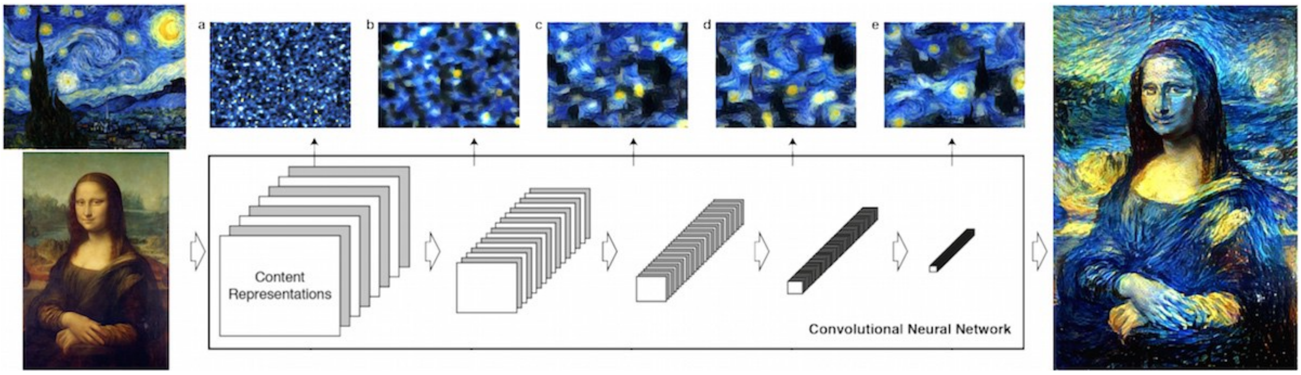
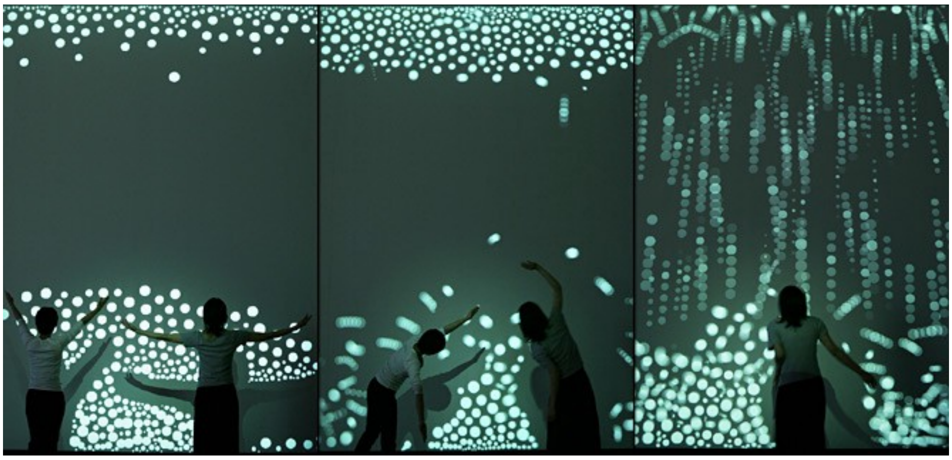


Machine learning for artists

This spring I will be teaching a course at NYU's Interactive Telecommunications Program (ITP) called “Machine Learning for Artists.” Since the subject is fairly uncommon outside of the realm of scientific research, I thought it would be helpful to outline my motivations for offering this class.

There have been a handful of courses introducing machine learning to students in creative fields, including ones by Heather Dewey-Hagborg, Patrick Hebron, and Rebecca Fiebrink. A few classes geared toward people in journalism, STS, and related disciplines exist as well. But in general, the topic is taught mostly to computer scientists and engineers. This disparity should not be taken for granted. To see why, it helps to compare machine learning (ML) to another branch of artificial intelligence which once followed a very similar path, the field of Computer Vision (CV).

. . .



Footfalls (2006, by Golan Levin & Zachary Lieberman) is an interactive audiovisual installation in which the stomping of the visitors' feet creates cascading avalanches of bouncy virtual forms

Machines start to see

The first person to demonstrate artistic potential in CV was Myron Krueger, who began to research its application to virtual and augmented reality, initially as a PhD student in computer science and continuing throughout the 70s and 80s. His groundbreaking work foreshadowed interactive installations and would greatly influence a generation of artists to come.

This embedded content is from a site that does not comply with the Do Not Track (DNT) setting now enabled on your browser.

Please note, if you click through and view it anyway, you may be tracked by the website hosting the embed.

Learn More about Medium's DNT policy

SHOW EMBED

Myron Krueger's Videoplace

Nevertheless, CV remained mostly the domain of computer science through the early 2000s, taught to scientists and engineers who used it primarily for automation, surveillance, and other industrial applications. Around that time, artists first began to experiment with the software trickling out of research labs, including OpenCV, the first major free and open source library for it.

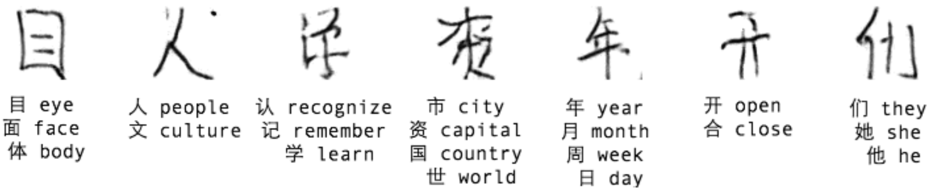
Today, there are numerous open source computer vision libraries, including at least one in every creative coding framework. CV is widely used in games, interactive installations, and many other contexts.

To be sure, there are significant differences between CV and ML. Most applications of the latter require immense amounts of data which are not always readily available. Additionally, the hardware requirements for CV—once prohibitive—are now typical of most consumer-level devices, including mobile, whereas ML, although also rapidly diminishing, remains steep for newer architectures.

On the other hand, there are reasons to suspect that ML could open up even more expeditiously than CV. For one, it is much more general, with applications to audio and natural language processing, data science, journalism, finance, and countless others. It is so general that it is beginning to even subsume CV itself, with a few techniques from deep learning—convolutional neural networks in particular—outperforming classical CV approaches in some vision tasks like object recognition.

Moreover, educational resources are vastly more abundant compared to computer vision at a similar stage. Online courses are plentiful [1][2][3][4][5], as are compilations, visual guides, tutorials, and many others [1][2][3]. Although they do not specifically target artists, they are general enough to be applicable across many disciplines. Similar resources now exist for CV as well, but having so many freely available would have seemed inconceivable when CV was first entering the public arena.

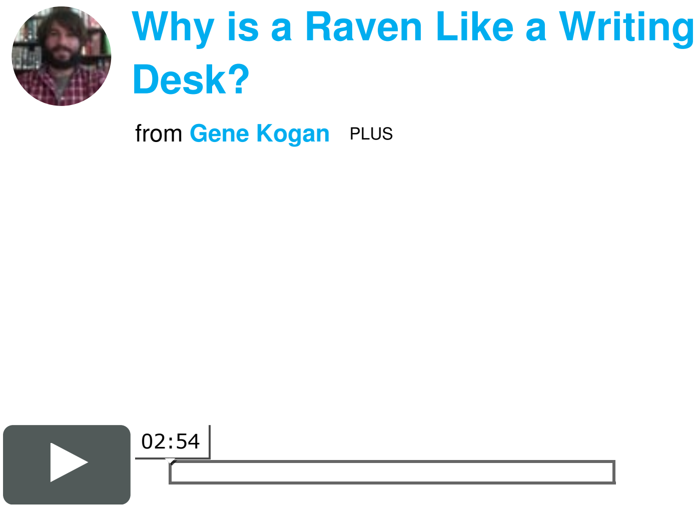
. . .



Excerpt from A Book from the Sky, Dec 2015

Machines start to dream

Like many, I've been excited by the rise of Deep Learning, a branch of ML which has recently achieved state-of-the-art results for a variety of standard tasks, and has shown a penchant for encoding representations of large, disorganized, unlabeled data like raw images, video, audio, and text. Artistic hacks of deep learning software rapidly emerged in 2015, and I participated by producing a number of new works.



Why is a Raven Like a Writing Desk? Stylenet + Alice in Wonderland, Sep 2015

The most recent, excerpted above, was "A Book from the Sky", in which I fed a large database of handwritten Chinese characters to a neural network which learned a generative representation of them, enabling it to "fantasize" fake samples of real characters, and render smooth interpolations among groups of complimentary characters. In September, I made "Why is a Raven Like a Writing Desk?", which applied the style transfer or "stylenet" technique to a scene in *Alice in Wonderland*, as part of a series of animations.

Both were shared by Yann LeCun [1][2], among others in the deep learning research community. My hope is that such public works motivate more researchers to release their software in an accessible way for people outside of academic research to spin off new projects.

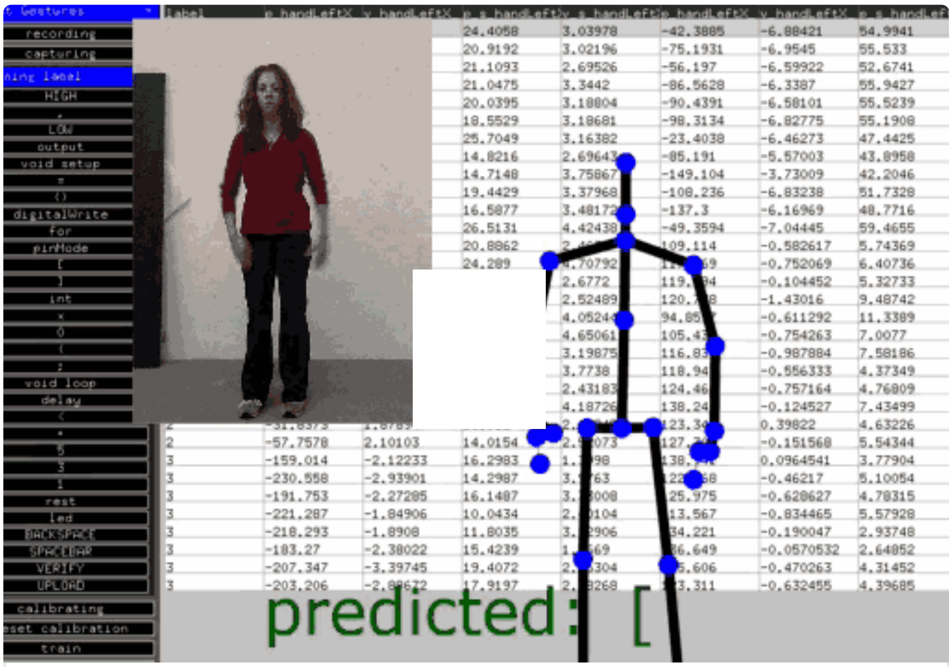
I wrote about this phenomenon of deep learning artistic research in a medium post last month.

From Pixels to Paragraphs

How artistic experiments with deep learning guard us from hype

medium.com

To learn shallow learning deeply or learn deep learning shallowly?



Gene Kogan
@genekogan

Follow

95% accuracy over 30 gesture categories using an SVM. deep learning shmeep learning github.com/genekogan/Blink pic.twitter.com/Ns8Z8DMRmv

10:58 PM - 13 Oct 2015

16 40

SVM FTW

Deep learning poses some pedagogical challenges. First, the prerequisite software can be difficult to install, characterized by numerous and sometimes obscure dependencies, unpredictable runtime errors, and instructions targeted towards people assumed to have a background in computer science or software engineering. For those without one, debugging can be very time-consuming if it can't be taken care of in advance, and distracts from the main educational objectives.

Additionally, the software contains few of the high-level abstractions found in creative coding libraries. The algorithms are expertly hand-crafted to effectively do one narrowly-defined task, and do it very well. Thus, it can be difficult to apply the software creatively in ways that are much different from how the original authors already demonstrated them. Furthermore, the computational expensiveness of most deep neural networks makes virtually all real-time applications impossible. A lack of desirable and large enough datasets, memory restrictions, and various other complications reduce it further.

For those reasons, I don't think it makes sense yet to structure an ML course for artists around the new deep learning libraries. It would be more effective to use more mature and stable tools to demonstrate applications of classification, regression, and clustering. For real-time and performance-based purposes, Wekinator is excellent for this, encapsulating most of the gritty details of ML and providing a convenient interface via open sound control (OSC), letting artists and musicians plug in their favorite tools and observe the essential functions of ML routines from a high-level perspective. For non-real-time tasks like data visualization and inference, scikit-learn is the consensus choice for its ease-of-use, documentation, and accompanying examples.

I am optimistic deep learning will become more practical to teach soon, perhaps even before the end of 2016. By then, students who have thoroughly studied "classical" machine learning will be much more prepared to dive into the deep end.