Stackademic · Follow publication

How A Stray Cat Inspired Me To Build A Cat Detection System

10 min read · 2 days ago

Partha Roy Follow

Listen Share ••• More

For the last month, a certain uninvited (yet deeply loved) guest has been visiting my doorstep every day, a stray cat with the confidence of royalty and the timing of a mystery thriller. Whenever we meet, he stares straight into my soul and meows as if to say,

> “Where art thou, my servant? Have you forgotten your sacred duty of feeding me?”

The Master

Do I even need to say it? I love cats. I've always loved cats.

So, as a loyal subject to my feline overlord, I made sure to offer food whenever possible. Pika, my housecat, is a ginger boy with one active brain cell and an attitude fit for a CEO. He always refuses to eat leftovers that are older than an hour.

Naturally, I started keeping the untouched food outside for the tabby gentleman. He would sweep the plate clean every time.

But there was one small problem. He never had a fixed time to arrive, and I never had a fixed schedule to spot him.

How a Hungry Cat Hijacked My Engineer Brain

Here's how it went. I'd be in a meeting or deep into debugging some weird async function, and meanwhile, this cat would be quietly sitting outside waiting for dinner service. If I happened to open the door, great; otherwise, he'd leave.

He looks & behaves somewhat like the one from the movie Flow

Now, Pika is an indoor cat, pampered and protected, and we couldn’t risk him catching an infection from our street visitor. So letting the stray inside was out of the question. But I also couldn’t stand the thought of him waiting outside hungry.

And that’s when my software engineer brain kicked in.

The Problem Statement

I had a Raspberry Pi 3B+ sitting around. It was gathering dust, silently judging me for buying it “for a project someday.” Well, someday had arrived.

My Raspberry Pi 3B+

> The question was simple:
> “How can I know when the cat arrives at my doorstep without being physically there?”

That became the foundation of this slightly ridiculous yet deeply satisfying project, **The Cat Detection System.**

The goal was to:

- Detect when the cat appeared outside.
- Notify me instantly so I can feed him.
- Eventually, collect enough data to understand his routine and predict his arrival times.

The High-Level Design

Here’s the design I created for the initial architecture, Sounds like overkill? Probably. But tell that to someone who's ever been meowed at like they've failed their cosmic purpose.

I started by attaching a small camera to my Raspberry Pi. I decided against video streaming because, well, the poor Pi has the processing power of a sleepy cat after a full-course afternoon meal. So I configured it to take still images at regular intervals.

Each captured image was then sent through a lightweight object detection model — specifically tuned for cats. When a cat was detected, the system would trigger an alert through Telegram (because, of course, I wanted to get a ping that says "Cat detected!") with sound and notification (without having to go through the hassle of setting up a whole client-side app)

And for testing, my QA engineer was obviously, Pika. He jumped in front of the camera multiple times, posing like he was auditioning for the cover of Cat Vogue.

Pi and Pika. (Ignore my messed up table)

Once I confirmed the model was accurate enough, I set up the device near the doorstep inside the dry food container hidden. To keep it wireless, I connected it to a power bank, making the setup portable and minimal.

I have wrapped the whole setup in a paper and kept it inside the food container (seperating the food from it) for testing purpose temporarily, will be moving it to a dedicated top bracket. kept it hidden as it’s a common hallway

The system quietly clicked pictures every few seconds, processed them locally, and notified me whenever it spotted my furry friend outside.

Finally, after keeping it on for some hour, I got a ping on Telegram,

The camera didn't have a proper vertical mount, so I had to install it upside down. As a result, every photo comes out inverted — but thankfully, our furry friend still gets spotted without a hitch. (later i flipped the image programmatically)

And finally, for what this whole system was built, as I got the notification, I was full of joy and went outside hurriedly, seeing our tabby boy sitting on the cushion on the shoerack, then I fed him.

Here are some glimpses of the art and the artist together at play,

The Fur Kiddo enjoying his feast!!

What I Learned (and Laughed At)

Now, before anyone gets too impressed, let's be clear, this was far from a production-grade system. It was a homemade DIY setup built for one purpose — ***to detect my stray cat and make sure he didn't go hungry.***

But here's the thing. The learning I got from this small weekend project was massive.

Lessons From My Cat Detection System

1. **Necessity drives creativity.** I didn't plan to build a pet surveillance system, *but hunger (and guilt) are powerful motivators.*
2. **Keep it simple.** Sometimes, taking pictures at intervals beats running a heavy meesage queues with LLMs in real time. *You don't need a gun to kill a mosquito.*
3. **Test with patience.** Having Pika walk in front of the camera repeatedly was an experience in itself. He was more interested in the lens than the logic.
4. **Embrace imperfect setups.** I didn't have the best camera or the perfect power source, but the system worked well enough.
5. **Data tells a story.** After a week, I could roughly predict when the stray cat would show up. His schedule was more consistent than some of my developer stand-ups in some past companies.

The complete system at play

A Bit About the Tools

Here's what I used to build this mini engineering-meets-cat-care system:

- Raspberry Pi 3B+ — the tiny brain behind the setup, I used this one, but would recommend anyone who is buying anew to go for a newer version device with a good amount of RAM (>4GB), e.g. Raspberry Pi 4

- **Pi Camera Module** — for capturing the still images, I wanted to say one thing about this particular camera module. If you want a value for money pi camera within budget in India, don't look anywhere else, I settled down for this after a good amount of research, trust me, this Raspberry Pi Camera board model is the best one within 350rs only.
- **SuperFast Flash Storage** — for storing the OS and temp data.
- **Memory Card Reader** — for flashing the Pi from MacBook
- **Power Bank** — because outdoor sockets aren't exactly cat-safe
- **YOLO Object Detection Model** — lightweight pre-trained model for cats
- **Telegram API** — to send real-time notifications to my phone

And that's it — a simple system that made both my life and the cat's life a little easier.

Here's the setup & code

First, I had to set up an isolated Python environment,

```
Run these commands in order:

# Make sure venv module is installed
sudo apt install python3-venv -y
```

```
# Create a virtual environment (any folder name is fine)
python3 -m venv ~/catenv

# Activate the venv
source ~/catenv/bin/activate

# Now you can safely install Python packages
pip install requests schedule


# now our shell prompt changes to:

(catenv) username@pie_host:~ $


# Now we're inside an isolated environment where pip install works normally.
```

Had to get this done right off the bat, as from experience I have seen sometimes global env creates problems such as **PEP 668** behaviour.

Now we write our code,

Capturing an Image

First, attach the camera module and reboot, and then after sshing to the pi on the terminal do the following,

We can make sure our system is up to date: `sudo apt update && sudo apt full-upgrade -y`

Then we install **rpicam**

```
sudo apt update
sudo apt install rpicam-apps -y
```

This will give us:

- rpicam-hello
- rpicam-still

- rpicam-vid
- rpicam-jpeg

To check if it’s working properly,

```
rpicam-still -o test.jpg
```

log like this will come and a image will be captured

Now, if you are using ssh for accessing the Pi, then to transfer it to a different device with a display, we can run the command below in our system in a separate tab, and it will ask for the Pi device password.

```
scp roy@MY_IP:/home/roy/test.jpg ~/Desktop/

# Note - To get current directory, pwd
# to get raspberry's IP, hostname -I
```

Check if the image is alright, and arrange the angle and that's all, now let’s write our scripts photo capturing component code,

```python
def capture_photo():
    """Captures a photo using the Raspberry Pi camera."""
    output_dir = "/home/roy/photos"
    os.makedirs(output_dir, exist_ok=True)
    photo_path = os.path.join(output_dir, f"photo_{timestamp}.jpg")
    os.system(f"rpicam-still -t 1000 -o {photo_path}")
    return photo_path
```

The above script will take a photo and save it to a temporary path.

Image Analysis

First, I tried with a TensorFlow Lite model and OpenCV Haar Cascade locally on Pi, but since the device is very underpowered and quite old, it wasn't running smoothly.

My Pi config

Then I used **YOLOv8 ONNX** on my Raspberry Pi 3B to build a small vision-based script, as it's a very lightweight model and it runs entirely offline, capturing an image every minute, detecting objects locally, and sending me a Telegram alert if a cat is found.

For understanding YOLO, this video is great —

It’s a super light setup that shows how powerful edge AI can be — real-time object detection, no cloud required.

Here’s my code,

- ***Gist Link*** (Can’t embed here as it’s bit big)

> *I haven’t covered the inner workings of the model in this post — but if that sounds interesting, comment below and I’ll write a beginner-friendly walkthrough.*

Since the image quality, lighting and resolution weren’t great, and the model is very lightweight for this quality image, it had some false positives and true negatives.

Therefore, I am writing a cloud service where the models (bigger) will be hosted and be used via REST/grpc.

For now, I have used perplexity’s pre-existing ***sonar-pro model*** *to do that part (anyway, I got the pro version from Airtel offer), and for image hosting, I used a temporary cloud storage; one can use Cloudinary or S3 too.*

The Image upload and telegram message sending functions were very straightforward, then I had set up the cron job, which runs every minute,

```
# ---- SCHEDULER ----
schedule.every(1).minutes.do(job)

print("🚀 Cat detection running every 1 minute...")
while True:
    schedule.run_pending()
    time.sleep(1)
```

And that's all. Now I run,

```
python3 cat-v6.py
```

> *If you need the full script with more details, please comment, and I will update that.*

The system in action to detect outside cats

When a Cat Meets a Coder's Heart

This wasn’t about building something flashy or complicated. It was about using what I knew best, **tech,** to solve a small, meaningful problem.

> Sometimes we overthink innovation as something that changes the world at scale. But honestly, even a tiny project that makes one life better, or in this case, one cat less hungry — is worth the effort.

The whole experience also reminded me why I fell in love with building/engineering in the first place. It’s the ability to turn ideas (or random emotional moments involving cats) into something real and functional.

Here’s couple of snaps from our cat detection system

Finally,

The system now quietly does its job. Whenever the stray cat shows up, I get a Telegram message. I check the camera feed, smile, grab the food, and step out to meet my little friend, who still meows like he owns the place.

Pika, on the other hand, is quite proud of his assistant role in all this, or maybe he’s just jealous that I’m feeding someone else. Hard to tell.

QA engineer doing his QC (ignore random stuff in the setup, trust me all those has some purpose)

As of now, I still haven't named the stray cat. So if you have any ideas, feel free to suggest one. For now, I just call him *Sir Tabby King*, because he's definitely got royal manners.

ciao!

A message from our Founder

Hey, Sunil here. I wanted to take a moment to thank you for reading until the end and for being a part of this community.

Did you know that our team run these publications as a volunteer effort to over 3.5m monthly readers? **We don't receive any funding, we do this to support the community.** ❤️

If you want to show some love, please take a moment to **follow me on LinkedIn, TikTok**, **Instagram**. You can also subscribe to our **weekly newsletter**.

And before you go, don't forget to **clap** and **follow** the writer!

Raspberry Pi · Python · Object Detection · AI · Artificial Intelligence

Follow

Published in Stackademic

65K followers · Last published just now

Stackademic is a learning hub for programmers, devs, coders, and engineers. Our goal is to democratize free coding education for the world.

Follow

Written by Partha Roy

221 followers · 81 following

Full Stack Engineer & Tech Mentor. Writing about code, product thinking, and real-world lessons from a decade of building software.

No responses yet

Bgerby

What are your thoughts?

More from Partha Roy and Stackademic

In JavaScript in Plain English by Partha Roy

Tried React 19’s Activity Component — Here’s What I Learned

Learn everything about the React Activity Component — how it works, why it matters, key benefits, pros and cons, and where to use it.

Oct 22 306 5

In Stackademic by Subodh Shetty

15 Git Terms That Confuse Developers (and What They Actually Mean)

I’ve been in the trenches for 17 years now - reviewed hundreds of pull requests, fixed a fair share of messy repos, and mentored devs who…

Oct 22 431 4

In Stackademic by Somendradev

10 Niche Developer Tools You Didn't Know Existed

Let's be real — the developer world moves fast. Every week, a dozen new tools launch promising to "boost productivity" or "make your life...

Oct 15 310 5

Partha Roy

Best USB Type-C Hubs/Docking Stations for MacBooks in India — A Guide from Personal Usage [2025]

I went down the rabbit hole of researching USB-C hubs available in India, and after comparing a bunch of options, I’ve boiled it down to 5.

May 29 👏 91

See all from Partha Roy

See all from Stackademic

Recommended from Medium

In Entrepreneurship Handbook by Joe Procopio

AI Porn Is Here—and We’re All Pretending It’ll Be Fine

Sam Altman has decided that five years is enough of a head start for the porn companies

✦ Oct 31 2.2K 101

In Data And Beyond by Adham Khaled

I Don’t Use Microsoft Word for Math Anymore. Gemini’s LaTeX Upgrade Changed Everything.

I tried Gemini’s new LaTeX features — here’s how they fix math for students, engineers, and creators (and why you should care).

Oct 23 330 4

In Python in Plain English by Adonis

You Don’t Understand Python — You Just Memorized Syntax

Knowing def and lambda doesn’t make you a Python developer — understanding what happens under the hood does.

6d ago 31 2

In Coding Nexus by Code Coup

Claude Desktop Might Be the Most Useful Free Tool You'll Install This Year

I didn't expect much when I first saw the announcement for Claude Desktop. Another AI wrapper, I thought. Maybe with a shiny UI.

✦ Oct 23 👏 967 💬 38

Jannis

MS Office Without MS Office — The Free Web App That Does It All

Bring Word, Excel, and PowerPoint to your browser — no subscriptions, no sign-ins, no strings attached.

✦ Oct 30 👏 751 💬 13

Tosny

7 Websites I Visit Every Day in 2025

If there is one thing I am addicted to, besides coffee, it is the internet.

✦ Sep 23 7.5K 263

See more recommendations