Snakes on a train!

09 July 2026 · ai, testing, critical thinking, snakes

Think back to the months after COVID, we were getting out and about again. Maybe going to the office, a couple of days a week. If you'd seen a demo of some software, let's assume a serious application, not just some fabricated startup pitch - a real mission critical system.

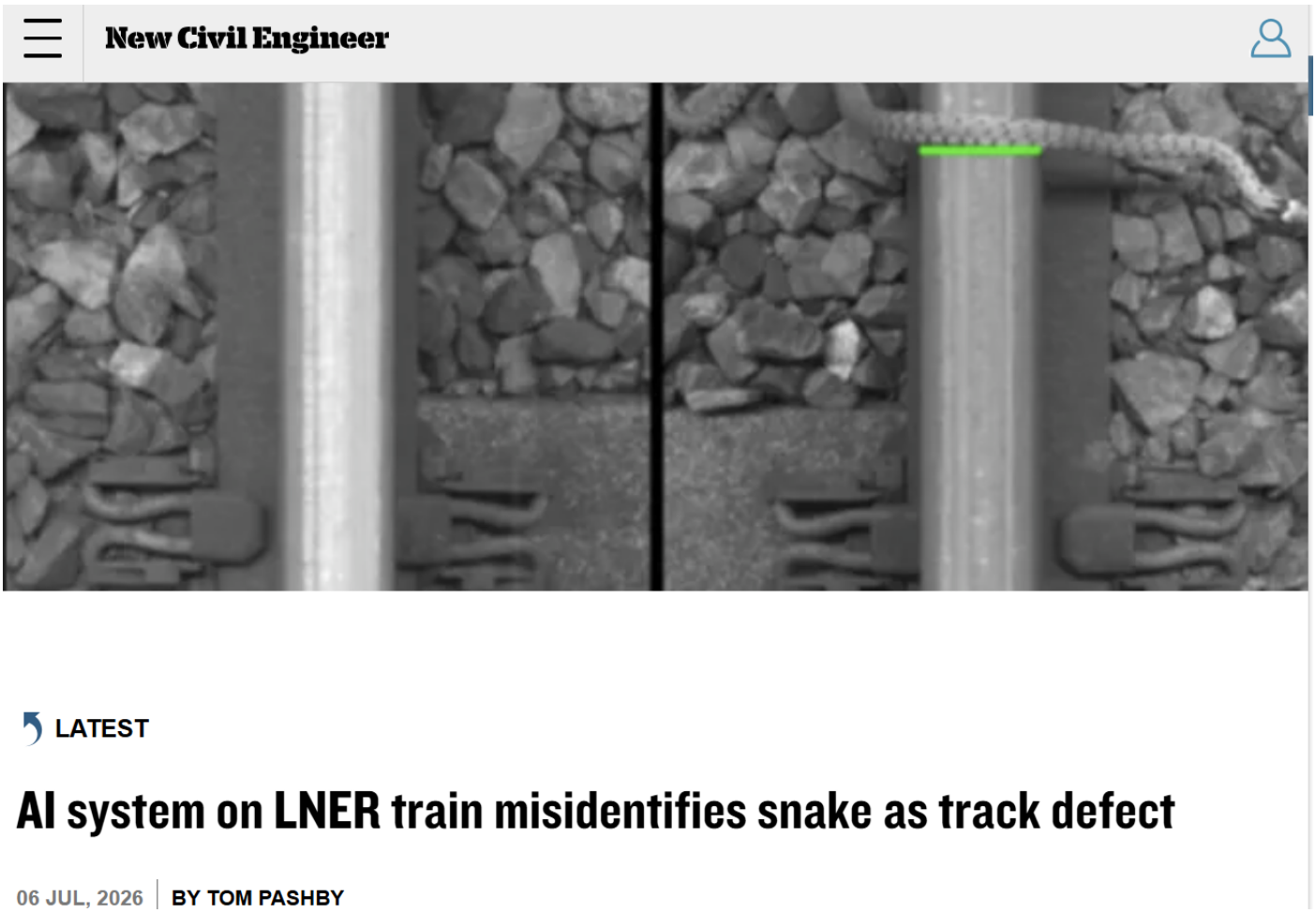
How would you have felt if it didn't work? It made a category error and fundamentally got something wrong. You'd feel different probably, depending on your background. You might feel awkward, annoyed, let down, or maybe overcome by an overwhelming desire to say "A-ha! Its buggy!" (No comment on which one I'd feel.)

But we'd all accept it's a bug, even if the vendor tried to excuse it away. We'd know the sales team had a buggy product. How buggy? Whether those bugs would mean the software wasn't worth the price, who knows? We might even use the bug as a bargaining chip in procurement "Hey we know it's kinda new and it's got issues, as we'll be testing it on the job… let's have a 90% discount!"

But since Covid, a new gambit has hit teams, vendors and their sales reps. I refer to "Hallucinations" or to give it its more accurate technical name "a bug".

How do these bugs come into being? It's all down to the fact that AI models, be they generative or classifiers etc, do one thing: Predict. They make predictions, they don't calculate like a traditional 'deterministic' application. They find the most likely outcome. whether that's the next word or whether a picture is of a broken train track.

Which brings me to this article (https://www.newcivilengineer.com/latest/ai-system-on-lner-train-misidentifies-snake-as-track-defect-06-07-2026/), about a train track maintenance system that mistook a snake (yes wet and cold England does have a few snakes!) for a crack in the track. That's a bad prediction by the model, causing a bug in the application. Once again for those at the back… The mission critical system for detecting cracks in train tracks has at least one bug.



As an aside, it's probably ok to have a few miss-fires or 'false positives', if it means there are fewer 'false negative' where a broken track is classified as a snake or twig. That causes an extra expense, but probably little actual harm.

But one does not mean the other, the system can be buggy on both sides of the problem, seeing lots of snake-cracks and lots of crack-snakes… who knows?

I'll tell you who knows, your tester. the person putting the system through its paces, detecting those bugs. Discovering how does the system actually behave? Giving you facts about what fails and what works. So that you are forewarned and for armed.

Why? If you think your data science team or external vendor is giving your accurate success/fail rates and bug reports, then I'm sorry to say you **are hallucinating**.