**VISAKHAPATNAM**

**OCTOBER 2019**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM**

**(Deemed to be University)**

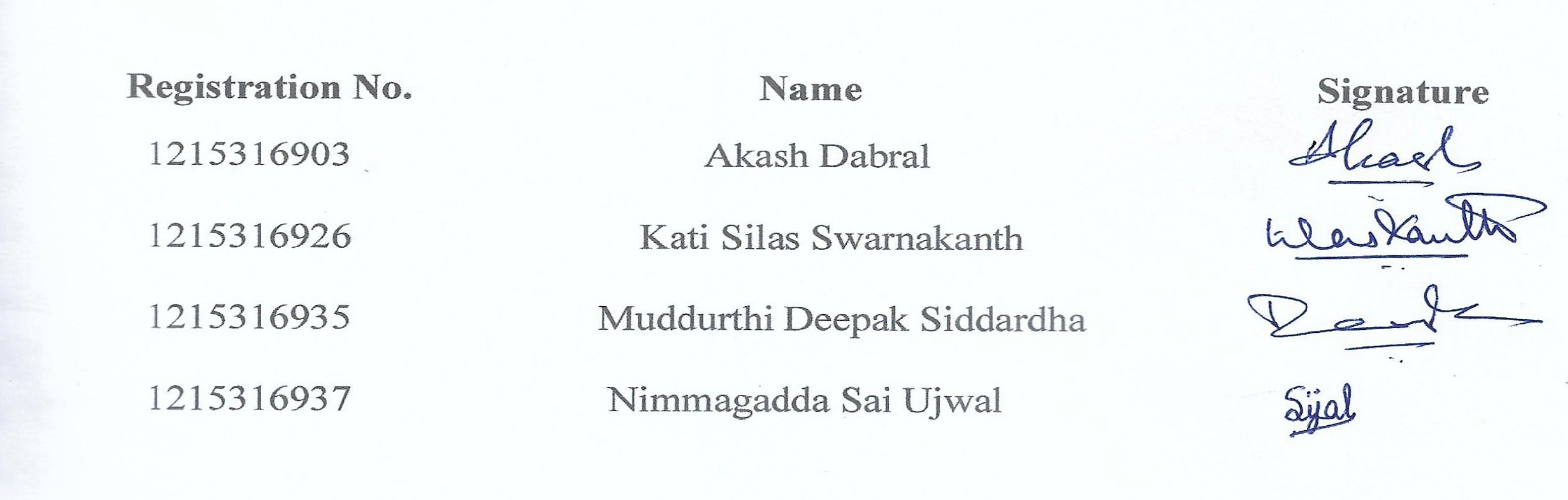
****

**DECLARATION**

We, hereby declare that the mini project work entitled “**ACTIVITY RECOGNITION USING KERAS**” is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfilment of the requirements for the award of the degree of Bachelors of Technology in Computer Science and Engineering.

The work has not been submitted to any other college or university for the award of any degree or diploma.

Date: 14th October 2019



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

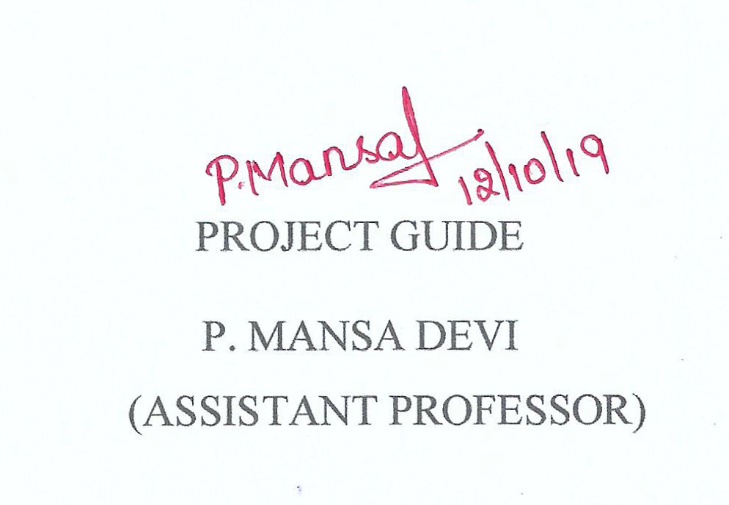
**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM**

**(Deemed to be University)**

****

This is to certify that the mini-project report entitled **“ACTIVITY RECOGNITION USING KERAS”** is a bonafide record of work carried out by **AKASH DABRAL (1215316903), KATI SILAS SWARNAKANTH (1215316926), MUDDURTHI DEEPAK SIDDARDHA (1215316935), NIMMAGADDA SAI UJWAL (1215316937)** students submitted in partial fulfilment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.



**ACKNOWLEDGEMENT**

We express our deepest gratitude and special thanks to Ms P. Mansa Devi, Assistant Professor, C.S.E Department who in spite of being extraordinarily busy with her duties, took time out to hear, guide and keep us on the correct path during the whole project.

We express our sincerest thanks to Mr P. N. R. L. Chandra Sekhar, Associate Professor, C.S.E Dept. and to Mrs K. Neelima Santoshi, Assistant Professor, C.S.E. Dept. for taking part in useful discussions along with giving us necessary advice and guidance. We choose this moment to acknowledge their contribution to the completion of our project.

We would also like to thank our A.M.C., Mrs B. Soujanya, Assistant Professor, C.S.E Dept. and our HoD, Mr. Konala Thammi Reddy, C.S.E. Dept., who gave us this opportunity to do a mini project and helped us in the successful completion of our report.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| 1. | Abstract | 1 |
| 2. | Introduction | 2 |
| 3. | Literature Survey  3.1 Sensors in HAR  3.2 Classification Models | 4  4  5 |
| 4. | Problem Statement   * 1. Model (ResNet)   2. Input   3. Output | 7  7  8  8 |
| 5. | Problem Details  5.1 Dataset Collection  5.1.1 JavaScript  5.1.2 Python  5.2 Project Structure  5.3 Training Phase  5.4 Testing Phase  5.5 Limitations | 9  9  9  9  11  11  13  14 |
| 6. | Conclusion | 15 |
|  | References | 16 |

1. **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. In this project, we have implemented a simple procedure to detect the activity performed in the input video stream.

The project consists of two parts, learning and prediction. The trained model can be used to detect sports activities in the video streams.

We have employed a residual learning based convolutional neural network (ResNet-50) taken from Keras library to train our model with different sports images which we scraped from the web using Google Images 2 Data Set (gi2ds). After training the model with the dataset gathered, our model predicts the type of sport being played in the video. The video stream given as input to the model is processed using the OpenCV library.

Our model can be further used help a machine understand activities similar to how a human being perceives them. Our model is able to detect the activity it has been trained on with high accuracy.

1. **INTRODUCTION**

Human activity recognition is an ability to interpret human body gesture or motion via sensors and determine human activity or action. Most of the human daily tasks can be simplified or automated if they can be recognized via HAR system. Typically, HAR system can be either supervised or unsupervised. A supervised HAR system requires some prior training with dedicated datasets while unsupervised HAR system needs to be configured with a set of rules during development.

Human Activity Recognition (HAR) is one of the active research areas in computer vision as well as human computer interaction. However, it remains a very complex task, due to unresolvable challenges such as sensor motion, sensor placement, cluttered background, and inherent variability in the way activities are conducted by different human.

HAR is considered as an important component in various scientific research contexts i.e. surveillance, healthcare and human computer interaction (HCI).

1. Surveillance System

In surveillance context, HAR is adopted in surveillance systems installed at public places i.e. banks or airports. Human activity prediction is used to prevent crimes and dangerous activities from occurring at public places. Such approaches are able to recognize ongoing human-human interactions at the earlier stage. For example, Legion: AR, a system that provides robust, deployable activity recognition by supplementing existing recognition systems with on-demand, real-time activity identification using inputs from the crowds at public places.

1. Healthcare

HAR is employed in healthcare systems installed in residential environment, hospitals and rehabilitation centres. HAR is used widely for monitoring the activities of elderly people staying in rehabilitation centres for chronic disease management and disease prevention. HAR is also integrated into smart homes for tracking the elderly people's daily activities. Besides, HAR is used to encourage physical exercises in rehabilitation centres for children. Other than that, the HAR is adopted in monitoring patients at home such as estimation of energy expenditure to aid in obesity prevention and treatment. HAR is also applied in monitoring other behaviours such as stereotypical motion conditions in children with Autism Spectrum Disorders (ASD) at home, abnormal conditions for cardiac patients and detection for early signs of illness and it provided the clinicians with opportunities for intervention.

1. Human Computer Interaction

In the field of human computer interaction, HAR has been applied quite commonly in gaming and exergaming such as Kinect, Nintendo Wii, full-body motion-based games for older adults, etc. Through HAR, human body gestures are recognized to instruct the machine to complete dedicated tasks.

Elderly people and adults with neurological injury can perform a simple gesture to interact with games and exergames easily. HAR also enables surgeons to have intangible control of the intraoperative image monitor by using standardized free-hand movements.

1. **LITERATURE SURVEY**

**3.1 SENSORS IN HAR**

Generally, the sensor(s) in a conventional HAR plays an important role in recognizing human activity. Figure 1 illustrates the process of how a human activity is recognized when a body gesture is given as input. The sensor(s) capture the information acquired from human body gesture and the recognition engine analyses the information and determines the type of activity has been performed.

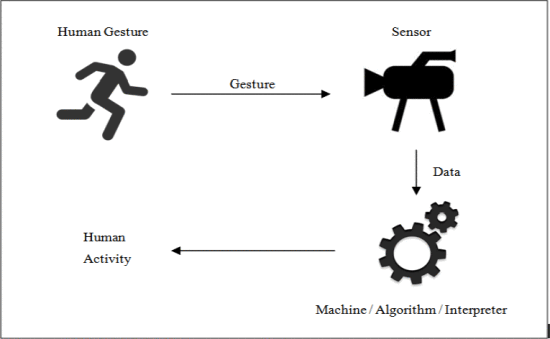


Fig. 1: General structure of HAR system

We mainly talk about three types of sensors used for HAR which are mentioned below:

1. **RGB Camera:** Recognizing human activity using RGB camera is simple but with low efficiency. An RGB camera is usually attached to the environment and the HAR system will process image sequences captured with the camera. Most of the conventional HAR systems using this sensing technology are built with two major components which is the feature extraction and classification. Besides, most of the RGB-HAR systems are considered as supervised system where trainings are usually needed prior to actual use. Image sequences and names of human activities are fed into the system during training stage. Real time captured image sequence is passed to the system for analysis and classification by dedicated computational/classification algorithms.
2. **Depth Sensor:** The depth sensor also known as infrared sensor or infrared camera is adopted into HAR systems for recognizing human activities. In a nutshell, the depth sensor projects infrared beams into the scene and recapture them using its infrared sensor to calculate and measure the depth or distance for each beam from the sensor. The reviews found that Microsoft Kinect sensor is commonly adopted as depth sensor in HAR. Since the Kinect sensor has the capability to detect 20 human body joints with its real-world coordinate, many researchers utilized the coordinates for human activity classification.
3. **Wearable Sensor:** HAR using wearable-based requires single or multiple sensors to be attached to the human body. Most commonly used sensor includes 3D-axial accelerometer, magnetometer, gyroscope and RFID tag. With the advancement of current smart phone technologies, many works use mobile phone as sensing devices because most smart phones are equipped with accelerometer, magnetometer and gyroscope. A physical human activity can be identified easily through analysing the data generated from various wearable sensing after being process and determine by classification algorithm.

**3.2 CLASSIFICATION MODELS**

HAR system can be either supervised or unsupervised. A supervised HAR system requires some prior training with dedicated datasets. HAR system usually is implemented using the classification model. Since the HAR classifier takes input from a sensor, the choice of algorithm hugely depends on the type of data collected. Since the dataset can be huge and there might be many labels to be detected, it would be viable to use deep neural networks (DNNs) for our model.

Since the dataset for RGB-HAR is mainly images or videos, we will be needing convolutional neural networks (CNNs) for our system. There are many CNNs to do the job but we need to select the one which best suits our need.

When it comes to deep learning-based recognition, there are the following networks:

* R-CNN and their variants, including the original R-CNN, Fast R- CNN, and Faster R-CNN
* Single Shot Detector (SSDs)
* YOLO
* ResNet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Properties | R-CNN | SSD | YOLO | ResNet |
| Working | Region based learning. | Runs I/P once and generates feature map. | Runs I/P once and divides it into segments to learn class probabilities. | Based on Residual Learning. |
| Computational Need | High | Medium | Less | Medium |
| Speed | Slow | Fast | Fast | Fast |

Table 1: Differences between the various CNNs.

R-CNNs are one of the first deep learning-based object detectors and are an example of a two-stage detector. The problem with the standard R-CNN method is that it was very slow and not a complete end-to-end object detector.

To help increase the speed of deep learning-based object detectors, both Single Shot Detectors (SSDs) and You Only Look Once (YOLO) use a one-stage detector strategy. These algorithms treat object detection as a regression problem, taking a given input image and simultaneously learning bounding box coordinates and corresponding class label probabilities. In general, single-stage detectors tend to be less accurate than two-stage detectors but are significantly faster.

A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or short-cuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between.

1. **PROBLEM STATEMENT**

With our advancement in giving machines the ability to think, behave and work like humans, it is important for us to let them learn continuously and keep giving them newer abilities to be more like humans. In that conquest, we want the machine to learn the ability to recognise and perceive the activities that are happening around it just as a human does. This will help simplify or automate most of the daily tasks of the human.

**4.1 MODEL (RESNET)**

Very deep neural networks are hard to train as they are more prone to vanishing or exploding gradients. To solve this problem, the activation unit from a layer could be fed directly to a deeper layer of the network, which is termed as a skip connection. This forms the basis of residual networks or ResNets.

A building block of a ResNet is called a residual block or identity block (Fig. 2). A residual block is simply when the activation of a layer is fast-forwarded to a deeper layer in the neural network.

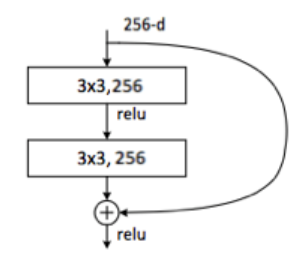


Fig. 2: Residual Block

In Fig. 2, the activation from a previous layer is being added to the activation of a deeper layer in the network. This feature allows training much deeper neural networks.

In theory, the training error should monotonically decrease as more layers are added to a neural network. In practice however, for a traditional neural network, it will reach a point where the training error will start increasing. This is called the problem of degrading accuracy. Deep networks are hard to train because of the notorious vanishing gradient problem — as the gradient is back-propagated to earlier layers, repeated multiplication may make the gradient extremely small. As a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly.

ResNets do not suffer from this problem. The training error will keep decreasing as more layers are added to the network. In fact, ResNets have made it possible to train networks with more than 100 layers, even reaching 1000 layers.

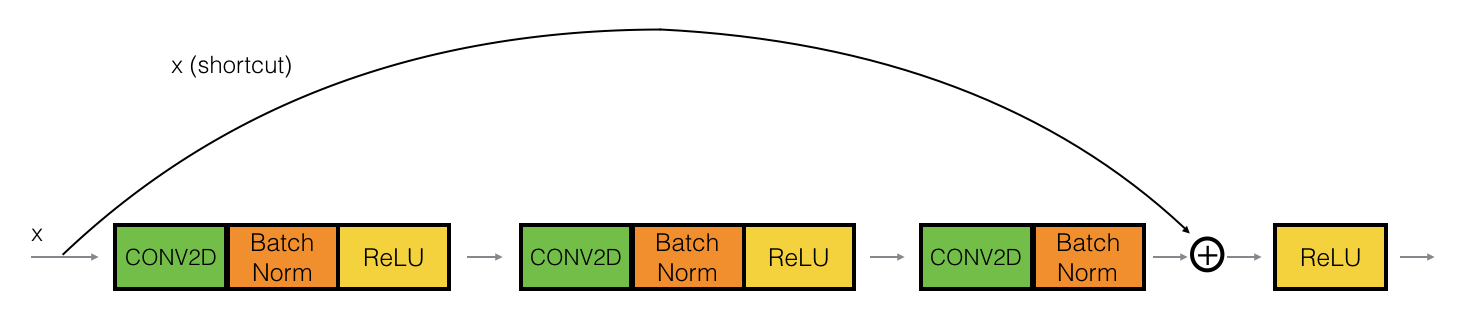


Fig. 3: When x and x\_shortcut are the same shape

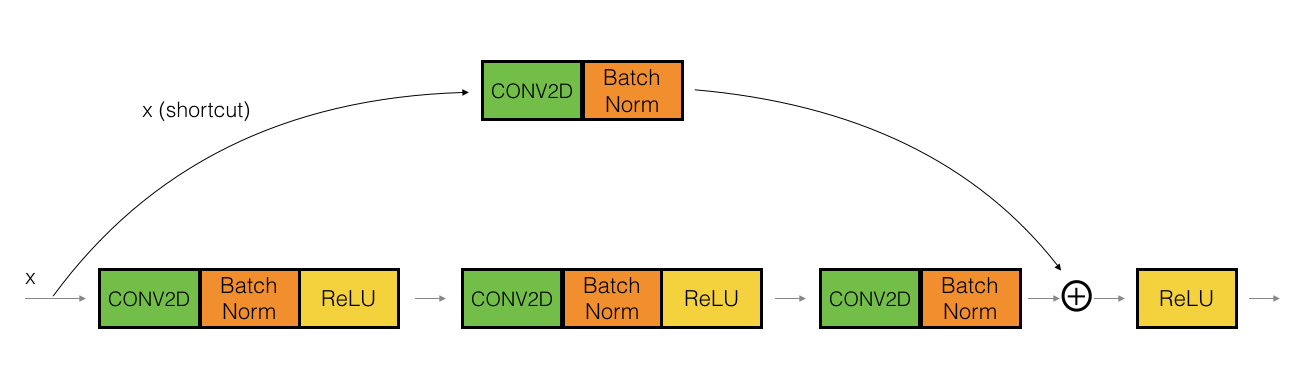


Fig. 4: x\_shortcut goes through convolution block

**4.2 INPUT**

For our project we give a video stream (.mp4) as input to the model. Since the model takes lot of time for prediction on CPU, we have kept the videos short so that we will get faster output.

**4.3 OUTPUT**

The output is in the form of a video stream (.avi) same as that to the input one, but here the output will also consist of the recognised activity on the top left corner.

1. **PROBLEM DETAILS**

**5.1 DATASET COLLECTION**

For our project, we planned to detect various sport activities. We used Google Images to collect our dataset. We used JavaScript and Python to help automate the task to download large number of images.

**5.1.1 JAVASCRIPT**

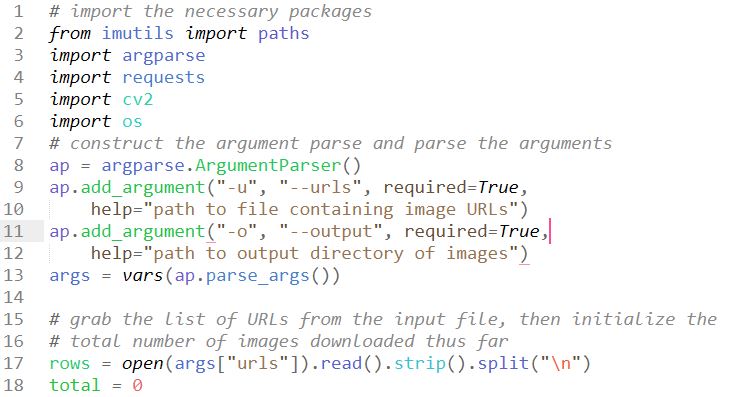
The JavaScript snippet (Fig. 5) using JQuery was used to gather the URLs from Google Images for our dataset.



Fig. 5: JavaScript code to download image URLs form Google Images

**5.1.2 PYTHON**

Python was used to download the images from the URLs we gathered. We used the request library and OpenCV library in python (Fig. 6).



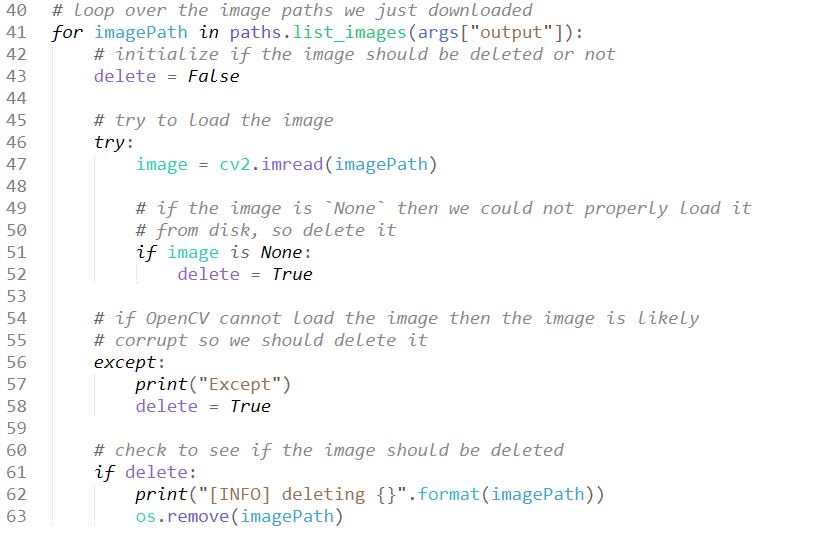
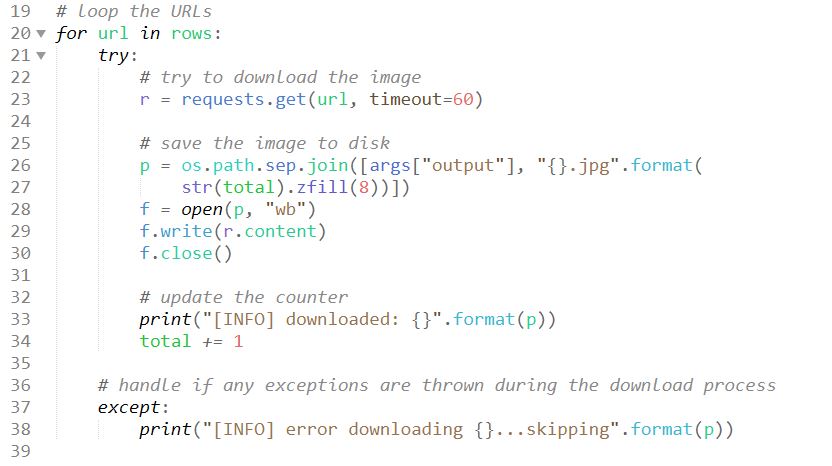


Fig. 6: Python code to download images from the URLs

This procedure was repeated for every sport we collected. For our project, we collected data for 22 different sports.

**5.2 PROJECT STRUCTURE**

Our whole project workspace (Fig. 7) consists of the datasets, python code for training of model and prediction.

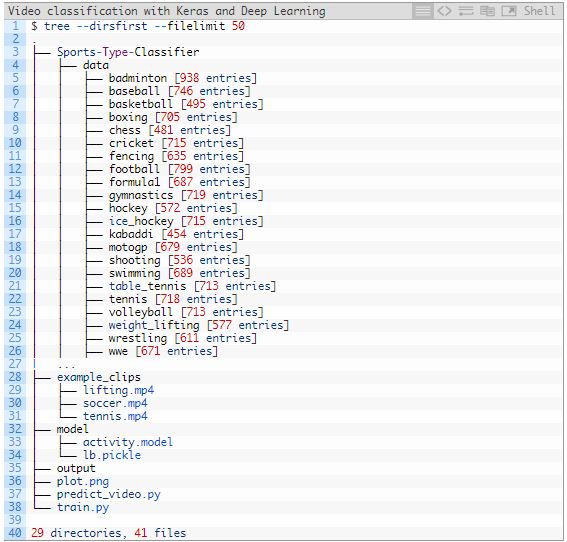


Fig. 7: Complete project directory

**5.3 TRAINING PHASE**

With the dataset gathered, we need to train our model. With ResNet50 as our model, we send our data to create the model we need.

Due to time constraints, we trained our neural network on only three datasets (Weight lifting, Football, Tennis). Our training lasted for over 20 hours.

**Resources:**

Laptop – HP Pavillion 15 (Model: hp 15ac120tx)

Processor: Intel i3 5005U – 2.00GHz

RAM: 16GB

**Libraries:**

TensorFlow: 2.0.0

Keras: 2.2.8

**Training Statistics**

Time: 22 hrs

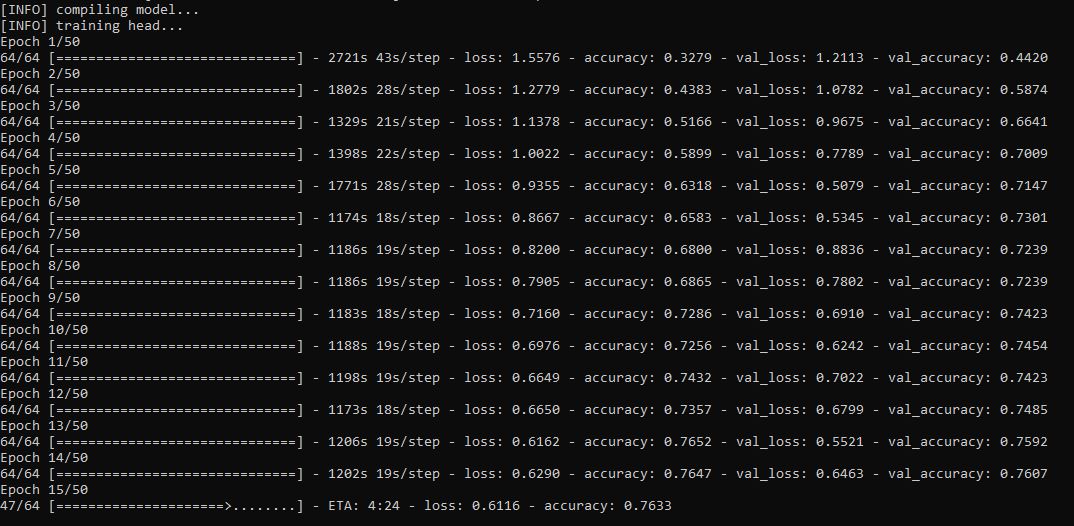


Fig. 8: 15th Cycle of Training

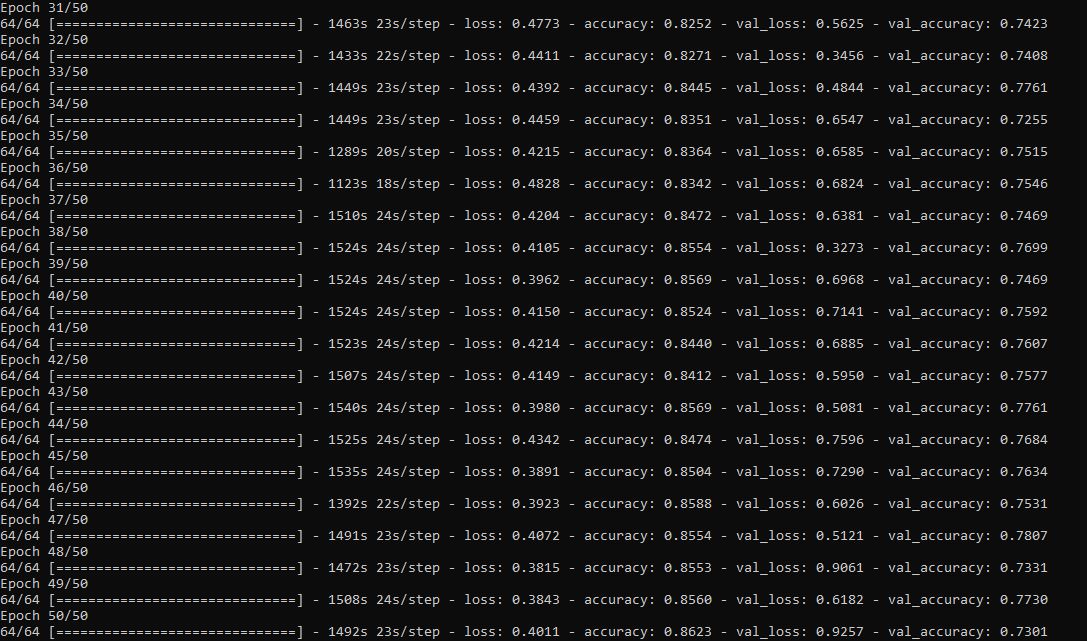


Fig 9: 50 Cycles of Training Completed

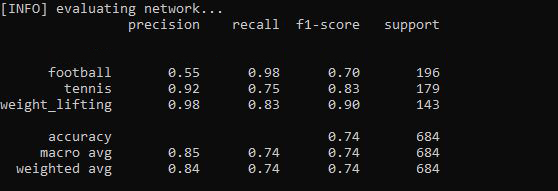


Fig. 10: Training Statistics



Fig. 11: Training Loss and Accuracy on Dataset

**5.4 TESTING PHASE**

After we trained our model, we tested it on few videos to check whether the model classified the activity properly or not. Time taken for testing depends on the length of the video.

Our model is properly recognising the activity in the test video.

**Resources:**

Laptop – HP Pavillion 15 (Model: hp 15ac120tx)

Processor: Intel i3 5005U – 2.00GHz

RAM: 16GB

**Libraries:**

TensorFlow: 2.0.0

Keras: 2.2.8

**Input:** lifting.mp4 (10 sec)

**Output:** lifting.avi (10 sec)

**Time:** 2 mins 40 secs



Fig. 12: Result video after recognition.

**5.5 LIMITATIONS**

The model can only recognise activities it has been trained for.

Addition of any new activity needs the training of the whole model again.

The recognition processing time hugely depends on the length of the input video.

1. **CONCLUSION**

In our project we are successfully able to train the machine to learn activities from the given image dataset using ResNet50 CNN and then let the machine recognise what activity is being performed from the input video.

Our model once trained, can be used on various platforms to be able to recognise activities.

If any other activity needs to be recognised, the whole model has to be trained again to be able to detect the new activity.

This project displays the real time operation on the recognition and then saves the video file.

Training time can be reduced using a high-performance CPU or a GPU or training on clouds like Amazon AWS, Microsoft Azure or Google Colab.

**References**

1. **Journals / Articles**
2. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), **pp. 580-587, (2014).**
3. Ross Girshick, “Fast R-CNN”, 2015 IEEE International Conference on Computer Vision (ICCV), (**7-13 December 2015).**
4. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, IEEE Transactions on Pattern Analysis and Machine Intelligence, **Volume: 39, Issue: 6, (June 1 2017).**
5. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (**27-30 June 2016).**
6. Joseph Redmon, Ali Farhadi, “YOLOv3: An Incremental Improvement”, Computer Vision and Pattern Recognition (cs.CV), arXiv:1804.02767, **(8 Apr 2018).**
7. Ong Chin Ann, Lau Bee Theng, “Human activity recognition: A review”, 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014), **(28-30 Nov. 2014).**
8. Stefan Oniga, József Sütő, “Human activity recognition using neural networks”, Proceedings of the 2014 15th International Carpathian Control Conference (ICCC), **(28-30 May 2014).**
9. **E-websites**
10. https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33
11. https://towardsdatascience.com/hitchhikers-guide-to-residual-networks-resnet-in-keras-385ec01ec8ff
12. https://keras.io/