Motivation: Traditional Pet Cameras

- Pet cameras are increasingly popular as a way to ease pet owners' minds when not at home
- However, owners must actively watch (or review) the feed in order to catch bad behaviors or get a sense of their pets' activity, which can be inconvenient does

Introducing PetSTAR (System to Track And Report)

- With a combination of software and signal processing, we aim to creating a pet monitoring system with added convenience:
  - Detect and report bad behaviors as they happen
  - Differentiate between multiple pets in the home
  - Provide a concise log of the pet's activity through the day

Requirement: vision algorithms should be *fast* and *accurate*

- Identify when an animal has entered the frame within 5 seconds
- Pinpoint the animal's location within 1 foot of its actual location
- Detect if the animal has entered a forbidden zone within 1 second
- Classify between animals with 90% accuracy
- Determine which animal it is within 5 seconds of it being in full view

Requirement: Web application and overall system should be *intuitive* and *user friendly*

- Notification from the camera to the user should take less than 10 seconds
- The overall system should take no more than 5 minutes to set up
- At least 95% of users should be able to complete core tasks on the website with little or no instruction
- Granularity of the forbidden zone should be sufficiently fine for at least 95% of use cases
- The overall system should cost no more than $100

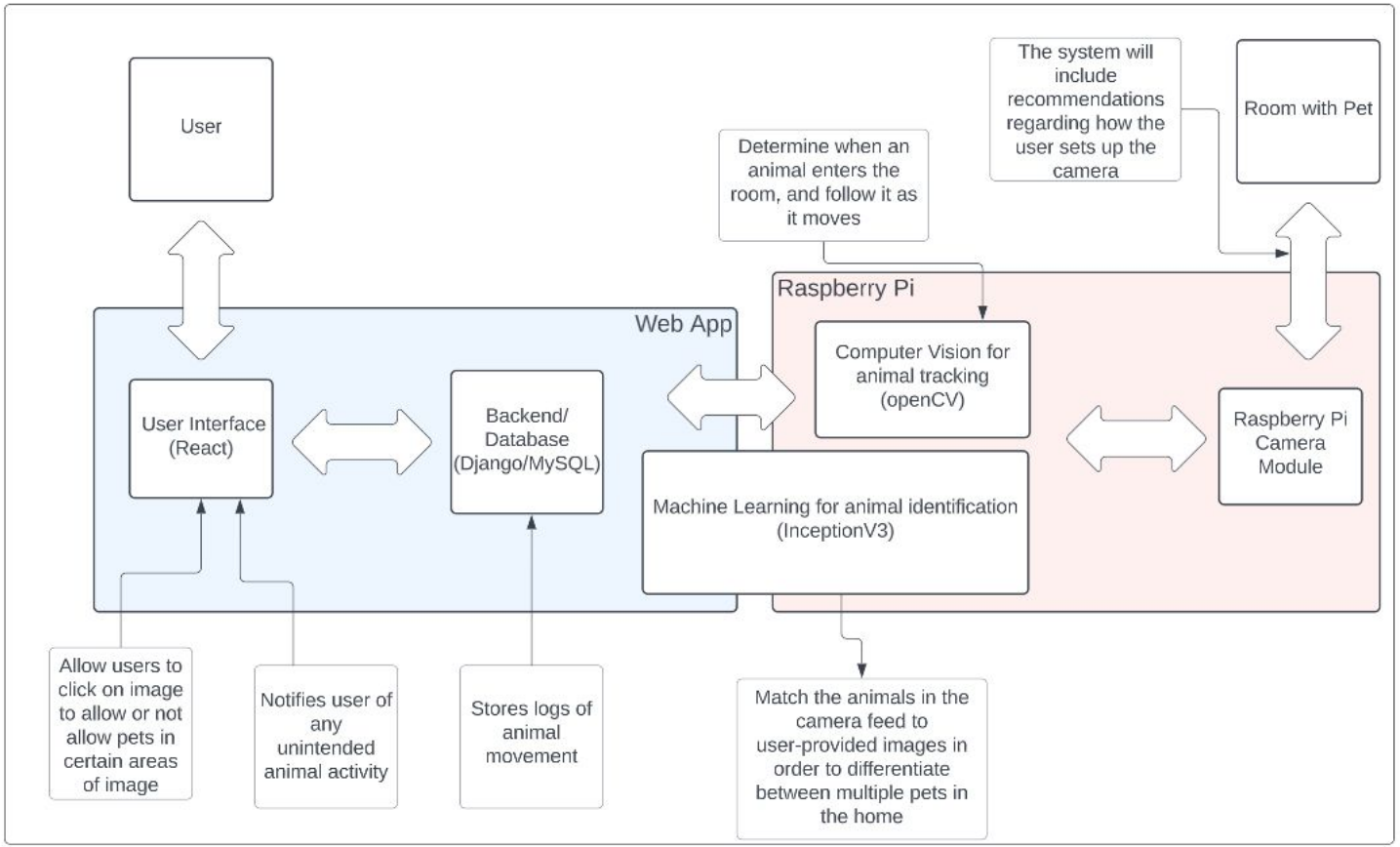
Technical Challenges

- What if we need to differentiate between two animals which look incredibly similar?
- What if a forbidden zone is at the edge of the frame?
- What hardware will it take to meet ML accuracy and timing requirements?
- What data structures should be used to store forbidden area and pet activity data in the database?
- How to tell the web application from the Raspberry Pi to send a notification to the user that the pet has entered a forbidden zone?

Solution Approach

- Hardware - Raspberry Pi 4 Model B
  - Camera Module - supply video feed
  - OpenCV Library - locate and track the animal
- Machine Learning - Convolutional Neural Network
  - InceptionV3 - base ML architecture
  - Cats & Dogs Breed Classification Datasets - initial training data
  - User supplied images - supplemental training data from web application
- Software - Web application
  - React - frontend
  - Django - backend
  - MySQL - database

Solution Approach, Continued



Testing and Verification - CV and ML

- Phase 1: Printed images
  - Can we track an animal as it moves?
  - Can we differentiate between different images?
  - Do the necessary notifications send correctly?
  - How does the activity report look?
- Phase 2: Stuffed animals
  - Does tracking still work if the animal changes direction/profile?
  - Can we still differentiate between multiple plushies at different angles?
- Phase 3: Live animal testing, if possible (Rebecca's apartment, 1 cat)
  - How does it perform under real world conditions? (single pet household)

User Testing - Web App

- Phase 1: Volunteer users interact with website
  - Are they able to accomplish simple tasks?
  - Is anything confusing?
- Phase 2: Volunteer full-system set up
  - Can they successfully set it up?
  - How long did it take?

Tasks and Division of Labor

Brandon (Web App)	Rebecca (Computer Vision)	Max (Machine Learning)
• Create Login using Google OAuth • Display live video feed • User can select sections of the camera view that are forbidden to pets • Notifies user of any disallowed activity from the pet • Gives user animal activity on request • Store logs of animal activity in database	• Set-up OpenCV on the Raspberry Pi • Detect when an animal has entered the frame • Track its movements through the environment • Flag if it enters a forbidden area	• Adapt current CNN architecture for animal classification • Integrate user supplied images with existing datasets for improved accuracy • Validate flags prior to notification • Validate animal activity logs

Schedule

							PHASE ONE														PHASE TWO										
WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE	WEEK 1					WEEK 2					WEEK 3					WEEK 4					WEEK 5					
							M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M
1	Test User can define forbidden area on image	All																													
1.1	User can upload image	Brandon	2/6/23	2/12/23	6	0%																									
1.2	User can click parts of image that are forbidden	Brandon	2/13/23	2/19/23	6	0%																									
1.3	Web app stores in database what parts of image are forbidden	Brandon	2/13/23	2/19/23	6	0%																									
2	Validation of Flags	All																													
2.1	Animal Classification	Max	2/6/23	2/12/23	6	0%																									
2.2	Flagging operation	Max	2/13/23	2/19/23	6	0%																									
2.3	Forbidden zone setup	Max	2/13/23	2/19/23	6	0%																									
3	Animal Classification	All																													
3.1	Operational CNN	Max	2/6/23	2/12/23	6	0%																									
3.2	Trained CNN	Max	2/13/23	2/19/23	6	0%																									
4	Forbidden zone setup by user	All																													
4.1	User can upload image	Brandon	2/6/23	2/12/23	6	0%																									
4.2	User can click parts of image that are forbidden	Brandon	2/13/23	2/19/23	6	0%																									
4.3	Web app stores in database what parts of image are forbidden	Brandon	2/13/23	2/19/23	6	0%																									
5	Trained CNN	All																													
5.1	Get trained data	Max	2/6/23	2/12/23	6	0%																									

In Summary

- We aim to create a pet monitoring system which identifies each animal in frame, tracks its position, and communicates to the user via a web app
- Key challenges/requirements will be the accuracy and speed of the detection, and communicating with the user in a straightforward and intuitive way
- We plan to test our system with images and stuffed animals, before hopefully graduating to a live animal