Artificial Intelligence & Image Generation

George Legrady © 2022
Experimental Visualization Lab
Media Arts & Technology
University of California, Santa Barbara

May 5 **Introduction to the Topic**
Weihao will present on Convolutional Neural-Networks (CNN)

May 10 **Review of Conferences, artistic applications, cultural critique of**

May 12 **Fabian Offert** guest lecture on CLIP and Dalle-E 2 from OpenAI

Neural Texture Synthesis demo (bring photos)

May 14 **Examples of DeepFakes & Social Implications**

Deep Dream, Alex Mordvintsev (2015)

STEVEN LEVY BACKCHANNEL 12.11.2015 12:00 AM

Inside Deep Dreams: How Google Made Its Computers Go Crazy

Why the neural net project creating wild visions has meaning for art, science, philosophy — and our view of reality

Inceptionism: Going Deeper into Neural Networks

Wednesday, June 17, 2015

Posted by Alexander Mordvintsev, Software Engineer, Christopher Olah, Software Engineering Intern and Mike Tyka, Software Engineer

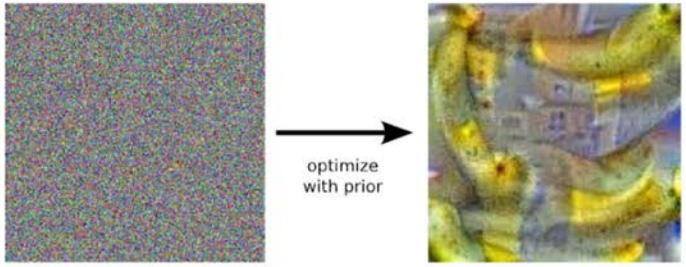
Update - 13/07/2015

Artificial Neural Networks have spurred remarkable recent progress in image classification and speech recognition. But even though these are very useful tools based on well-known mathematical methods, we actually understand surprisingly little of why certain models work and others don't. So let's take a look at some simple techniques for peeking inside these networks.

We train an artificial neural network by showing it millions of training examples and gradually adjusting the network parameters until it gives the classifications we want. The network typically consists of 10-30 stacked layers of artificial neurons. Each image is fed into the input layer, which then talks to the next layer, until eventually the "output" layer is reached. The network's "answer" comes from this final output layer.

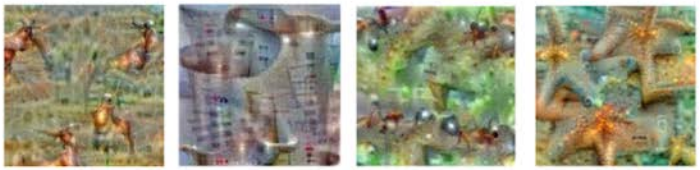
One of the challenges of neural networks is understanding what exactly goes on at each layer. We know that after training, each layer progressively extracts higher and higher-level features of the image, until the final layer essentially makes a decision on what the image shows. For example, the first layer maybe looks for edges or corners. Intermediate layers interpret the basic features to look for overall shapes or components, like a door or a leaf. The final few layers assemble those into complete interpretations—these neurons activate in response to very complex things such as entire buildings or trees.

One way to visualize what goes on is to turn the network upside down and ask it to enhance an input image in such a way as to elicit a particular interpretation. Say you want to know what sort of image would result in "Banana." Start with an image full of random noise, then gradually tweak the image towards what the neural net considers a banana (see related work in [1], [2], [3], [4]). By itself, that doesn't work very well, but it does if we impose a prior constraint that the image should have similar statistics to natural images, such as neighboring pixels needing to be correlated.



So here's one surprise: neural networks that were trained to discriminate between different kinds of images have quite a bit of the information needed to
generate

images too. Check out some more examples across different classes:

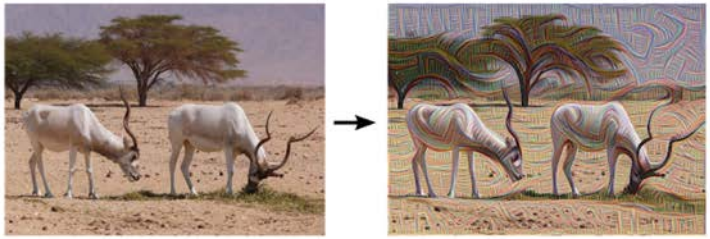


Why is this important? Well, we train networks by simply showing them many examples of what we want them to learn, hoping they extract the essence of the matter at hand (e.g., a fork needs a handle and 2-4 tines), and learn to ignore what doesn't matter (a fork can be any shape, size, color or orientation). But how do you check that the network has correctly learned the right features? It can help to visualize the network's representation of a fork.

Indeed, in some cases, this reveals that the neural net isn't quite looking for the thing we thought it was. For example, here's what one neural net we designed thought dumbbells looked like:

There are dumbbells in there alright, but it seems no picture of a dumbbell is complete without a muscular weightlifter there to lift them. In this case, the network failed to completely distill the essence of a dumbbell. Maybe it's never been shown a dumbbell without an arm holding it. Visualization can help us correct these kinds of training mishaps.

Instead of exactly prescribing which feature we want the network to amplify, we can also let the network make that decision. In this case we simply feed the network an arbitrary image or photo and let the network analyze the picture. We then pick a layer and ask the network to enhance whatever it detected. Each layer of the network deals with features at a different level of abstraction, so the complexity of features we generate depends on which layer we choose to enhance. For example, lower layers tend to produce strokes or simple ornament-like patterns, because those layers are sensitive to basic features such as edges and their orientations.



Left: Original photo by Zachi Evenor. Right: processed by Günther Noack, Software Engineer

Convolution Neural Networks

Convolutional neural networks (CNNs or convnets for short) – are at the heart of **deep learning**, emerging in recent years as the most prominent strain of neural networks in research.

They have revolutionized computer vision, achieving state-of-the-art results in many fundamental tasks. Diverse applications:

- **detecting and labeling objects**, locations, and people in images
- **converting speech into text** and synthesizing audio of natural sounds
- **describing images and videos** with natural language
- **tracking roads and navigating around obstacles** in autonomous vehicles
- analyzing video screens to **guide autonomous agents** playing them
- “hallucinating” images, sounds, and text with **generative models**

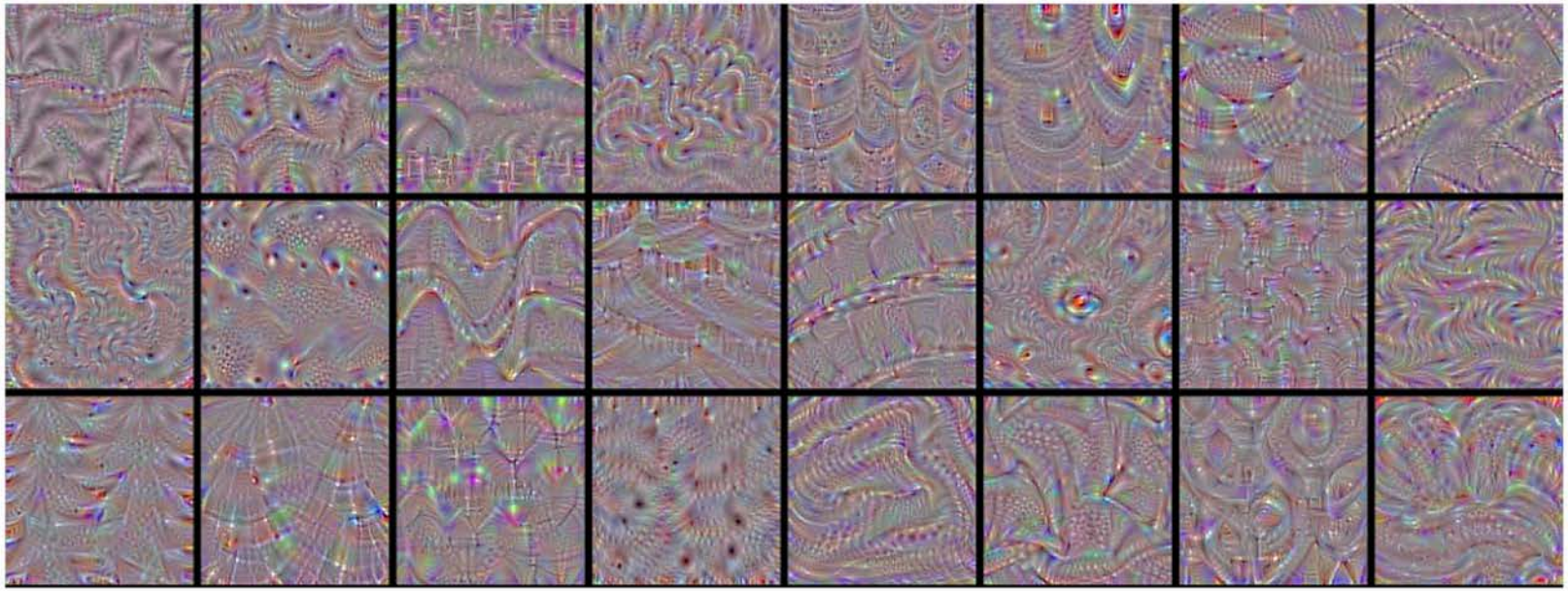
How convolutional neural networks see the world

Note: this post was originally written in January 2016. It is now very outdated. Please see this example of how to visualize convnet filters for an up-to-date alternative, or check out chapter 9 of my book "Deep Learning with Python (2nd edition)".

Sat 30 January 2016
By Francois Chollet
In Demo.

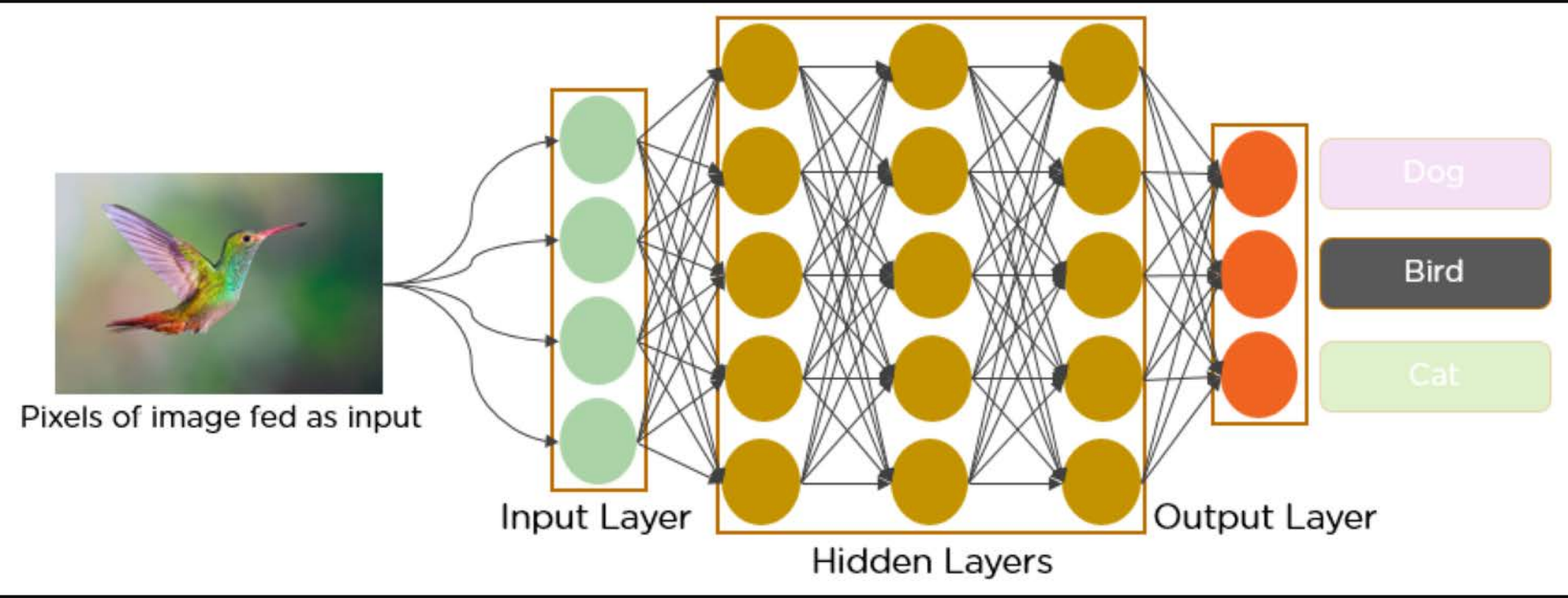
An exploration of convnet filters with Keras

In this post, we take a look at what deep convolutional neural networks (convnets) really learn, and how they understand the images we feed them. We will use Keras to visualize inputs that maximize the activation of the filters in different layers of the VGG16 architecture, trained on ImageNet. All of the code used in this post can be found on Github.



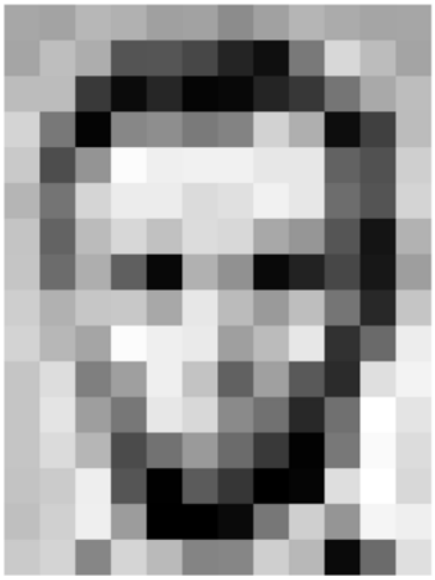
VGG16 (also called OxfordNet) is a convolutional neural network architecture named after the Visual Geometry Group from Oxford, who developed it. It was used to win the ILSVR (ImageNet) competition in 2014. To this day is it still considered to be an excellent vision model, although it has been somewhat outperformed by more revent advances such as Inception and ResNet.

Introduction to Convolutional Neural Networks (CNN), Manav Mandal (2021)



https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

Digital image *made up of pixels* is a multi-dimensional data structure



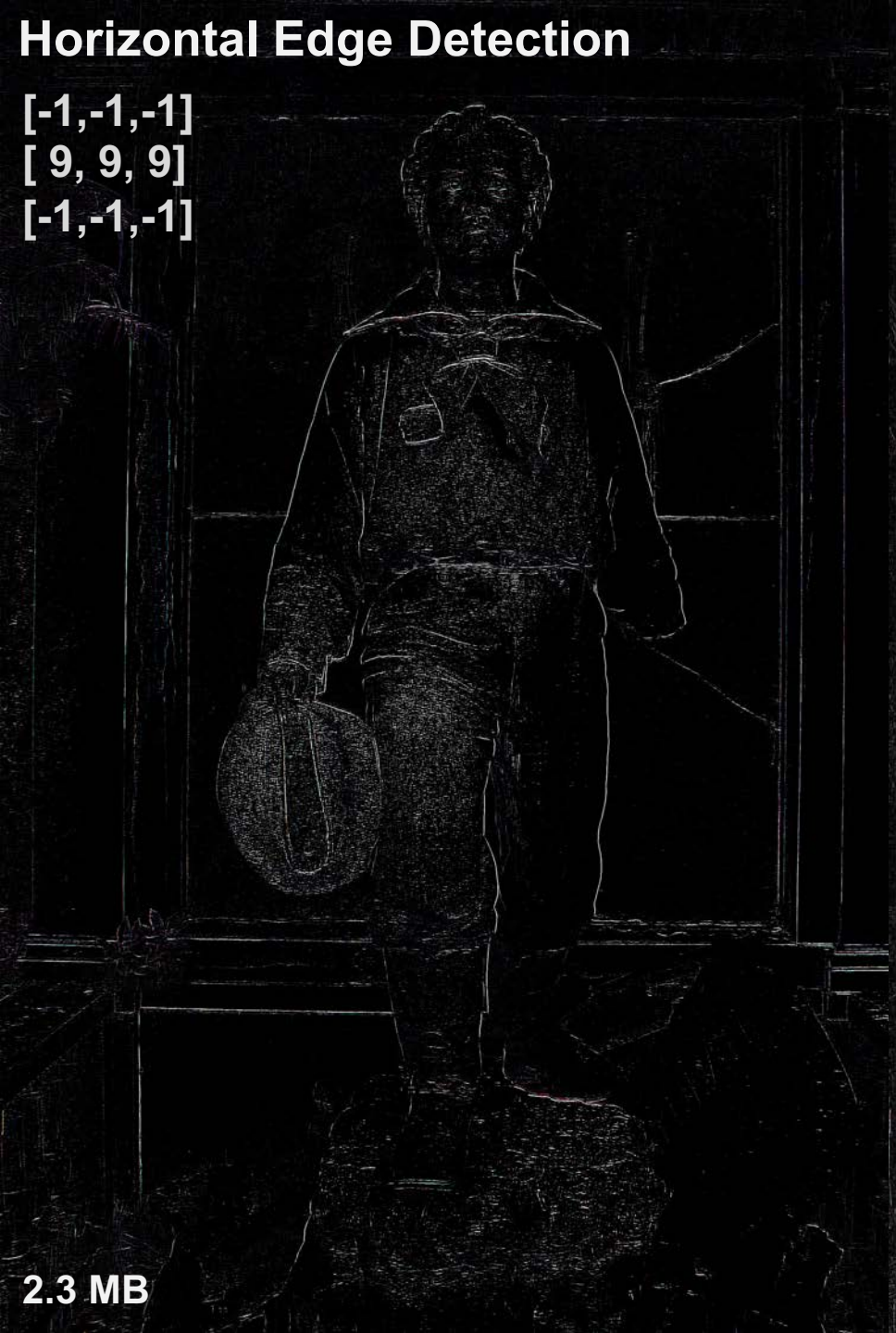
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	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

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

- Pixel ***Horizontal*** location
- Pixel ***Vertical*** location
- Pixel ***Red*** color value
- Pixel ***Green*** color value
- Pixel ***Blue*** color value
- Pixel ***Alpha*** (transparency) value
- The whole image has a ***BitDepth*** resolution (2bit, 16bit, etc.)

Horizontal Edge Detection

[-1,-1,-1]
[9, 9, 9]
[-1,-1,-1]



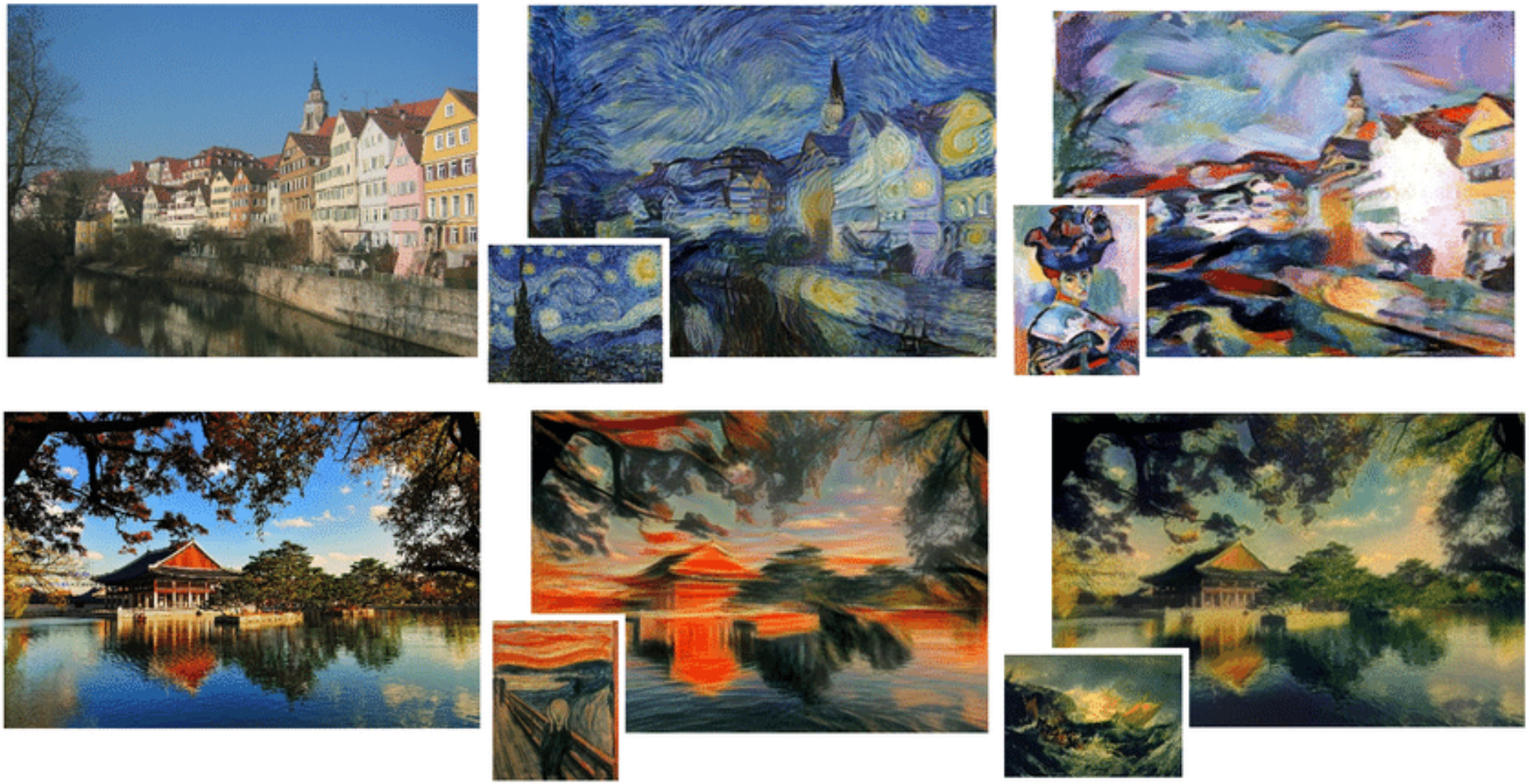
2.3 MB

Vertical Edge Detection

[-1,9,-1]
[-1,9,-1]
[-1,9,-1]

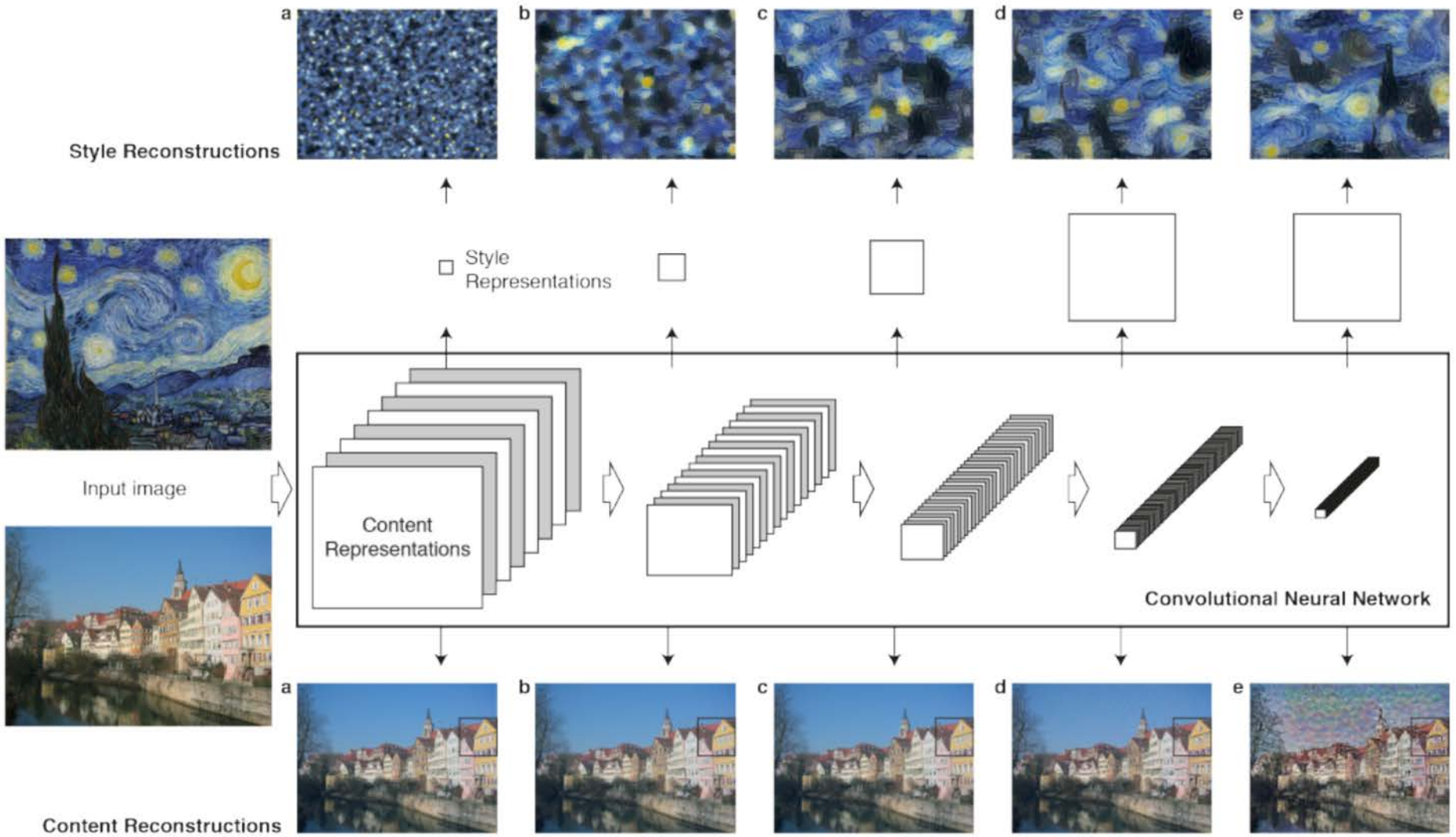
4.4 MB

“A Neural Algorithm of Artistic Style”, (Style Transfer), Leon Gatys (2015)



https://arxiv.org/pdf/1508.06576.pdf

“A Neural Algorithm of Artistic Style”, (Style Transfer), Leon Gatys (2015)

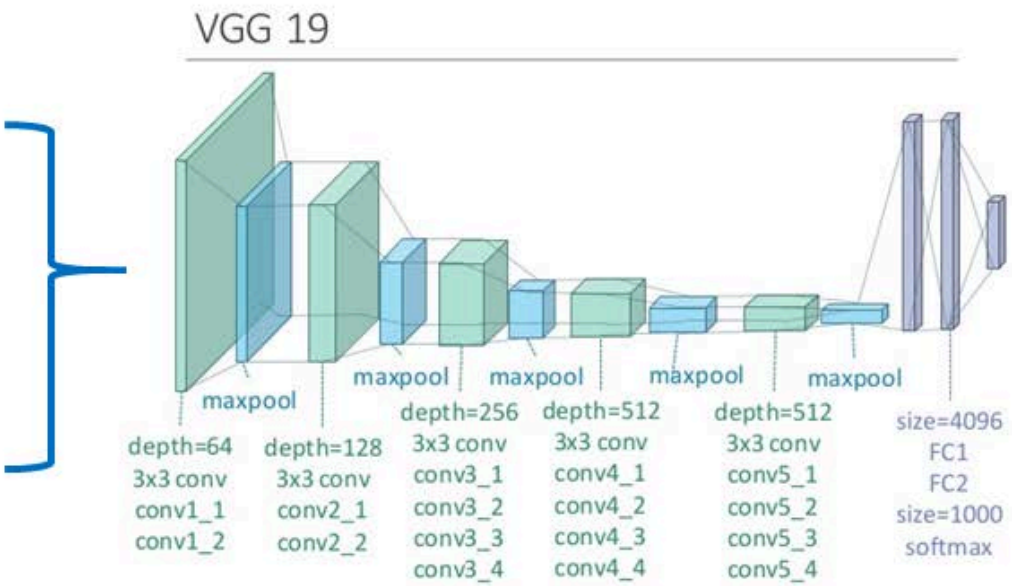


https://github.com/amenglong/art_generation_cnn

VGG-19 (19 layers of convolution calculations)

Barcelona city

Impressionist style painting



Barcelona painting with impressionist style

https://github.com/amenglong/art_generation_cnn

Very Deep Convolutional Networks for Large-Scale Visual Recognition

Karen Simonyan and Andrew Zisserman

Overview

Convolutional networks (ConvNets) currently set the state of the art in visual recognition. The aim of this project is to investigate how the ConvNet depth affects their accuracy in the large-scale image recognition setting.

Our main contribution is a rigorous evaluation of networks of increasing depth, which shows that a significant improvement on the prior-art configurations can be achieved by increasing the depth to 16-19 weight layers, which is substantially deeper than what has been used in the prior art. To reduce the number of parameters in such very deep networks, we use very small 3×3 filters in all convolutional layers (the convolution stride is set to 1). Please see our publication for more details.

Results

ImageNet Challenge

The very deep ConvNets were the basis of our ImageNet ILSVRC-2014 submission, where our team (VGG) secured the first and the second places in the localisation and classification tasks respectively. After the competition, we further improved our models, which has lead to the following ImageNet classification results:

Model	top-5 classification error on ILSVRC-2012 (%)	
	validation set	test set
16-layer	7.5%	7.4%
19-layer	7.5%	7.3%
model fusion	7.1%	7.0%

Generalisation

Very deep models generalise well to other datasets. A combination of multi-scale convolutional features and a linear SVM matches or outperforms more complex recognition pipelines built around less deep features. Our results on PASCAL VOC and Caltech image classification benchmarks are as follows:

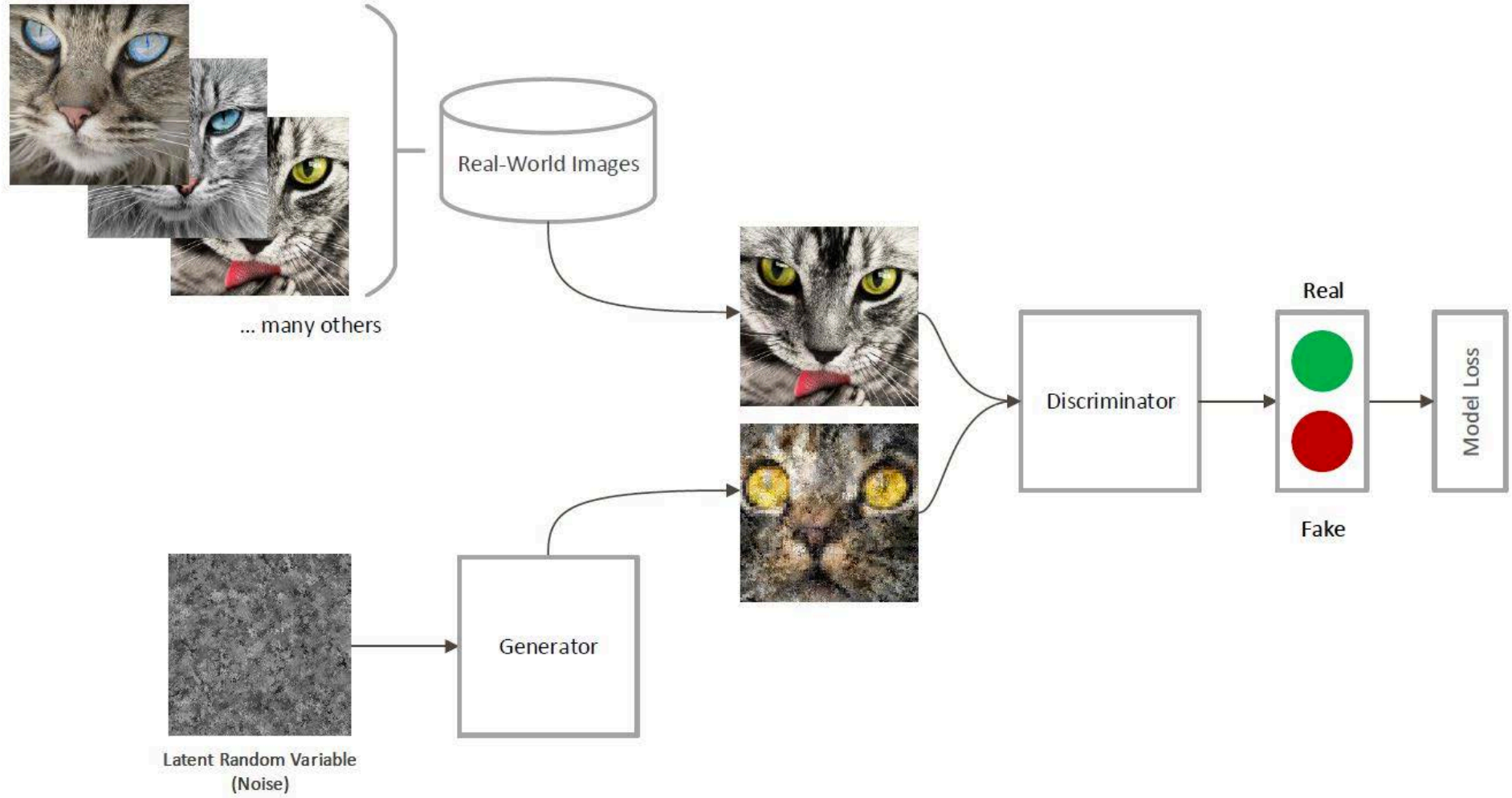
Model	VOC-2007 (mean AP, %)	VOC-2012 (mean AP, %)	Caltech-101 (mean class recall, %)	Caltech-256 (mean class recall, %)
16-layer	89.3	89.0	91.8±1.0	85.0±0.2
19-layer	89.3	89.0	92.3±0.5	85.1±0.3
model fusion	89.7	89.3	92.7±0.5	86.2±0.3

https://www.robots.ox.ac.uk/~vgg/research/very_deep//amenglong/art_generation_cnn

StyleGans: Generate Realistic faces

https://towardsdatascience.com/how-to-train-stylegan-to-generate-realistic-faces-d4afca48e705

Generative Adversarial Network



A GAN network is made up of three components: real-world data, a discriminator, and a generator. The "generative" node of a GAN typically creates text, images, or video. It begins with random data, and generates progressively-better samples, to try and trick the discriminator into believing that the sample is real-world data. The generator and discriminator are two discrete networks competing against each other. Of these, the discriminator network is trained using true, real-world, data. This component's job is to answer the question "Is this real or manufactured?".

https://financeandriskblog.accenture.com/risk/how-generative-adversarial-networks-can-impact-banking?c=acn_glb_financeandriskblinkedinelevate_11010921&n=smc_0819

Xavier Snelgrove

Twitter Github Chronology/CV

I design algorithms to understand the world.

My go-to metaphor is spatial.

I'm a partner at Probably Studio where we work primarily with computer vision techniques in diverse areas such as biomedical imaging and creative tools.

I'm also Creative Technologist in Residence at the BMO Lab in Creative Research in the Arts, Performance, Emerging Technologies and AI, where we are supporting interdisciplinary collaborations with emerging technology.

Previously, I worked on uncertainty modelling and the explainability of AI at Element AI. Still earlier I cofounded Whirlscape. We built Dango, using neural networks to embed emoji in 1000 dimensional semantic space. We built Minuum, searching for words in 10 dimensional keyboard space.

I love to explore the role of computation in creative work. If "*We make our tools and our tools make us*", so building computational models of reality causes us to experience it differently. For instance, I built a toy to simulate the refraction of light through curved surfaces. Now I notice these patterns everywhere.

Lately I've been using neural networks to create images, and I'm newly sensitive to the textures of the world.

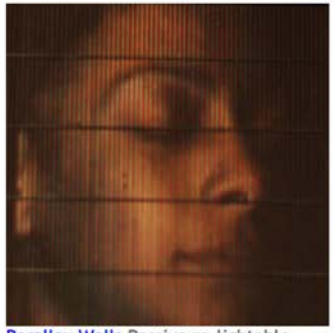
In this spirit, I regularly give talks, teach workshops, organize art galleries at major computer vision conferences and organize the quasi-annual GenArtHackParty, where we teach people how to build generative art, and have a party to show it off. Many past winners had never programmed before, which is a point of pride.

Some Projects

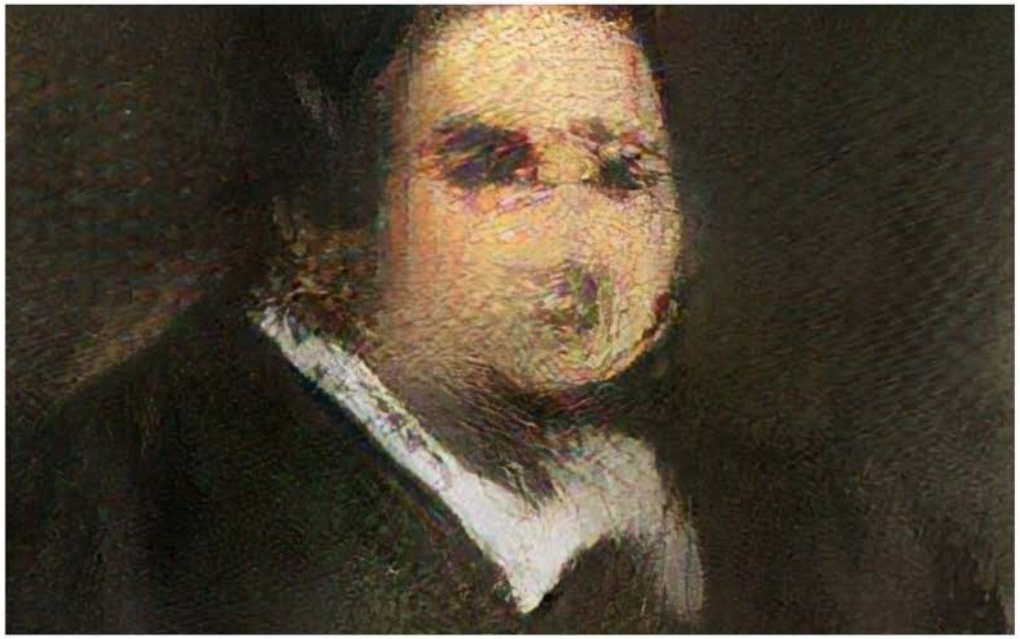
This is incomplete and relatively infrequently updated, some work comes out via conferences, other gets posted with minimal documentation to my Twitter. Some work is also documented on my Google Scholar page.

Multi-Scale Neural Texture Synthesis
High-resolution synthesis!
2017

Studies Visual Studies
2014



Parallax Walls Passive re-lightable display technology (Disney Research)
2013



Is artificial intelligence set to become art's next medium?

AI artwork sells for $432,500 — nearly 45 times its high estimate — as Christie's becomes the first auction house to offer a work of art created by an algorithm

12 December 2018

PHOTOGRAPHS & PRINTS | AUCTION PREVIEW

Main image:

Portrait of Edmond Belamy (detail) created by GAN (Generative Adversarial Network), which will be offered at Christie's on 23-25 October. Image © Obvious

Highlighted sale

The portrait in its gilt frame depicts a portly gentleman, possibly French and — to judge by his dark frockcoat and plain white collar — a man of the church. The work appears unfinished: the facial features are somewhat indistinct and there are blank areas of canvas. Oddly, the whole composition is displaced slightly to the north-west. A label on the wall states that the sitter is a man named Edmond Belamy, but the giveaway clue as to the origins of the work is the artist's signature at the bottom right. In cursive Gallic script it reads:

image-net.org

IMAGENET

14,197,122 images, 21841 synsets indexed

Explore Download Challenges Publications Updates About

Not logged in. Login | Signup

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.

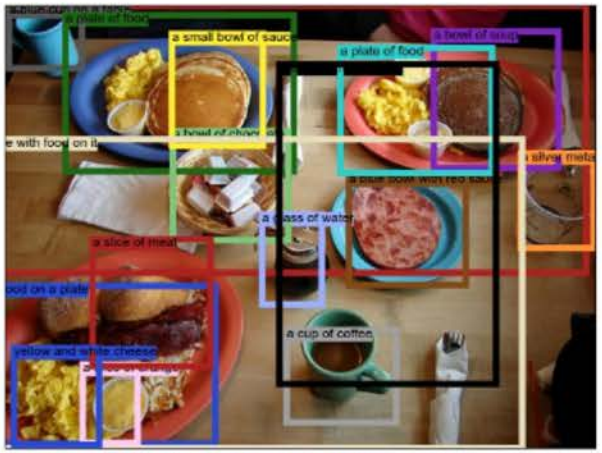
What do these images have in common? *Find out!*

Research updates on improving ImageNet data

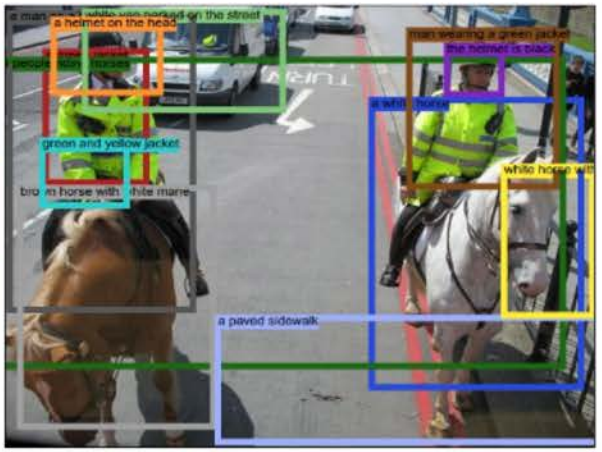
Example Results: Dense Captioning

Example predictions from the model. We slightly cherry picked images in favor of high-resolution, rich scenes and no toilets.
Browse the **full results** on our interactive predictions visualizer page (30MB) (visualizer code also included on Github).

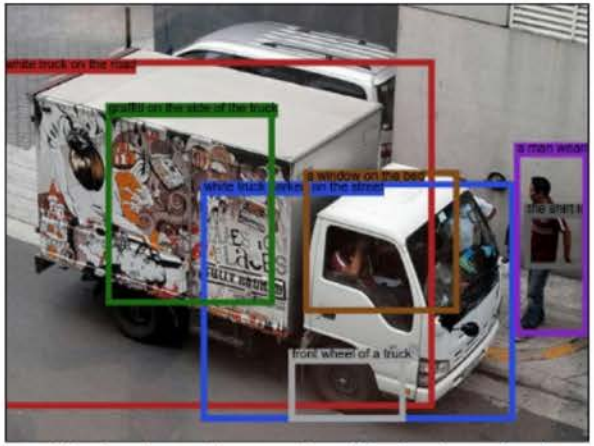
bus parked on the street. a city street scene. front windshield of a bus. man walking on sidewalk. a silver car parked on the street. a city scene. a green traffic light. a building in the background. the bus has a number. a large building. a brick building. red brick building with windows. a blue sign with a white arrow. white lines on the road.



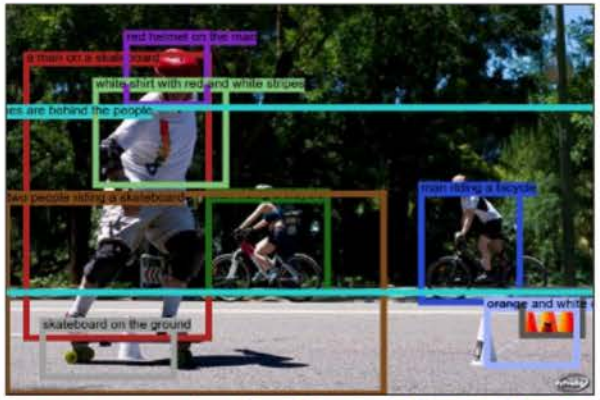
a plate of food. food on a plate. a blue cup on a table. a plate of food. a blue bowl with red sauce. a bowl of soup. a cup of coffee. a bowl of chocolate. a glass of water. a plate of food. a silver metal container. a small bowl of sauce. table with food on it. a slice of orange. a table with food on it. a slice of meat. yellow and white cheese.



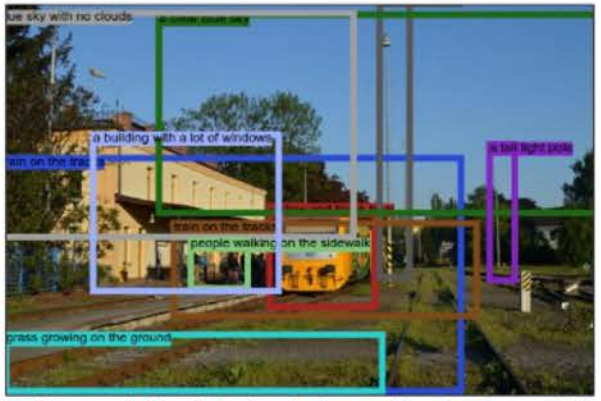
a green jacket. a white horse. a man on a horse. two people riding horses. man wearing a green jacket. the helmet is black. brown horse with white mane. white van parked on the street. a paved sidewalk. green and yellow jacket. a helmet on the head. white horse with white face.



a white truck on the road. white truck parked on the street. the shirt is red. graffiti on the side of the truck. a window on the bed. a man wearing a black shirt. front wheel of a truck.



a man on a skateboard. man riding a bicycle. orange cone on the ground. man riding a bicycle. two people riding a skateboard. red helmet on the man. skateboard on the ground. white shirt with red and white stripes. orange and white cone. trees are behind the people.



yellow and black train. train on the tracks. a tall light pole. a clear blue sky. train on the tracks. a tall light pole. a blue sky with no clouds. people walking on the sidewalk. a building with a lot of windows. grass growing on the ground.

CLIP: Connecting Text and Images

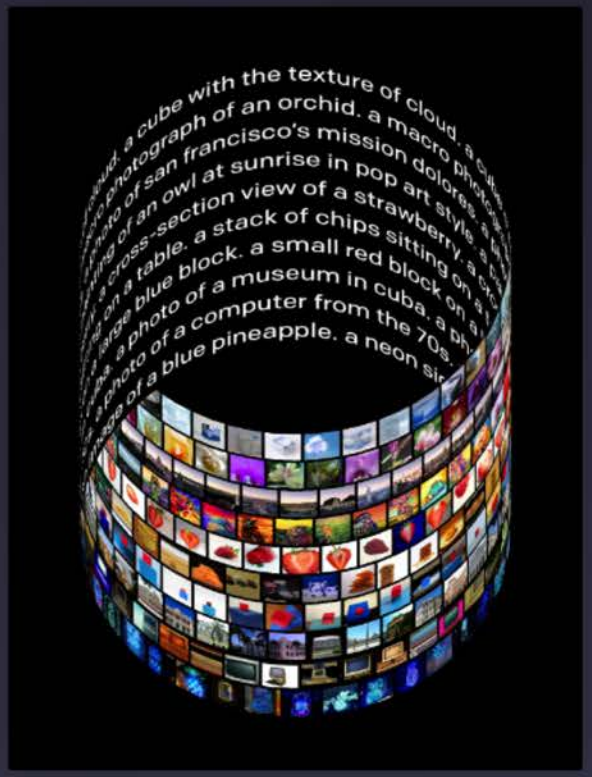
We're introducing a neural network called CLIP which efficiently learns visual concepts from natural language supervision. CLIP can be applied to any visual classification benchmark by simply providing the names of the visual categories to be recognized, similar to the "zero-shot" capabilities of GPT-2 and GPT-3.

January 5, 2021
15 minute read

https://openai.com/blog/clip/

DALL·E: Creating Images from Text

We've trained a neural network called DALL·E that creates images from text captions for a wide range of concepts expressible in natural language.

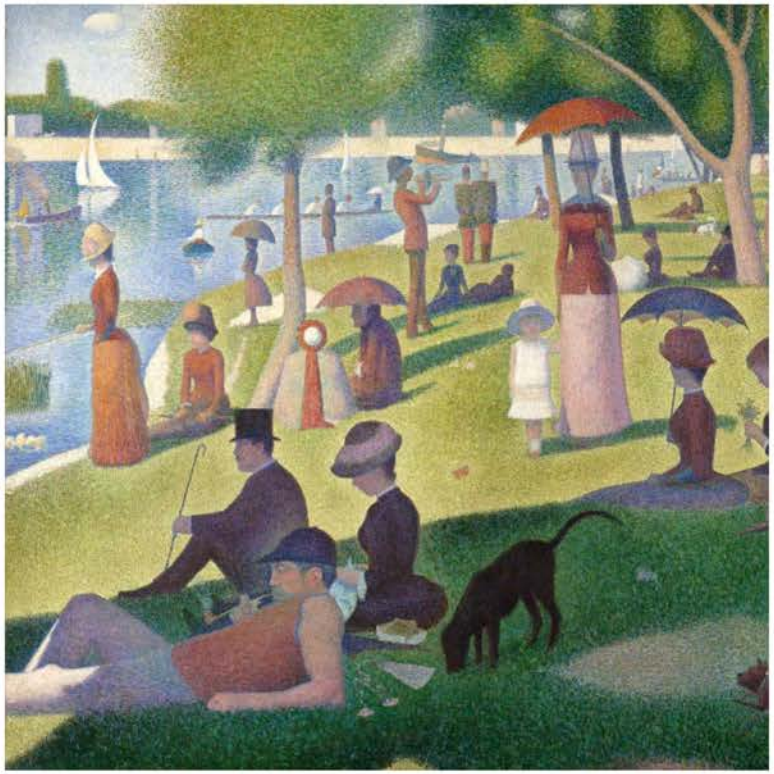


January 5, 2021
27 minute read

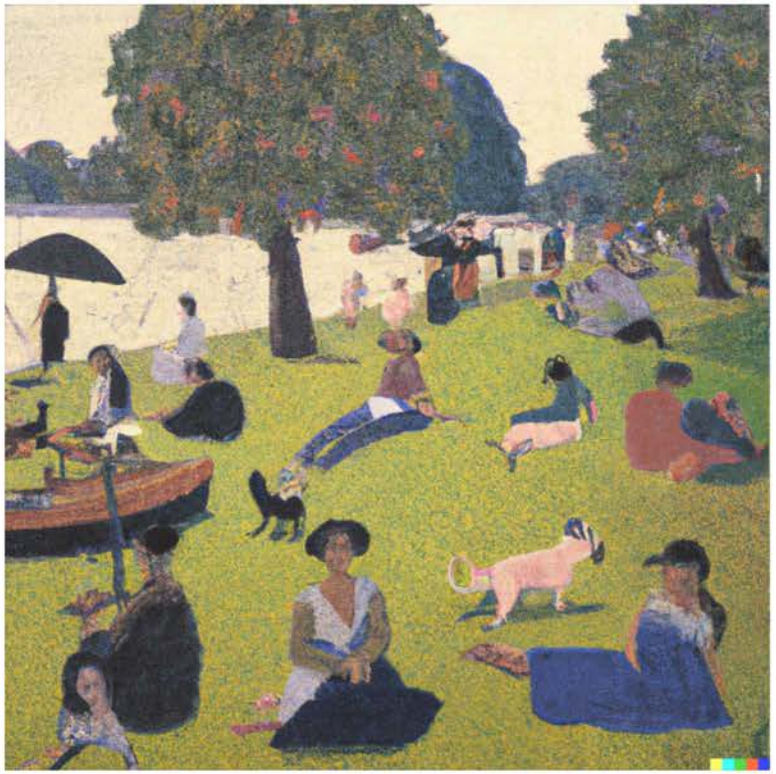
https://openai.com/dall-e-2/

DALL·E 2 can take an image and create different variations of it inspired by the original.

ORIGINAL IMAGE



DALL·E 2 VARIATIONS



https://openai.com/dall-e-2/

Term Definitions

Artificial Intelligence: Computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages (Google Translate)

Machine Learning: Autonomous learning without explicit human guidance. Identifies and extracts patterns from data

Supervised Learning: Maps an output to an input. A process of teaching a model by feeding it input data where the systems knows what to expect in the output.

Unsupervised Learning: Self-learning, iteratively increases its performance. There is no target attribute to compare the results to as in supervised learning.

Convolution: A mathematical operation of two functions that produces a third result

CNN (Convolutional Neural Network): A type of artificial **neural network** used in image recognition and processing that is specifically designed to process pixel data

Deep Learning: A subset of machine-learning with an increased, unsupervised

GANS (Generative Adversarial Networks): A machine learning model where two neural networks compete with each other to produce more accurate results to the training data. A generator produces an outcome and the discriminator evaluates if it matches the input data, functioning as a feedback loop, forcing the generator to continuously increase its performance.

Style Transfer: Two source images influence each other by imposing the style of one image onto the form of the other image (a Van Gogh image is used to make a photo of a street scene look like a van Gogh painting)

Deep Fakes: A person in an existing image or video is replaced with someone else's lineness and behavior.

Artificial Intelligence & Image Creation

George Legrady © 2022
Experimental Visualization Lab
Media Arts & Technology
University of California, Santa Barbara