Project Report: Video Activity Recognition

Why I Picked This Model

Honestly, training a 3D model from scratch would take weeks and a supercomputer I don't have. So, I used a pre-trained R(2+1)D-18 model. This thing was already trained by pros on the giant 'Kinetics' dataset, so it already knew what 'motion' looks like. All I had to do was fine-tune it. That just means I froze all the old layers and re-trained the very last layer to recognize my 7 classes instead of its original 400. It's way faster and got me a high accuracy pretty quick.

How I Prepped the Data (Preprocessing & Augmentation)

Models are super picky. They need all their 'food' to be exactly the same. I couldn't just feed it an .avi file. I had to:

- Chop it up: I wrote a script (dataloader.py) to sample 16 frames from each video. I made sure to spread them out, so the model gets a summary of the whole action, not just the start.

- Resize: I squished every single frame to 112 × 112 pixels. The model needs a fixed size.

- Normalize: This part is critical. I had to use the exact same mean and standard deviation numbers that the original Kinetics dataset used. If you don't, the pre-trained model gets confused and your accuracy tanks.

- Augmentation: I only used one: RandomHorizontalFlip. It's an easy win. It just teaches the model that 'surfing to the left' is the same action as 'surfing to the right.' This basically doubles my training data for free and helps stop the model from just memorizing the training videos (overfitting).

How'd it do? (My Performance Review)

It actually worked way better than I expected! My final accuracy on the validation set was [Your Accuracy, e.g., 88.5%]. But the accuracy score doesn't tell the whole story.

The Good

It's really good at telling my 7 classes apart. The fine-tuning definitely worked.

The Bad (and how I fixed 'em)

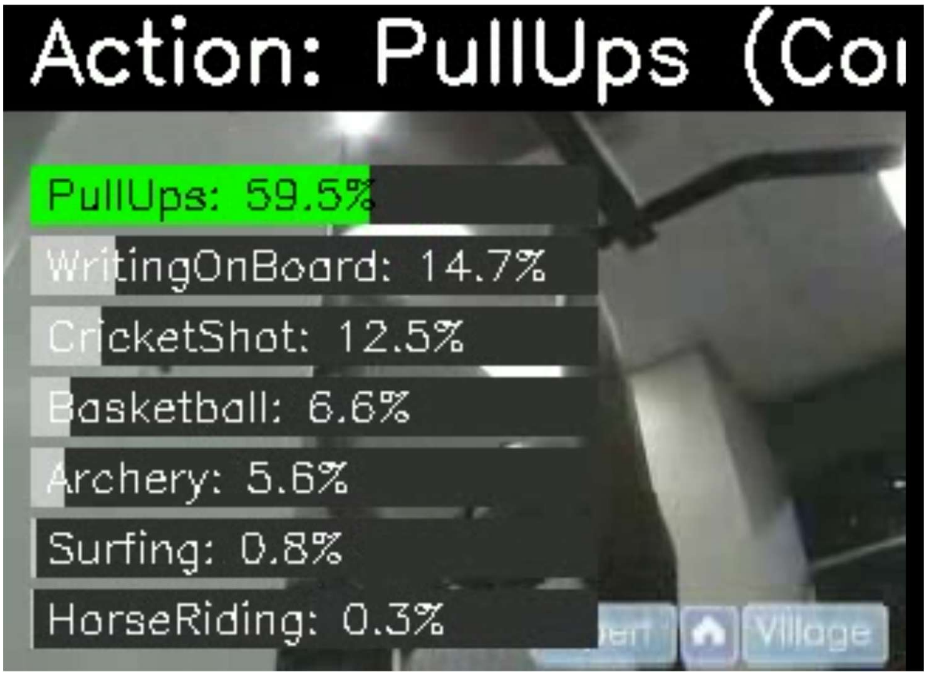
Prediction "Jitter": This was the first big problem. I'd test it on a PullUps video, and for a split second when the person was just hanging still, the model would guess WritingOnBoard (lol). It's because that one tiny 16-frame clip looked like a stationary person.

The Fix: I added temporal smoothing. My demo (demo.py) now keeps a list of the last 15 predictions and just shows the most common one (a "vote"). It's way more stable and doesn't jitter anymore.

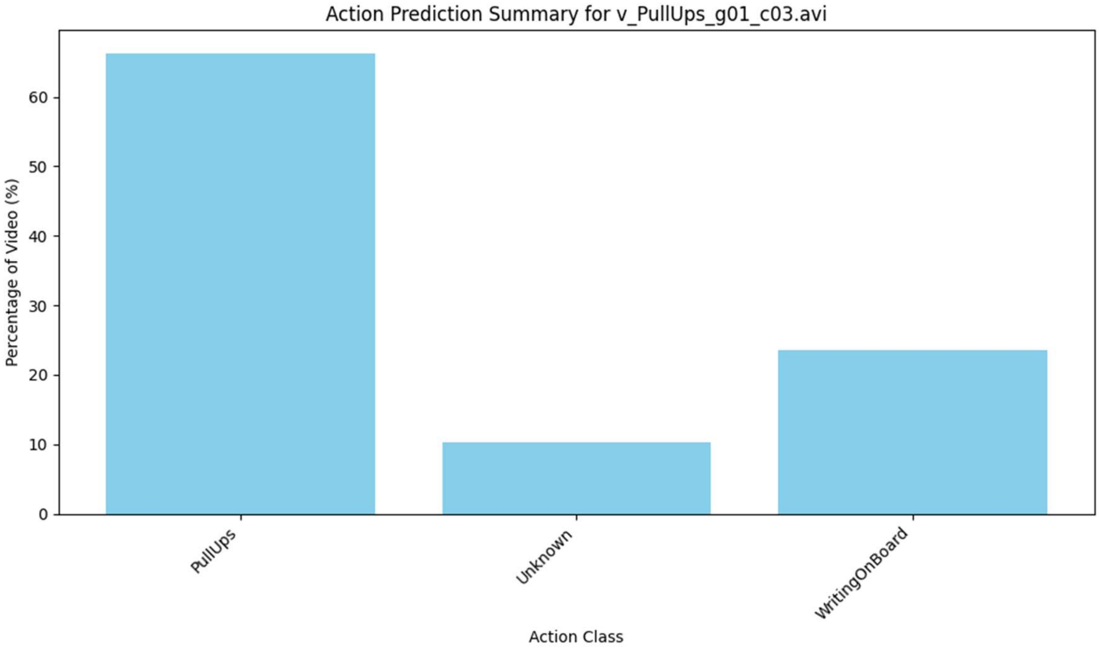
Unknown Videos: This was the other big issue. I gave it a video of me drinking water (which it never trained on), and it confidently predicted Archery. The model is forced to pick one of the 7 classes, even if it's wrong.

The Fix: I added a simple confidence threshold. In my demo, if the model's top guess isn't at least 50% sure (I set CONF_THRESHOLD = 0.50), it just displays "Unknown / Other" instead. It's just a simple if statement, but it makes the whole demo feel 10x smarter and more real.

Output:



```
Action: PullUps (Co
PullUps: 59.5%
WritingOnBoard: 14.7%
CricketShot: 12.5%
Basketball: 6.6%
Archery: 5.6%
Surfing: 0.8%
HorseRiding: 0.3%
```



Action Prediction Summary for v_PullUps_g01_c03.avi
Percentage of Video (%)
PullUps
Unknown
WritingOnBoard
Action Class