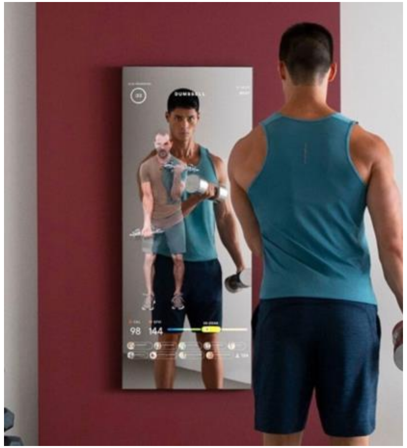
Falcon: the Pro Gym Assistant

Team Ao
Vishal Baskar
Albert Chen
Venkata Vivek Thallam

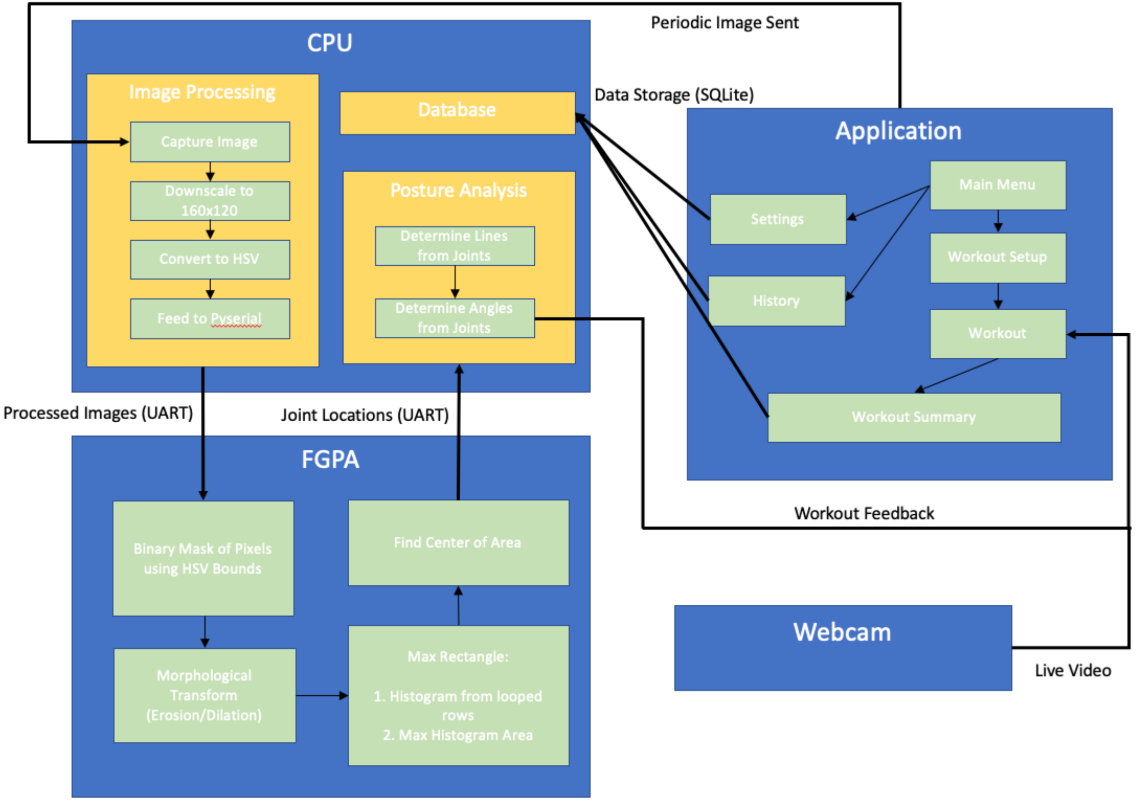
Use Case

- Advanced at-home workout system that provides:
  - Demonstration of exercises
  - Rep counter
  - Calorie estimator
  - Live stream of themselves
  - Ability to get customized workouts that comprise of the following:
    - Leg Raises, Pushups, Lunges
  - Real-time feedback regarding posture **(Unique to Falcon)**
- Workout system involves a display and a side camera
- Processing done on an FPGA to address privacy concerns
- Areas Covered:
  - Software Systems, Signals and Systems, Hardware Systems

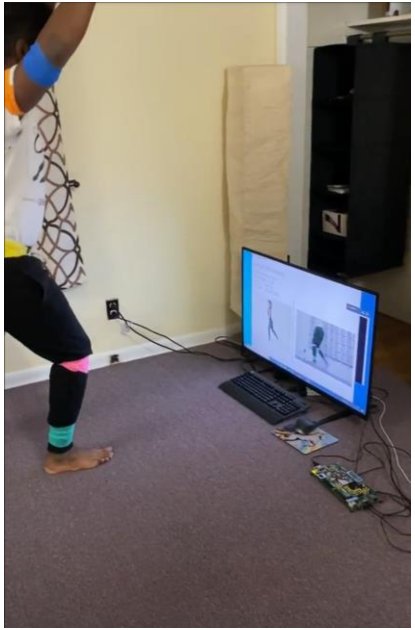


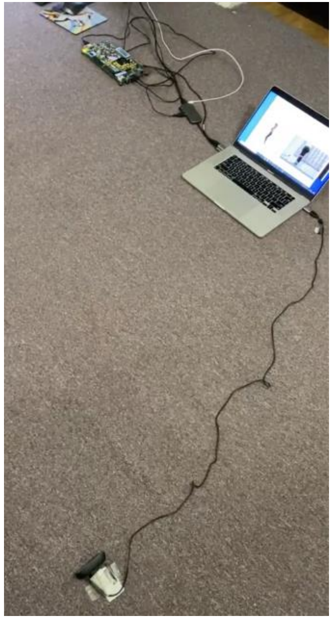
Mirror: A popular at-home workout tool

Solution Approach

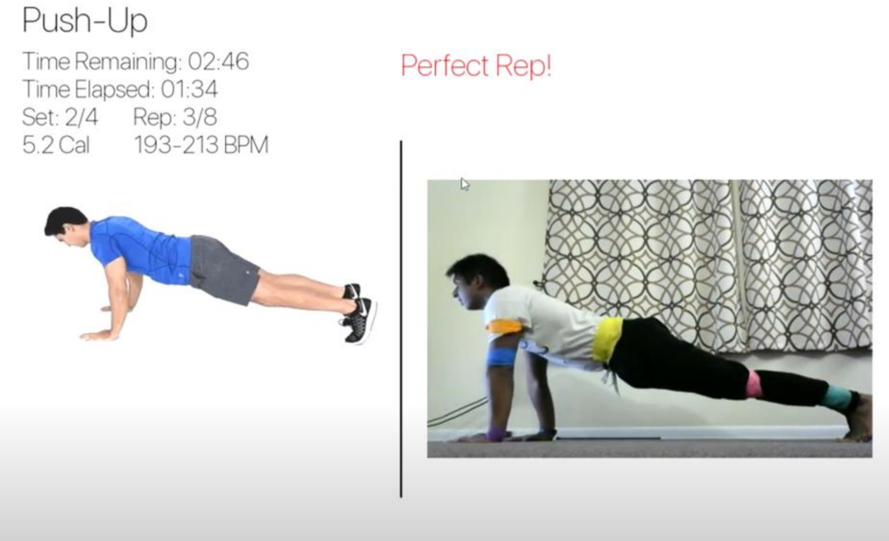


Hardware Setup





Complete Solution



Push-Up

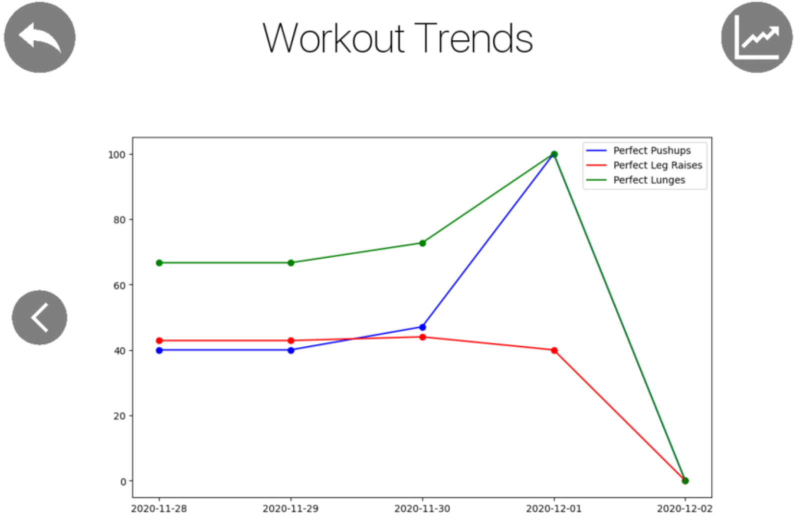
Time Remaining: 02:46

Time Elapsed: 01:34

Set: 2/4 Rep: 3/8

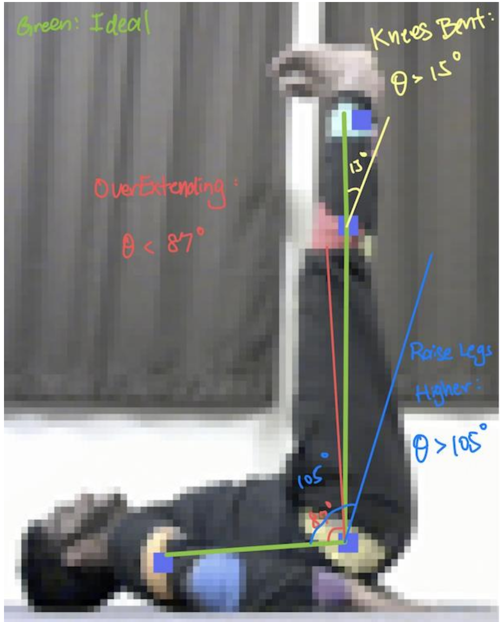
5.2 Cal 193-213 BPM

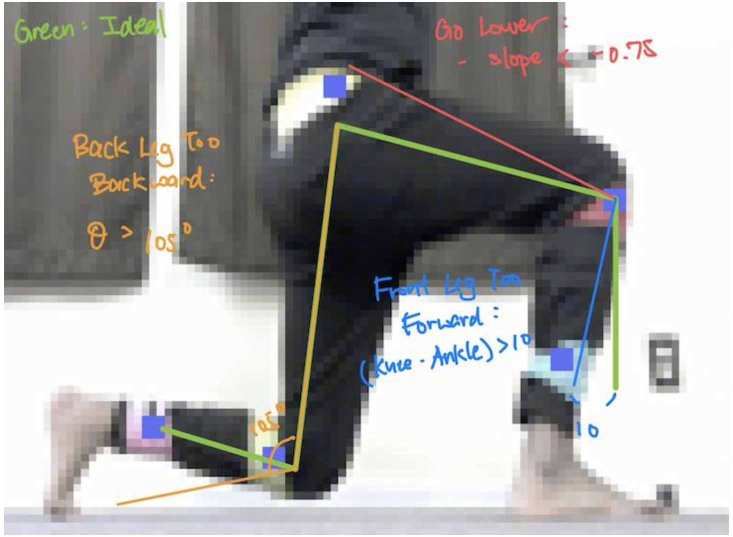
Perfect Rep!



Workout Trends

Complete Solution





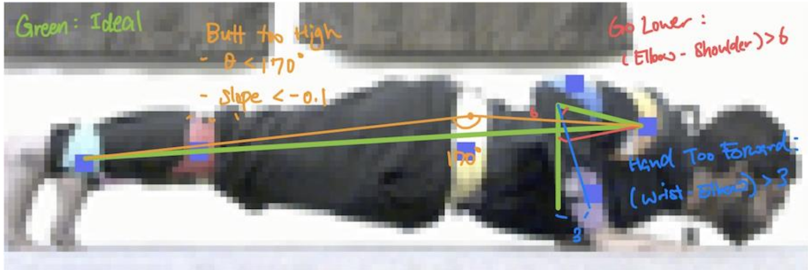


Image Processing Metrics

	Test Inputs	Testbench	Target Output (%)	Actual Output (%)
Downscale + Conversion	4 Different size Images	- HSV Bounds Check to ensure the pixels are all within the legal detectable bounds - Created a Counter for 19,200 pixels and Visual Confirmation for the Quality	100%	4 / 4 = 100%
Tracker Detection	Live User Test: 3 set of 8 reps = 24 reps	- Fed the application 3 * 8 live images - Observed the Posture Analysis's Error Handling or Invalid Joints Detected	21 / 24 = 87.5%	22 / 24 = 91.67%

Hardware Metrics

	Test Inputs	Testbench	Target Output	Actual Output
Data Accuracy	11 sample images	- Generated txt files of the HSV values of each pixel for each of the sample images - Sent the values to the FPGA, which echo'd the values back and compared the logs	100%	100%
Latency	11 sample images	- Started a timer before using Pyserial to send an image to the FPGA - Recorded the average time it took for the FPGA to receive and store the image and send back couple bytes of information	< 1 s	~0.808s

Feedback Metrics

	Test Inputs	Testbench	Target Output	Actual Output
Posture Analysis	Live User Test: 9 Bad Posture Picture (1 per feedback check)	- Manually pinpoint the joints to generate the calculations to ensure the thresholds generates the expected feedback from our designed models.	100%	9 / 9 = 100%
Delay	10 end to end reps	- Started a timer before reading saved image - Recorded the average time it took to: - Pre-Process image - Transfer to FPGA through UART - Image Process and find coordinates - Send coordinates back to CPU - Determine and Display Feedback	< 1.5 s	~1.43 s

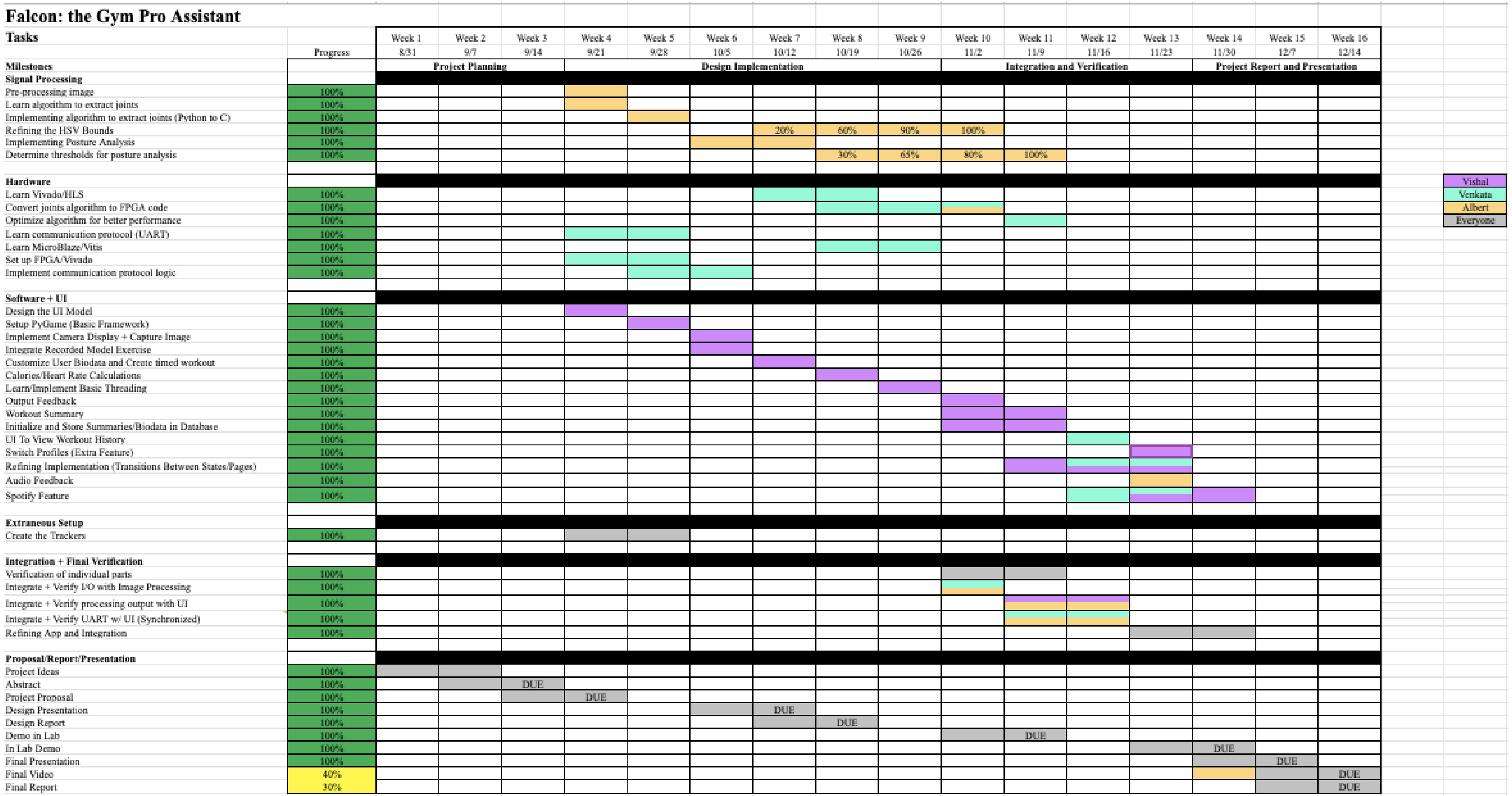
Trade-offs

- Color Tracking vs RFID vs Human Pose Estimation
  - Human Pose hardware translation difficult
  - RFID signal corruption
- HLS vs RTL
  - Handshaking logic with the BRAMs is complicated
  - Feedback requirement of 1.5 s fairly relaxed
- Determining the number of joints to process
  - Was able to meet feedback time by processing only 5 joints
  - Detecting 5 joints provided us with enough feedback regarding the posture

Testing, Verification and Metrics

Requirement	Testing Strategy	Metrics
Downscaling & Conversion	Software testbench (analyze size and quality of resulting image)	100% size match
Detect trackers	Software testbench (analyze trackers over various images)	1 rep to be misclassified every set
Communication between computer and FPGA	Hardware testbench (analyze various packets of data sent)	Latency < 1s & 100% data accuracy
Posture Analysis	Software testbench (analyze various positions to extract info)	100% accuracy according to our designed models
UI (Workout Data + Feedback)	Human Eye (analyze the metrics are met from what is done)	Workout Data: ~tracker accuracy Feedback Delay < 1.5 secs

Updated Schedule



Falcon: the Gym Pro Assistant

Tasks	Progress	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16
	Progress	8/31	9/7	9/14	9/21	9/28	10/5	10/12	10/19	10/26	11/2	11/9	11/16	11/23	11/30	12/7	12/14
Milestones		Project Planning			Design Implementation						Integration and Verification				Project Report and Presentation		
Signal Processing																	
Pre-processing image	100%				🟧												
Learn algorithm to extract joints	100%				🟧												
Implementing algorithm to extract joints (Python to C)	100%					🟧											
Refining the HSV Bounds	100%							20%	60%	90%	100%						
Implementing Posture Analysis	100%						🟧	🟧									
Determine thresholds for posture analysis	100%								30%	65%	80%	100%					
Hardware																	
Learn Vivado/HLS	100%							🟩	🟩								
Convert joints algorithm to FPGA code	100%								🟩	🟩	🟩🟧						
Optimize algorithm for better performance	100%											🟩					
Learn communication protocol (UART)	100%				🟩	🟩											
Learn MicroBlaze/Vitis	100%								🟩	🟩							
Set up FPGA/Vivado	100%				🟩	🟩											
Implement communication protocol logic	100%					🟩	🟩										
Software + UI																	
Design the UI Model	100%				🟪												
Setup PyGame (Basic Framework)	100%					🟪											
Implement Camera Display + Capture Image	100%						🟪										
Integrate Recorded Model Exercise	100%						🟪										
Customize User Biodata and Create timed workout	100%							🟪									
Calories/Heart Rate Calculations	100%								🟪								
Learn/Implement Basic Threading	100%									🟪							
Output Feedback	100%										🟪						
Workout Summary	100%										🟪	🟪					
Initialize and Store Summaries/Biodata in Database	100%										🟪	🟪					
UI To View Workout History	100%												🟩				
Switch Profiles (Extra Feature)	100%													🟪			
Refining Implementation (Transitions Between States/Pages)	100%											🟪	🟩🟪	🟩🟪			
Audio Feedback	100%													🟧			
Spotify Feature	100%												🟩	🟩🟪	🟪		
Extraneous Setup																	
Create the Trackers	100%				▨	▨											
Integration + Final Verification																	
Verification of individual parts	100%										▨	▨					
Integrate + Verify I/O with Image Processing	100%										🟩🟧						
Integrate + Verify processing output with UI	100%											🟪🟧	🟪🟧				
Integrate + Verify UART w/ UI (Synchronized)	100%											🟩🟧	🟩🟧				
Refining App and Integration	100%													▨	▨		
Proposal/Report/Presentation																	
Project Ideas	100%	▨	▨														
Abstract	100%		▨	DUE													
Project Proposal	100%			▨	DUE												
Design Presentation	100%						▨	DUE									
Design Report	100%							▨	DUE								
Demo in Lab	100%										▨	DUE					
In Lab Demo	100%													▨	DUE		
Final Presentation	100%														▨	DUE	
Final Video	40%														🟧	▨	DUE
Final Report	30%															▨	DUE

🟪	Vishal
🟩	Venkata
🟧	Albert
▨	Everyone