This challenge focuses on Super Smash Bros Melee and we're going to use it to test your knowledge on a couple of things. Mainly computer vision.

Here is a very popular clip from Evo 2014 showcasing one of the fastest games of competitive Melee ever.

On the computer vision side, we'd like you to do a couple of things in whatever language you'd like. (Python is cool though).

Required:

- Detection of what stage was picked (Fountain of Dreams).
- Detection of when the game starts (around 0:05).
- Detection of the player names as shown on the screen (GC Silentwolf and MOR RZR AXE).
- Detection of characters being played by both (Green Fox and Party Hat Pikachu).
- Detection of how many stocks each player has (*for every 10th frame)*.
- Detection of how much percent each player is at (*for every 10th frame)*.
- Detection of in-game time as shown by the timer at the top of the screen (*for every 10th frame)*.

Now, there are MANY ways to do all these things. It's up to you to choose the ways that will work the best in terms of accuracy and consistency. For example, if you want to detect when the game starts you could detect the big "GO!" sprite that pops up or you can detect when both players are at 4 stock and at 0 percent or you can train a neural net to detect certain features to see if a game just started. There are multiple ways to do every detection and we're curious to see the decisions you make.

Here's where this actually gets hard. At run time you know exactly where everything is in the Evo 2014 clip, things like the percents, stock icons, etc. But, we also want you to think about how you can create more *general algorithms*. For example, if we give you this video, will the code you wrote for the Evo 2014 clip work for this clip? Will your code still be able to detect things like characters, percents, and stage name despite everything being different? What about this one?

Creating CV algorithms that can generalize to different screen sizes, layouts, and resolutions is going to be a key issue you work on here if you join and we're trying to see if you're comfortable with the domain! So, if your program can only do detection properly on the Evo 2014 clip, it won't wow us as much for sure.

I recommend you first create the CV to work just for the Evo 2014 clip, and go from there!

Super extra-credit. *(If you do any of these, it'll seriously impress the team! But make sure to do what is required before hand.):*

- Using object detection, automatically draw bounding boxes around Pikachu and Fox. Kinda like this! Note: You are not allowed to use the Evo 2014 clip given to you as training data, you must use other clips!

Other note:

- Honestly, the **less** libraries,third-party projects, and APIs you use the better. We're cool with stuff like Tensorflow, OpenCV, and numpy. But if you just take someones pretrained OCR model and use that to do percent detection, well, it doesn't really show us if you can figure it out on your own!
- Clean, concise, commented code is best! If everything works, but the code is disorganized, we will be sad.
- Feel free to message us with questions at anytime while you work on this. Never hesitate when asking questions. Text Evan with any questions immediately @ 626-241-8919 or on Discord at evanfrawley#1220
- It might cost you some $ to train some deep learning algorithms! Assume your budget is 50$ for now. We can give you more as needed. A lot of this stuff can be done without deep learning though :).

GOOD LUCK!