UNIVERSITY OF ARKANSAS

Automatic Drone Tracking

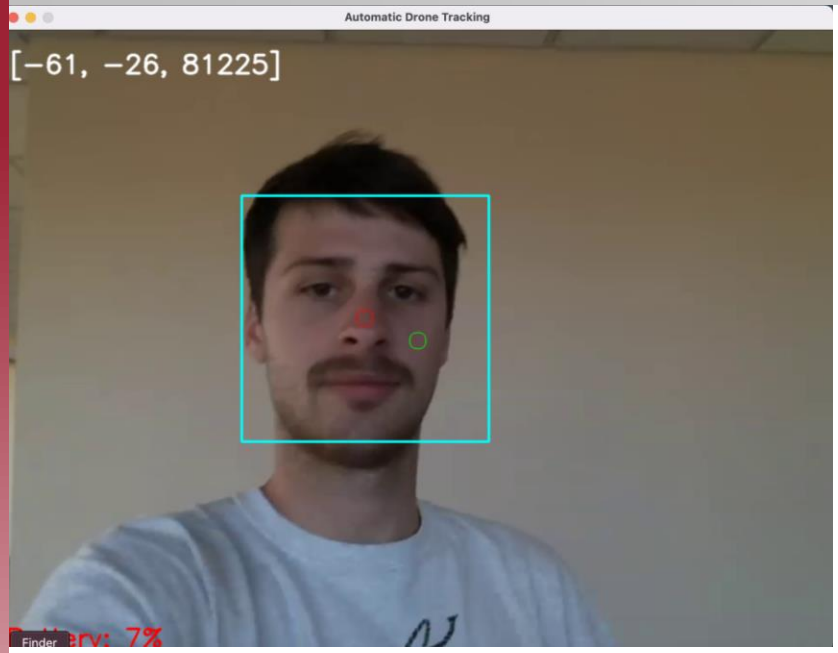
By: Byron Denham, Parker Weber, Zachery Gansz, Andre Fuentes, Corbett Stephens

Champion Advisor: Dr. Khoa Luu

Purpose

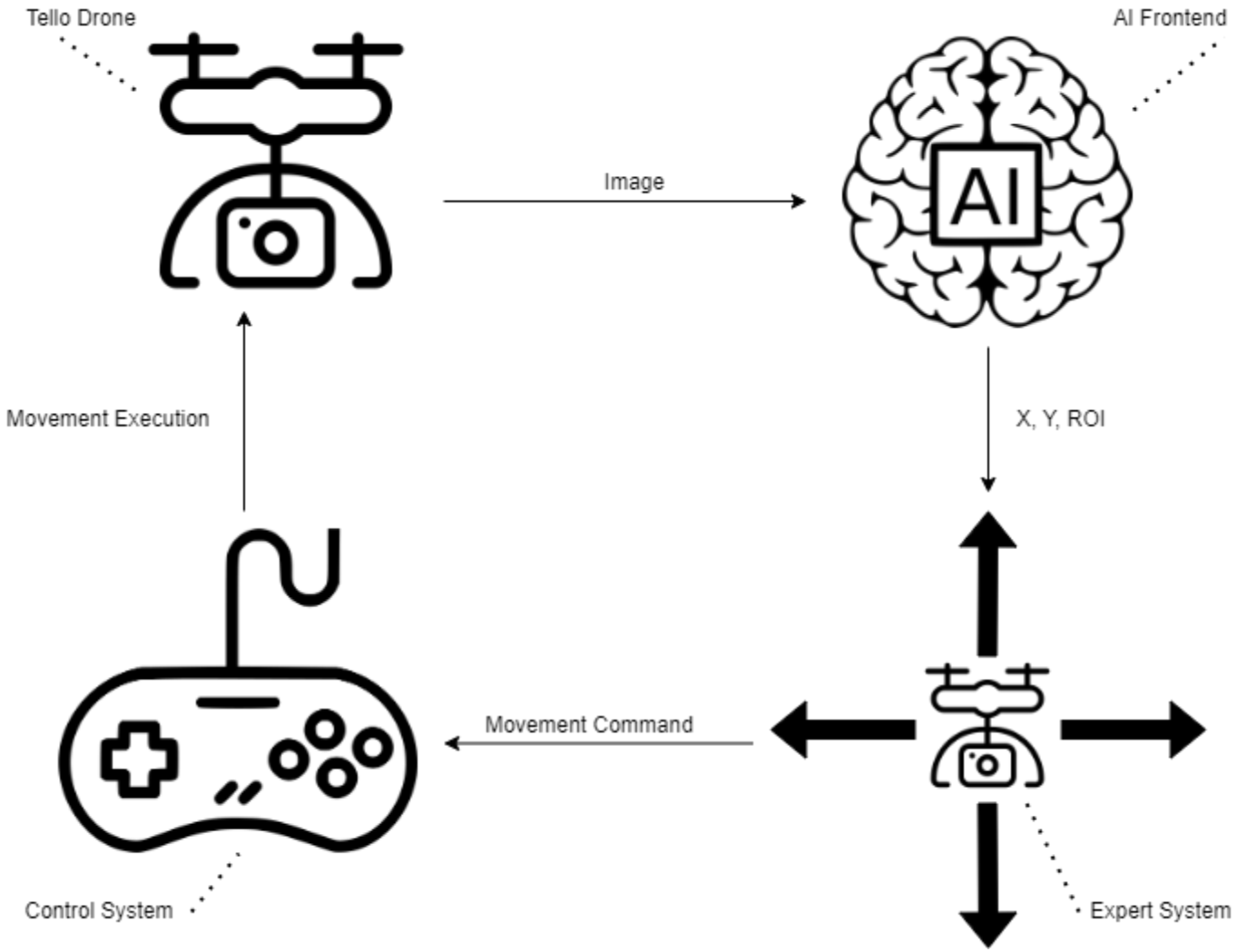
- Track a humans face autonomously
- Some use are videography and surveillance.
- Open Source
- Low cost

Goal

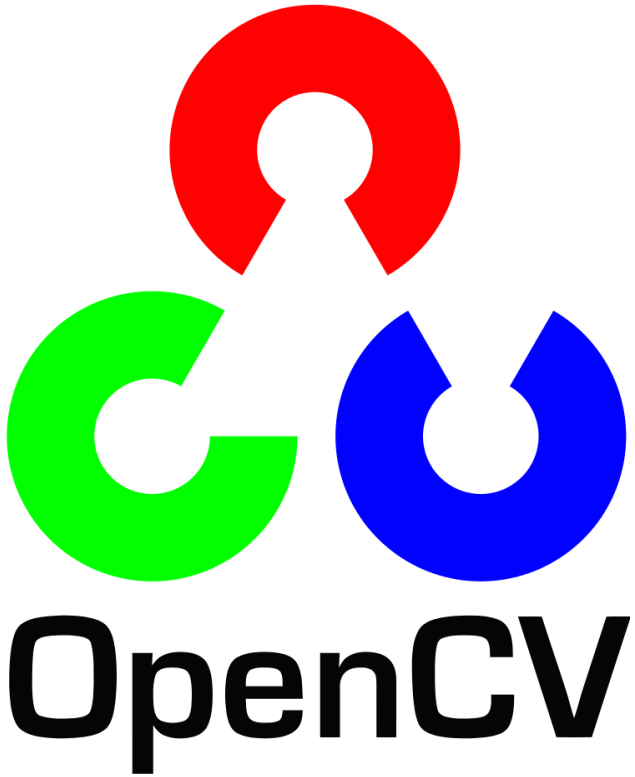


- Establish reliable communication between the drone and the backend
- Optimize expert system for quick decision making
- Real-time facial detection

Design Overview



AI Backend

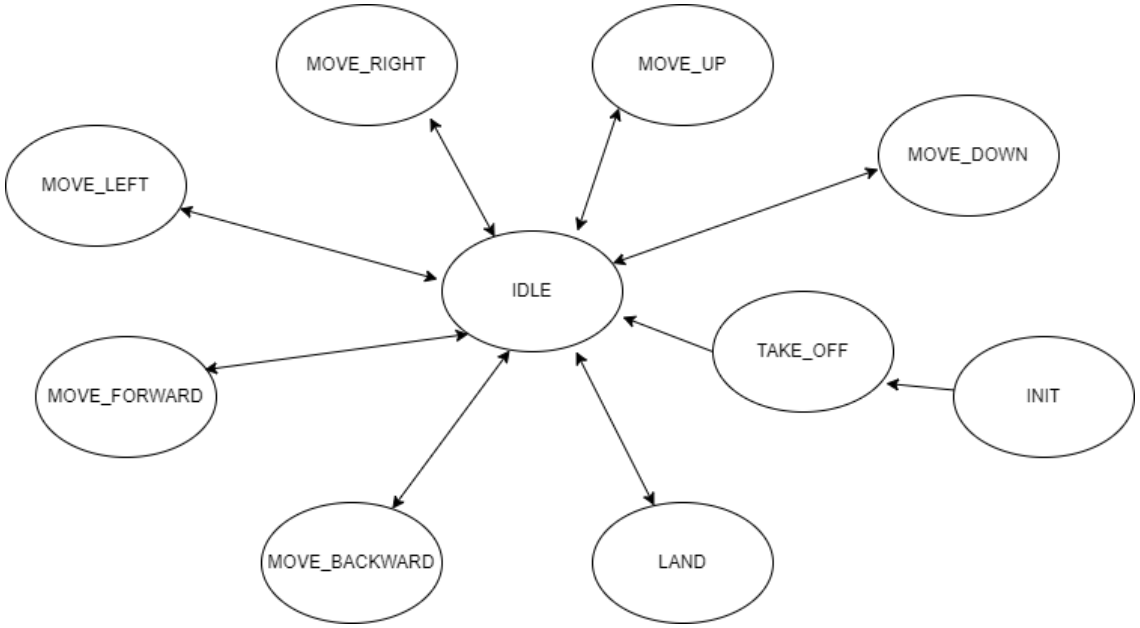


- Uses OpenCV: a programming library designed for real-time computer vision
- Creates a classifier for facial detection with OpenCV's CascasdeClassifier method and an xml file of pretrained weights
- Classifier takes a frame of video from drone's camera, attempts to identify all faces in frame, and returns list of faces
- The classifier identifies faces represented as tuples of (x, y, w, h)
- List is passed to a function that removes all faces from list other than the one with largest ROI, in essence allowing the drone to lock onto the closest face
- The values of the remaining face are then passed to the expert system and used to place indicators around the face and print pertinent info to the video stream

Controller

- Tello-Face-Recognition
- Djitellopy
- OpenCV problems
- PyGame
- Integrated FSM code

Expert System



- Finite-state machine model via Python
- Receives X, Y, and ROI from AI program.
- Makes decisions based on position relative to camera
- Sends commands to controller

UNIVERSITY OF ARKANSAS

Cython

- Benefits of C
- Benefits of Python
- Integration of Cython
  - Bridging
  - Cythonizer

Future Work

- Addition of Cython
- More FSM states to account for all possible scenarios
  - Multiple angles/faces
  - Advanced repositioning
- Addition of pre-trained weights

UNIVERSITY OF ARKANSAS

Demo

Thank you

- Sponsor: Dr. Khoa Luu
- TA's: Thanh-Dat Truong, Xuan Bac Nguyen, Pha Nguyen, Naga Venkata Sai Raviteja Chappa