Computer Vision & Vibe Coding

Andrés Caballero

RESEARCH & PARTNERSHIPS
Data Science Institute

Why?

Roots for Resilience Fellowship (R4R)

Trevor Paglen "Seeing Machines" workshop

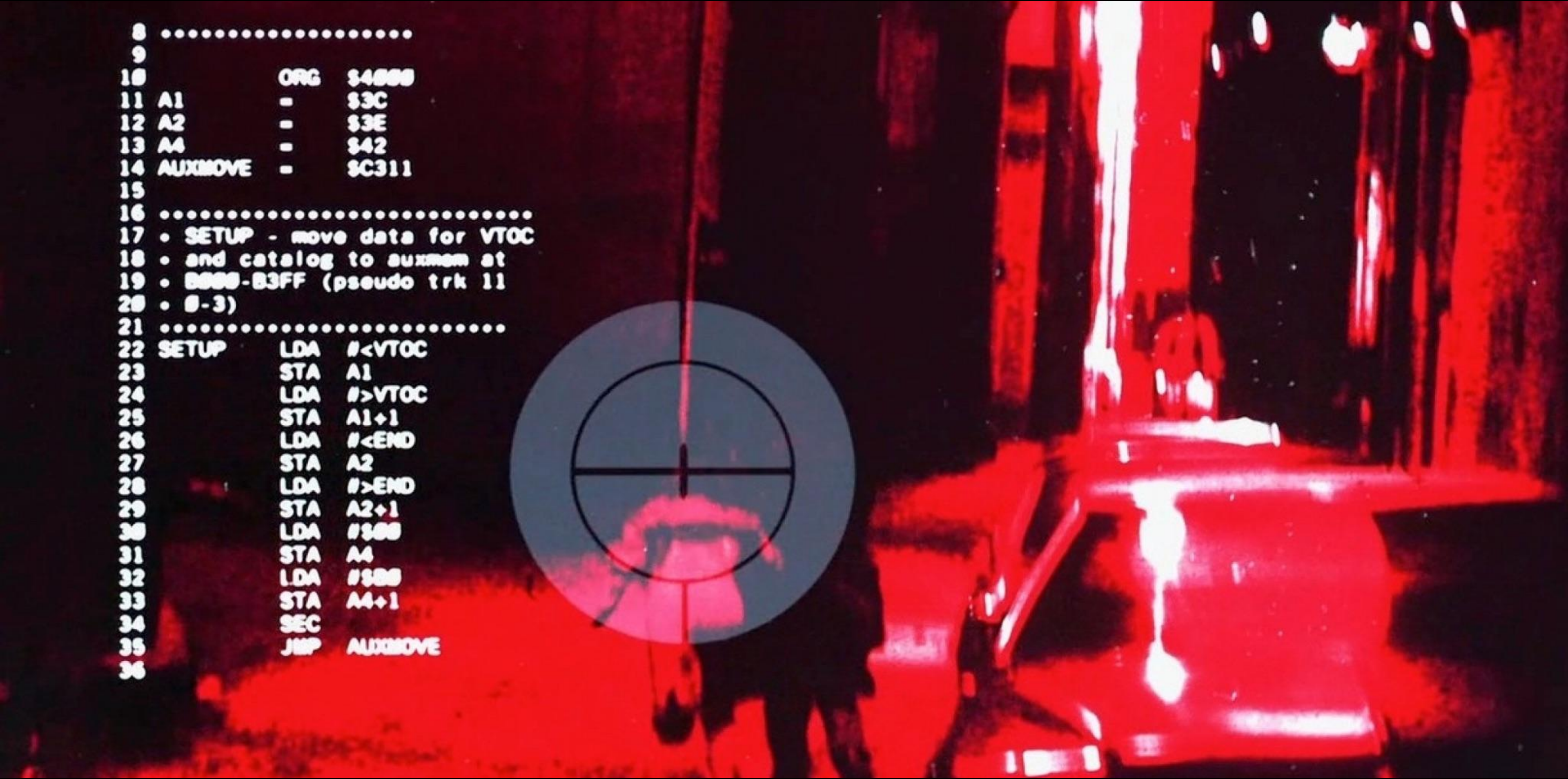
What we'll cover

Part one:

- → Concepts: *Computer Vision & Vibe Coding*
- → Our toolchain (Terminal + Python + OpenCV)
- → Applications for artistic practice & research

Part two:

- → Live Demo

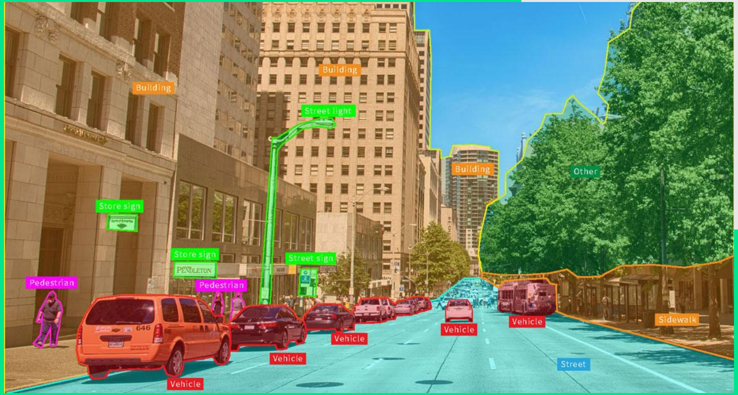


```
••••••••••••••••••

         ORG   $4000
A1       =     $3C
A2       =     $3E
A4       =     $42
AUXMOVE  =     $C311

••••••••••••••••••••••••••
• SETUP - move data for VTOC
• and catalog to auxmem at
• B000-B3FF (pseudo trk 11
• 0-3)
•••••••••••••••••••••••••
SETUP    LDA   #<VTOC
         STA   A1
         LDA   #>VTOC
         STA   A1+1
         LDA   #<END
         STA   A2
         LDA   #>END
         STA   A2+1
         LDA   #$00
         STA   A4
         LDA   #$B0
         STA   A4+1
         SEC
         JMP   AUXMOVE

```

COMPUTER VISION

- Computer vision is the field where computers interpret images and video.

- Instead of ‘seeing’ like humans, they work with pixels, numbers, and patterns.

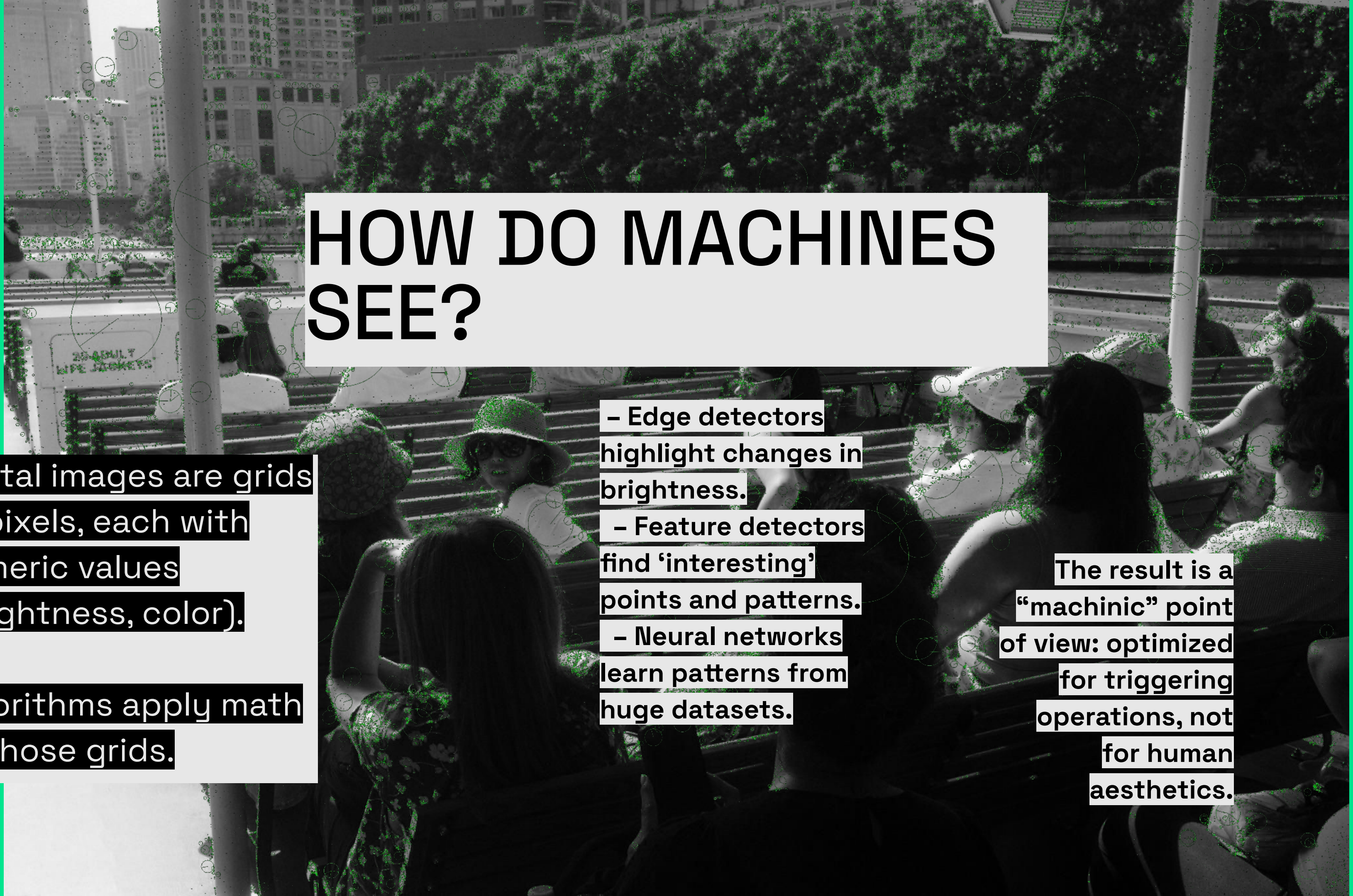


- Typical tasks:

  – Detecting edges, shapes, and motion.
  – Recognizing faces, objects, text, or gestures.
  – Segmenting an image into regions (sky, skin, road, etc.).

It powers phone cameras, face unlock, self-driving cars, CCTV, and more.

HOW DO MACHINES SEE?



Digital images are grids of pixels, each with numeric values (brightness, color).

Algorithms apply math to those grids.

– Edge detectors highlight changes in brightness.

– Feature detectors find ‘interesting’ points and patterns.

– Neural networks learn patterns from huge datasets.

The result is a “machinic” point of view: optimized for triggering operations, not for human aesthetics.

Cool Art

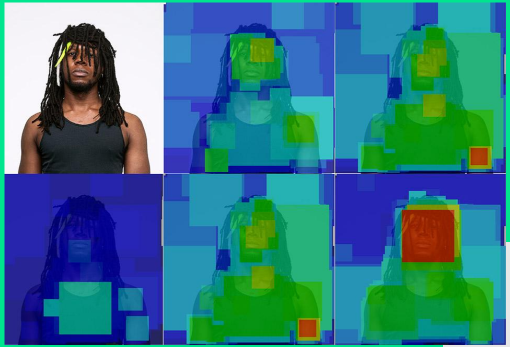
Surveillance, datasets, bias, and the politics of supposedly 'neutral' algorithms

→ Agency over tools

→ Misusing the tool intentionally, for a specific effect.

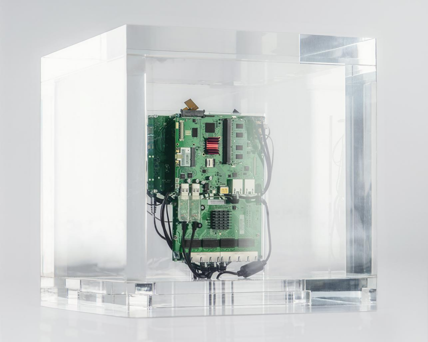
→ Visualizing the "pipeline"

→ Messing with the technology: Self-surveillance, flooding the system, hidden browsing

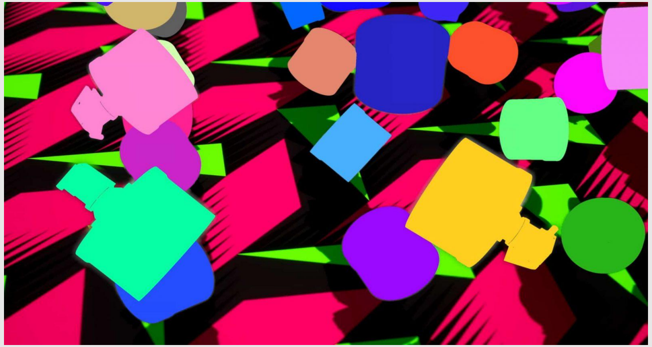


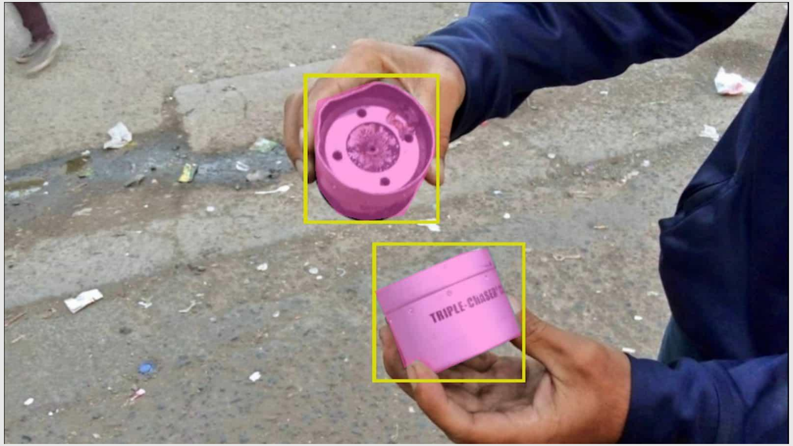
→ Adversarial Camouflage

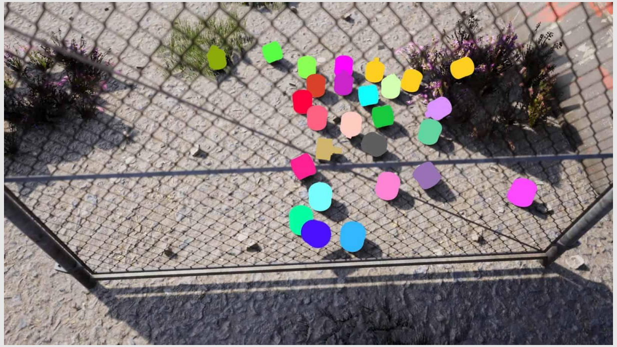
→ Accountability

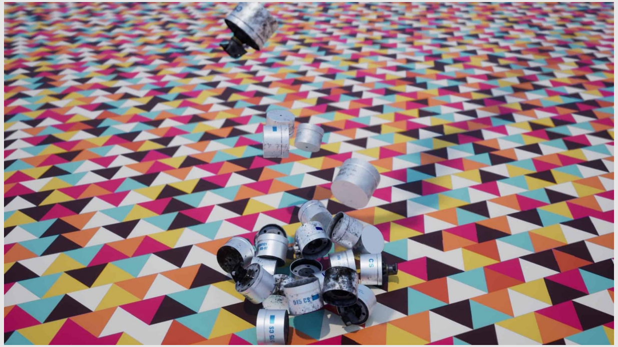


TRIPLE CHASER : Forensic Architecture (2019)

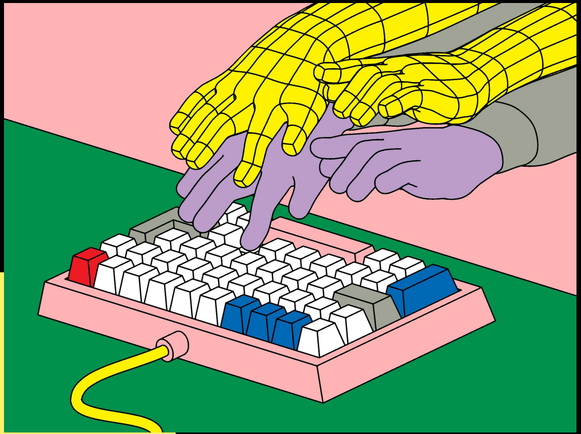








VIBE CODING



Describe what you want in plain language, let the LLM write the code for you.

Coding as sketching, improvising, and iterating – not perfectionism.

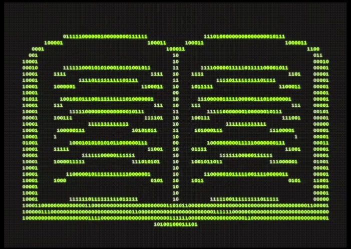
- ➔ Start rough, unsophisticated.
- ➔ Focus on behavior and experiments, not perfect architecture.
- ➔ Let the LLM handle technical details while you decide the artistic direction.

Key Words

- **Prototype** – a quick version that proves the idea.
- **Prompt** – the instructions you give your LLM co-coder.
- **Refactor** – cleaning and reorganizing existing code.
- **Debug** – figuring out why something is broken and fixing it.
- **Iteration** – make a version, test, adjust, repeat.

Treat code like a sketchbook, not a finished print.

Curiosity over correctness; collaboration over control.



Your Toolchain

Free and simple to use.

Terminal

The text interface where we run commands and scripts.

Python 3.10

Very popular, beginner-friendly programming language.

LLM*

Large Language Models. ChatGPT, Claude, Gemini; your main collaborator.

VS Code*

Code editor that makes writing and running scripts easier.

Open CV

Powerful computer vision library (edges, detection, tracking, etc.).

Homebrew*

Package manager for macOS that lets you easily install software through command-line tools.

*Paid version recommended

*Optional

*Optional

Collaborating with your LLM is key!

Your LLM is your best friend.

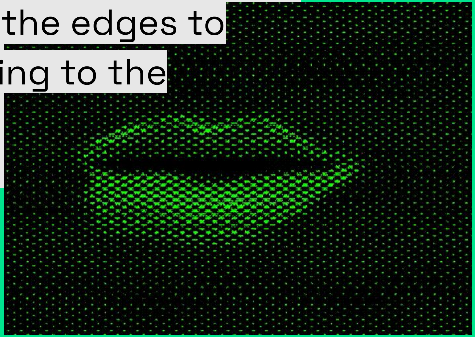
➔ They will help you:

- Turn ideas into working code quickly.
- Translate plain language to Python scripts.
- Debug error messages and weird behavior.
- Learn new techniques without tutorials.

Instead of memorizing syntax, you learn to write detailed prompts.

Categories of prompts you can use

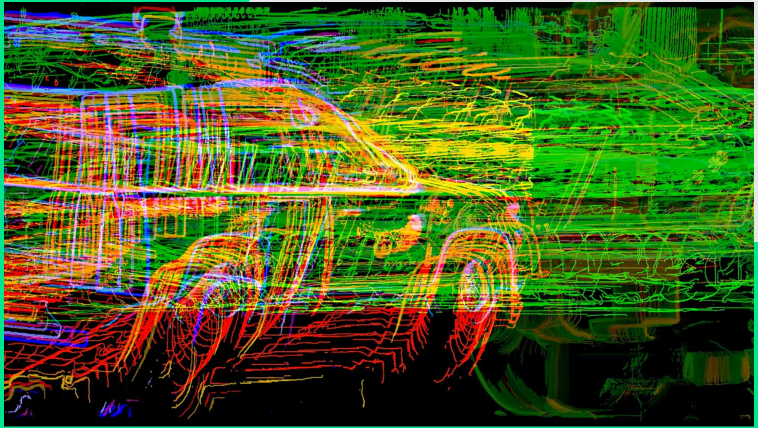
- I want to…
  “Write a Python script using OpenCV that turns portrait.jpg into black and white and saves the new file.”
- Fix this…
  “Here’s my error message, fix the code and explain what went wrong.”
- Explain this…
  “Explain what SIFT does, like I’m new to computer vision.”
- Change this…
  “Change the colors of the edges to be color coded according to the object detected.”



Continuation...

Mix context + goal + constraints:

"I'm an artist who barely codes. Write a Python script using OpenCV that draws motion trails on moving objects in a video. Explain which parameters I can change."



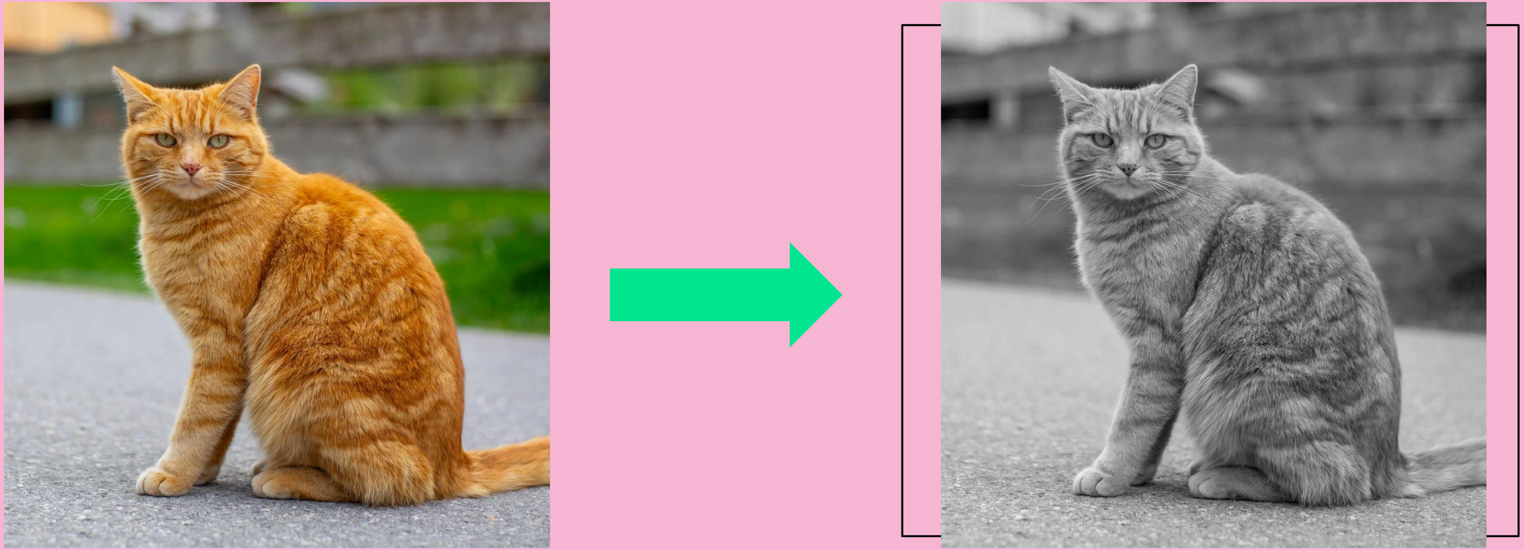
Learning together:

"What are some of the things OpenCV can do?"

"Explain three different ways to detect motion in a video with OpenCV".

What is OpenCV?

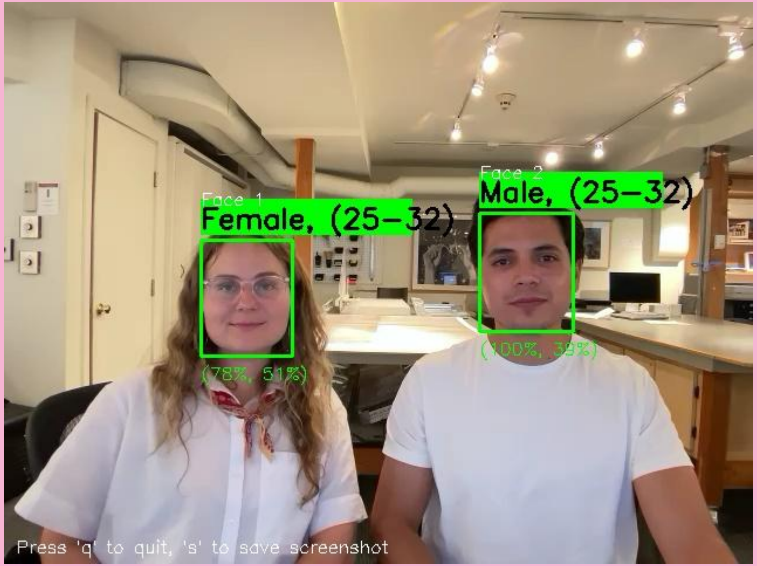
Live Demo: Photo



Live Demo: Video

Trevor Paglen: At the Expense of Everybody Else:
https://www.youtube.com/watch?v=Qmty4__lV30

Live Demo: Operational Images



Press 'q' to quit, 's' to save screenshot

Tools & Resources at the University

Think of these tools as another layer inside your existing infrastructure.

- ➔ CyVerse : for data-intensive or remote computing workflows. Imaging Computer
- ➔ **High Performance Computer, Data Visualization lab:** access to GPUs, storage, and consultation.
- ➔ **Imaging lab** : High-performance work station with professional GPUs for advanced research, rendering, and computation.
- ➔ **Catalyst Studio (VR/AR)**: to bring CV-generated imagery into immersive installations.

Software and software licences for UofA community

MORE REFERENCES

➔ **Harun Farocki:**
Eye / Machine (Full Video) Part I of III

➔ **Trevor Paglen:**

❖ Works:
-(Camouflage) :
https://paglen.studio/2020/04/09/autonomy-cube/
-(Visualizing the pipeline):
https://kadist.org/work/behold-these-glorious-times/ (Full Video)

❖ Writings:
-Invisible Images:
https://thenewinquiry.com/invisible-images-your-pictures-are-looking-at-you/
-Excavating AI:
https://excavating.ai/
-Society of the Psyop:
https://www.e-flux.com/journal/147/623330/society-of-the-psyop-part-1-ufos-and-the-future-of-media

➔ **Adam Harvey:**
https://adam.harvey.studio/cvdazzle
(Camouflage)

➔ **ScanLAB:**
https://scanlabprojects.co.uk/projects/dreamlife-of-driverless-cars/ (intentional glitches)

➔ **Hito Steyerl:** - How not to be seen

➔ Forensic Architecture (Website)

➔ Operational Images: From the Visual to the Invisual - Jussi Parikka (Book)

➔ E-flux (Website)

Special thanks to the R4R faculty!

Michele Cosi

Tina Johnson

Jeff Gillan