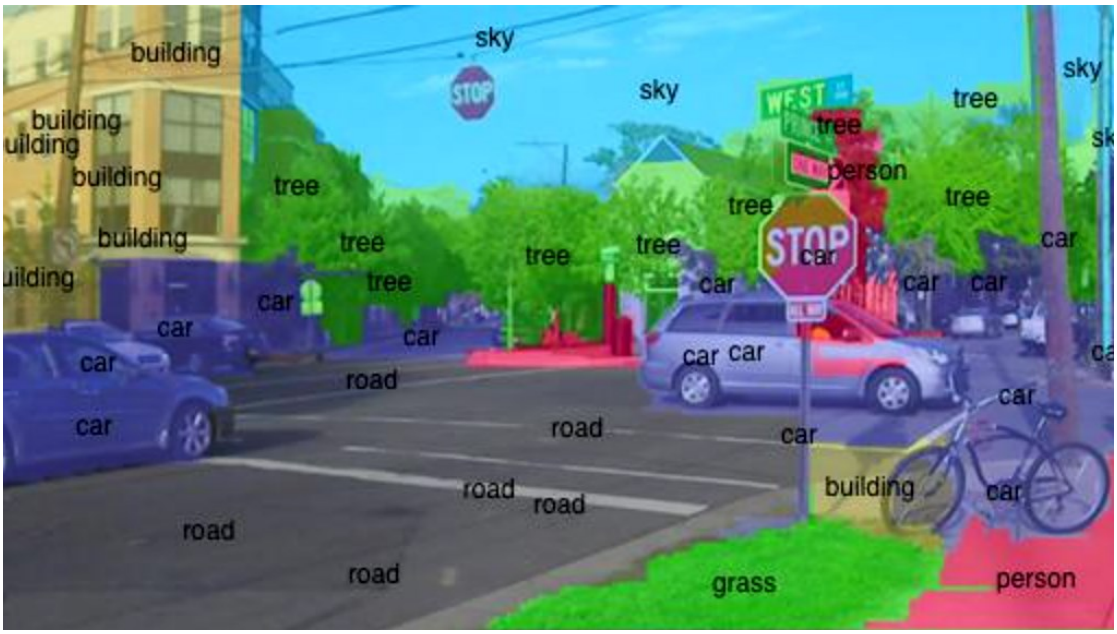
ALGORITHMIA

APRIL 2, 2018

Introduction to computer vision: what it is and how it works



Source: TechCrunch

Using software to parse the world's visual content is as big of a revolution in computing as mobile was 10 years ago, and will provide a major edge for developers and businesses to build amazing products.

Computer Vision is the process of using machines to understand and analyze imagery (both photos and videos). While these types of algorithms have been around in various forms since the 1960's, recent advances in Machine Learning, as well as leaps forward in data storage, computing capabilities, and cheap high-quality input devices, have driven major improvements in how well our software can explore this kind of content.

What is computer vision?

Computer vision is the process of using machines to understand and analyze imagery (both photos and videos). While these types of algorithms have been around in various forms since the 1960's, recent advances in Machine Learning, as well as leaps forward in data storage, computing capabilities, and cheap high-quality input devices, have driven major improvements in how well our software can explore this kind of content.

Computer vision is the broad parent name for any computations involving visual content – that means images, videos, icons, and anything else with pixels involved. But within this parent idea, there are a few specific tasks that are core building blocks:

- In **object classification**, you train a model on a dataset of specific objects, and the model classifies new objects as belonging to one or more of your training categories.
- For **object identification**, your model will recognize a specific instance of an object – for example, parsing two faces in an image and tagging one as Tom Cruise and one as Katie Holmes.

A classical application of computer vision is handwriting recognition for digitizing handwritten content (we'll explore more use cases below). Outside of just recognition, other methods of analysis include:

- Video **motion analysis** uses computer vision to estimate the velocity of objects in a video, or the camera itself.
- In **image segmentation**, algorithms partition images into multiple sets of views.
- **Scene reconstruction** creates a 3D model of a scene inputted through images or video (check out Selva).
- In **image restoration**, noise such as blurring is removed from photos using Machine Learning based filters.

Any other application that involves understanding pixels through software can safely be labeled as computer vision.

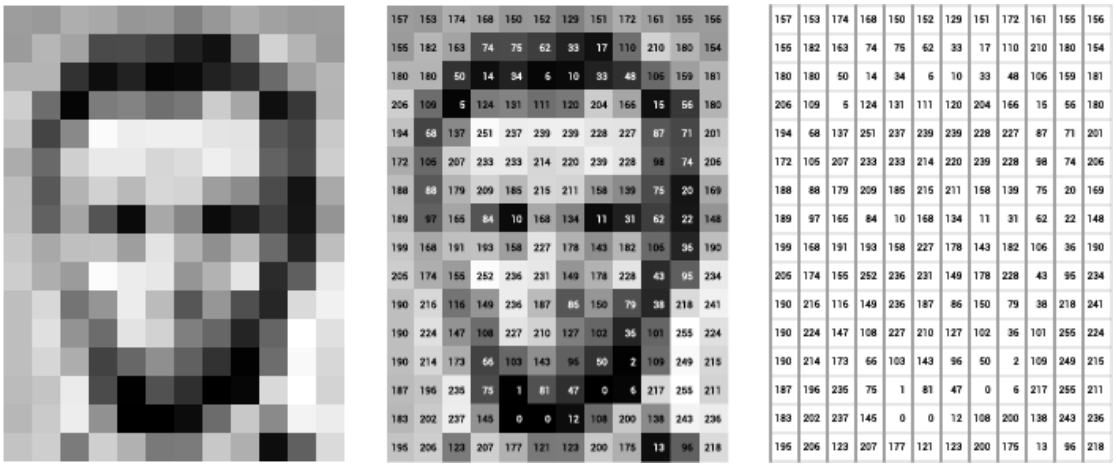
How computer vision works

One of the major open questions in both Neuroscience and Machine Learning is: how exactly do our brains work, and how can we approximate that with our own algorithms? The reality is that there are very few working and comprehensive theories of brain computation; so despite

the fact that Neural Nets are supposed to “mimic the way the brain works,” nobody is quite sure if that’s actually true. Jeff Hawkins has an entire book on this topic called On Intelligence.

The same paradox holds true for computer vision – since we’re not decided on how the brain and eyes process images, it’s difficult to say how well the algorithms used in production approximate our own internal mental processes. For example, studies have shown that some functions that we thought happen in the brain of *frogs* actually take place in the eyes. We’re a far cry from amphibians, but similar uncertainty exists in human cognition.

Machines interpret images very simply: as a series of pixels, each with their own set of color values. Consider the simplified image below, and how grayscale values are converted into a simple array of numbers:

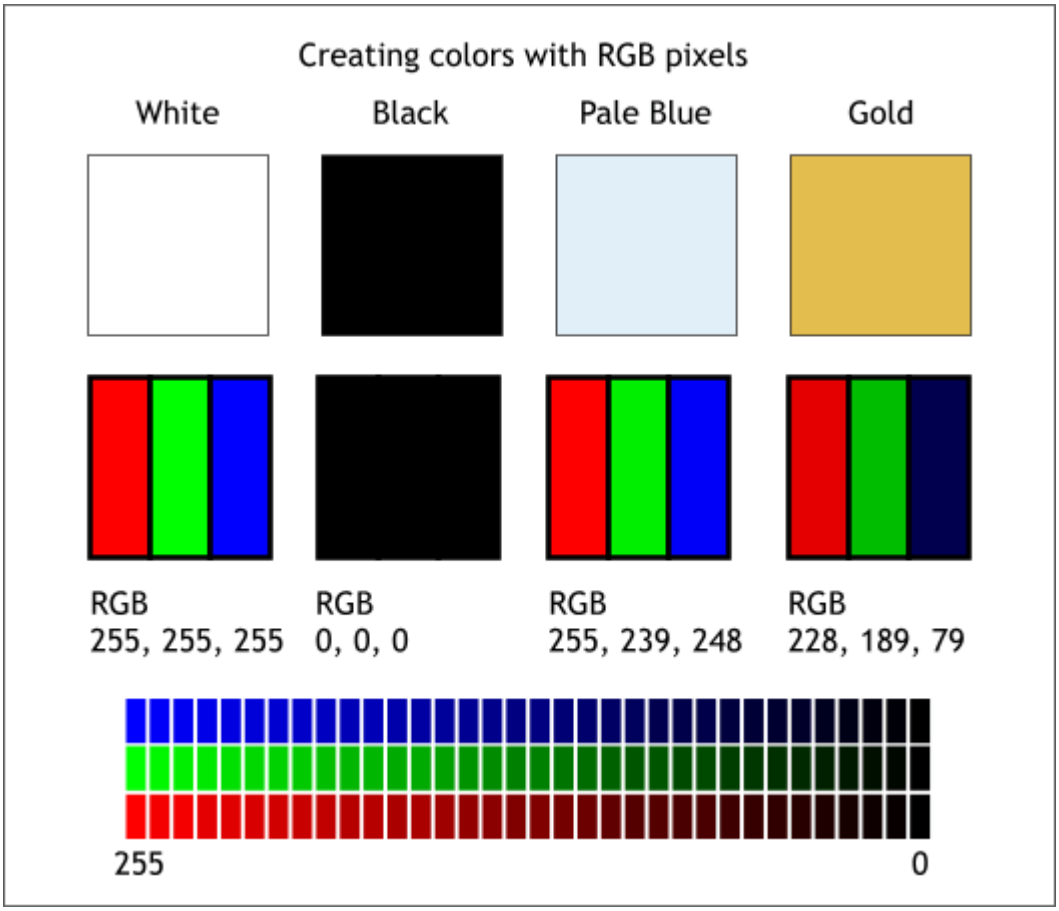


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Source: Openframeworks

Think of an image as a giant grid of different squares, or pixels (this image is a very simplified version of what looks like either Abraham Lincoln or a Dementor). Each pixel in an image can be represented by a number, usually from 0 – 255. The series of numbers on the right is what software sees when you input an image. For our image, there are 12 columns and 16 rows, which means there are 192 input values for this image.

When we start to add in color, things get more complicated. Computers usually read color as a series of 3 values – red, green, and blue (RGB) – on that same 0 – 255 scale. Now, each pixel actually has 3 values for the computer to store in addition to its position. If we were to colorize President Lincoln (or Harry Potter’s worst fear), that would lead to 12 x 16 x 3 values, or 576 numbers.



Source: Xaraxone

For some perspective on how computationally expensive this is, consider this tree:

- Each color value is stored in 8 bits.
- 8 bits x 3 colors per pixel = 24 bits per pixel.
- A normal sized 1024 x 768 image x 24 bits per pixel = almost 19M bits, or about 2.36 megabytes.

That's a lot of memory to require for one image, and a lot of pixels for an algorithm to iterate over. But to train a model with meaningful accuracy – especially when you're talking about Deep Learning – you'd usually need tens of thousands of images, and the more the merrier. Even if you were to use Transfer Learning to use the insights of an already trained model, you'd still need a few thousand images to train yours on.

With the sheer amount of computing power and storage required just to train deep learning models for computer vision, it's not hard to understand why advances in those two fields have driven Machine Learning forward to such a degree.

Business use cases for computer vision

Computer vision is one of the areas in Machine Learning where core concepts are already being integrated into major products that we use every day. Google is using maps to leverage their image data and identify street names, businesses, and office buildings. Facebook is using computer vision to identify people in photos, and do a number of things with that information.

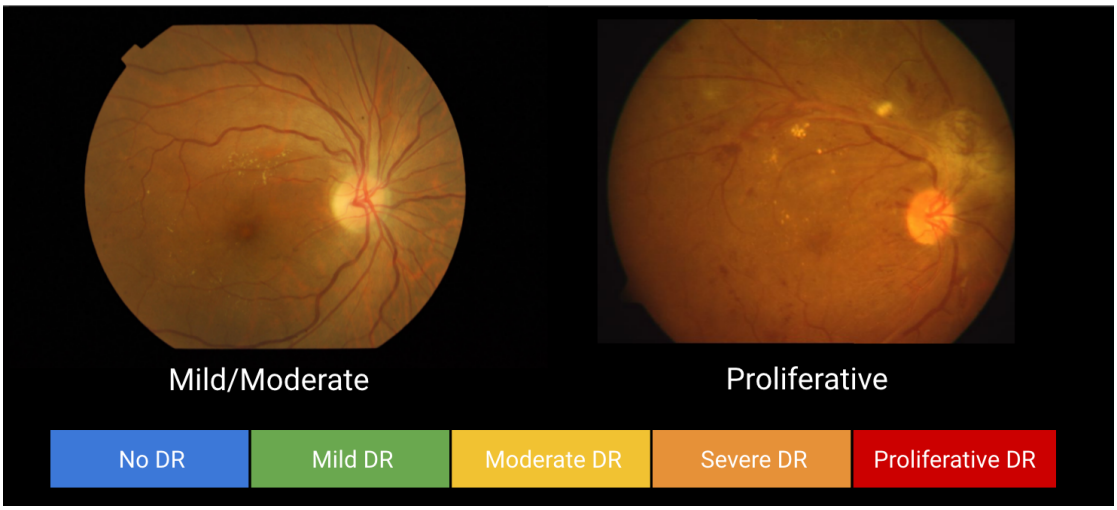
But it's not just tech companies that are leverage Machine Learning for image applications. Ford, the American car manufacturer that has been around literally since the early 1900's, is investing heavily in autonomous vehicles (AVs). Much of the underlying technology in AVs relies on analyzing the multiple video feeds coming into the car and using computer vision to analyze and pick a path of action.

Another major area where computer vision can help is in the medical field. Much of diagnosis is image processing, like reading x-rays, MRI scans, and other types of diagnostics. Google has been working with medical research teams to explore how deep learning can help medical workflows, and have made significant progress in terms of accuracy. To paraphrase from their research page:

"*Collaborating closely with doctors and international healthcare systems, we developed a state-of-the-art computer vision system for reading retinal fundus images for diabetic retinopathy and determined our algorithm's performance is on par with U.S. board-certified*

ophthalmologists. We've recently published some of our research in the Journal of the American Medical Association and summarized the highlights in a blog post."

Source: Research at Google



On a less serious note, this clip from HBO's Silicon Valley about using computer vision to distinguish a hot dog from, well, anything else, was pretty popular around social media.

But aside from the groundbreaking stuff, it's getting much easier to integrate computer vision into your own applications. A number of high-quality third party providers like Clarifai offer a simple API for tagging and understanding images, while Kairos provides functionality around facial recognition. We'll dive into the open-source packages available for use below.

computer vision and convolutional neural networks

Much of the progress made in computer vision accuracy over the past few years is due in part to a special type of algorithm. Convolutional Neural Networks are a subset of Deep Learning with a few extra added operations, and they've been shown to achieve impressive accuracy on image-associated tasks.

Convolutional Neural Networks (CNNs or ConvNets) utilize the same major concepts of Neural Networks, but add in some steps before the normal architecture. These steps are focused on feature extraction, or finding the best version possible of our input that will yield the greatest level of understanding for our model. Ideally, these features will be less redundant and more informative than the original input.

The CNN uses three sorts of filters for feature extraction. For more detail and interactive diagrams, Ujjwal Karn's walkthrough post on the topic is excellent.

Convolution

During the convolution process (perhaps why it's called a CNN) the input image pixels are modified by a filter. This is just a matrix (smaller than the original pixel matrix) that we multiply different pieces of the input image by. The output – often called a Feature Map – will usually be smaller than the original image, and theoretically be more informative.

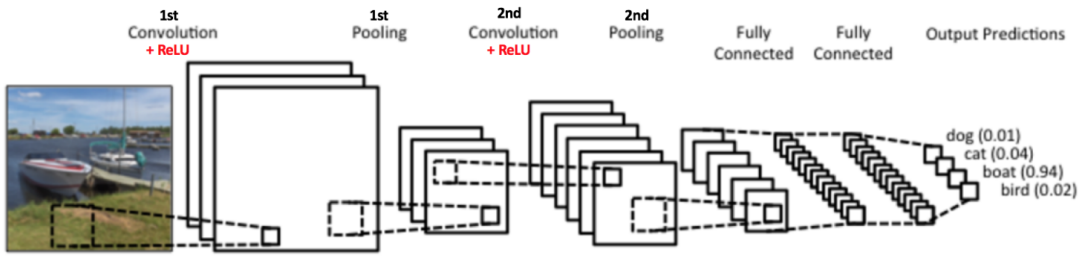
ReLU

This futuristic sounding acronym stands for Rectified Linear Unit, which is an easy function to introduce non-linearity into the feature map. All negative values are simply changed to zero, removing all black from the image. The formal function is $y = \max(0, x)$.

Pooling

In pooling, the image is scanned over by a set width of pixels, and either the max, sum, or average of those pixels is taken as a representation of that portion of the image. This process further reduces the size of the feature map(s) by a factor of whatever size is pooled.

All of these operations – Convolution, ReLu, and Pooling – are often applied twice in a row before concluding the process of feature extraction. The outputs of this whole process are then passed into a neural net for classification. The final architecture looks as follows:



Source: Ujjwal Karn

If you've gotten lost in the details, not to worry. Just remember:

- Convolutional Neural Networks (CNNs) are a special type of Deep Learning that works really well on computer vision tasks
- A lot of preprocessing work is done on the input images to make them better optimized for the fully connecgted layers of the neural net

And that's it!

Computer vision on Algorithmia

Algorithmia makes it easy to deploy computer vision applications as scalable microservices. Our marketplace has a few algorithms to help get the job done:

- SalNet automatically identifies the most important parts of an image
- Nudity Detection detects nudity in pictures
- Emotion Recognition parses emotions exhibited in images
- DeepStyle transfers next-level filters onto your image
- Face Recognition...recognizes faces.
- Image Memorability judges how memorable an image is.

A typical workflow for your product might involve passing images from a security camera into Emotion Recognition and raising a flag if any aggressive emotions are exhibited, or using Nudity Detection to block inappropriate profile pictures on your web application.

For a more detailed exploration of how you can use the Algorithmia platform to implement complex and useful computer vision tasks, check out our primer here.

Computer vision resources

Packages and frameworks

OpenCV – "*OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.*"

SimpleCV – "*SimpleCV is an open source framework for building computer vision applications. With it, you get access to several high-powered computer vision libraries such as OpenCV – without having to first learn about bit depths, file formats, color spaces, buffer management, eigenvalues, or matrix versus bitmap storage.*"

Mahotas – "*Mahotas is a computer vision and image processing library for Python. It includes many algorithms implemented in C++ for speed while operating in numpy arrays and with a very clean Python interface. Mahotas currently has over 100 functions for image processing and computer vision and it keeps growing.*"

Openface – "*OpenFace is a Python and Torch implementation of face recognition with deep neural networks and is based on the CVPR 2015 paper FaceNet: A Unified Embedding for Face Recognition and Clustering by Florian Schroff, Dmitry Kalenichenko, and James Philbin at Google. Torch allows the network to be executed on a CPU or with CUDA.*"

Ilastik – "*Ilastik is a simple, user-friendly tool for interactive image classification, segmentation and analysis. It is built as a modular software framework, which currently has workflows for automated (supervised) pixel- and object-level classification, automated and semi-automated object tracking, semi-automated segmentation and object counting without detection. Using it requires no experience in image processing.*"

Online courses and videos

Introduction to Computer Vision (Georgia Tech and Udacity) – "*This course provides an introduction to computer vision including fundamentals of image formation, camera imaging geometry, feature detection and matching, multiview geometry including stereo, motion estimation and tracking, and classification. We focus less on the machine learning aspect of CV as that is really classification theory best learned in an ML course.*"

Convolutional Neural Networks (Deeplearning.ai and Coursera) – "*This course will teach you how to build convolutional neural networks and apply it to image data. Thanks to deep learning, computer vision is working far better than just two years ago, and this is enabling numerous exciting applications ranging from safe autonomous driving, to accurate face recognition, to automatic reading of radiology images.*"

Introduction to Computer Vision (Brown) – "*This course provides an introduction to computer vision, including fundamentals of image formation, camera imaging geometry, feature detection and matching, stereo, motion estimation and tracking, image classification, scene understanding, and deep learning with neural networks. We will develop basic methods for applications that include finding known models in images, depth recovery from stereo, camera calibration, image stabilization, automated alignment, tracking, boundary detection, and recognition.*"

There are a number of good YouTube series available as well. Two of the most popular options include Fundamentals of Computer Vision and a Gentle Introduction to Computer Vision. Also check out Algorithmia's detailed tutorial around facial recognition using OpenFace.

Books

Computer Vision: Algorithms and Applications – "*Computer Vision: Algorithms and Applications explores the variety of techniques commonly used to analyze and interpret images. It also describes challenging real-world applications where vision is being successfully used, both for specialized applications such as medical imaging, and for fun, consumer-level tasks such as image editing and stitching, which students can apply to their own personal photos and videos.*"

Programming Computer Vision with Python (O'Reilly) – "*If you want a basic understanding of computer vision's underlying theory and algorithms, this hands-on introduction is the ideal*

place to start. You'll learn techniques for object recognition, 3D reconstruction, stereo imaging, augmented reality, and other computer vision applications as you follow clear examples written in Python."

Learning OpenCV (O'Reilly) – "*Learning OpenCV puts you in the middle of the rapidly expanding field of computer vision. Written by the creators of the free open source OpenCV library, this book introduces you to computer vision and demonstrates how you can quickly build applications that enable computers to "see" and make decisions based on that data.*"

Continue Learning

An Introduction to Deep Learning

Introduction to Natural Language Processing (NLP): What is NLP?

Introduction to Sentiment Analysis: What is Sentiment Analysis

---

Algorithmia

More Posts - Website

Follow Me:

Search

Enter your query here...

Algorithm spotlights

Case studies

Demos

Events

Machine learning

Newsletters

Recipes

Algorithmia

AI in every application.