White Paper

Human Activity Recognition

Fengyao Luo
Praveen Kasireddy
Sam Shih
Sean Campos

August, 2021

Contents

1. Abstract
2. Background
3. Dataset
4. EDA
5. Data Pipeline
6. Fine Tuning Models
7. Challenges
8. Next Steps
9. References

Abstract

This project aims at collecting metrics pertaining to the activities of a person or a group of people based on sensor observation in the public realm, in order to inform better planning decisions. The application of activity recognition can be used in calculating the time span for human activities in public spaces such as parks. By leveraging the data, stakeholders can enhance the environment as well as the utilization of public facilities including benches, playgrounds, food areas, etc, as well as plan for maintenance, capital improvements and events.

In this project, we use an edge device, Jetson with a web camera to shoot a video, frame a person in the video, identify body parts, and recognize human postures. The pipeline is constructed with 2 main parts: Body Pose Detector and Action Recognition. The body pose detector used the Resnet18 as our backbone and trained it with the COCO dataset. It is able to pinpoint the coordinates of 18 key body parts and draw connections to each, forming a skeleton of 2D human pose. The action recognition used an LSTM to track a sequence of frames with pose vectors and trained with NTU RGB+D 60 dataset. The dataset includes 60 categories and we only picked 5 categories that were germain to our problem domain to train our model. The Identified activities include drink and eat, sit and squat, phone and talk, walk, selfie. We achieved a f1 score of 0.83 on the test dataset. Each category's accuracy score has reached above 0.83 in the end.

In the ideal case, an activity is recognized regardless of the environment it is performed in or the performing person and data can be processed at a near real time pace. Instead of connecting a real time camera, we shoot a video which includes the actions to test our models, and output the measurable metrics, including actions, number of people, time span.

Background

Human activity recognition

Human activity recognition(HAR) has been increasing in popularity for several years. Applications of HAR include video surveillance, health care, and human-computer interaction. As the imaging technique advances and the camera device upgrades, novel approaches for activity recognition constantly emerge. Our project aims to provide a full line structure to the video-based human activity recognition.

Human activities have an inherent hierarchical structure that indicates two levels of it. First, there is an atomic element which configures the body parts of people. The skeleton output constitutes more complex human activities which comes as the second level.

When we expand the recognition scenario from a single person to a group of people in a video, recognition and framing technique on humans in a video is applied between the first and second level mentioned in the previous paragraph.

Challenges of HAR

Complex and Various Backgrounds

Most of these real-world videos have complex dynamic backgrounds. First, those videos, as well as the broadcasts, are recorded in various and changing backgrounds. Second, realistic videos abound with occlusions, illumination variance, and viewpoint changes, which make it harder to recognize activities in such complex and various conditions.

Multi-subject Interactions and Group Activities

low-level human activities such as jumping, running, and waving hands are relatively simple recognition. One typical characteristic of these activities is having a single subject without any human-human or human-object interactions. However, in the real world, people tend to perform interactive activities with one or more persons and objects. In one sample video in our training dataset, the video beginning with two people talked to each other. Another person joined their conversation in the middle of the film. Another sample video shows a man helping his son hanging on the bar in the playground.

HAR For Outdoor Activities

In this project, we focus on detecting human activities in outdoor environments. The activity recognition system is based on video processing components (NVIDIA Xavier jetson). The end goal of the project is to use image recognition to track the traffic of public spaces as well as the utilization of public facilities. Furthermore, we can further identify the behavior of citizens and answer questions such as what's the average time spent waiting for a bus, what kind of exercise they do in the park, what's the utilization of benches in the park, etc.

Another use case of HAR is place audit. For public resource groups like parks and recreation, when the responsible bureau has a budget and would like to invest in additional facilities such as benches, trash cans, or even coffee stands. Before execution, they would like to know what are regular activities of citizens in public spaces. Traditionally, they would cooperate with a vendor for a survey which we called place audit. Basically, the vendor will assign a surveyor, who sits in the public space and writes down the activities of citizens. This is quite costly and causes safety issues during COVID 19. However, with the help of HAR and edge devices, we can automate the manual process and identify human activities. Then, the cost of place audits can be cheap and thus scalable.The sample map below shows the type of activities and its duration in a public area.

Activation of place

< 10 mins 10–20 mins 20–30 mins 30+mins

Posture

384 37 3

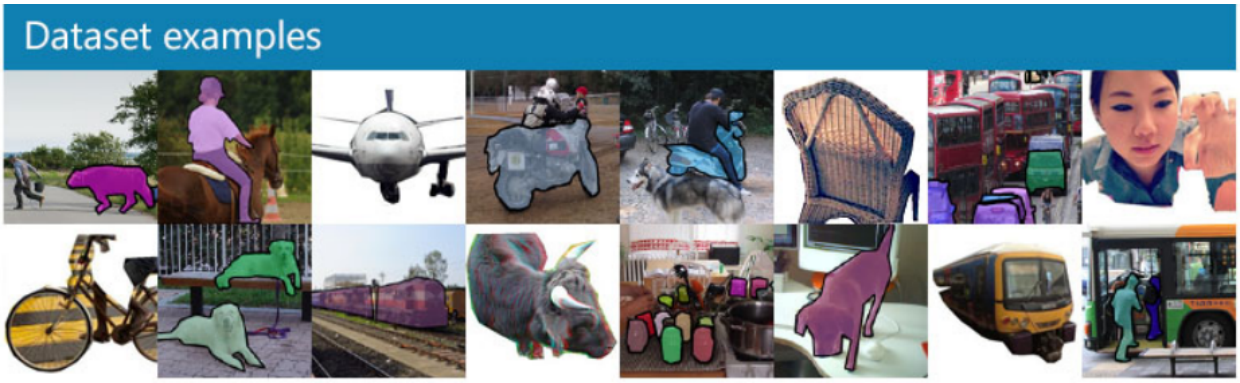
Behaviour

336 47 33 8

Dataset

Coco (Common Objects in Context)

COCO is a large-scale object detection, segmentation, and captioning dataset. It includes 81 categories, 330k images, 1.5 million object instances, 250, 000 people with keypoints, which contains basically all objects captured from everyday scenes. This dataset is good for object segmentation, recognition in context and superpixel stuff segmentation. For our project, we only choose pictures with humans from COCO datasets.



Train size

51,292 images containing 209,972 annotated human skeletons.

Val size

12,823 images containing 52,493 annotated human skeletons.

Test size

2,693 images containing 11,004 human skeleton annotations.

"NTU RGB+D"

"NTU RGB+D" datasets contain 56,880 and 114,480 action samples, respectively. This dataset includes 4 different modalities of data for each sample:

- RGB videos
- depth map sequences
- 3D skeletal data
- infrared (IR) videos

The resolutions of RGB videos are 1920×1080, depth maps and IR videos are all in 512×424, and 3D skeletal data contains the 3D locations of 25 major body joints at each frame. "NTU RGB+D" dataset contains 60 action classes. We adopted the following classes for the project:

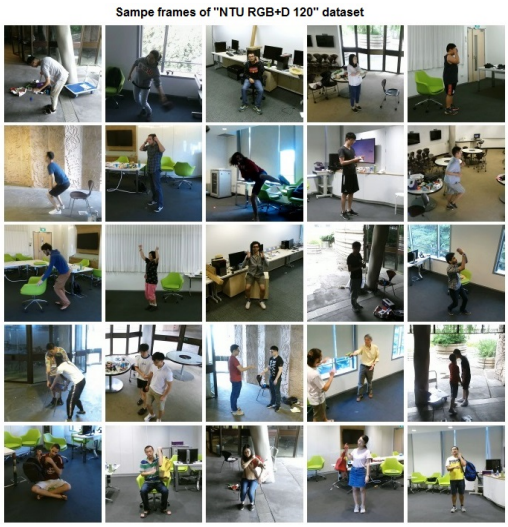
- Drink/Eat
- Sit/Squat
- Use a phone / Talk
- Take a selfie
- Walk

Although the NTU RGB+D includes skeleton data, it is in a higher dimension that requires specific convolutional layers not yet supported by TensorRT, which is necessary to optimize the model for use on an edge device. We labeled the NTU RGB+D videos with 2D skeletons inferred from the Resnet model trained on the COCO dataset, then used these skeletons to train against the action labels. This means that each sample consists of the skeleton vectors for a sequence of frames with an action label. We experimented with sequence lengths from 5 to 15 and found that length 9 had the best results.

Train Set: 185,799 sequences

Validation Set: 46,450 sequences

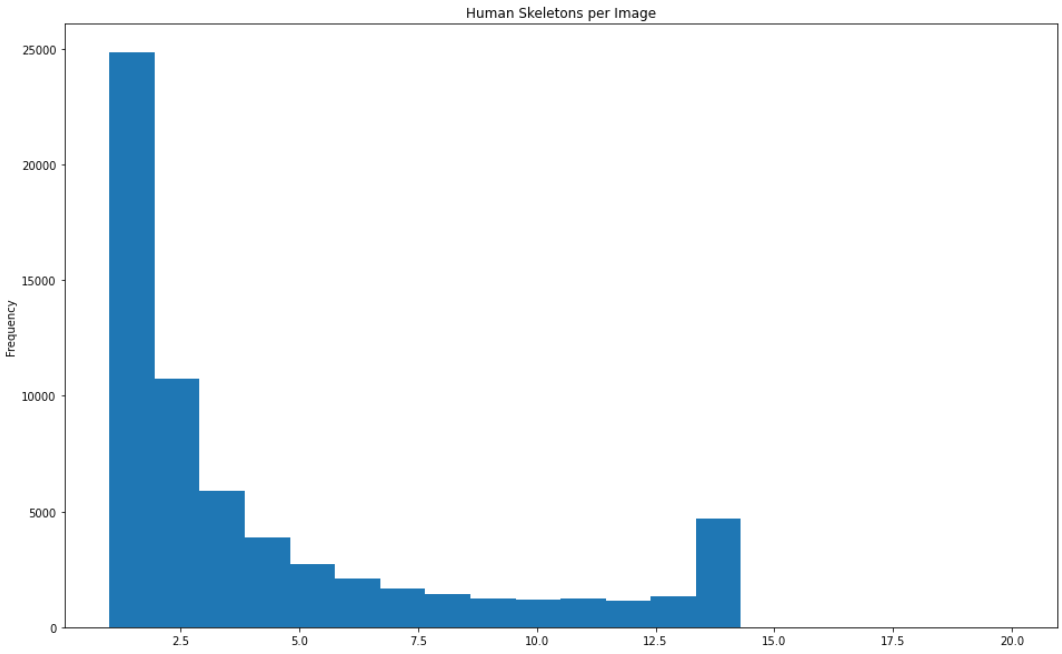
Test Set: 58,063 sequences



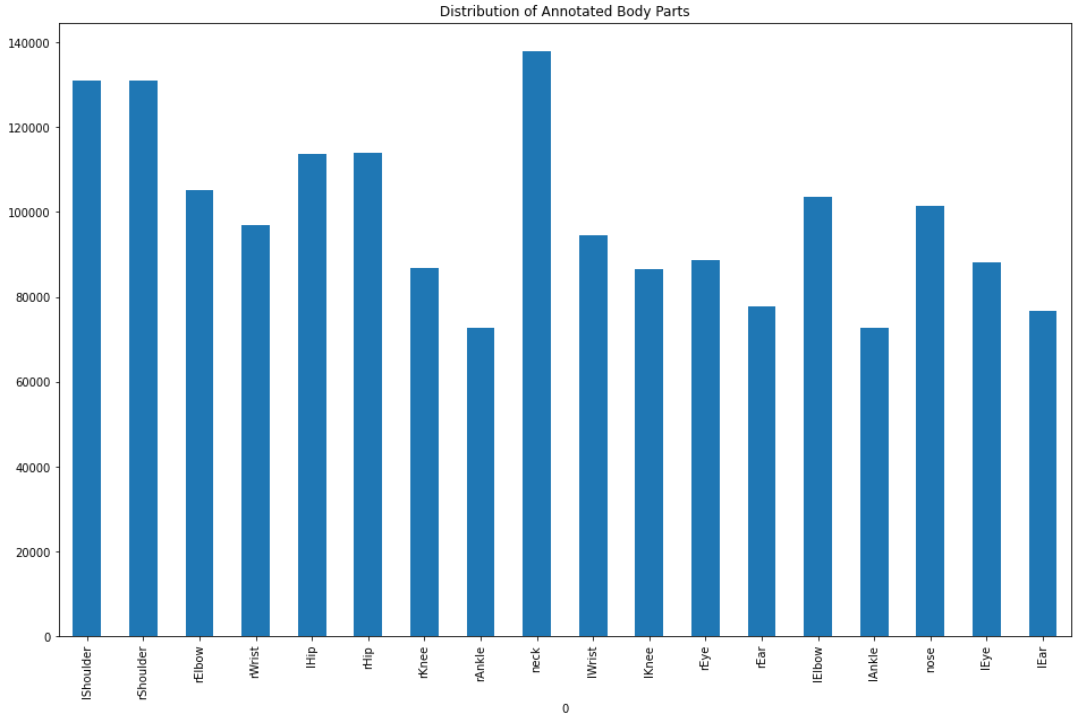
Exploratory Data Analysis

One of our requirements is to be able to individually classify multiple people in a frame at the same time. The COCO training dataset is not balanced between the specific number of people, but it is roughly balanced between a single person and all combinations of multiple people, which seems to be a reasonable distribution for our task.

For training set:

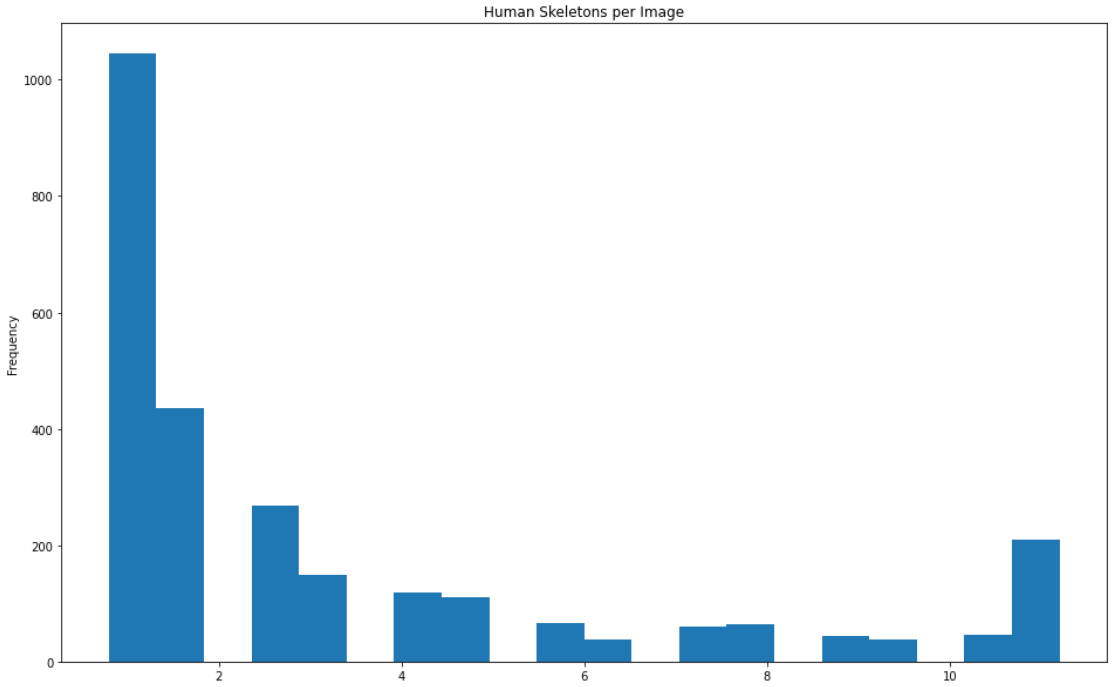


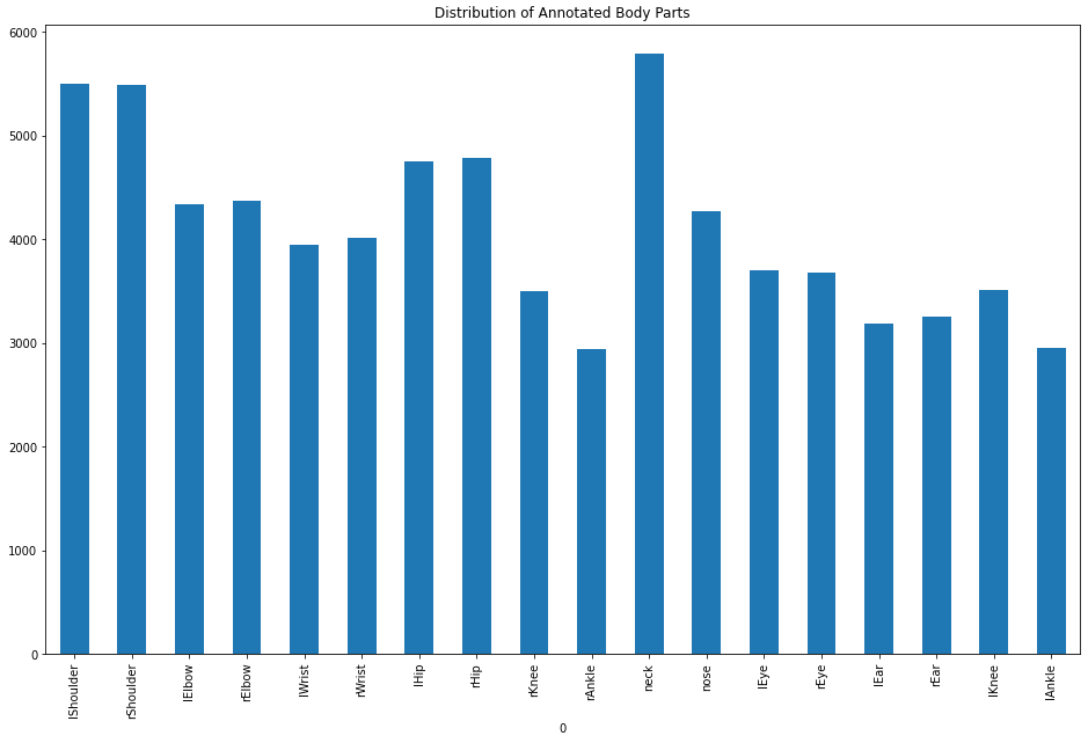
It is also important that we have a balanced distribution of body parts labeled on each human. The COCO training set isn't perfectly balanced, but does seem to be within a reasonable range for each body part.



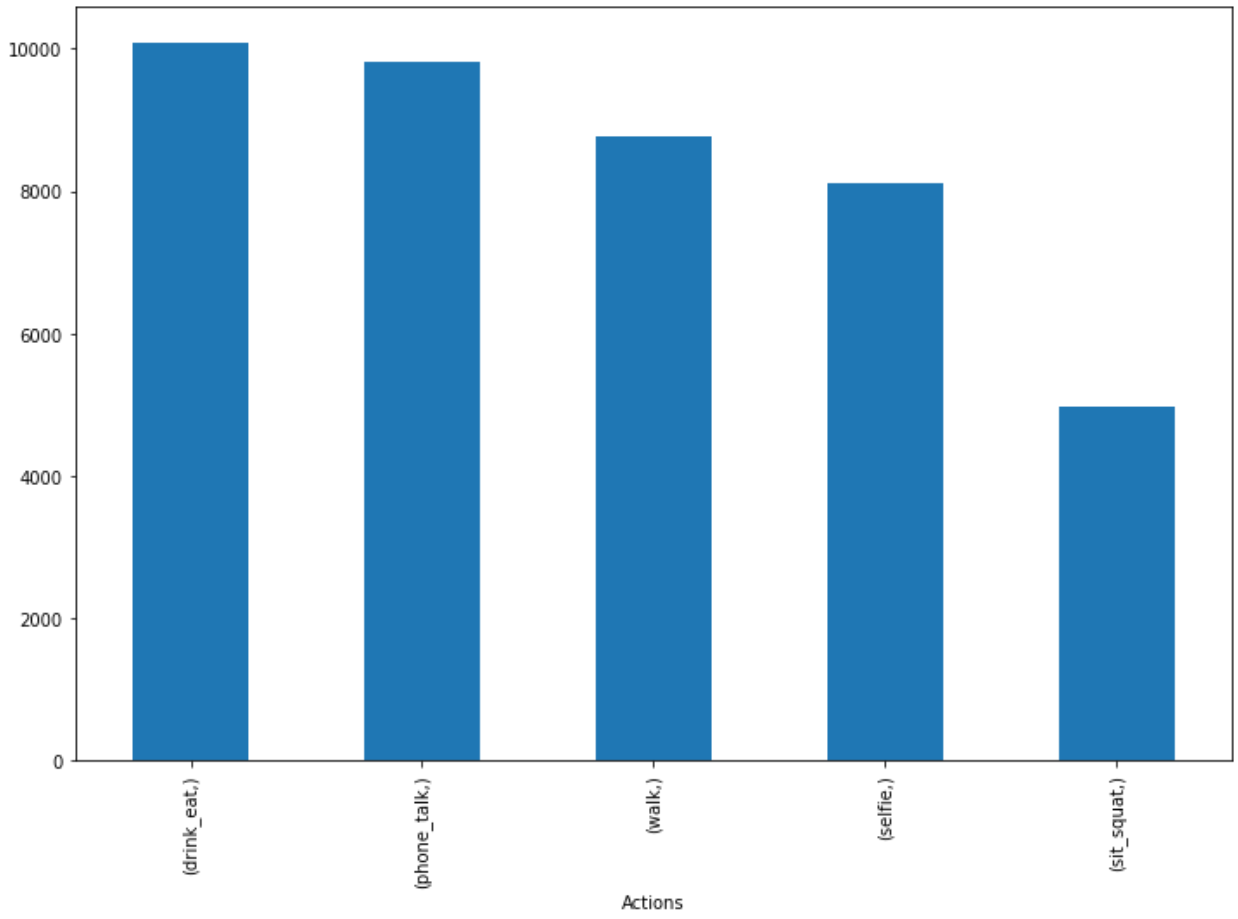
Additionally we verified that the validation distributions were not drastically different from the training data, although they were somewhat sparser, which is not unusual.

For validation set:

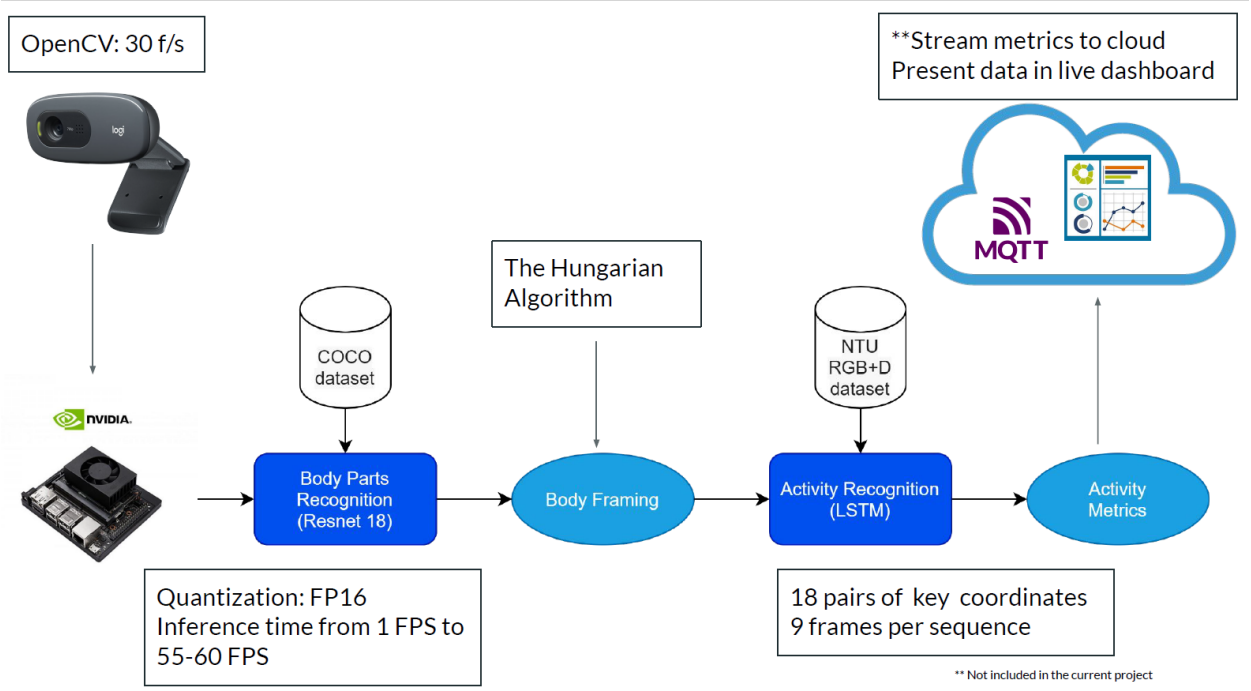




In the NTU RGB+D dataset, since our samples are sequences of frames, class balance is tied to the length of time required to perform an action. While most of our classes are roughly balanced, it's notable that sit/squat has fewer samples because it is a shorter action, which is why we included the 'squat' action in the same category to improve the number of classes as best we could with the available data.



Data Pipeline



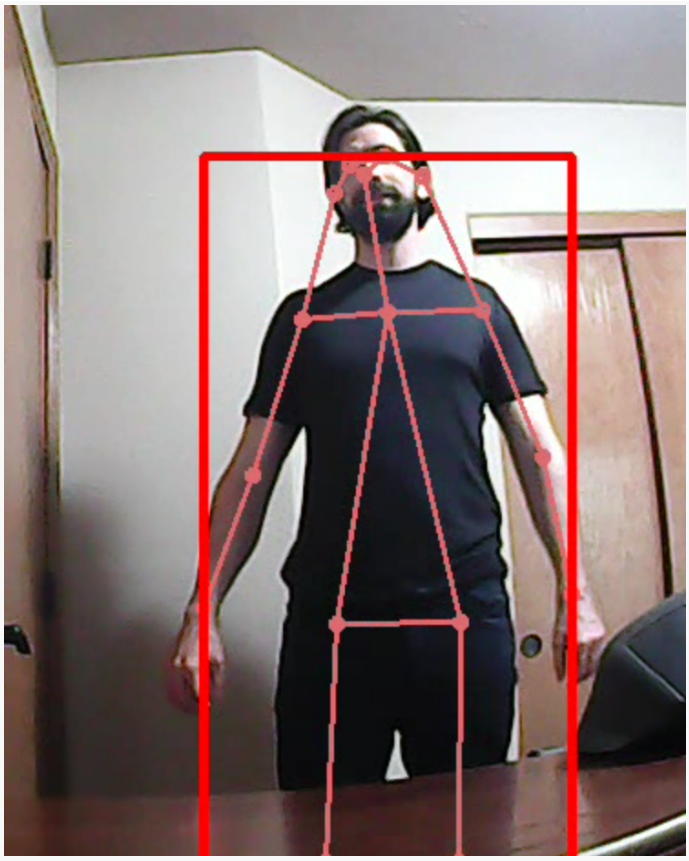
Camera

We use a camera which connects to Jetson and takes video clips. In our business case, the device should be put in the public space.

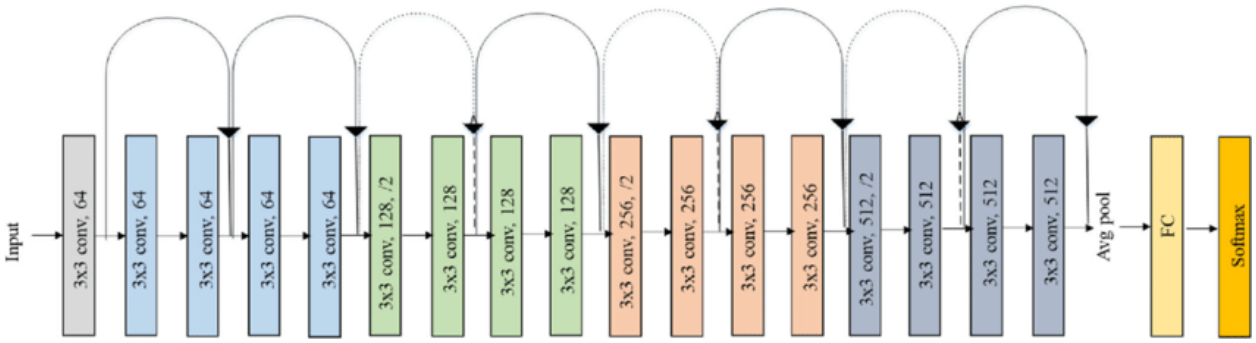
First level - Skeleton-Based Representations

We applied ResNet 18 trained on COCO dataset to develop our Skeleton-Based Representations. ResNet is a specific type of neural network that was introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in their paper “Deep Residual Learning for Image Recognition”. In order to enhance the recognition accuracy with limited datasets, we use transfer learning based on the ResNet Deep Neural Networks which results in improved accuracy and performance. The intuition behind adding more layers is that these layers progressively learn more complex features. As

shown in the following picture, we can see that we successfully identify the body parts for videos.



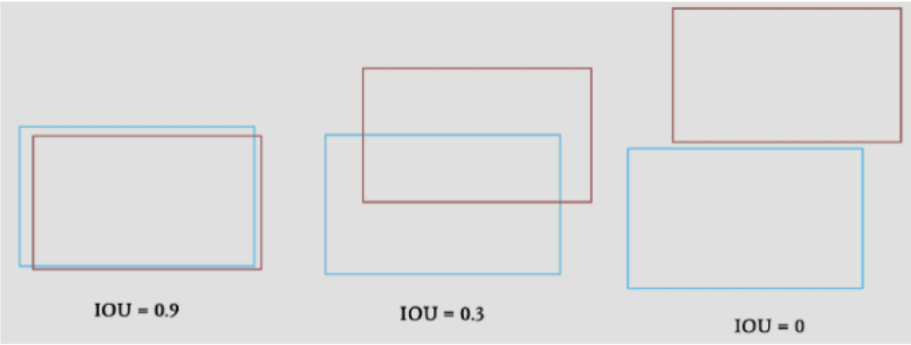
Structure of ResNet 18 :



This stage of our pipeline is by far the most computationally intensive and contains more than 99% of the parameters across the series of models. We used the torch2trt package to quantize our fine-tuned model to make it more suitable for real-time inference on the Jetson. Quantizing to float16 provides for less storage space, memory bandwidth, power consumption, lower inference latency and higher arithmetic speed. Before quantization, the Jetson achieved approximately 1 FPS inference speed, and afterwards the Jetson was able to achieve 55-60 FPS, making it more than sufficient for real-time inference in our pipeline.

Body Framing

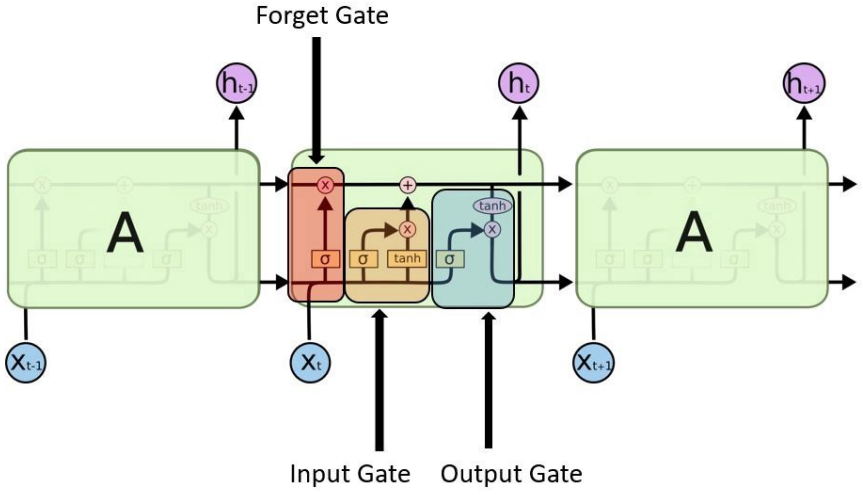
Hungarian Algorithm is applied to identify bodies that are showing continuously in a video. A crucial limitation of object detection algorithms is that they do not tell you if they're detecting the same object in sequential frames. To fulfill this requirement, we use the Hungarian algorithm, which can associate an object between one frame and the next. It calculates the intersection over union score, which is a value representing the amount of a bounding box that overlaps the previous frame. As illustrated in the following diagram, if the bounding box overlaps the previous one, it's probably the same object. For each frame, we compute a matrix of IOU scores for each box with the boxes from the previous frame and assume that the maximum scores are matches if they're above a specified threshold.



Second level - Activity recognition

For the second level, we have the body frame information along with key coordinates on each frame representing movement as the input from First level. We need to classify sequences of body pose coordinates as actions, and we know Long Short Term Memory (LSTM) network models are able to learn and remember over long sequences of input data. They are intended for use with data that consists of long sequences of data. So we built the second phase of the pipeline using LSTMs.

Long Short Term Memory (LSTM) is a special kind of RNN's, capable of learning long-term dependencies. LSTMs make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things. The information at a particular cell state has three different dependencies. i.e. The previous cell state; The previous hidden state; The input at the current time step. A common LSTM unit is composed of a **cell**, an **input gate**, an **output gate** and a **forget gate**. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. There are two states that are being transferred to the next cell; the **cell state** and the **hidden state**.



An LSTM model with 2 layers is used, as multi-layer models provide better structural representation of input i.e. body frame and activity. Also multi layer models learn

hierarchical features layer by layer. A dropout layer is implemented to avoid overfitting and achieve stronger generalization. The NTU RBG +D dataset is used to train the LSTM model to classify sequences of keypoint movements as actions, and thus is able classify actions from the body poses inferred from the previous stage.

Fine Tuning Models

We conducted a series of experiments to fine tune the body pose estimation (Resnet 18) parameters and action recognition model (LSTM). Please see the listed details in this spreadsheet.

For the body pose estimation, we tried the parameters in the following range:

- Image shape: 224 x 224 ; 368 x 368; 256 x 256
- Batch size: 64
- IoU: 0.5 - 0.95
- Area: all, medium, large
- maxDets: 20
- Optimizer: Adam

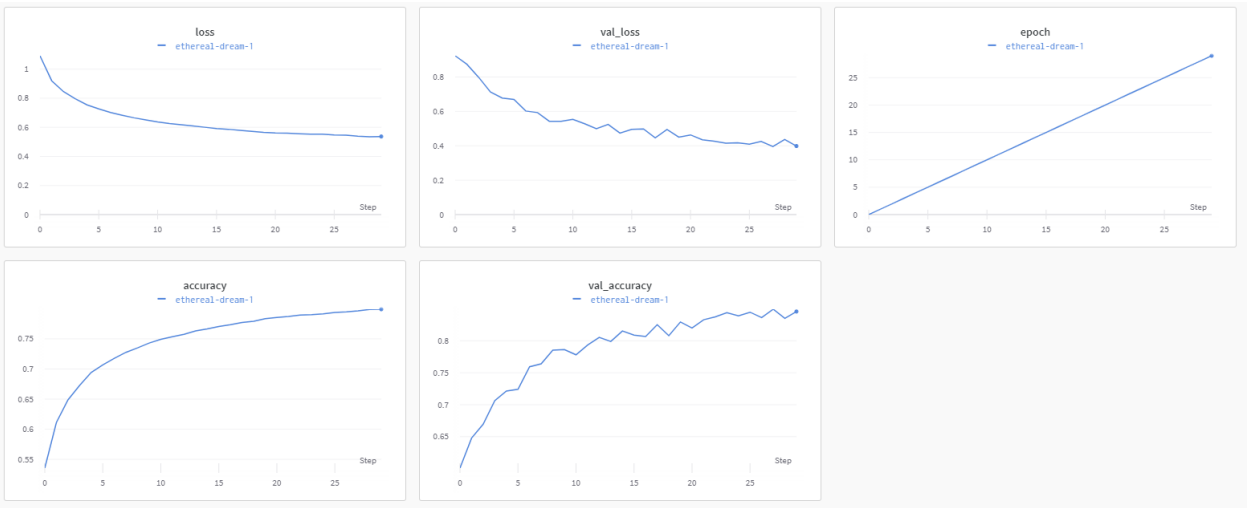
For the action recognition model, we tried the following parameters:

- Batch size: 32 - 96
- Windows: 9
- Dropout: 0.2 - 0.4
- Optimizer: RMS prop

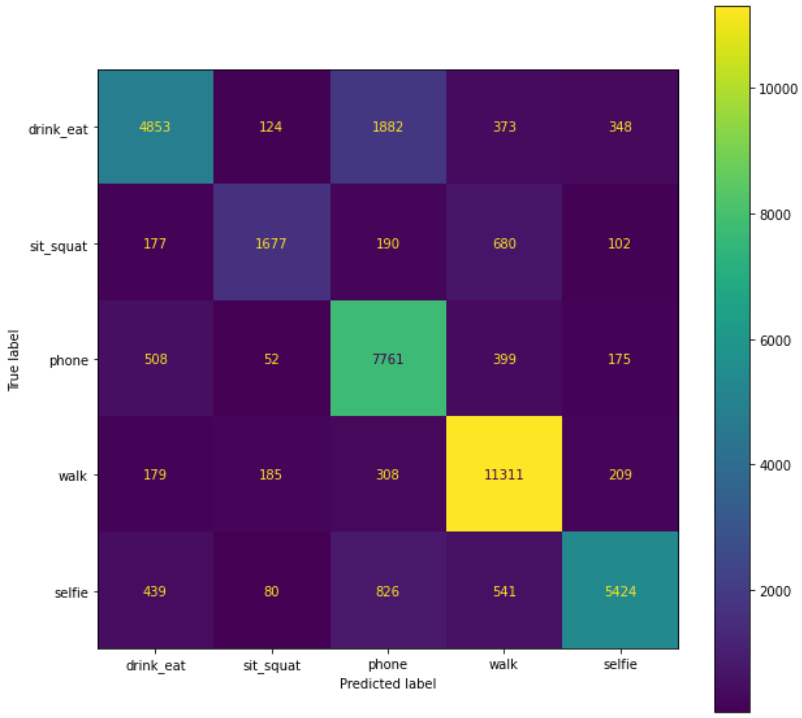
As our dataset is really big, in order to find out the performance for different combinations, we used the 30 epochs to test the val. After testing different combinations of the parameters, the following parameter sets had the best performance: for the body pose estimation, image shape 368 x 368, batch size 64, IoU: 0.5, area: all, max Dets: 20, optimizer, Adam; for the action recognition model, layer depth: 128 x 64, windows: 5, batch size: 96, Dropout: 0.2.

With applying the best set of performance for 30 epochs for LSTM, our final model achieved a validation loss as 0.3985, validation accuracy as 0.8458.

Here is the validation accuracy over time for training of our best model:



Here is the confusion matrix after we run the model on the validation set:



The confusion matrix indicates that we did a good job of recognizing the actions. We also noticed that lots of our actions are labelled as “walk”. Part of the reason is that there are more videos in the dataset is labelled as “walk”, so the distribution of video categories is not well balanced. Category “phone” and “drink_eat” are easily confused with each other. In the further research,

we can add more labelled videos for the categories other than “walk”, and need extra work to better distinguish between “phone” and “selfie”, as well as “phone” and “drink_eat”.

Below is the classification report for our held out test dataset.

	precision	recall	f1-score	support
drink/eat	0.88	0.73	0.80	13746
phone	0.72	0.89	0.80	13319
selfie	0.91	0.79	0.85	13319
sit/squat	0.80	0.82	0.81	6884
walk	0.90	0.93	0.91	12817
accuracy			0.83	58063
average	0.84	0.83	0.83	58063

Challenges

One of the big challenges that we have seen in the project is identifying people's actions from available dataset. This challenge can be broken down into three parts: 1) There are multiple people in the videos who are doing different actions, models easily get confused and are trained that they are all having the same action. For example, there are 2 people in one video which is labeled as “walk”. One of them is waiting at the same place, while another is actually walking. After we passed the videos into the models, the output shows that lots of “standing” people are mislabelled as “walking”. In order to solve the issue, we switched from HMDB51 dataset, which includes multiple people in one video, to NTU RGB+D dataset, which only includes one person per video for a single action and 2 people for a paired action (such as shaking hands). Using this NTU RGB + D dataset significantly improved our accuracy and confusion matrix, however, the downside is that the action categories provided by NTU dataset are too specific. Based on the business usage of our project, we are looking for some genetic actions such as “walk”, “run”, “talk”, “take photos”, etc. Therefore, out of 60 provided categories in the NTU dataset, we picked 5 categories which are most relevant to our use case.

2) Action detection involves a sequence of human pose changes in frames. In our final model, we used 9 frames per window to run our train and val datasets. It is because, according to the most recent research, 5-7 frames (0.3-0.5 seconds of video) will be enough to recognize actions (Schindler, Van Gool). Originally, we would like to use an open source tool MMAction to conduct Spatio Temporal Action Detection Models SlowFast, while their “export to ONNX” function for the SlowFast model is still under development and thus we have to implement LSTM to implement the action recognition.

3) Action recognition model requires a large amount of data to train. Although NTU RGB+D is a large scale dataset, we only chose 5 relevant categories to train our model. Ideally, a large amount of clean formatted, clear labelled, and relevant dataset will help with our accuracy. Our experience states again how important the data is to the model training.

Next Steps

For our next steps, at the scope of model, we can implement Proposed Part-Aware LSTM, which has been mentioned in the research paper “NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis” that P-LSTM got the highest accuracy (70.27%) among all experiments.

In the perspective of the project, the next steps could be to have the model recognize the gender, age range, and other demographic information. It can help the park administration to make decisions whether they can add more kids' playground, or youth education center etc. Some datasets have been labeled as “smile” and we can pass videos with different facial expressions to train the model to track people’s emotions such as “laugh” “smile” “cry” etc.

Other than adding more features to the model, we can also stream the human activity metrics to a dashboard to help businesses or organizations to make decisions. As this data pipeline outputs the number of people, actions, and their time span, we can pass the data and build a dashboard which can show the live time change of park visits, time spent in a certain area. Entertainment parks, such as Disneyland or Universal Studio, can have these implemented, and then use the dashboard to better arrange the stores or shops, or have special events arranged at certain times of the day.

References

Github Repo: Human Activity Monitoring in Public Places
https://github.com/fengyaoluo/Human-Activity-Monitoring-in-Public-Places

Demo video:
https://github.com/fengyaoluo/Human-Activity-Monitoring-in-Public-Places/blob/main/demo/human_activity_demo.mp4

2D Skeleton Pose Estimation
https://docs.nvidia.com/isaac/isaac/packages/skeleton_pose_estimation/doc/2Dskeleton_pose_estimation.html

COCO DataSet
https://cocodataset.org/#home

Illustrated Guide to LSTM's and GRU's: A step by step explanation
https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Schindler, Van Gool, Action Snippets: How many frames does human action recognition require?
https://ethz.ch/content/dam/ethz/special-interest/baug/igp/photogrammetry-remote-sensing-dam/documents/pdf/schindler08cvpr.pdf

Skeleton-Based Activity Recognition: Preprocessing and Approaches
https://link.springer.com/chapter/10.1007/978-3-030-68590-4_2

MMAction2's documentation
https://mmaction2.readthedocs.io/en/latest/index.html#

"NTU RGB+D" Action Recognition Dataset
https://github.com/shahroudy/NTURGB-D

Real-Time Action Recognition Using Multi-level Action Descriptor and DNN
https://www.intechopen.com/chapters/61855

A review on applications of activity recognition systems with regard to performance and evaluation
https://journals.sagepub.com/doi/10.1177/1550147716665520

Intelligent Video Surveillance: Recent Trends And What Lies Ahead
https://www.alliedtelesis.com/en/blog/intelligent-video-surveillance-recent-trends-and-what-lies-ahead

The Hungarian Algorithm

https://towardsdatascience.com/computer-vision-for-tracking-8220759eee85

Torch2trt

https://github.com/NVIDIA-AI-IOT/torch2trt