**NAGARJUNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

(An Autonomous College under VTU, Belagavi)

|  |  |
| --- | --- |
|  | E:\Raghu\18CSP83\Project111\VTU logo.png |

A Mini Project Report

on

**“TITLE OF THE PROJECT”**

submitted in partial fulfillment for the course 19CSP77 of the degree

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted by,

Name USN

Name USN

Under the guidance of

Dr./Mr./Mrs. FACULTY NAME

Designation

Dept. of CSE, NCET

**DEPARTMENT OF COMPUTER SCIENCE & ENGNIEERING**

**NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY**

(An Autonomous College under VTU, Accredited by NAAC with “A+” Grade**)** Mudugurki (V), Venkatagirikote (P), Devanahalli (T), Bengaluru-562164.

**2022-23**

|  |  |  |  |
| --- | --- | --- | --- |
| **NAGARJUNA COLLEGE OF ENGINEERING & TECHNOLOGY**  **(An Autonomous College under VTU, Accredited by NAAC with “A+” Grade**)  **Bengaluru-562164, Karnataka, India**  **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**    **CERTIFICATE**  This is to certify that the Project Phase-1 work entitled **“TITLE OF THE PROJECT”** carried out by NAME bearing USN, NAME bearing USN, NAME bearing USN, NAME bearing USN arebonafide students of Nagarajuna College of Engineering and Technology, an autonomous institution under Visvesvaraya Technological University, Belagavi in partial fulfillment for the course Mini Project (20CSP57) of Bachelor of Engineering in **Computer Science and Engineering** during the academic year 2022-2023. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report.   |  |  |  | | --- | --- | --- | | **Name & Signature of the Guide**  Mr. Faculty Name  Designation,  Dept. of CSE | **Name & Signature of the HOD**  Dr. Anil Kannur  Professor,  Dept. of CSE | **Name & Signature of the Principal**  Dr. B V Ravishankar  Principal,  NCET | |

|  |
| --- |
| **ACKNOWLEDGEMENT**  It is my proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparations of this project. It would not have been possible to prepare this project, in this form without their valuable help, cooperation and guidance.  I would like to thank **NGI Management** for constant support and facilities provided in the carrying out of project in college premises.  I wish to record my sincere gratitude **Dr. B V Ravishankar, Principal,** Nagarjuna College of Engineering and Technology for his constant support and encouragement in preparation of this project and for providing library and laboratory facilities needed to prepare this project.  I would like to thank **Dr. Anil Kannur, HOD,** Department of Computer Science and Engineering, Nagarjuna College of Engineering and Technology for his valuable suggestions and guidance throughout the period of this project.  I thank my Mini project coordinator **Prof. Swathi S, Sr** Asst. Professor, Department of CSE, Nagarjuna College of Engineering and Technology for your valuable suggestions and all the encouragement that lead towards completion of our project.  I would like to thank my project guide **Dr.** **/** **Prof. Faculty Name,** Designation, Department of CSE, Nagarjuna College of Engineering and Technology for your valuable guidance and all the encouragement that lead towards completion of our project.  Last but not least, we would like to thank our parents, friends, teaching and non-teaching staff of NCET.  **STUDENT NAME1 (USN)**  **STUDENT NAME2 (USN)**  **ABSTRACT**  Understanding the activities of human from videos is demanding task in Computer  Vision. Identifying the actions being accomplished by the human in the video  sequence automatically and tagging their actions is the prime functionality of  intelligent video systems. This project is about identifying or predicting the activity  someone is performing based on the sensor data recorded. Human activity  recognition is applicable in scenarios where we require knowledge of an individual's  activity in real time. The goal of activity recognition is to identify the actions and  objectives of one or more objects from a series of examination on the action of object  and their environmental condition. The major applications of Human Activity  Recognition vary from Content-based Video Analytics, Robotics, Human-Computer  Interaction, Human fall detection, Ambient Intelligence, Visual Surveillance, Video  Indexing etc. The Experimental Evaluation of various papers are observed efficiently  with the various performance metrics like Precision, Recall, and Accuracy.  **TABLE OF CONTENTS**    INTRODUCTION  This Project uses video and images dataset for the recognition of the human activity in the assigned dataset. Vision-based human action recognition has received increasing attentions in computer vision and pattern recognition, and has made significant progress in recent years. In general, it can be considered as labelling video with action classes. By applying image processing and machine learning techniques, relevant features are extracted from videos and are subsequently used to model and recognize human actions. In this project, we intend to design a cost-effective and faster Human Activity Recognition System that can process both video and picture to identify the activity being conducted in it, thus assisting end-users in various applications such as surveillance, helping purpose, etc. This system will not only be cost-effective but also as a utility-based system that can be integrated into a wide variety of applications that can save time and support in different activities that need recognition process and save a lot of time with good accuracy.  LITERATURE SURVEY      OBJECTIVES  The primary objective of this project is to solve human centered problems from healthcare to security by inferring several simple human activities. Through this project we will be able to identify the actions of several objects and their environmental behavior.  The main objectives of the design of the Human Activity Recognition System are:   * To understand different models and techniques of Human Activity Recognition Systems based on research papers. * To recognize various Human Activities from video or image data. * To provide a cost-effective and faster Human Activity Recognition system. * To provide with a HAR program that can be incorporated into a variety of different real-time applications like surveillance, aiding blind people etc. * To automate the process of Activity recognition from video stream or an image.   SYSTEM REQUIREMENTS AND SPECIFICATIONS FUNCTIONAL REQUIREMENTS:   * System must be able to read video sequence as input * System should be able to extract each frame from the video input for processing * System must be able to pre-process the frames extracted from the input and resize or crop it to the required threshold size. * System should be able to compare the frames with the trained weights. * After comparing, the system should be able to categorize the input sequence into various classes with acceptable accuracy.   NON-FUNCTIONAL REQUIREMENT’S  Security no outside entity shall be allowed to modify content of code without proper authorization  SOFTWARE AND HARDWARE REQUIREMENT’S  Python based Computer Vision and Deep Learning libraries will be exploited for the development and experimentation of the project. Tools such as Anaconda Python, and libraries such as OpenCV, TensorFlow, and Keres will be utilized for this process  SYSTEM ANALYSES  EXISTING SYSTEM:   * + System using sensors like tri-axial accelerometer and gyroscope from smartphone to estimate the human activity   + System that recognizes human activity based on feature points   + System sensing with respect to Wi-Fi and walls   + System that performs background reduction and correlation feature enhancement   PROPOSED SYSTEM:   * + No setup required before implementation   + Sensor cost eliminated   + Dependability on sensors is eliminated   + Fewer resources are required as compared other   + Easy to enhance and add activities to current model without any additional hardware requirements   SYSTEM DESIGN:   * The Face Recognition System DFD (Data Flow Diagram) describes the overall “flow” of data on the project. It is used to document the transformation of data (input-output) for project development. The face recognition system DFD consists of DFD levels 0, 1, and 2.The **Face Recognition System DFD (Data Flow Diagram)** describes the overall “flow” of data on the project. It is used to document the transformation of data (input-output) for project development.     METHODOLOGY:  A flowchart is a diagram that represents an algorithm, workflow, or process. The flowchart shows the steps of various kinds and their order by connecting them with arrows . In the Human Activity Recognition System, data is first passed through the pre-processing function which makes it suitable to fed as input to the neural network, the data then goes through the network which then gives us the output.    SYSTEM IMPLEMENTATION  These instructions will show how to prepare your image data, train a model, and deploy the model to classify human action from image samples. See deployment for notes on how to deploy the project on a live stream.  Prerequisites   * [TensorFlow 2.0](https://www.tensorflow.org/) * [scikit-learn](https://scikit-learn.org/stable/) * [OpenCV](https://opencv-python-tutroals.readthedocs.io/en/latest/) * [pandas](https://pandas.pydata.org/) * [pillow](https://pillow.readthedocs.io/en/stable/)   Installing  We recommend using a virtual environment to avoid any conflicts with your system's global configuration. You can install the required dependencies via pip:  Jetson Nano Installation  We use the [trt\_pose repo](https://github.com/NVIDIA-AI-IOT/trt_pose) to extract pose estimations. Please look to this repo to install the required dependencies. You will also need to download these zipped [model assets](https://drive.google.com/open?id=1SkPn4vzZofCtwReodtAsnwYgVkONR5-G) and unzip the package into the models/ directory.  # Assuming your python path points to python 3.x  $ pip install -r requirements.txt  All pre-processing, training, and deployment configuration variables are stored in the conf.py file in the config/ directory. You can create your own conf.py files and store them in this directory for fast experimentation.  The conf.py file included imports a Linear Regression model as our classifier by default.  Example  After pre-processing your image data using the preprocess.py script, you can create a model by calling the action Model () function, which creates a scikit-learn pipeline. Then, call the train Model () function with your data to train:  # Stage your model  pipeline = action Model (config. classifier ())  # Train your model  model = train Model (config.csv\_path, pipeline)  Data processing  Arrange your image data as a directory of subdirectories, each subdirectory named as a label for the images contained in it. Your directory structure should look like this:  ├── images\_dir  │ ├── class\_1  │ │ ├── sample1.png  │ │ ├── sample2.jpg  │ │ ├── ...  │ ├── class\_2  │ │ ├── sample1.png  │ │ ├── sample2.jpg  │ │ ├── ...  . .  . .  Samples should be standard image files recognized by the pillow library.  To generate a dataset from your images, run the preprocess.py script.  $ python preprocess.py  This will stage the labelled image dataset in a csv file written to the data/ directory.  Training  After reading the csv file into a data frame, a custom scikit-learn transformer estimates body key points to produce a low-dimensional feature vector for each sample image. This representation is fed into a scikit-learn classifier set in the config file. This approach works well for lightweight applications that require classifying a pose like the [YogAI](https://www.hackster.io/yogai/yogai-smart-personal-trainer-f53744) use case:  Run the train.py script to train and save a classifier  $ python train.py  The pickled model will be saved in the models/ directory  To train a more complex model to classify a sequence of poses culminating in an action (i.e.. squat or spin), use the train\_sequential.py script. This script will train an LSTM model to classify movements.  $ python train\_sequential.py  Deployment  We've provided a sample inference script, inference.py, that will read input from a webcam, mp4, or rstp stream, run inference on each frame, and print inference results.  If you are running on a Jetson Nano, you can try running the iva.py script, which will perform multi-person tracking and activity recognition like the demo gif above *Getting Started*. Make sure you have followed the Jetson Nano installation instructions above and simply run:  $ python iva.py 0  # Or if you have a video file  $ python iva.py /path/to/file.mp4  If specified, this script will write a labelled video as out.mp4. This demo uses a sample model called lstm\_spin\_squat.h5 to classify spinning vs. squatting. Change the model and motion dictionary under the RUNSECONDARY flag to run your own classifier.  TEACHABLE MACHINE  We've also included a script under the experimental folder, teachable\_machine.py, that supports labelling samples via a PS3 Controller on a Jetson Nano and training in real-time from a webcam stream. This will require  these extra dependencies:   * [Pygame](https://www.pygame.org/docs/ref/joystick.html) * [PS3 Controller](https://docs.donkeycar.com/parts/controllers/#ps3-controller)   To test it, run:  # Using a webcam  $ python experimental/teachable\_machine.py /dev/video0  # Using a video asset  $ python experimental/teachable\_machine.py /path/to/file.mp4  This script will also write labelled data into a csv file stored in data/ directory and produce a video  asset out.mp4.  OUTPUT    TESTING:  In extraction process of human activity recognition first input image is read by the camera on the client side .then it detects the key points from the input which his read by the camera and detect the activity of the human by the system as shown below    FUTURE ENHANCEMENT  This project has a tremendous scope in future. Firstly, video recognition code can be further fined tuned using transfer learning and much bigger datasets can be used to further increase the accuracy of the model. Moreover, web and mobile apps can be built which can call these python scripts via an API call to provide activity recognition on users mobile, and can also aid the elderly and blind people to understand and interact with their surroundings etc. and many more real-time applications of activity recognition system.  CONCLUSION  Human activity analysis is a popular activity in the growing industry and we have applied different machine learning algorithm in them logistic regression gave good result. The limitations of this work is though the efficiency of neural network is good the model is not dynamic. The results can be used for monitoring elderly people, prison inmates or anyone who needs constant supervision |