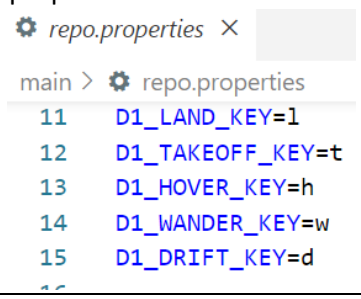
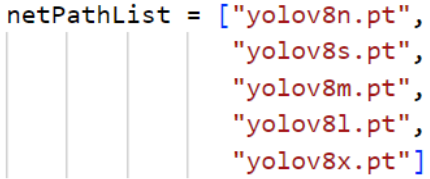
Requirements & Evidence:

User Challenges	System Requirements	Current Status
The user is unable to find and edit source code to monitor different aspects of the drone during flight.	Adjustable settings for telemetry	**Not met*** The adjustable telemetry settings could not be implemented through the GUI due to time constraints, but they have been made adjustable in the code with instructions provided to the user.
	Settings are stored for future iterations.	**Met** Device IDs are stored in a ".json" file, and other settings are stored in a ".properties" file.
	Adjustable key mappings for commands	**Met** Key binding can be changed in the properties file. 
	Vision models are read automatically.	**Met** The YOLOv8 CV model downloads automatically, shown in proof videos. Video title: cv_models_downloading.mp4
	Vision models are selectable through GUI.	**Not met**** Due to time constraints and low priority, the option to select different models through the GUI was not implemented, however, the user can adjust its size in the system according to these options: 
The user is unable to monitor telemetry from multiple drones.	Telemetry from multiple drones is displayed.	**Partially Met** Both drone camera feeds can be displayed, just external to the GUI page.
The user is unable to control multiple drones independently.	Drone connection can be managed in GUI.	**Met** The GUI "connectivity page" allows the user to easily input IP addresses. Video title: connectivitypage.py

	Send individual commands to each drone.	**Met**
		`def operator_override(self):`
		`  if(key.is_pressed('1')):`
		`  self.override_drone(self.drone1)`
		`  if(key.is_pressed('2')):`
		`  self.override_drone(self.drone2)`
		Users can toggle control of drones based on pressing keys 1 or 2 if flying multiple. Visible evidence can be seen in video examples.

The user must initially set up a control loop, wasting time.	The general control loop implements an extensible Finite State Machine and automatically takes care of drone telemetry and command overrides.	**Met** Evidence is found in the control logic shown under "Finite State Machine in System Components in the ReadMe.
Current safety features hardcoded into the Tello are not preventative.	Provide a base with built-in safety controls.	**Met** The drone must go through multiple safety checks before the flight
	Users can interrupt at any point for all drones simultaneously	**Met** Override is available through the keyboard
Users spend too much time looking for scattered code examples.	Read-Me contains references to other projects with descriptions.	**Met** We include references to any other repository we borrowed from in our ReadMe.
	Read-Me contains examples/suggestions to implement with our current repository.	**Met** There are examples of how to create a new reaction and new state in our ReadMe.

Requirements not met:

***Adjustable Telemetry**

We were not able to implement adjustable telemetry settings through the GUI due to time and priority constraints. Currently, the user must manually go through the code to adjust. However, we have made them adjustable and easily found in the .properties file.

****Selectable Vision Models**

Again, we were not able to implement this through the GUI due to time constraints. The user can adjust the size of the model through the .properties file.