



# Human Activity Recognition in Videos- Minor Project Report

27.04.2024

---

Abhinav Mahajan- IMT2020553

Agastya Thoppur- IMT2020528

## Problem Statement and Introduction

“Human Activity recognition from Videos”

- This is a fundamental problem in Computer Vision, and involves learning deep semantic understanding to classify abstract tasks such as the one at hand. Our starting point was : [“DCapsNet: Deep capsule network for human activity and gait recognition with smartphone sensors”](#) [1]. But as it was paid and its preprint was not available for us to read, we found its closest match, [“Human Activity Recognition Based on a Modified Capsule Network”](#) [2].
- We chose the UCI-HAR dataset since it was the most commonly used dataset in our literature survey. All our experiments conducted henceforth are on this dataset.
- We expanded our horizons with a few survey papers, such as [3, 4] which gave us a gentle introduction towards how HAR (Human Activity Recognition) has progressed from traditional Machine Learning and EDA models towards Deep Learning based methods and which papers have made the most impact. In this project, we have had a chance to experiment with various such techniques, come up with some novelties, and achieve SOTA results!

## Problem Motivation

There has been a definite plateau in Computer Vision when it comes to static Images, as we have achieved a point where for every niche task, we have models for it. Be it detection, segmentation or for generative purposes. However, for Videos, we are lagging behind. Human activity analysis is one such example, where this problem cannot be solved by a single frame(Image) but the underlying temporal relations have to be accounted for as well.

- Human Activity Recognition is imperative in many applications such as AI virtual instructors (rehabilitation instructors, gym instructors etc.). In fact, this was our project once (we implemented the work done by Ziyi Zhao et al [8])!

•Other use cases involve: enhancing Surveillance Systems, Healthcare Monitoring, Behaviour Analysis and Understanding and Assistive Technologies for benefiting individuals with disabilities or elderly populations by providing context-aware assistance

## Literature Survey

•To extend our understanding, we also read the following survey papers on Human Activity Recognition: -

1. ["Video-based Human Action Recognition using Deep Learning: A Review"](#) [9]
2. ["Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances"](#) [10]

•These survey papers act as treasure trove of information, as it summarizes the work done by 100-200 projects along with their inferences and what scope there is left to improve on and what papers have pushed the needle.

•In our exploration, we further stumbled upon the following research papers, which propose interesting avenues which can be further studied and worked on: -

1. ["Virtual Fusion with Contrastive Learning for Single Sensor-based Activity Recognition"](#) CVPR 2024' [12]
2. ["BCL: A Branched CNN-LSTM Architecture for Human Activity Recognition Using Smartphone Sensors"](#) [11]
3. ["Human activity recognition from sensor data using spatial attention-aided CNN with genetic algorithm"](#) [13]

•We will summarize our inferences through the course of the presentation. But a very blatant observation was that state of the art models have achieved 96% Accuracy and F1 scores on the UCI-HAR dataset, so the scope of improvement is not much, however, we have a few experiments in mind to see if we can even slightly improve it.

## Dataset Analysis

- The dataset chosen by us is extremely apt, as this is not a conventional video dataset, but instead it is a dataset made by various subjects performing 6 activities (Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, Laying) and they are wearing a sensor strapped on their arm, which records 9 signals which shall be elaborated on later.
- The advantage is that we tremendously reduce the size of the dataset, and our corresponding models will also be smaller. This is pivotal for us students, as we do not have access to big storage Computing resources as well as high end GPUs and is ideal for running these small models and dataset locally or on google colab/Kaggle cloud servers.
- The dataset is broken down with the following steps.
- Firstly, both the training and test data along with the test labels are included. In each set(training and test), there are raw inertial signals(from the sensors), as well as hand crafted signals from the raw signals.
- There are 7352 training “videos” and 2947 testing ones. For each sample or “video”, we have most importantly, the raw inertial signals from the sensors. Each sample has a signal for 128 timesteps and each time step has 9 signals or features which are the X, Y, Z acceleration, X, Y, Z gyroscope and X, Y, Z Total Acceleration parameters.
- The hand crafted features are also available, but it helps more for EDA purposes. All works in literature solely use Raw inertial time varying signals, without EDA. Sometimes normalization is performed for some signals, but that’s it.
- And as mentioned in the previous slide, for each sample, we have to predict whether the subject is Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing OR Laying.

## Experiments conducted

- 1.Plain CNN [4]

2.Plain LSTM [5]

3.CNN+LSTM [5, 11]

4.CNN+Capsule Networks [1,2,6]

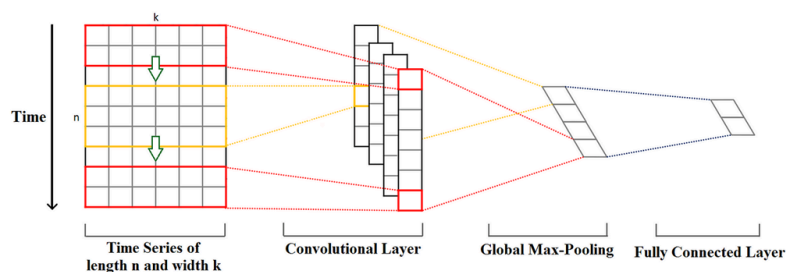
5.CNN+LSTM+Capsule Networks

6.HART (Human Activity Recognition Transformer) (Transformer Baseline for HAR)[7]

7.HART + Capsule Networks

## Plain CNN

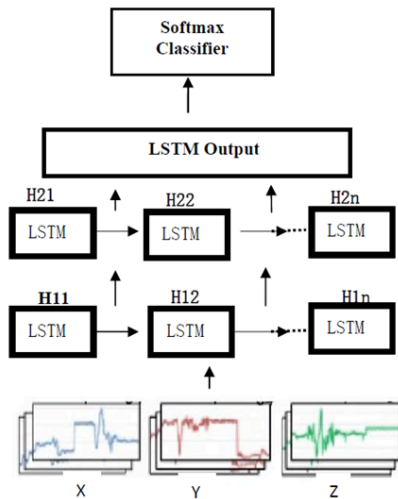
- The features are not spatially correlated! (Unlike images). Imagine an image is reduced to one column or one row, and instead of R,G, B channels, there are 9 features (the sensor parameters).
- Therefore, we use Conv1D and we are using it to exploit temporal redundancies.
- It performs really well, getting an accuracy of 93% (by our experiments) (and even in literature[4])



## Plain LSTM

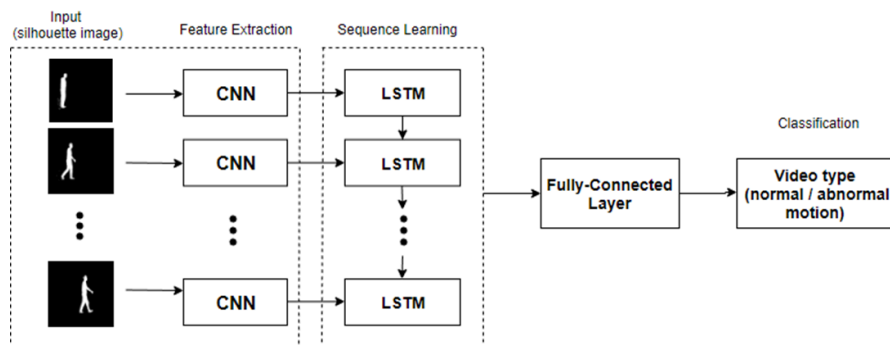
- Simpler to understand why LSTM's are used as we now simply feed in our 9 features at each timestep.

- The LSTM calculates the current state using the previous states, and the current time step input. Due to the design there are gradient highways, which ensures there are long term correlations learnt and not just short term connections.



## CNN-LSTMs

- CNN-LSTMs are predominantly used in HAR for actual videos, where data is in the form of frames rather than signals. The CNN is used to encode the frame/image at each time step and the LSTM takes care of the Temporal redundancy. It's a very neat and tidy design, but in the case of time varying signals, where both CNNs and LSTMs try to take care of temporal redundancy, how will it work?



- Simple solution provided by [5], Since we have 128 timesteps, let's rewrite it as  $32(n\_length) \times 4(n\_steps)$ . Now we perform convolution on the reduced 32 dimensions, and

feed the encoding to the LSTM. And at  $(t+1)$  of the LSTM, we feed in the CNN embeddings of the next 32 timesteps. This is a very neat work around and has been seen in literature to give splendid results.

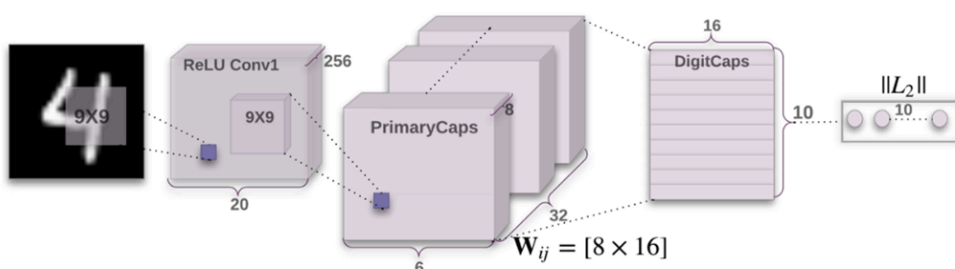
- However, we didn't get much of an improvement and in general from all our experiments, LSTMs never fared so well. You will see from experiment 5 just how poorly it can be sometimes. This problem is due to the nature of data, the input to the LSTM or you could say the field of view of it is too less. CNN+LSTM aims to bridge that by increasing the field of view but still there are methods which perform better on this simple dataset.

## CNN + Capsule Networks

- Capsule Networks were introduced in 2011, and the idea came from none other than Geoffrey E. Hinton, the father of Computer Vision. And only in 2017, was he able to actually implement it! (["Dynamic Routing Between Capsules"](#) [6])

- The architecture of Capsule Networks differ slightly from other Deep Learning frameworks, as there are various transformations, and dynamic routing mechanisms to learn hierarchical and deeper correspondences. It is very underutilized in literature so far.

- CNN-Capsule Networks consist of traditional convolutional layers, primary capsules and digital capsules (activity capsules in the context of this project). The primary capsules process the output of the conv layers and return a vector consisting of the presence and orientation of a particular feature it detects. It uses a process called "dynamic routing" to "vote" for which capsules in the activity layer are most compatible with the detected features. The digital or activity capsules output a vector that encodes the presence of a feature or an activity.



•To implement this for our dataset is however challenging. The various challenges we overcame are: -

1. There is no original implementation in pytorch or tensorflow. And the existing methods are built on top of 2D Convolution Filters, not 1D in the four time varying signal case.

Therefore, we need to write it from scratch, tweaking the transformation equations to suit our dimensions.

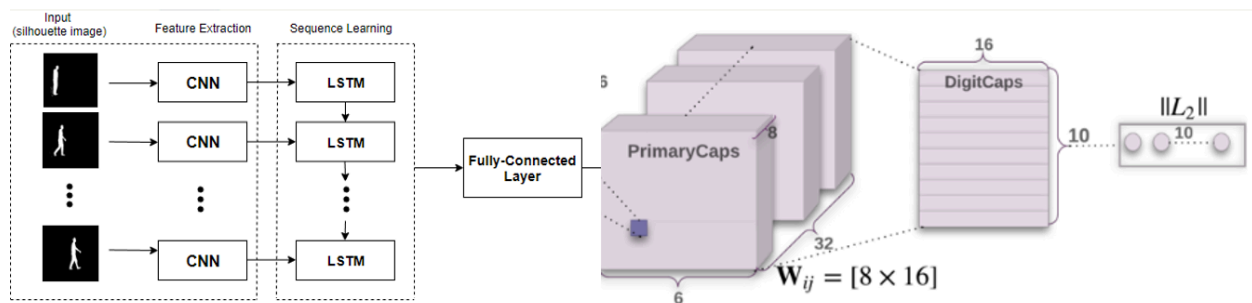
2. They use a hinge loss and try to predict the presence of multiple objects, with clutter and occlusion. However, we want to only predict one of the labels, where the presence of one, impacts the presence of others.

3. [6] used a reconstruction branch as well on the embeddings, to enforce learning more global features. And the loss function was a weighted loss of the Hinge Loss and reconstruction loss(mse). However, reconstruction loss doesn't work well for time varying signals, and it provides negligible if not negative effect, mostly because reconstructing abstract signals simply are not as meaningful as reconstructing an image.

## CNNs + LSTMs + Capsule Networks

•This was a novel architecture, with no paper talking about such a work. We combined our CNN-LSTM split as we did in method 3, and used our 1D Capsule Networks to finally predict the label.

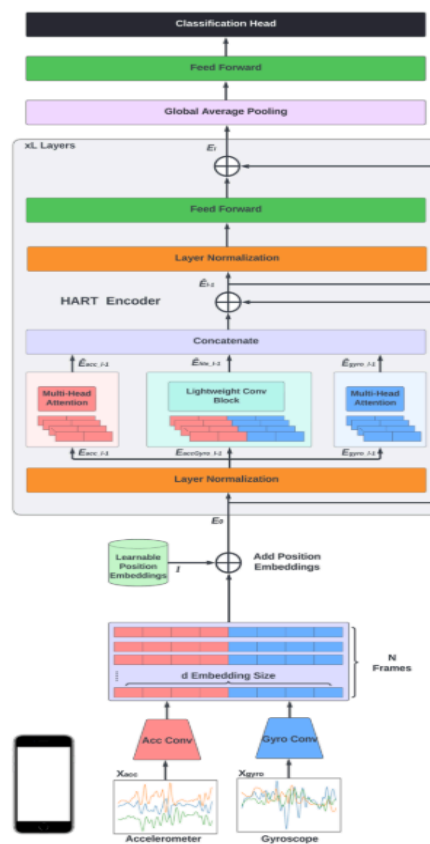
•As mentioned before, LSTMs don't seem to work so well. And reducing the final output to the state of the LSTM + 32 frames instead of 128 kills the performance.





## HAR Transformer

- This wasn't mentioned in the project proposal, and we went a step ahead and tried to experiment on this, as we were excited at the possibility of merging this with good performing models.
- This has been studied in literature[7], and our baseline uses the following architecture but it has the following drawbacks: -
  1. They are using only 2/3<sup>rd</sup> the dataset and have abolished the TotalAcceleration signals, they are only using the acceleration and gyroscope signals (X, Y, Z for each).
  2. Therefore, we revamp the architecture in our upcoming experiment and we get a paper worthy result.



## HART + Capsule Networks

- This is our novel architecture, and we overcome the baselines shortcomings by: -
  1. Introducing the TotalAcceleration branch along with the Acceleration Branch and Gyroscope branch. (TotalAcceleration is not the same as acceleration, they are separately recorded signals).
  2. We slice the initial embedding layer differently, to get the extra multi-head attention branch for TotalAcceleration, and also added the TotalAcceleration embedding to the lightformer at stage one.
  3. At the decoding step, we further add the Capsule network immediately, we arrange the MLP head to primary capsules and then implement the 1D dynamic routing Digit Capsules algorithm to form the digicaps, which are then squashed to return the logits for predicting the class.

## Conclusion

- During the course of the project we learnt how to attempt solving Human Activity Recognition without the use of Computer Vision but using deep learning methods, also how computer vision methods can actually be implemented in cross modal scenarios. (Capsule Networks, CNN+LSTMS etc.)
- We were able to extend and make the HART baseline better with a few nuances and novelties and achieve SOTA results.
- With more experimentation, and more datasets, this can surely be extended to paper, and both of us are ready to go that extra mile.
- UCI-HAR is a relatively simple dataset, hence simple methods like CNN's can get amazing results. However, if we experiment with more complicated datasets, surely we will be able to see the distinction between the models, as the difference between models seems very low right now.


## Tabulated Results

Models	Accuracy Percentage
Plain CNN	93%
Plain LSTM	92%
CNN + LSTM	93%
CNN + Capsule Network	93.5%
CNN + LSTM + Capsule Network	70%
HAR Transformer	90%
HART + Capsule Networks	95.6%

**Github repo-** <https://github.com/Abhinav-Mahajan10/Human-Activity-Recognition>

## Bibliography

- [1] [“DCapsNet: Deep capsule network for human activity and gait recognition with smartphone sensors”](#)
- [2] [“Human Activity Recognition Based on a Modified Capsule Network”](#)
- [3] [“UCI HAR dataset”](#)
- [4] [“Real-time human activity recognition from accelerometer data using convolutional neural networks”](#)
- [5] [“A cnn-lstm approach to human activity recognition”](#)
- [6] [“Dynamic Routing Between Capsules”](#)
- [7] [“Lightweight Transformers for Human Activity Recognition on Mobile Devices”](#)

- 
- [8] ["3D Pose Based Feedback For Physical Exercises"](#)
  - [9] ["Video-based Human Action Recognition using Deep Learning: A Review"](#)
  - [10] ["Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances"](#)
  - [11] ["BCL: A Branched CNN-LSTM Architecture for Human Activity Recognition Using Smartphone Sensors"](#)
  - [12] ["Virtual Fusion with Contrastive Learning for Single Sensor-based Activity Recognition"](#)
  - [13] ["Human activity recognition from sensor data using spatial attention-aided CNN with genetic algorithm"](#)