ShotSense

A Cricket Visualisation Tool

Umair Ali Khan & Muhammad Fahad

Problem Statement

- Casual cricketers are hindered by lack of equipment to self evaluate their game
- Tech in cricket at international level is very expensive
- Million-dollar setups are inaccessible to amateurs
- The unavailability of technology for self-evaluation may demotivate street cricketers who aspire to be recognized

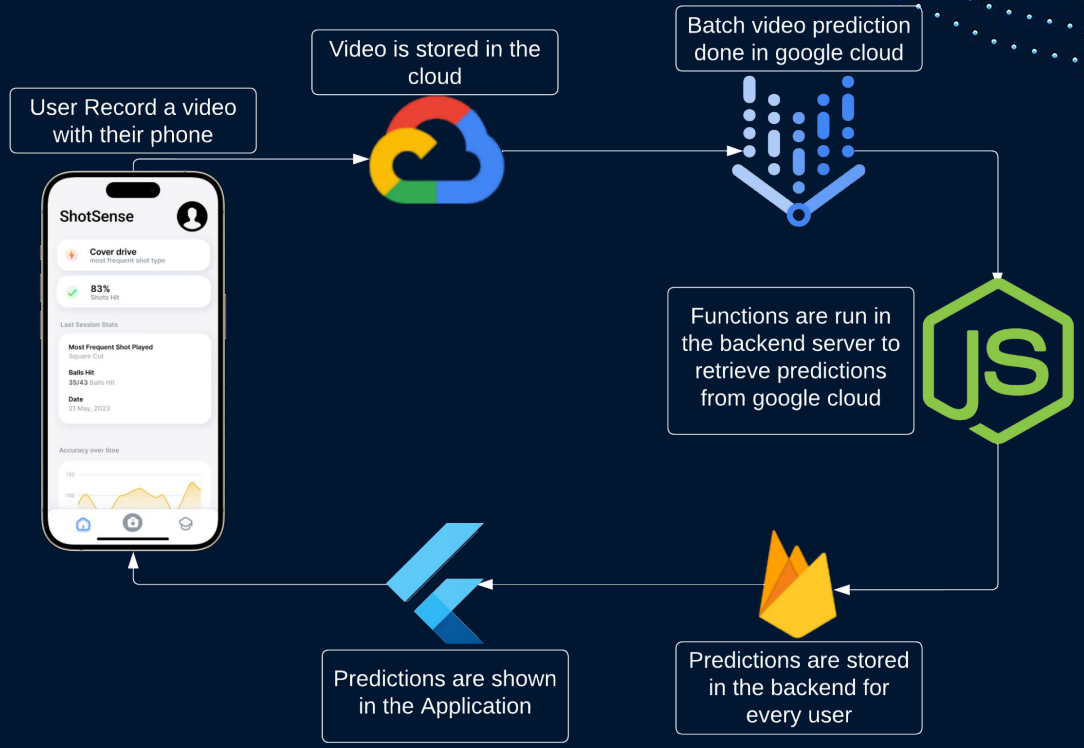
Solution

- A mobile application that provides tools for visualisations
- ShotSense imitates the visualizations seen in international broadcasts
- ShotSense identifies cricket shots and tracks ball deliveries for visualizations
- Visual plots provides overview for the complete session

Methodology

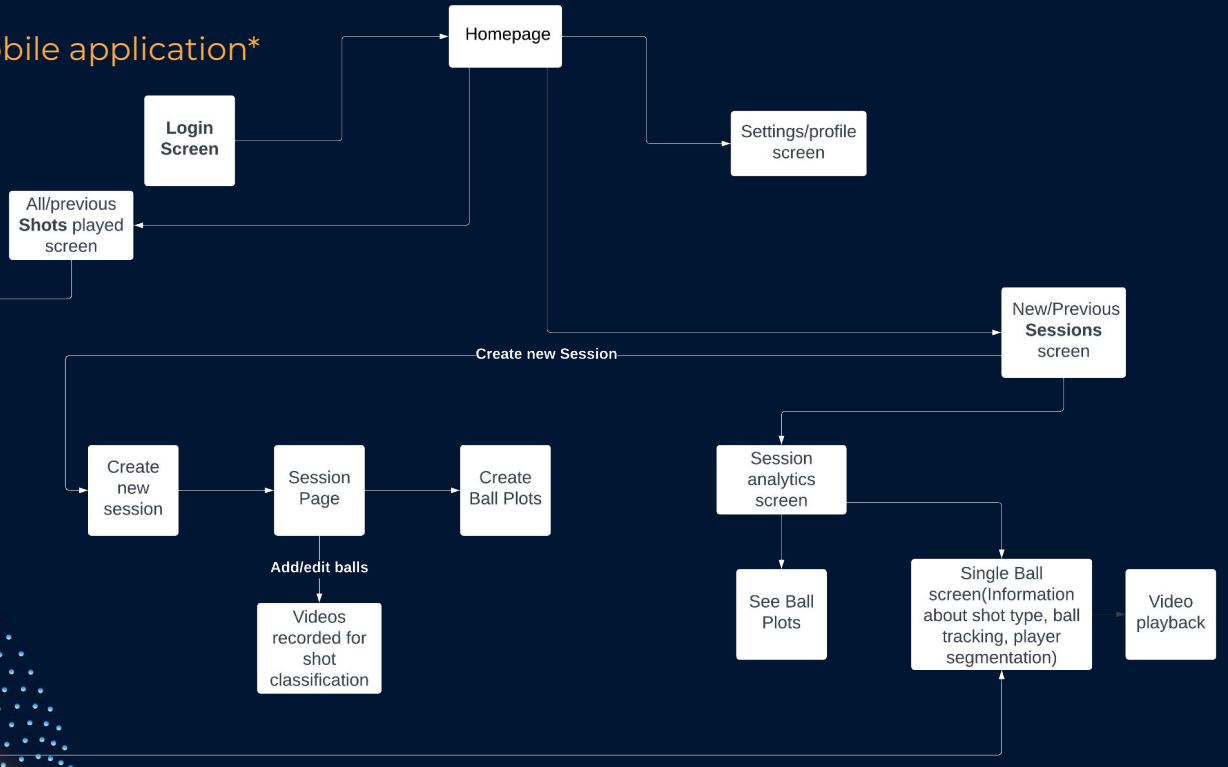
- **User research**
  - Talked to BNU coach to discuss basic types of shots to target for shot classification
  - Researched on cricket visualisation tools used in broadcasts
  - Shortlisted tools we can imitate in app
- **Custom Models**
  - Explored different pre processing techniques
  - Explored 3D ConvNets
  - PreTrained Models such as MoViNets, EfficientNet, VGG16
  - Evaluated test results on each of the models to choose the most appropriate on
- **Google Cloud Vertex AI**
  - Enabling us to train and deploy our model on the cloud
  - Running batch predictions on the cloud avoiding on-device processing (faster)
  - Using google cloud storage as a storage for videos and their predictions

High Level Architecture



Mobile Application Architecture

The high level view of our mobile application*



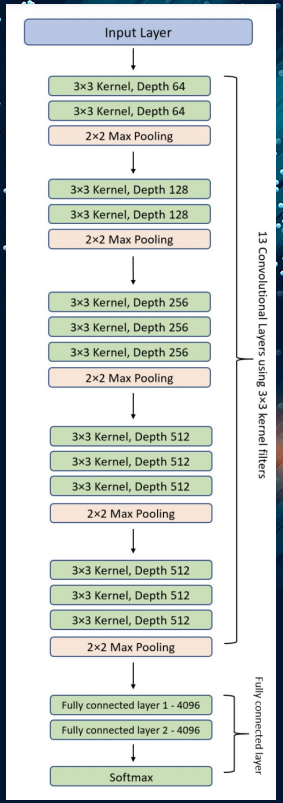
*subject to changes in FYP-II with improved requirements

Interactive Demo on Mobile Device

ML Implementation

Transfer Based Learning

- Used a pre trained VGG-16 Model on our custom dataset
- Used the model to extract features from our videos
- Once these features were extracted we introduced the model to our fully connected layers for classification
- Achieved an accuracy of 70% on our test set
- Converted our model to TF Lite for on device



ML Implementation contd.

Google Cloud Vertex AI

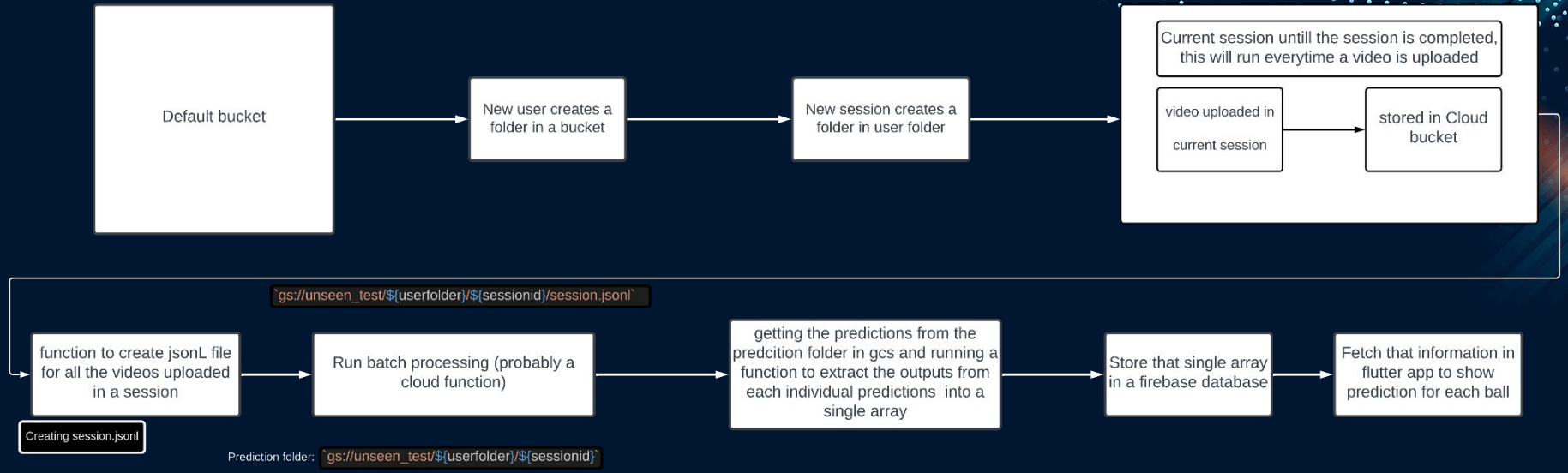
- Uploaded our data to Google Cloud Storage
- Created a dataset by manually annotating timestamps for when shots are occurring in videos (required for action recognition tasks)
- Trained an AutoML model
- Achieved an F1 score of 0.88
- Ran batch predictions on unseen dataset to check results on real life examples. Achieved accuracy of over 70%
- Designed a backend to process user uploaded session videos as batch predictions

ML Implementation Conclusion

Why Choose Google Cloud Vertex AI?

- Google Cloud Platform allows us to create an end-to-end ML pipeline
- Google Cloud Storage enables secure and efficient data storage
- Vertex AI streamlines training, development and deployment of the model
- Faster inferences than on-device predictions

Cloud Backend Design Architecture



*subject to changes in FYP-II with improved requirements

Issues Faced

- VGG16 with trainable parameters need substantial computing resources
- Compatibility issues, especially on iOS devices not supporting tensorflow lite
- Online prediction, which creates endpoints for video prediction
- Limited balance for google cloud(300$)

Solution

- Moving to a cloud infrastructure
- Creating a backend system to get the shot predictions using batch processing
- Creating multiple Google accounts for more credit

Technologies Used

Flutter

Code once use everywhere approach

Flutter integrates seamlessly with Firebase

Deployments to multiple platforms making it accessible for all users

Firebase

A Unified Backend Ecosystem

Integration of authentication with various different methods

Creating databases for users, shot predictions

Storing videos of shots played across the sessions

Node APIs

Custom node apis are used to connect in flutter

Functions are created to use to list all the json files

Connecting vertex Ai to trigger batch prediction

Cloud Vertex AI

Created a shot classification dataset

Utilized Googles AutoML to train a model

Model stored in google clouds model registry

Running batch predictions on user uploaded session videos

*subject to changes in FYP-II with improved requirements

Plans for FYP-II

Ball Tracking

- Show types of deliveries faced

Visual Plots

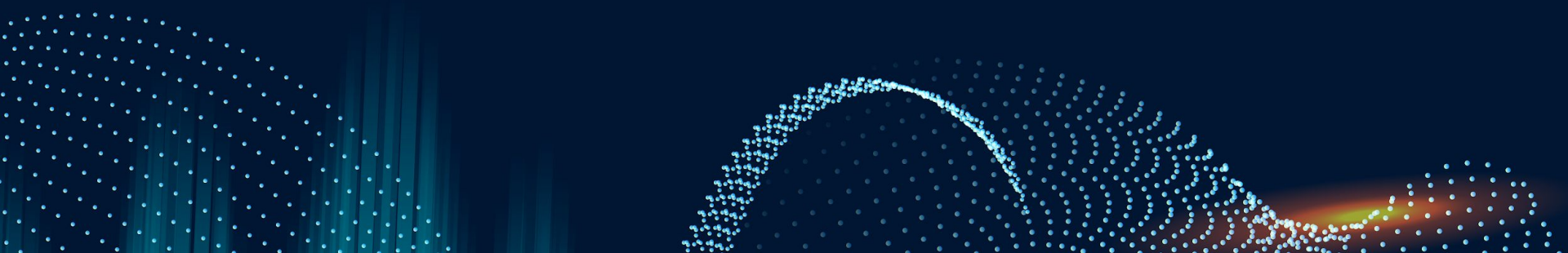
- Player Segmentation
- Pose Tracking
- Ball Plots

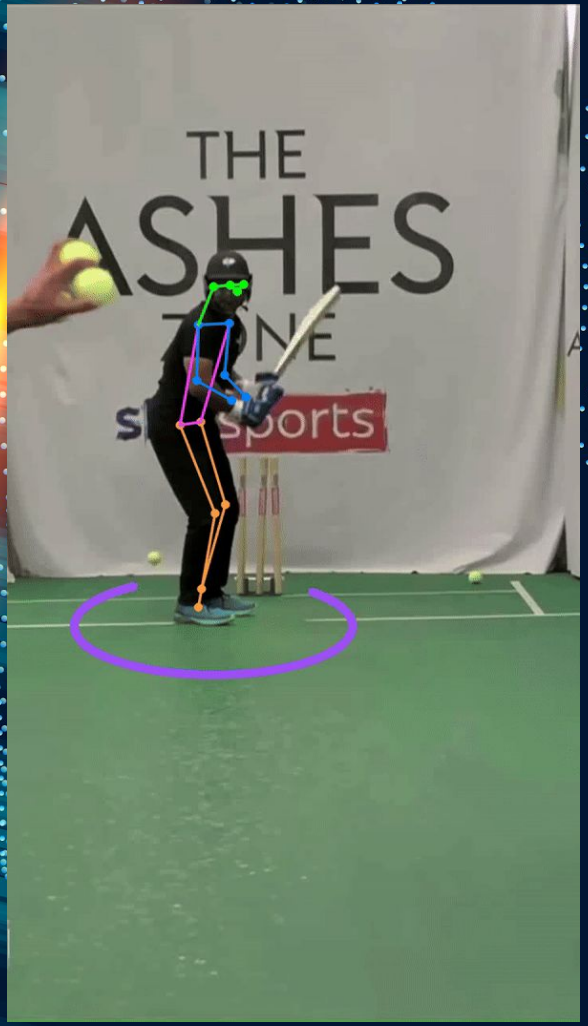
Key Analytics

- Most frequent shots
- Shot Accuracy

User Testing

- Deploying the app
- Test with BNU team





Player Segmentation & Pose Tracking

Thank You!