RunMo

Run Motion Capture and Analysis on the Cloud and on Edge

James Wall, Tosin Akinpelu, and Mumin Khan

THE TEAM

James Wall

Mumin Khan

Tosin Akinpelu

INTRODUCTION

- Running is one of the world's most democratized sport
- Yet access to tailored mentorship is often expensive and exclusive
- Poor running posture can lead to:
  - Plantar fasciitis
  - ITB syndrome
  - Runner's knee
  - Shin splints
  - Achilles tendonitis
  - More serious muscle and tendon tears

EXISTING SOLUTIONS

- Interactive workout apps
- Human coaching
- Physical posture correctors
- Sensor based correctors

CAN WE DO BETTER?

RunMo was built to bridge the gap between general AI workout and tailored running solutions

HIGH LEVEL OVERVIEW

01 Record User on Edge

Users can use any camera device with internet to stream themselves running

02 Persist Data on Cloud

Data should be collected and stored off-device for future re-analysis

03 Learn User's Features on Cloud

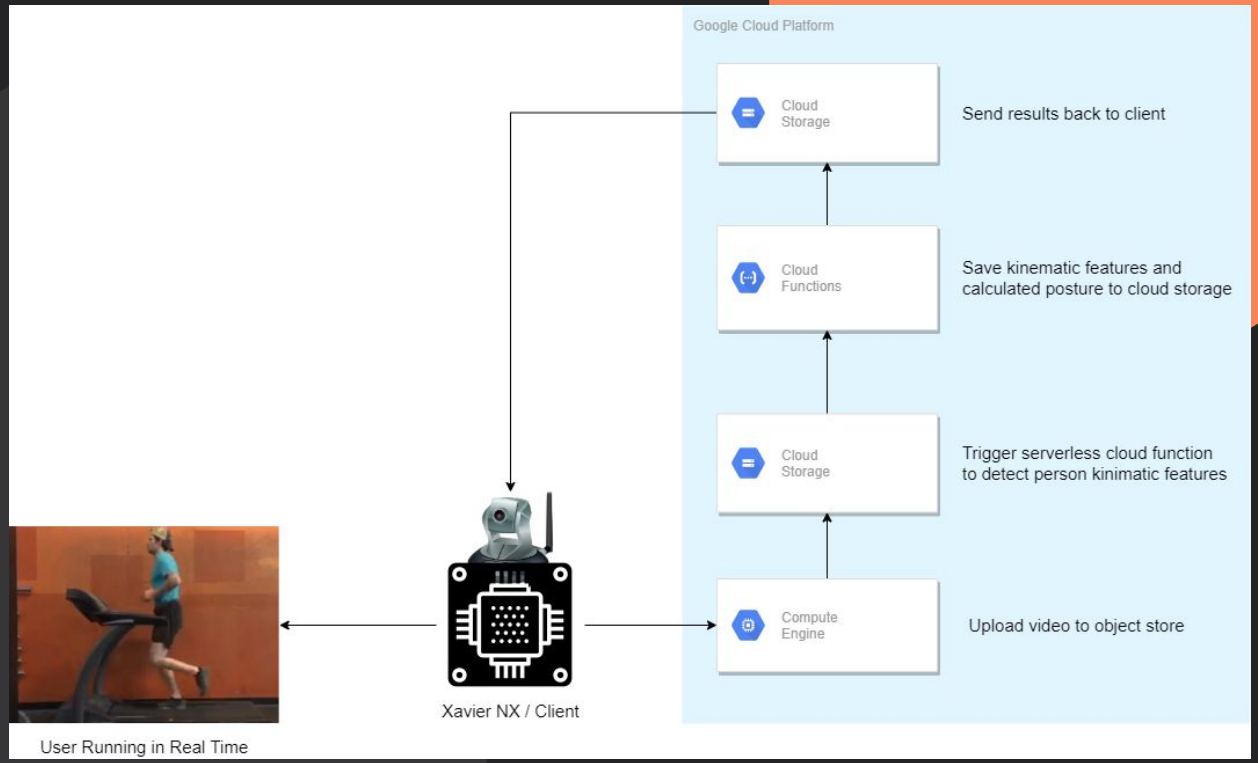
RunMo should provide near real-time recommendations

04 Send Results to Edge

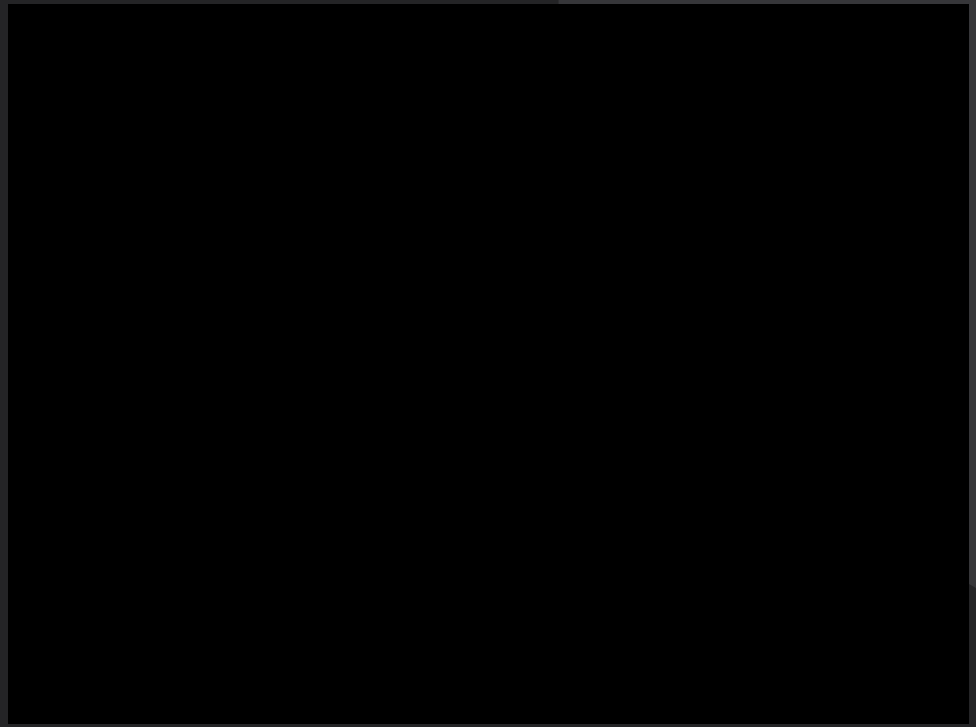
Users get to quickly consume the results of their analysis

Begin James

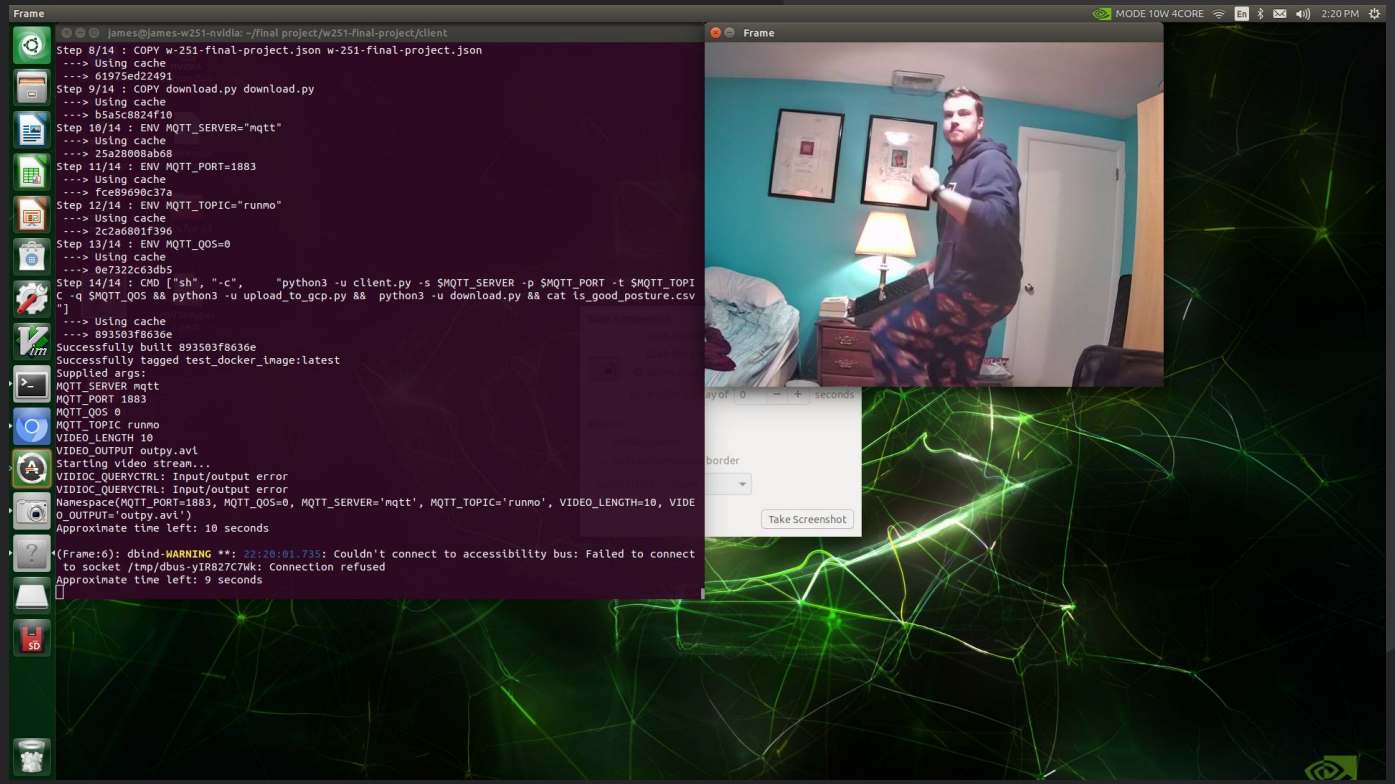
ARCHITECTURE



VIDEO DEMO



CAPTURE ON NX XAVIER

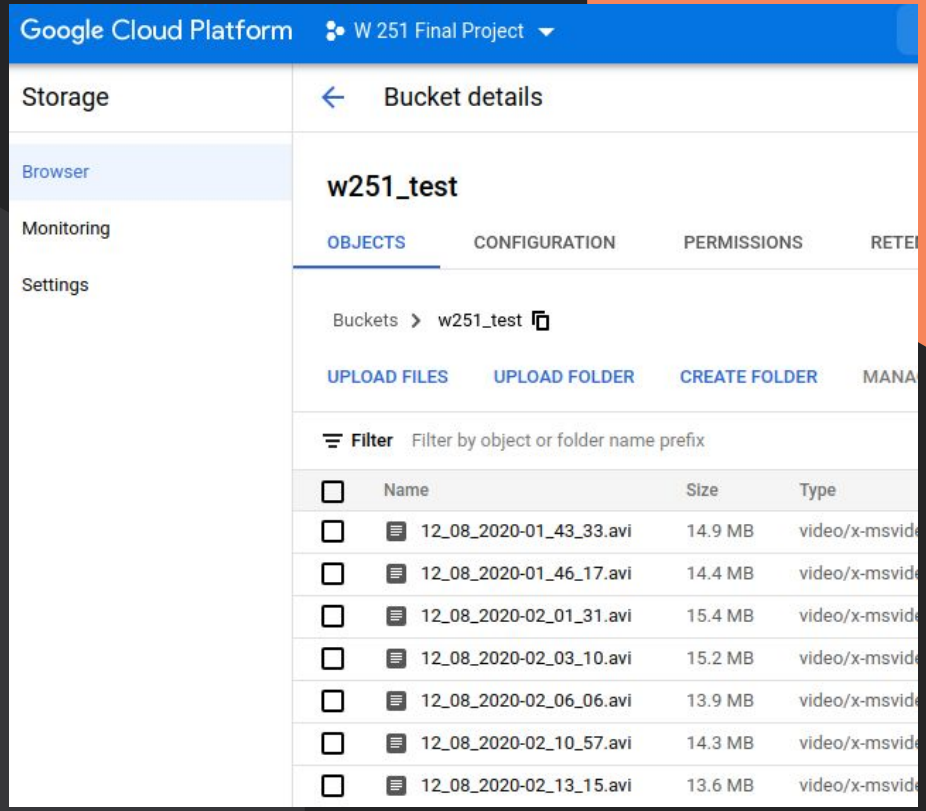


CLOUD INTERFACE

```
Approximate time left: 10 seconds

(Frame:6): dbind-WARNING **: 22:20:01
 to socket /tmp/dbus-yIR827C7Wk: Conn
Approximate time left: 9 seconds
Approximate time left: 8 seconds
Approximate time left: 7 seconds
Approximate time left: 6 seconds
Approximate time left: 5 seconds
Approximate time left: 4 seconds
Approximate time left: 3 seconds
Approximate time left: 2 seconds
Approximate time left: 1 seconds
Upload credentials set
Supplied args:
UPLOAD_OBJ outpy.avi
DESTINATION_PATH 12_08_2020-22_20_13.
BUCKET_NAME w251_test
Upload started
```

DATA PERSISTENCE



PROCESSING MODEL FEATURES

Cloud Functions | Function details | EDIT | DELETE | COPY

w-251-final-project-test | Version 1, deployed at Dec 5, 2020, 6:05:45 PM...

METRICS | DETAILS | SOURCE | VARIABLES | TRIGGER | PERMISSIONS | LOGS | TESTING

Runtime : Python 3.7 **Entry point** : main

main.py

requirements.txt

```python
import datetime
import math
import pandas as pd
from time import sleep
from google.cloud import videointelligence_v1p3beta1 as videointelligence

class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y


def getAngle(a, b, c):
    ang = math.degrees(math.atan2(c.y-b.y, c.x-b.x) -
                       math.atan2(a.y-b.y, a.x-b.x))
    return ang


def computeRightElbowAngle(row):
    wrist = Point(row[f'right_wrist_x'], row[f'right_wrist_y'])
    elbow = Point(row[f'right_elbow_x'], row[f'right_elbow_y'])
    shoulder = Point(row[f'right_shoulder_x'], row[f'right_shoulder_y'])
    return getAngle(wrist, elbow, shoulder)

def computeRightShoulderAngle(row):
    elbow = Point(row[f'right_elbow_x'], row[f'right_elbow_y'])
    shoulder = Point(row[f'right_shoulder_x'], row[f'right_shoulder_y'])
    hip = Point(row[f'right_hip_x'], row[f'right_hip_y'])
    return getAngle(hip, shoulder, elbow)


def computeRightKneeAngle(row):
    hip = Point(row[f'right_hip_x'], row[f'right_hip_y'])
    knee = Point(row[f'right_knee_x'], row[f'right_knee_y'])
    ankle = Point(row[f'right_ankle_x'], row[f'right_ankle_y'])
    return getAngle(ankle, knee, hip)
```

POSTURE CALCULATION

```python
def isGoodPosture(row):
  stride_length = row['stride_length']
  ideal_stride = row['ideal_stride']

  right_elbow_angle = row['right_elbow_angle']
  # right_shoulder_angle = row['right_shoulder_angle']
  right_knee_angle = row['right_knee_angle']

  left_elbow_angle = row['left_elbow_angle']
  # left_shoulder_angle = row['left_shoulder_angle']
  left_knee_angle = row['left_knee_angle']

  back_angle = row['back_angle']

  # Calculate weighted metric on whether good posture or not
  # Amount added ranges from [0,1]. Higher added values indicates more importance for posture
  weighted_posture_metric = 0

  # TODO get rid of hardcoded weights - maybe have standardized low, medium, and high weight variables defined before this method?
  if isGoodElbowAngle(right_elbow_angle):
    weighted_posture_metric = weighted_posture_metric + 0.4
  if isGoodElbowAngle(left_elbow_angle):
    weighted_posture_metric = weighted_posture_metric + 0.4

  if isGoodKneeAngle(right_knee_angle):
    weighted_posture_metric = weighted_posture_metric + 0.8
  if isGoodKneeAngle(left_knee_angle):
    weighted_posture_metric = weighted_posture_metric + 0.8

  if isGoodBackAngle(back_angle):
    weighted_posture_metric = weighted_posture_metric + 0.8

  if isStrideLength(stride_length, ideal_stride):
    weighted_posture_metric = weighted_posture_metric + 0.9

  # Make call whether posture is good enough or not
  # TODO check if there is a method to this so it is less arbitrary
  # Currently highest possible total is 4.3
  # Probably missing two 0.8s is ok, so we'll say over 2.7 is good
```

```
DOWNLOAD_OBJ is_good_posture.csv
DESTINATION_PATH 12_08_2020-22_20_40.xlsx
BUCKET_NAME w251_test
Download started
Blob is_good_posture.csv downloaded to is_good_posture.csv.
Download finished
TIMESTAMP,is_good_posture?
0,TRUE
0.01,FALSE
0.02,TRUE
0.03,TRUE
0.04,TRUE
0.05,TRUE
0.06,FALSE
0.07,FALSE
0.08,TRUE
0.09,TRUE
0.1,FALSE
0.11,TRUE
0.12,TRUE
0.13,TRUE
0.14,TRUE
0.15,FALSE
0.16,FALSE
0.17,FALSE
0.18,TRUE
0.19,FALSE
0.2,FALSE
0.21,TRUE
0.22,TRUE
0.23,FALSE
```

RESULT REPORTING

Nvidia Xavier NX receiving recommendation from RunMo pipeline

Begin Tosin

LEARNINGS

- No need to reinvent the wheel under time crunch; explored available MLaaS cloud solutions:
  - Amazon, IBM, Microsoft & Google offer video analytics REST APIs (some in beta versions)
  - These platforms leverage advances in Deep Learning under the hood
  - We chose Google Video Intelligence for their Pose Detection service which is relevant to our project
- Ideas are great but execution is ‘everything’

CHALLENGES

THE APP DESIGN

- MQTT protocol (a core feature of our project workflow) does not support video robustly
- API did not work when all points were not present
  - Robust error handling for when ankle, etc. were missing from frame

MURPHY'S LAW

- NX breaks just in time, house moves, covid-19

Conclusion

- RunMo makes live running advisory service accessible to runners with minimal hardware investment
- The solution can be extended to wider application areas leveraging Deep Learning in the Cloud.

Thank you!

Questions, Comments, or Concerns?