

An Introduction to Computer Vision

Nsumba Solomon & Akera Benjamin
AI Research Lab, Makerere University

June 4, 2019

can you see me?

Figure: computer vision at its best



Introduction

Humans are visual creatures we understand things best by seeing them. We have two eyes, each of which captures a two dimensional version of our surroundings, and somehow our brains can stitch those images together into a three dimensional worldview. But what if we replace the two eyes with digital cameras, and the brain with a computer: can we teach a computer to see the world around it? We are going to see just how some tasks in computer vision are quite easy, and how others are impossibly difficult, through demonstrations that you can take back home, replicate, and build upon.

What is Computer Vision

Computer vision has to do with teaching computers to recognize images, to interpret them in some way. Maybe that means identifying objects as tables and chairs and cars and people, or maybe that means dividing the image into foreground and background, or maybe that means classifying textured regions as either grass or brick. It could be anything; literally, anything that we do with our own eyes.

What to expect

We're going to look at a bunch of different examples of different things we can do with computer vision. I should point out that, as you might expect with a subject this broad, the amount of work that's been done in computer vision already does not even begin to compare with the breath of potential applications. You can definitely imagine tasks that we humans do with relative ease (how about estimating an object's depth from the camera?) that no one has really managed to do well with computers. It's an open area of research that's what makes it so cool!

Applications

- Medical image analysis
- Aerial photo interpretation
- Vehicle exploration and mobility
- Material handling
- Inspection For example, integrated circuit board and chip inspection
- Assembly
- Navigation
- Human-computer interfaces
- Multimedia
- Telepresence/Tele-immersion/Tele-reality

Implementation Requirements

The demos are all written in Python, which is a freely available programming language intended for quick and easy development.

Figure: Tools



Computer Vision Algorithms

Edge operators

These are probably some of the most fundamental tools in computer vision. As the name suggests, the goal of these operators (we call them filters) is to determine where the boundaries are in the image. Where does one object end, and another begin? This is critical for a lot of different tasks, like image segmentation.

example

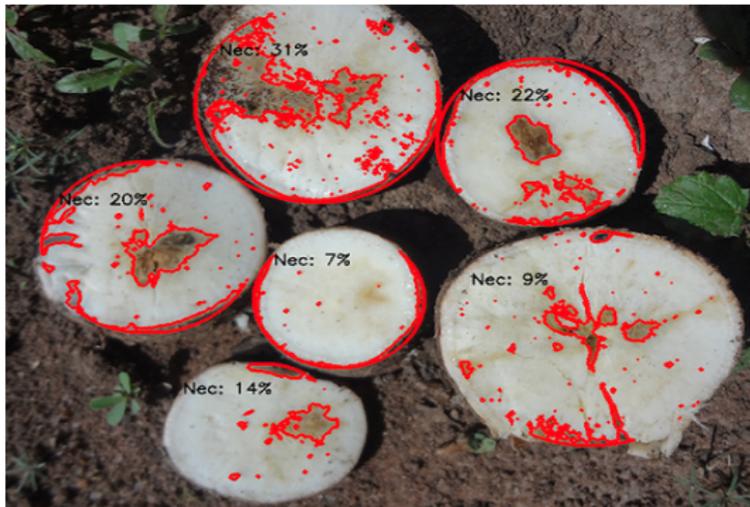


Figure: image segmentation to determine root necrosis

Canny edge detection

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.



Figure: Original image on the left Processed image on the right

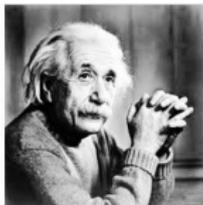
Entropy

This is a term that comes from chemistry, where it is used to describe the disorder of a sample. The higher the disorder, the greater the entropy. Actually, the second law of thermodynamics can be stated as the universe tends towards increasing entropy that's why if I never make an effort to clean up my room, it just keeps on getting messier. In computer vision, we often use local entropy measured in some neighborhood of a pixel to measure the texture in that area. Higher entropy means more texture.

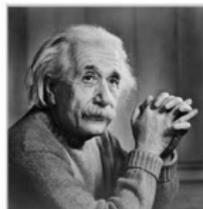
Histogram equalization

This is a class of algorithms that seek to improve the contrast of an image.

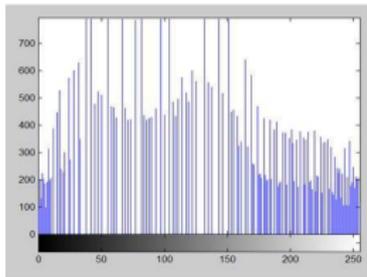
New Image



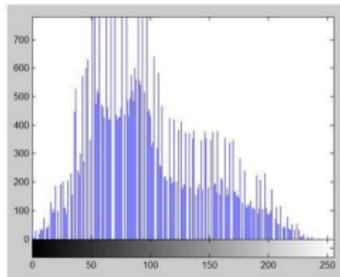
Old image



New Histogram



Old Histogram



Adaptive thresholding

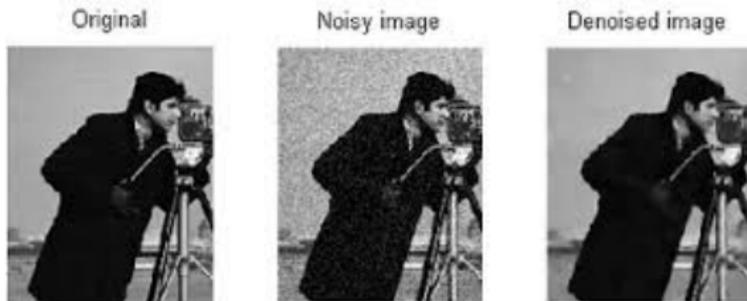
This is a technique for separating the pixels in every small region of the image into two classes, white and black, depending on their intensity. Compare adaptive to global thresholding notice that global thresholding is not robust to gradual shifts in background pixel intensity (e.g. due to lighting changes or shadows), while adaptive thresholding is more sensitive to local variations in intensity.

Tinting a grayscale image

This is a common filter you can apply in photo editing suites like Photoshop. It may seem complicated its changing the entire color of the image after all but it turns out to be super simple. Basically, you can transform the traditional RGB (red, green, blue) color representation space into what is called HSV (hue, saturation, value/ luminance). Hue and saturation are like the angle and radius on a color wheel, and value/ luminance is exactly the grayscale pixel intensity. Clearly then, all we need to do to tint a grayscale image is set value equal to grayscale intensity, then select appropriate, constant hue and saturation values depending on what color tint we want.

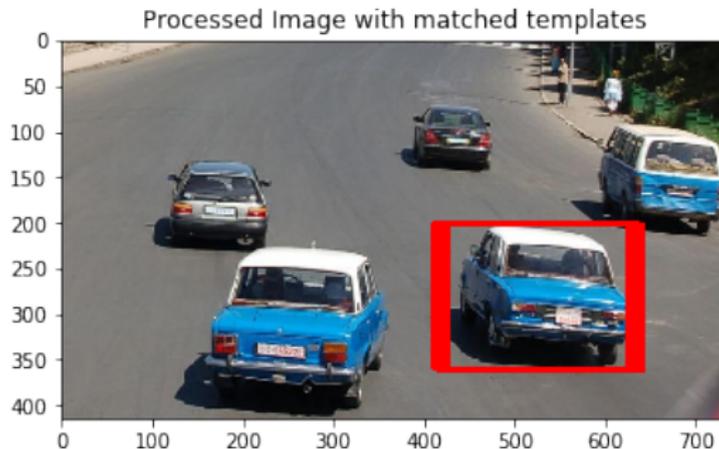
Image denoising

This is one of the classic applications of computer vision. Often times images are what we call "noisy." They have random fluctuations that make the image look like its corrupted. There are a lot of different denoising algorithms out there, but basically all of them try to identify pixels that dont look like their surroundings, then interpolate some value for those pixels based on the surrounding pixels.



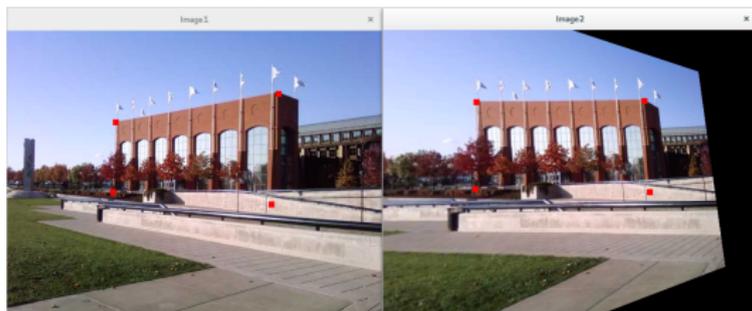
Template matching

This is a key element of object recognition. The idea is to look for a particular object in an image, given that you know what it should look like (i.e. you have a template) beforehand. One way to do this is just to slide the template across the image and look for when most of the pixels match up, but of course that is not necessarily robust to things like rescaling or lighting changes.



Geometric transformations

Are probably the coolest thing we've seen so far. Suppose you're looking at an image of a hallway, and you want to see what it would look like from a different perspective. If we were literally standing there with the camera, of course we would just move to a new location and take a new picture, but it turns out that you can computationally change the perspective as well, without taking a new image. This is called "warping." It relies on some simple linear algebra, which is fancy language for matrix multiplication.



Fourier analysis

Incredibly useful for understanding some of the deeper, hidden information content of images. Basically, Fourier analysis allows us to decompose an image (or any function, in any number of dimensions) into a set of oscillating functions called sinusoids.

Tutorial

Working with images of Amharic Numbers....



የአማርኛ ቁጥሮች ለማሳያ		
1	አንድ	Aand
2	ሁለት	Hulett
3	ሶስት	Sosst
4	አራት	Aarat
5	አምስት	Ammes
6	ፊት	Seddes
7	ሥንጠረዥ	Sabat
8	ጠቅላይ	Semme
9	ዓምስት	Zatang
10	አስር	Assan