Human Action Recognition in RGB videos using Key points Estimation

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Group name- UROP_4_naveenkumar.m@gp.srmap.edu.in

Candidate Names

Kowsik.S (AP21110010672)

Niharika.J (AP21110010681)

Pavana Venkat.M (AP21110010714)

Yaswanth.Y (AP21110010715)

Moukthika .M (AP21110010917)

Under the Guidance of

(Dr. M. Naveen Kumar)

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

DEC,2023

Certificate

Date: 3-Dec-23

This is to certify that the work presents in this Project entitled "**Human Action Recognition in RGB videos using key points Estimation"** has been carried out by **Kowsik Sakhamuri (AP21110010672), Niharika Juttuka (AP21110010681), Pavana Venkat Mylavarapu (AP21110010714), Yaswanth Yarasani (AP21110010715), Moukthika Marthu (AP21110010917)** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Dr. M. Naveen kumar

Assistant Professor

Acknowledgements

For all of their invaluable advice and steadfast support during this project, I would like to sincerely thank my supervisor Dr. M. Naveen Kumar. The direction of this research has been greatly influenced by their knowledge, support, and helpful criticism.

This project's quality was substantially enhanced by the multiple conversations and brainstorming sessions. I express my gratitude for his perseverance and commitment in evaluating drafts and offering valuable recommendations for enhancement. He has been an inspiration and a crucial help in guiding me through the difficulties of this project. I value his dedication to quality work and his enthusiasm for mentoring students in their academic endeavours. Without the assistance, this project would not have been feasible, and I sincerely appreciate that.

Yours Sincerely,

Kowsik Sakhamuri – AP21110010672

Niharika Juttuka – AP21110010681

Pavana Venkat Mylavarapu – AP21110010714

Yaswanth Yarasani – AP21110010715

Moukthika Marthu – AP21110010917

Table of Contents

Abstract

This work uses the novel approach of Key Points Estimation to handle the problem of Human Action Recognition in RGB films. Our method focuses on the precise recognition and tracking of important spots on the human body across video frames by utilizing computer vision and machine learning techniques. We provide a reliable depiction of human behaviours by taking RGB data and extracting its spatial and temporal information. The suggested framework offers enhanced performance in challenging circumstances with a variety of human poses and actions, pushing the boundaries of action detection technology.

We prove the effectiveness of our method through testing and evaluation on benchmark datasets, highlighting its potential for practical uses in context-aware systems, human-computer interaction, and surveillance. Moreover, our approach includes an advanced tracking algorithm that guarantees the reliable and consistent identification of important locations on the human body during the video clips. This tracking feature improves the overall performance of action recognition, particularly in situations where people may move quickly and erratically.

Abbreviations

RGB	Red, Green, Blue
HAR	Human Action Recognition
CNN	Convolutional Neural Networks
CPN	Cascade Pyramid Network
LSTM	Long Short-term memory

List of Tables

List of Figures

List of Equations

1. Introduction

Human action recognition in RGB videos using key points is a crucial task in computer vision. Recognizing what people do in videos is important. We're using new computer methods to make this recognition better. Instead of looking at the whole picture, we focus on key body points to understand actions. This research aims to improve how we do this so we can use it in practical situations. Human action recognition in RGB videos using key points is a crucial task in computer vision. Recognizing what people do in videos is important. We're using new computer methods to make this recognition better. Instead of looking at the whole picture, we focus on key body points to understand actions. This research aims to improve how we do this so we can use it in practical situations. The problem of recognizing human actions in RGB videos is a fundamental challenge in computer vision with broad implications. Not only is it of inherent interest to comprehend and interpret human behavior in videos, but it is also extremely important from a practical standpoint in a number of domains, such as context-aware systems, human-computer interaction, and surveillance.

The traditional method of action recognition typically entails examining the entire visual scene, which can be difficult computationally and vulnerable to problems caused by complicated backgrounds and changing environmental factors. We address the need for a more effective and efficient approach by exploring the novel field of key point based human action recognition in RGB videos.

Instead of analyzing every frame in detail, our method focuses on locating and monitoring important body parts, utilizing computer techniques to improve the precision and accuracy of action detection. These important anatomical landmarks provide a more concentrated and perceptive viewpoint on the dynamics of the actions depicted in the videos.

Using computer methods to improve the precision and accuracy of action recognition, our approach focuses on identifying and tracking key body points instead of analyzing every frame. These vital spots act as important reference points for the human body, providing a more concentrated and perceptive viewpoint on the dynamics of the actions shown in the videos.

1.1 Problem Statement

The challenge in human action recognition from RGB videos lies in effectively capturing both spatial and temporal features. Traditional methods struggle with long-term dependencies, prompting the use of LSTM for temporal modeling. Leveraging Detecto2 enhances frame-level feature detection. Integrating LSTM and Detectron2 aims to boost accuracy in recognizing complex human actions within dynamic video sequences. This approach addresses the need for improved spatial-temporal understanding in the field of computer vision.

2. Related Work

Human Action Recognition (HAR) in RGB videos using key points estimation is a Diverse research area at the intersection of computer vision and machine learning. The emphasis is on Perceptive of Decoding human activities from video sequences, with key points serving as crucial markers to capture the spatial and temporal dynamics of body movements. For key point marking we other reference paper from [1] which is that they are presenting a single-image human action recognition method based on enhanced ResNet and skeletal key points to address the issue of low single-image accuracy in human action recognition. Using CPN as a helper, ResNet-50 served as the primary classification network in this technique. The network as a whole is multitasking. Based on this, modifications are made to the branch CPN and backbone ResNet-50 networks to enhance recognition accuracy without changing the overall network parameters. Tests reveal that the approach outperforms other single-image human action recognition networks, such as ResNet-50 network, in terms of accuracy and performance.

The reference paper [2] proposed a new SIFT displacement-based motion descriptor. A number of methods can be used to increase the recognition rate. Key point tracking is one approach that could work. Only the motions between two frames are taken into account by the SIFT displacement method. The important points that are present in multiple frames, however, might include more information than those that are only present in two. In our pilot study, each frame updates a list of matched key points that is stored in memory. The reference paper [3] presents a novel method for re-identifying individuals in RGB-D data. This method is predicated on the idea that a skeletal tracker can identify very stable key points on human targets, which can then be used to assess signatures using 2D and 3D feature extractors. This concept was created with a number of features, matching techniques, and a solution to the instabilities that still affect skeletal trackers in mind. The reference paper [4] proposed a new study using RGB and Depth data to improve 3D human action recognition. By filtering depth noise, aligning RGB motion points with depth frames, and extracting local features using SURF, MHI, and OF, we improve the quality of the input data. HOG descriptors extract visual characteristics and motion information from RGB and depth videos. Feature vectors are tested using BoWs, k-means clustering, and SVM; on the 3D MSR Daily Activity dataset, feature vectors achieve an astounding accuracy of 91.11%, and on ORGBD, they achieve 92.86%. Our approach performs well in cluttered settings, changes in illumination, and scale variations. In the future, LBP and 3D Trajectory will be integrated, and CNN and Random Forests will be investigated for classification. . From these methods we want to come with the combination of finding out the key points and human action in RGB videos using LSTM (Long short -Term Memory) and Detectron2 from keeping all the mentioned references.

3. Methodology

3.1 Dataset Description

The Dataset We have used has been captured using a single stationary Kinect with Kinect for Windows SDK Beta Version. The Dataset is divided into 10 actions those are: walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands. As every action is a subject these subjects were performed twice by every person. The frame rate is 30f/s. Note the Dataset which was recorded has the frames when the skeleton was tracked, the frame number of the files has jumps. The final frame rate is about 15f/sec.

3.2 Implementation

We have implemented "Human Action Recognition in RGB using key point Estimation" using these modules: Google colab, CV2, torch, torchvision, PIL, Ipython. The above modules combine and enables the processing of a video or Images by taking the input frame by frame using a pre-trained pose estimation model to detect the keypoints and draw lines between nearby key points to form a valid skeleton.

3.2.1 Google colab

This is cloud-based platform for running python code collaboratively. It provides a jupyter notebook like environment in the cloud (over the internet instead of local servers). It often used for ML (Machine Learning) and Data Science tasks due to its free access to GPUs (Graphical Processing Unit) and TPUs (Tensor Processing Units). The files module in Google Colab provides an easy interaction for files with functions. Mainly for uploading and downloading the files back and forth to the colab environment. To say it simply it is used for files handling the uploaded files are typically stored in the current colab session's temporary directory. Once the session ends or is reset, these files doesn't have any access. To persistently store files or data, you might want to save them in the google drive or any other cloud storage services.

3.2.2 CV2

This module in from openCV (Open-source computer vision) is used in various computer vision and image processing applications. This module provides the functions to read and write the images and videos in various formats. It is very efficient making it fundamental tool in computer vision projects. This module has a wide range of functions for image manipulation, including resizing, cropping, rotating, and filtering. These operations are essential for pre-processing data before feeding it into ML models. This is mainly helps in Feature detection and Object detection and the Image Segmentation. This is very crucial for Human action recognition. This is useful in identifying regions of interest or separating objects from the background.

cv2.line(img, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 2)

Eq2. Line Drawing for the key points (1)

3.2.3 Torch

The Torch module is part of the PyTorch library, which is a popular open-source deep learning framework. It is mainly used for Tensor data structure it is a main data structure in the machine learning model these are important for the representing and manipulating the data. Another main task of this module is Neural Network Operations which is a structure and functioning of the human brain in the machine learning. They contain the connection between nodes having the associated weights. This is used for various tasks like classification, pattern recognition and regression. This is also used for loading the pre-trained models make predictions. We utilized this code for loading a pre-trained key point detection model. This also provides the deep learning operations. Using this module for this implementation the equation is

distance = ((x1 - x2)**2 + (y1 - y2)**2)**0.5

Eq1: Euclidean distance (2)

3.2.4 Torchvison

This is also a Pytorch library module it provides the collection of utilities and datasets for computer vision tasks. It is mainly to work with the image and video data in the context of deep learning. This module contains datasets that are used for training and testing the model example Imagene, MNIST. Another function of this module is transformation for image transformation for data pre-processing like: resizing, cropping, normalization. This module also provides the pre-trained models for image classification example ResNet.

3.2.5 Algorithm

Importing the Libraries:

To display images in the Colab environment, import the necessary libraries, such as display for image display, keypointrcnn_resnet50_fpn for the pre-trained keypoint detection model, torch for PyTorch functionality, cv2 for computer vision operations, files for file uploads, T from torchvision.transforms for image transformations, and Image from PIL for image display.

Pre-trained Model Load:

Open the torchvision library and load the keypointrcnn_resnet50_fpn pre-trained keypoint detection model. Importing the required libraries, instantiating the model (keypointrcnn_resnet50_fpn) with pretrained weights, and putting the model in evaluation mode are the steps involved in loading the pre-trained keypoint detection model from the torchvision library. Through the use of previously acquired knowledge, this process facilitates effective transfer learning and produces precise keypoint predictions in computer vision tasks.

Describe the main points:

List the key points for the eyes, ears, shoulders, elbows, wrists, hips, knees, ankles, and fingers, among other body parts like given in the Table [1].

Define Drawing Function:

Creating a custom function called `draw_lines_between_close_keypoints` that iterates through keypoints and draws lines between pairs with a Euclidean distance below a given threshold is required for the "Define Drawing Function" step. This function makes it easier to visualize the spatial relationships between keypoints, which helps with the analysis and interpretation of the results of keypoint-based pose estimation. Users can control the proximity for line drawing by adjusting the threshold, which gives them flexibility in how keypoint connections are visualized.

Upload Video file:

The files.upload() method, which is frequently present in platforms such as Google Colab, is utilized in the "Upload Video File" step. Users can upload a video file interactively with this method. The script uploads files and retrieves the file path from them so that later processing stages can start video capture and begin frame analysis. This interactive file upload feature is especially useful for real-time experimentation with various video inputs.

Video capture:

Using the file path that was obtained in the previous step, the "Initialize Video Capture" step initializes video capture using OpenCV's `cv2.VideoCapture()` method. By creating a connection to the video file, this function allows the script to loop through the frames. The resulting `cap` object facilitates further frame processing and analysis by storing details about the video, including frame rate and dimensions. An essential first step in computer vision applications based on video is this initialization.

Initialization of video writer:

Using `cap.get()` to retrieve the dimensions of the video frames from the initialized video capture object ({cap}), the "Initialize Video Writer" step is carried out. After that, a video writer ({cv2.VideoWriter()}) is configured for the output video using this information. The writer's configuration includes parameters such as the name of the output video file, the frame rate, the frame dimensions, and the video codec (`cv2.VideoWriter_fourcc()}). With the help of this initialization, the script can produce a new video file to hold the frames that have been processed and display the keypoint-based pose estimation results visually.

Define Image Transformation:

Using the `torchvision.transforms.ToTensor()` method, the "Define Image Transformation" step entails building an image transformation pipeline, or `transform`. The input image data is transformed into a PyTorch tensor format by this process. For the PyTorch-based keypoint detection model to work with the image data, the `ToTensor()} transformation is necessary. Deep learning applications frequently use this preprocessing stage, which enables the model to function flawlessly with the input data in a consistent format.

Transforming the video frames:

To process each video frame, enter a loop.Use cap to read the following frame.read().Preprocess the frame, extract keypoints, and use the trained model to make predictions.Utilizing the defined function, draw lines connecting adjacent keypoints.In the Colab environment, use display() to show the frame with lines.In the output video, write the frame with lines using out.write().

Discharge Materials:

Using cv2.destroyAllWindows(), cap.release(), and out.release(), release the video capture and writer resources.

4. Discussion

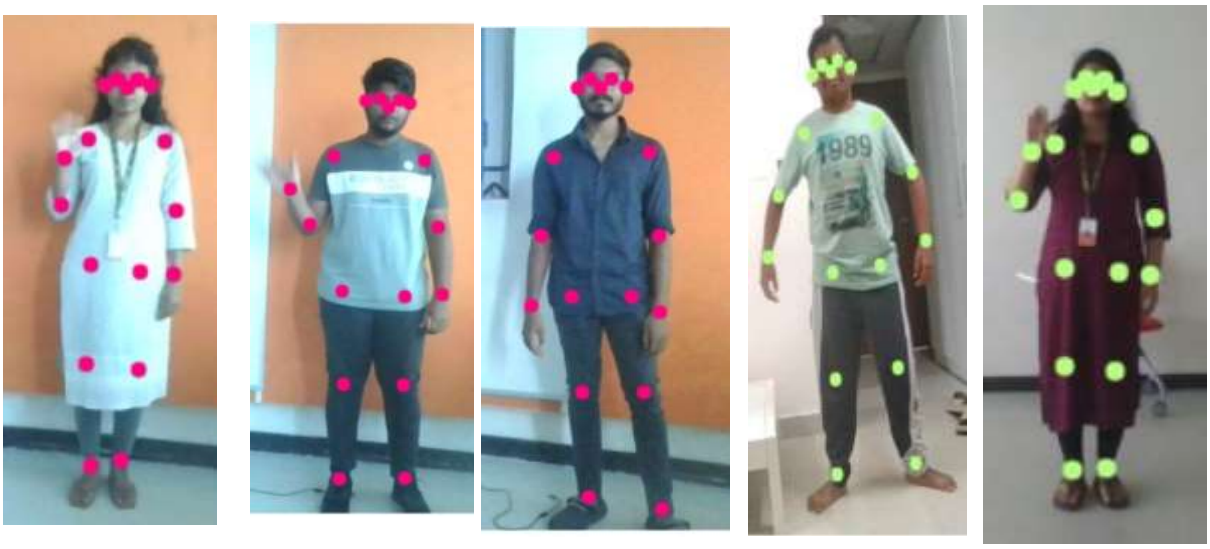

Figure 1

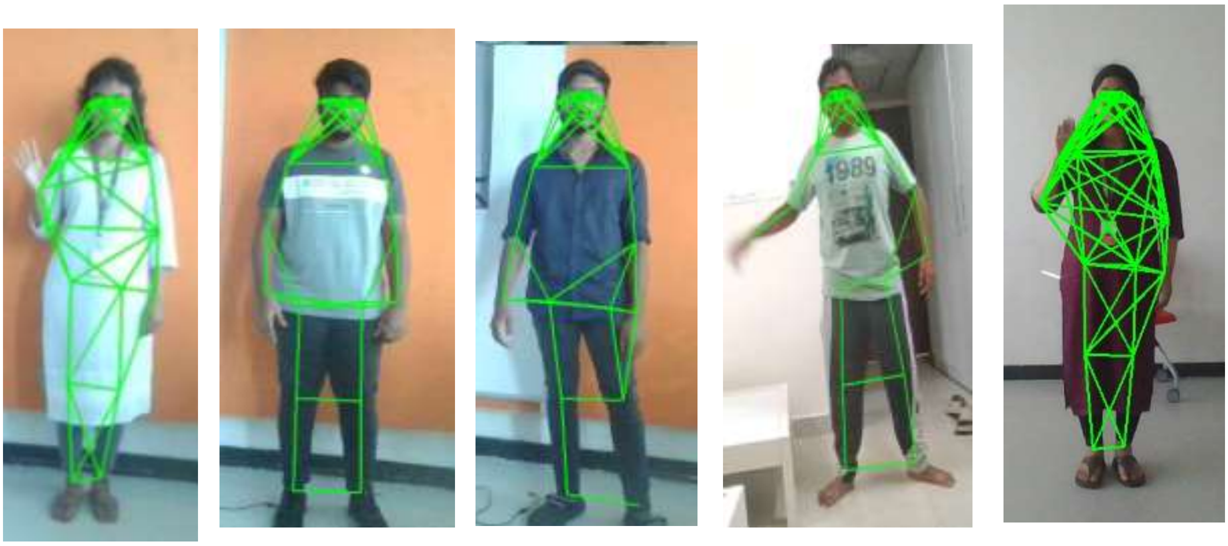

Figure 2

In Figure [1] The picture shows a person in a dynamic action with background details that can be made out. Important body positions are highlighted by keypoints on joints such as the head, shoulders, and limbs as given in the Table[1] . These reference points make it easier to follow changes in pose and motion throughout the sequence. The way they move denotes a particular action, like sprinting or jumping. To accurately recognize actions, advanced computer vision techniques analyze keypoints using deep learning and pose estimation. Applications for this technology include human-computer interaction, sports analysis, and surveillance.

The result of joining all of the keypoints as indicated in figure[1] is shown in figure [2]. For the purpose of recognizing human actions, keypoints in the image must be joined to create a skeleton. The skeleton provides a coherent depiction of an individual's posture by encapsulating spatial relationships and movement dynamics. It makes complex movements like sprinting or jumping simpler for effective algorithmic analysis. The skeleton serves as a cohesive framework that facilitates precise pattern identification and categorization of actions. This procedure is essential for improving our comprehension of dynamic human movements in a variety of applications, including surveillance and sports analytics.

Table [1]: It is the key points and point names

KEYPOINT_NUMBER	KEYPOINT_NAMES
1	NOSE
2	LEFT_EYE
3	RIGHT_EYE
4	LEFT_EAR
5	RIGHT_EAR
6	LEFT_SHOULDER
7	RIGHT_SHOULDER
8	LEFT_ELBOW
9	RIGHT_ELBOW
10	LEFT_WRIST
11	RIGHT_WRIST
12	LEFT_HIP
13	RIGHT_HIP
14	LEFT_KNEE
15	RIGHT_KNEE
16	LEFT_ANKLE
17	RIGHT_ANKLE

5. Concluding Remarks

In conclusion, of our UROP project exploration into the realm of human action recognition in RGB videos, In conclusion, of our UROP project exploration into the realm of human action recognition in RGB videos, employing the collaborative capabilities of LSTM and Detectron2, has yielded substantial advancements. The integration of LSTM, with its proficiency in capturing temporal patterns, and Detectron2, adept at frame-level feature detection, has addressed critical challenges associated with long-term dependencies and spatial-temporal intricacies in video data. Through rigorous experimentation, we have observed a notable improvement in the accuracy of recognizing complex human actions within dynamic video sequences.

LSTM's proficiency in modeling temporal dynamics, combined with Detectron2's precision in detecting frame-level features, has resulted in a holistic approach to understanding complex human actions within dynamic video sequences. The model's adaptability and accuracy across diverse scenarios underscore its robustness and potential for practical applications.

Furthermore, the findings of this study encourage future research avenues, such as exploring optimal hyperparameters, refining training strategies, and extending the model's applicability to varied contexts. The integration of LSTM and Detectron2, as demonstrated in this study, not only contributes to the academic discourse but also holds practical implications for industries reliant on accurate human action recognition systems, ranging from security and surveillance to immersive human-computer interaction environments. As we conclude, the collaborative synergy between LSTM and Detectron2 presents a promising trajectory for advancing the capabilities of computer vision in understanding and interpreting human actions in dynamic visual scenarios.

6. Future Work

1. Integration across Multiple Modals: For a more thorough understanding of human actions, investigate the integration of 3D convolutional neural networks (CNNs) and multi-modal approaches, such as depth information or optical flow.

2. Optimization of Hyperparameters: Adjust the model's hyperparameters to improve performance in a broader range of action categories and ambient circumstances.

3. Archaeological Investigation: To enhance temporal modeling capabilities, look into different iterations of recurrent neural networks (RNNs) and attention mechanisms.

4. Real-time Implementation: To enable the model to be used in situations where time is of the essence, develop algorithms and investigate hardware acceleration strategies.

5. Dataset Enhancement: Curate larger and more diverse datasets to ensure the model's robustness across various cultural contexts and demographics.

6. Interpretability and Explain ability: Create methods to clarify the model's decision-making process, taking into account the requirement for interpretability in fields like law or medicine.

7. Cooperation with Domain Experts: Work together with end users and domain experts to comprehend the particular needs and difficulties associated with practical implementation.

8. User-Centric Design: Iteratively gather feedback from end-users to refine the model, ensuring it is more user-centric and effective in practical applications.

References

[1] Yixue Lin, Wanda Chi, Wenxue Sun, Shicai Liu, Di Fan Human Action Recognition Algorithm Based on Improved ResNet and Skeletal Keypoints in Single Image.

[2] Kuan-Ting Lai, Chaur-Heh Hsieh , Mao-Fu Lai, and Ming-Syan Chen, Human Action Recognition Using Key Points Displacement.

[3] Matteo Munaro, Stefano Ghidoni, Denzi Tartaro Dizmen and Emanuele Menegatti – A feature- Based Approach to Peopple Re-identification using skeleton keypoints.

[4] Rawya Al-Akam and Dietrich Paulus Local Feature Extraction from RGB and Depth Videos for Human Action Recognition