NVIDIA Jetson AI Certification Project

Outdoor Cat-A-Logger

Deep Learning on an NVIDIA Jetson Nano Keeps Track of Outdoor Cats Health, Movements and Reproductive Status

Purpose

Outdoor cats are present nearly everywhere in the world. Some regions allow them to roam freely, some want them vaccinated, and others want them to be spayed or neutered. Outdoor cats are notoriously difficult to track and manage without specialized equipment, personnel, training, and procedures.

This project proposes a solution using Deep Learning to keep track of outdoor cats. It can distinguish cats from other objects and ultimately allow the user to track metadata on individual cats.

Metadata can include but is not limited to: The coming and goings of a particular cat over time, their vaccination, health, and reproductive status.

The Problem

Deep Learning can be used to distinguish cats from other critters, objects, and noise. A model can also be trained to recognize individual cats, but **there are no publicly available models of cats taken with a night vision camera nor are their publicly available images with which to train such a model.**

Images need to be generated, found, collected, or otherwise captured in order to train the model.

The initial model was bootstrapped with a highly augmented dataset and built up over time using real captured video to retrain the model. The Jupyter Notebooks and Datasets are publicly available on Kaggle.

Today’s State of the Art Motion Detection

The initial thought was to gather images using the Surveillance Cameras built-in motion detection to collect images of cats during the day and at night.

Retail Motion Detection Cameras are virtually useless for this because they tend to trigger on almost anything.

They are getting better at detecting people which makes them more useful, but that does not help with our cat imaging problem.

Another solution was needed.

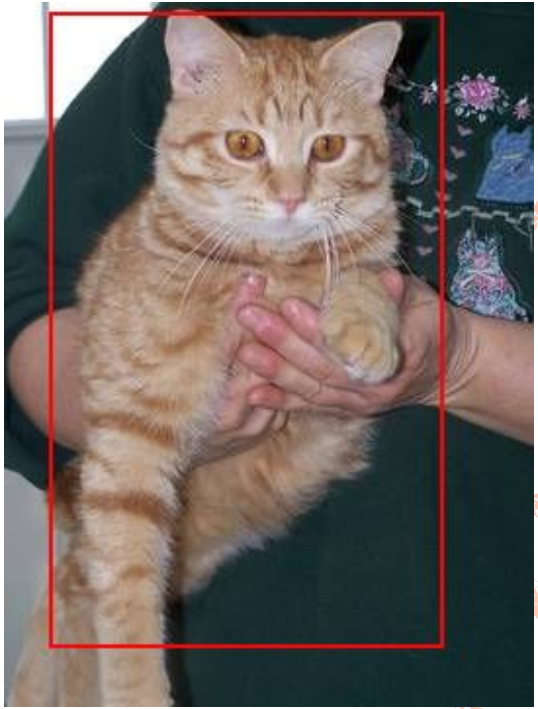
The Solution - Synthetic Data

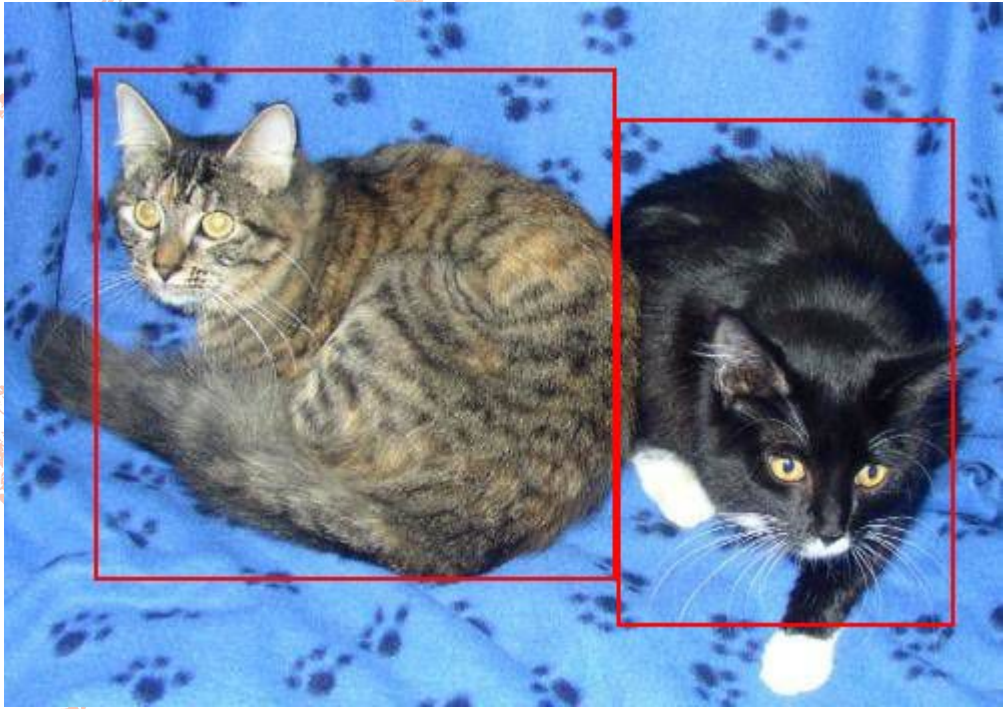
Build a model from publicly available images of cats and attempt to simulate images taken by a Night Vision Camera hoping to “trick” the initial model into recognizing images of cats at night. The original dataset was created as follows:

- Crop publicly available daylight images extracting only cats
- Convert the images to black and white
- Invert the colors to make them negative images
- Enhance the images by randomly sizing, skewing and adding noise

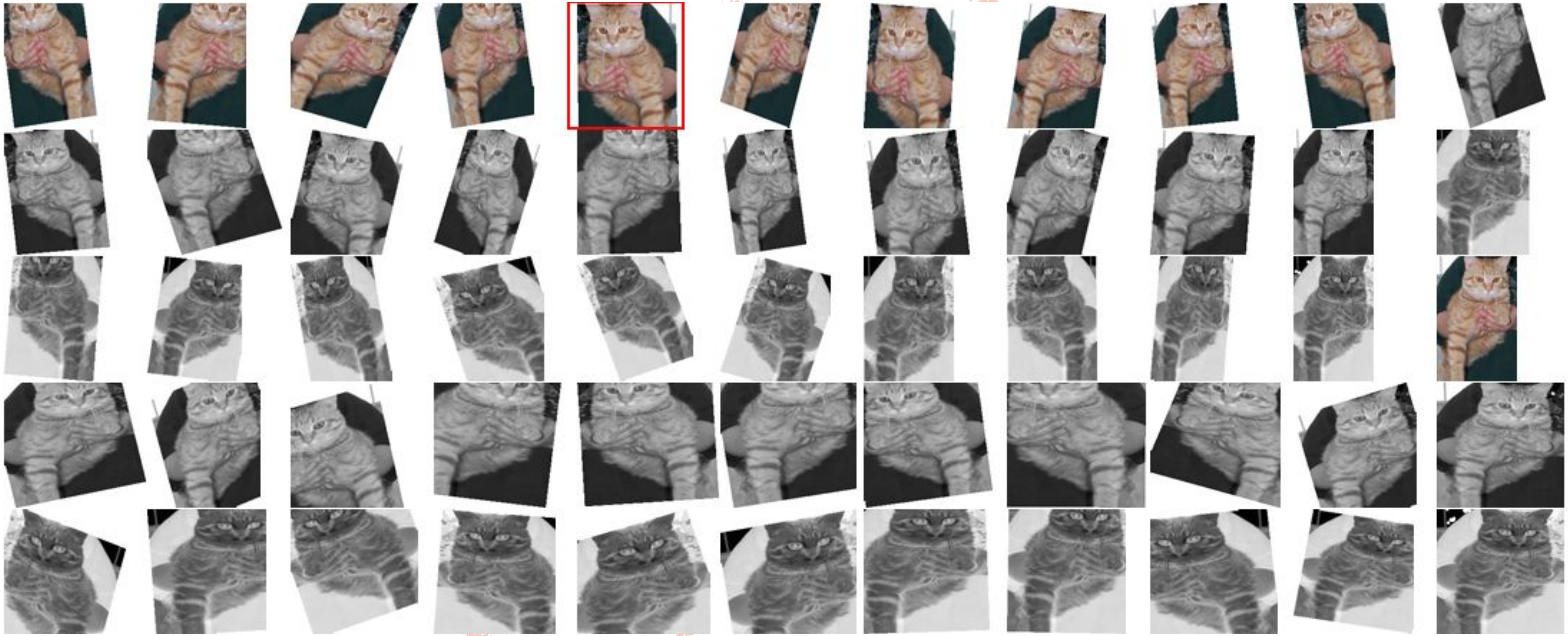
To make a larger more robust dataset, the entire set of source, intermediate, and final images were randomly paired with random backgrounds which were randomly augmented. The program also created metadata such as bounding boxes and labels to inform the model where the cats are in the composited images. The final images were used to train the model. This algorithm currently produces 55 unique output images per input image.

Extract Cats from Dataset Using YOLOv3

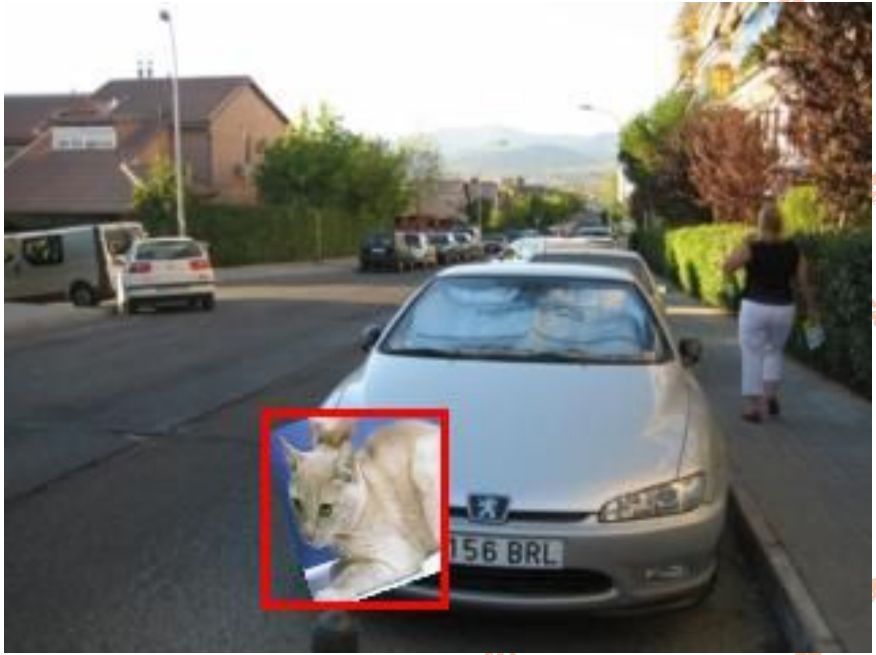


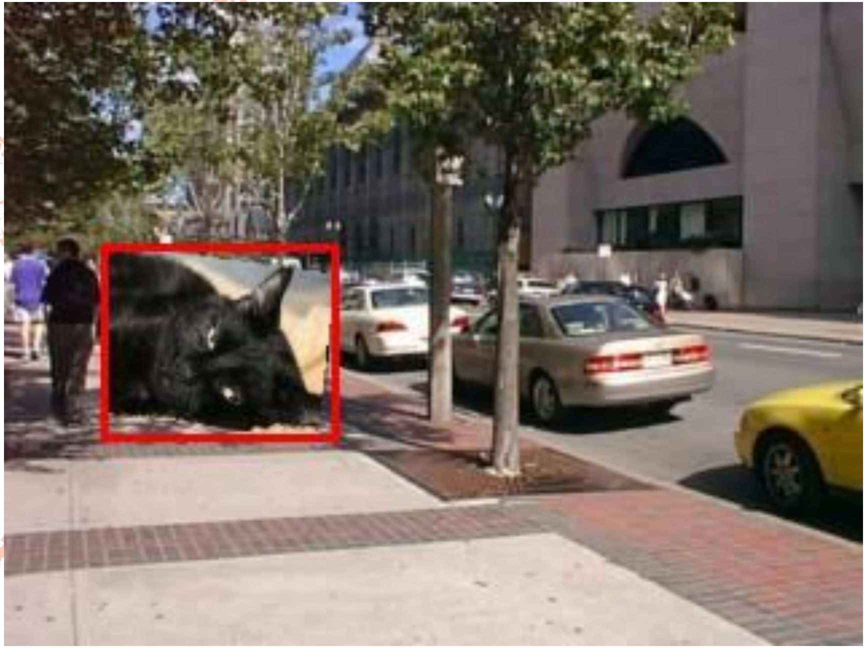


Augment Images

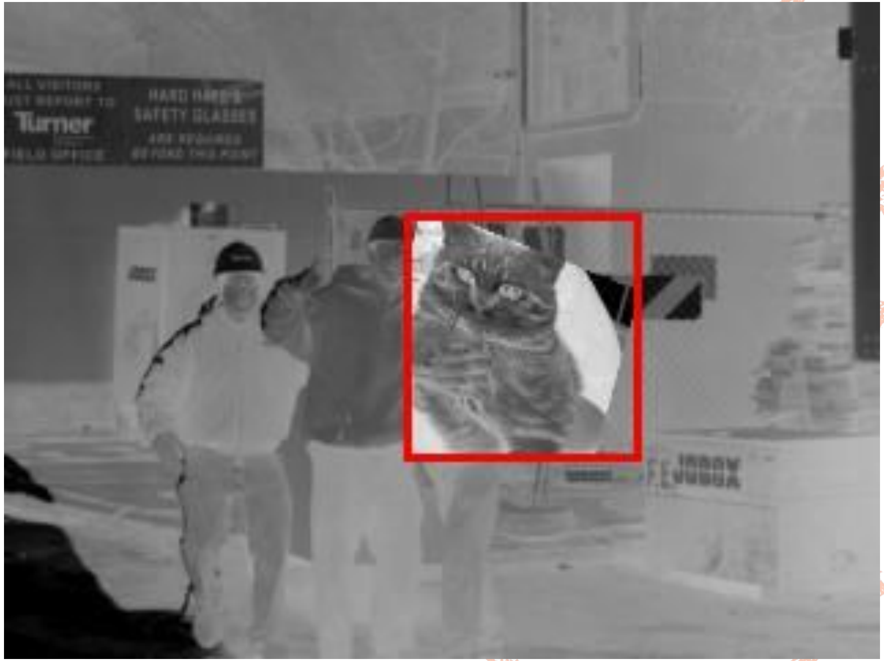


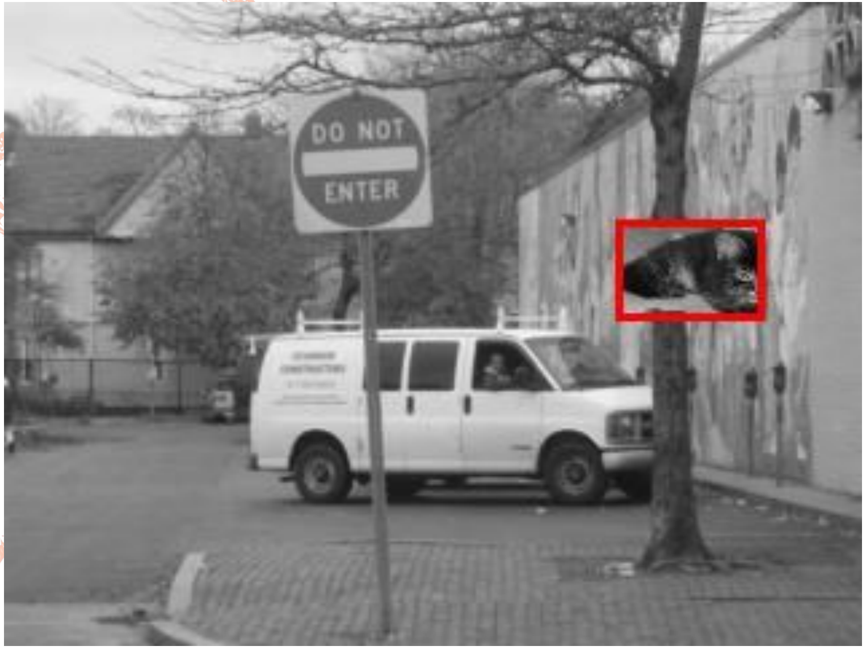
Composite Images and Emit Bounding Boxes





Negate and Grayscale Some of Them





The Cat Box

To capture images of cats, a unit was built to house a night vision capable surveillance camera and a Wifi-based monitoring system. The cats were baited in order to get them to come close to the Cat Box for imaging.

The unit is portable, battery powered, wireless, and can be moved to various locations as needed. Currently, the system requires Wifi but could support embedded cellular or satellite communications. The initial unit is rather large and power hungry but could be significantly miniaturized and made to be completely solar powered for remote operation.

To reduce bandwidth for cellular or satellite communications and AI Inference Device could be placed inside of the Cat Box so it would only transmit pertinent data.

Server & Inference Device

The NVIDIA Jetson Nano was chosen as the Server and Inference Device because of its relatively small size and low-cost to performance ratio

- This procedure can be refined significantly, but for this demonstration, the Nano receives video full-time as an RTSP Stream but does not process it until triggered over Wifi by the ESP32 connected PIR Motion Detector in the Cat Box
- The ESP32 and its Sensors use an RTOS to significantly reduce power consumption by Parallelization via Threading for use in a Low-SWaP Environment.
- When the Server is triggered, it begins recording and running inferences to determine whether the heat signature detected by the PIR belongs to a cat. Recorded video and detections are manually reviewed for accuracy and the results fed back into the model for further training
- In addition, each hour the ESP32 in the Cat Box sends battery voltage and vital statistics to the server. This acts as a Ping to inform the server the Cat Box is alive and well
- Communication from the ESP32 is performed over Wifi using a rudimentary REST API on the Jetson