

Deep Art: Python, deep-learning, art

Robert Arbon (@bertiearbon), 8th March 2019, Bristol Doctoral College
‘Skill Share’



University of
BRISTOL

Itinerary

Overview of course

Deep learning

Python programming

Break

Play time

Jean Golding Institute

ASK JGI: Data science support

GET IN TOUCH

ask-jgi@bristol.ac.uk



UK Reproducibility Network

Visit:

[http://www.dcn.ed.ac.uk/camarades/
ukrn/](http://www.dcn.ed.ac.uk/camarades/ukrn/)

Join local Bristol group, email:
robert.arbon@gmail.com

Twitter: @ukrepro



Course overview

Some questions

What will we be learning today?

Why Python?

Where can I find out more?

What will we be learning today?

Programming concepts, techniques and tools

How machines think about audio and images

Deep learning and convolutional neural networks

Some algorithms for deep art

Why Python?

Easy to learn programming language

Lots of tutorials, courses and example code

Many deep learning frameworks (e.g. Tensorflow) have a Python ‘front end’

Where can I find out more?

Github: [https://github.com/RobertArbon/deep art course](https://github.com/RobertArbon/deep_art_course)

Links to other resources

Leave comments and feedback

Slack: <https://bit.ly/2SGDAmG>

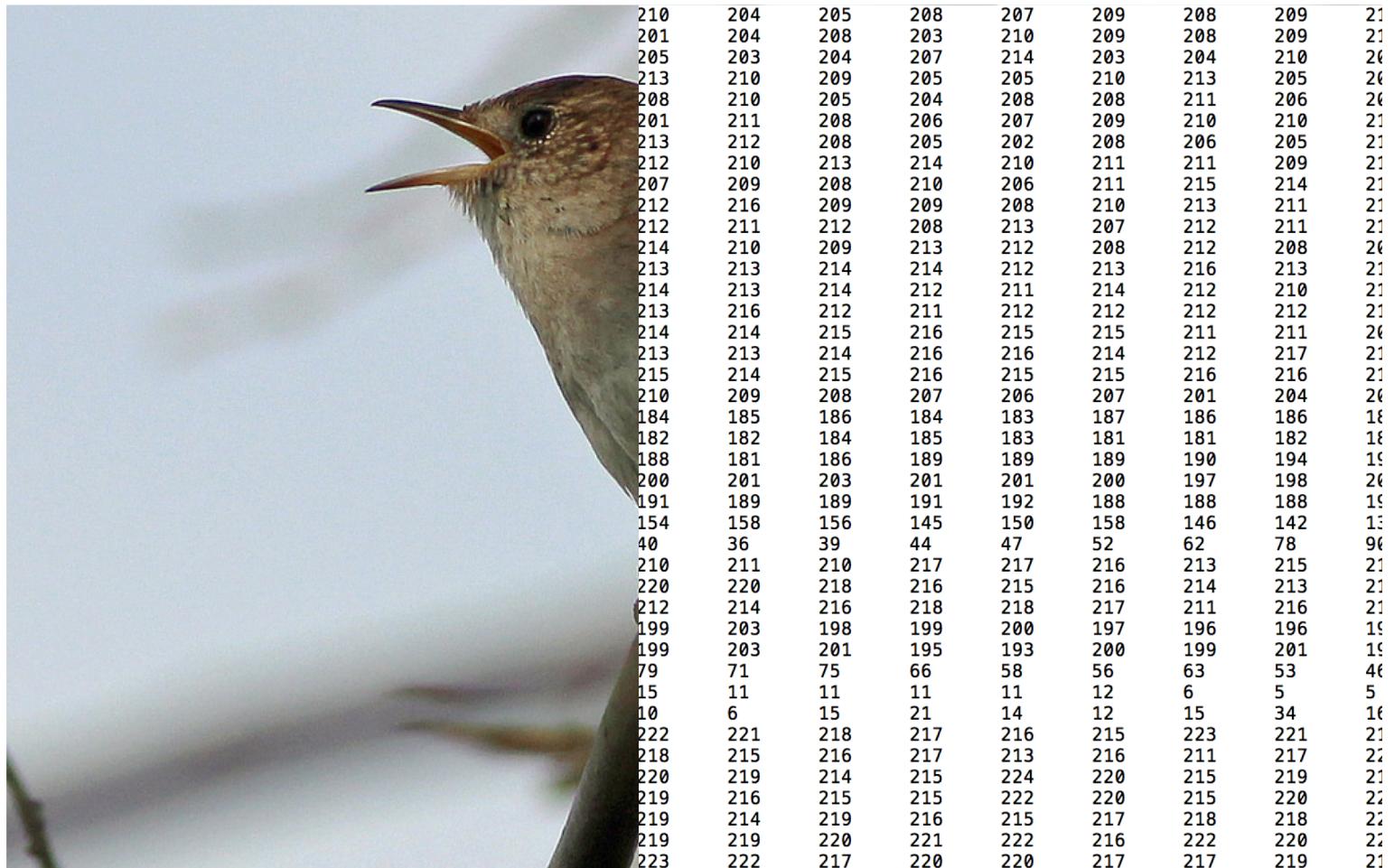
Discuss and share resources

Deep Learning

What are images?

Images are just arrays of numbers between 0 – 255 (or 0 – 1)

Each number is a pixel representing a color and its intensity



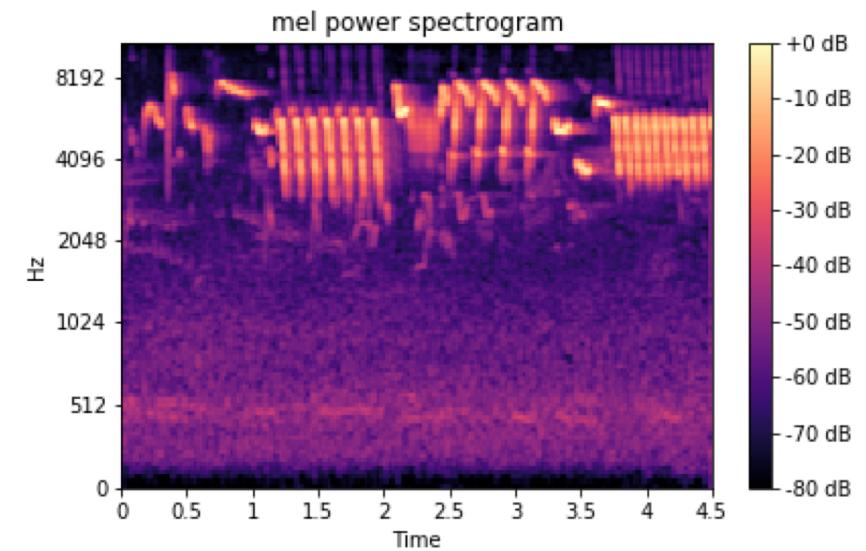
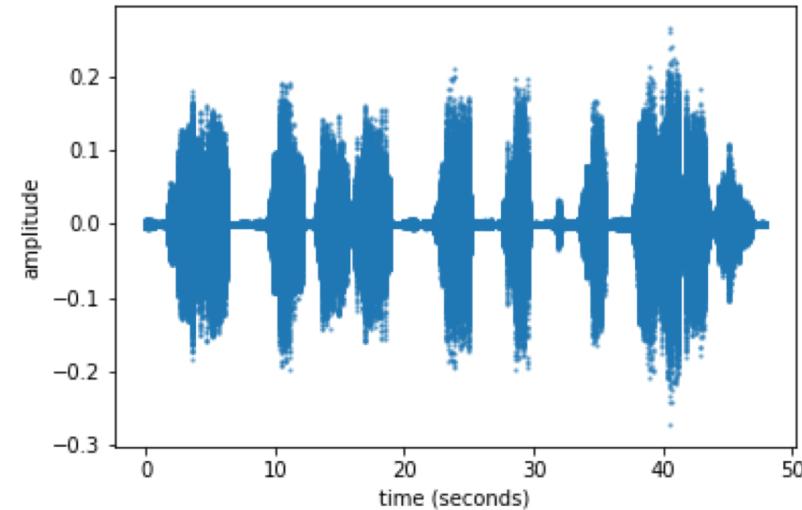
What are sounds?

Array of numbers

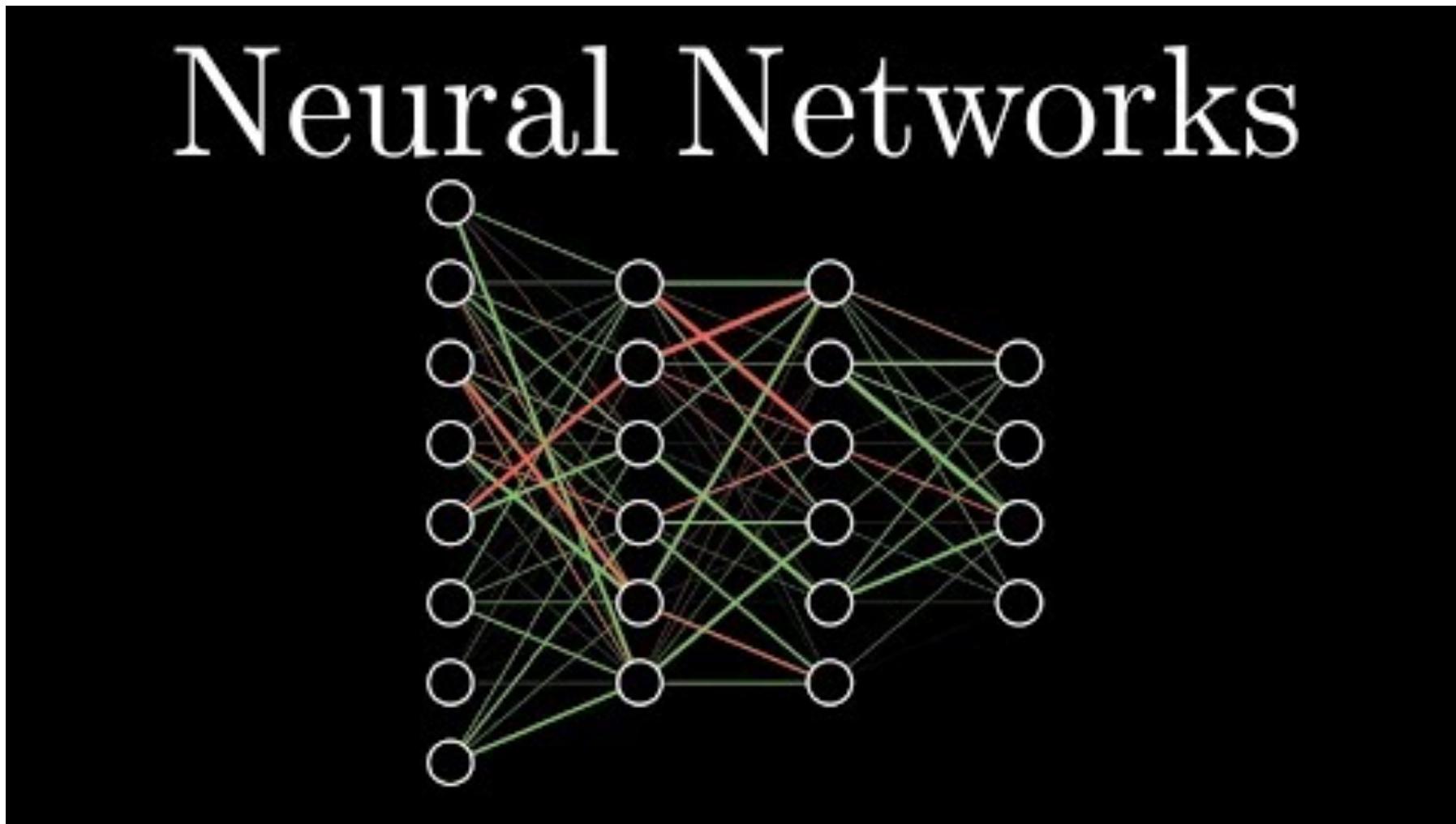
Values → recorded audio amplitude

Recorded 20,500 times a second
(20.5kHz sampling rate)

10 seconds of audio → $20,500 \times 10 =$
205,000 values

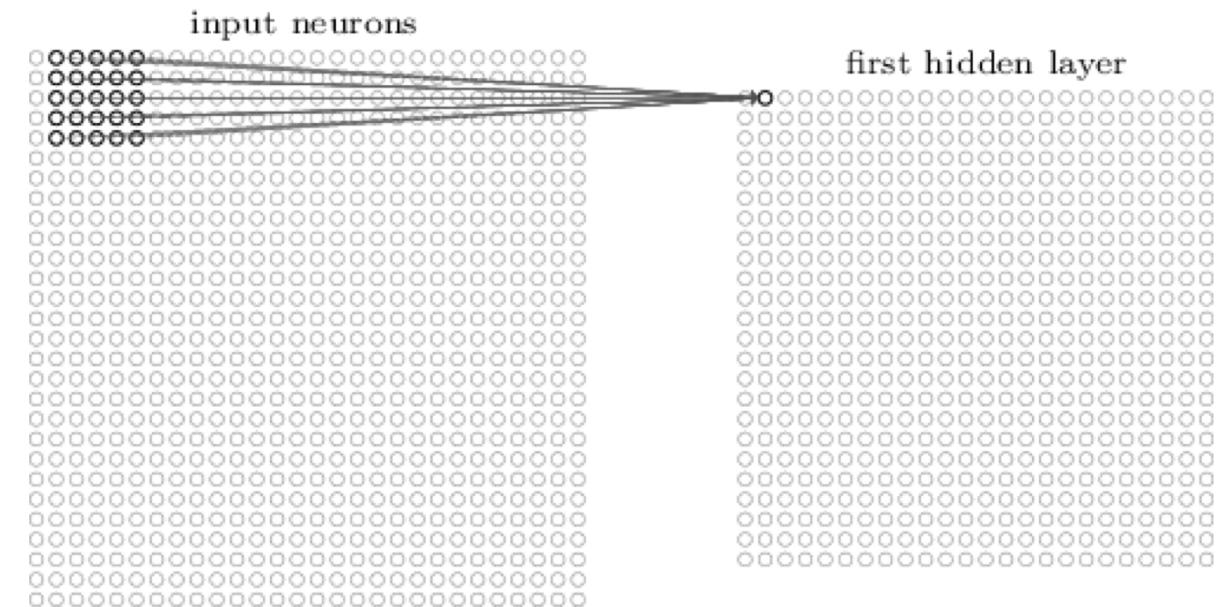
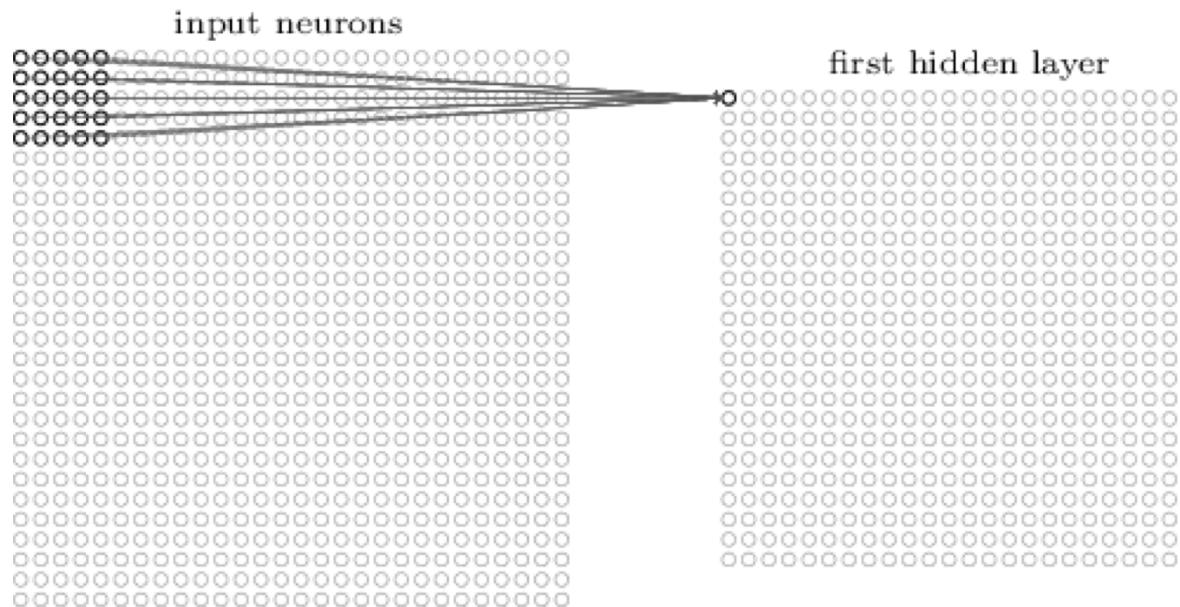


Neural networks



Attribution: <https://www.youtube.com/watch?v=aircAruvnKk>

Convolutional neural networks (CNNs)



Attribution: <http://neuralnetworksanddeeplearning.com/chap6.html>

ImageNet Large-Scale Visual Recognition Challenge 2014

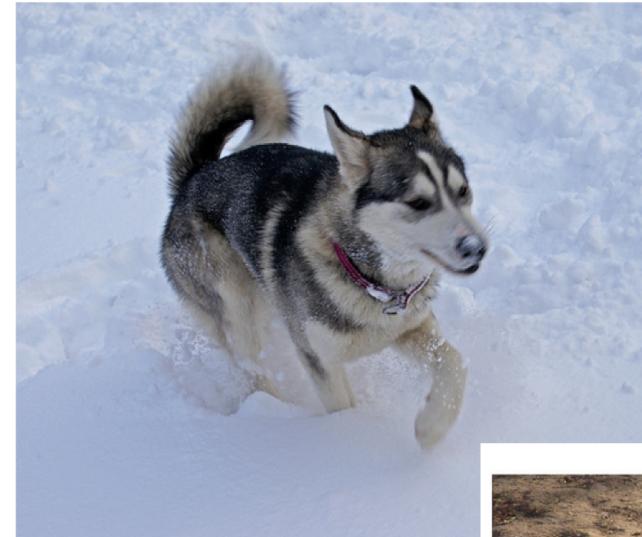
1000 categories

100,000 test images

GoogLeNet (Inception v1): 7% error

Human: approx. 5% error

Attribution: <https://arxiv.org/abs/1409.4842>

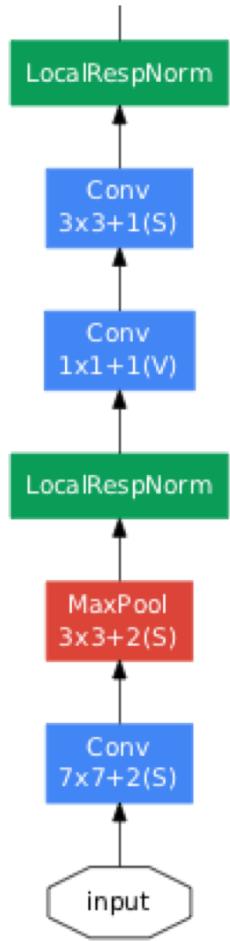


(a) Siberian husky

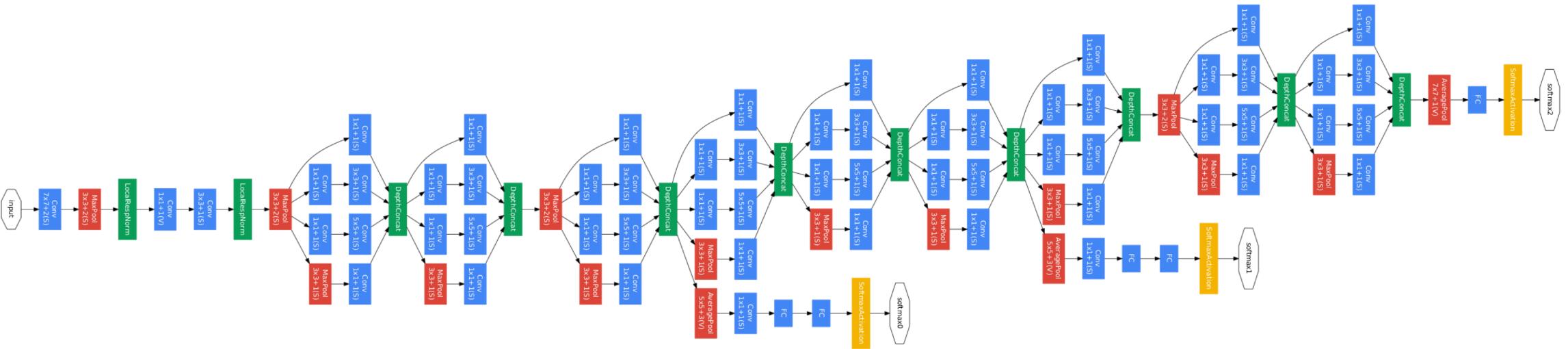


(b) Eskimo dog

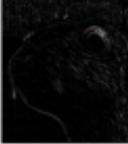
Inception v1

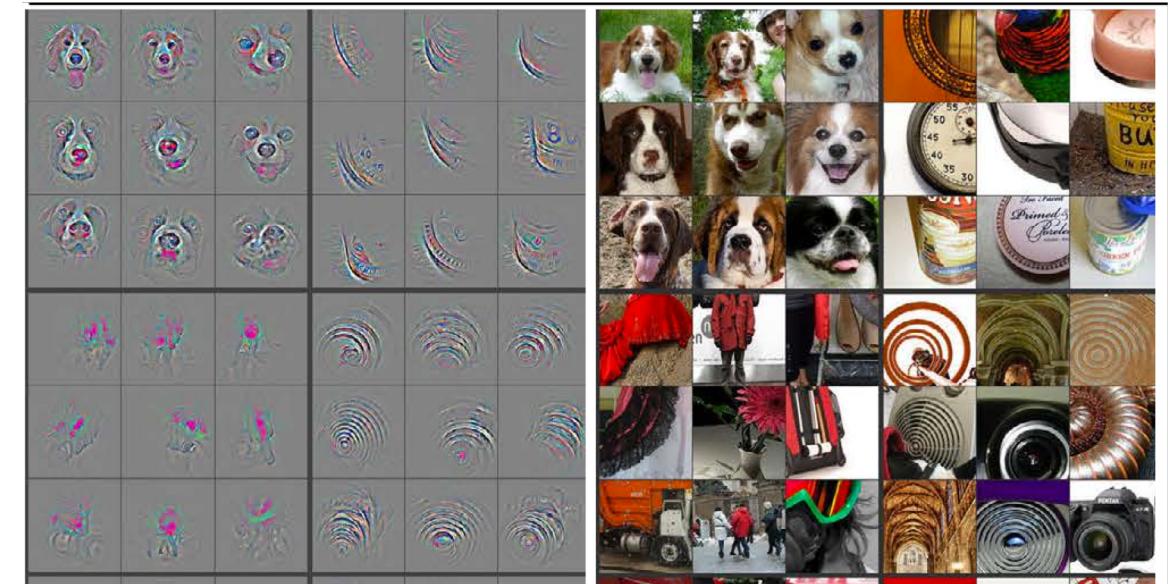


Inception v1



What do CNNs see?

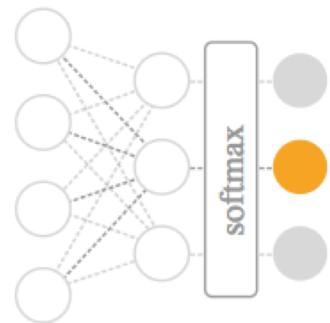
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	



<https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>

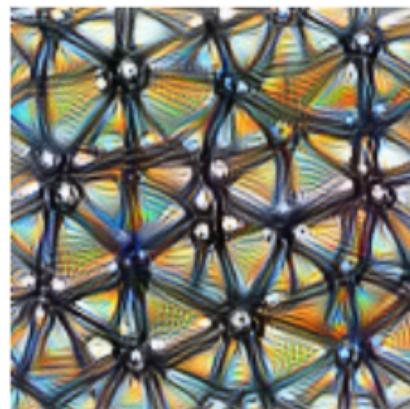
<https://jhui.github.io/2017/03/16/CNN-Convolutional-neural-network/>

What makes CNNs ‘turn on’?



Neuron

```
layer_n[x,y,z]
```



Channel

```
layer_n[:, :, :, z]
```



Layer/DeepDream

```
layer_n[:, :, :, :]2
```



Class Logits

```
pre_softmax[k]
```



Class Probability

```
softmax[k]
```

A neural algorithm of artistic style



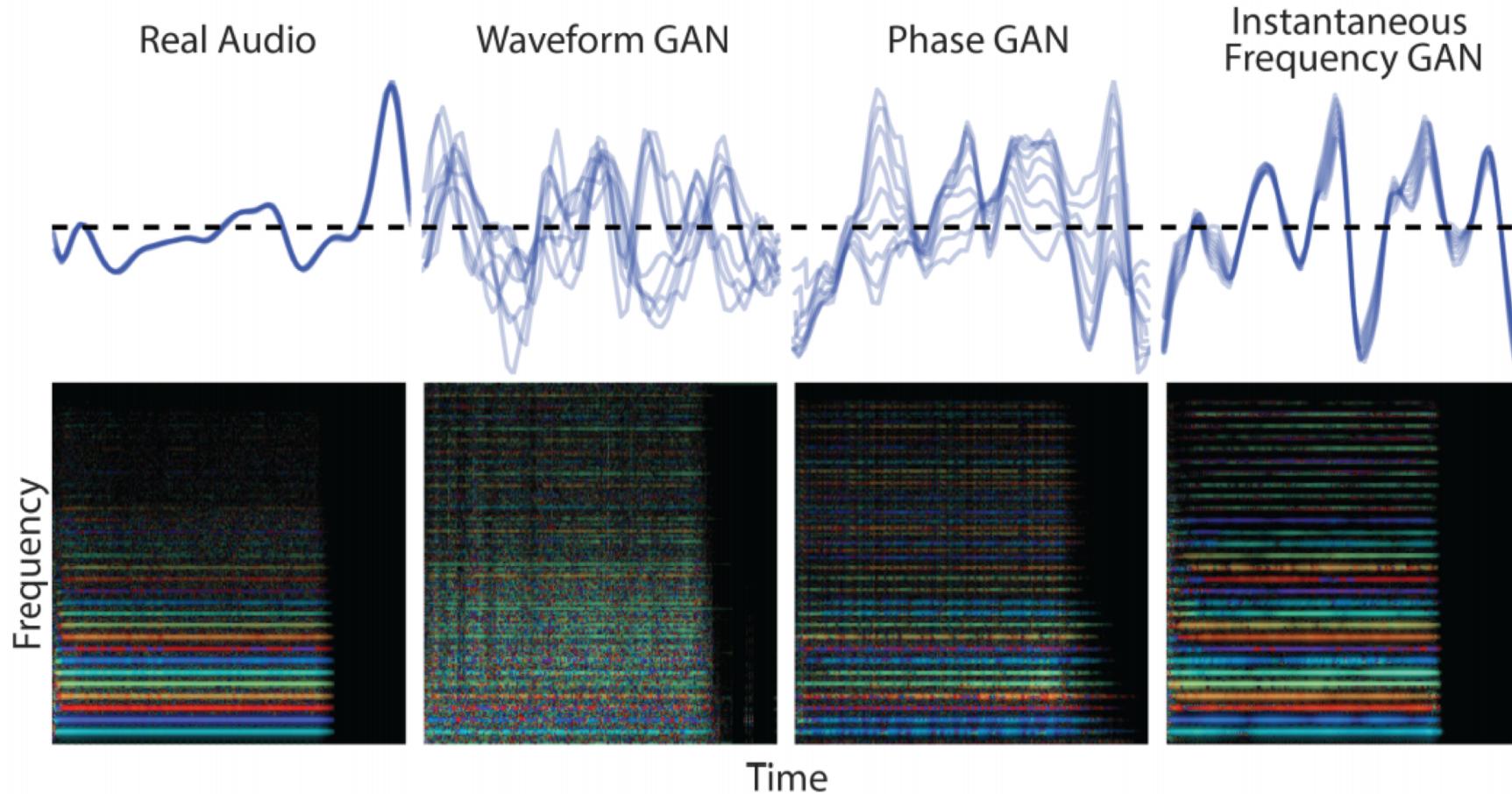
<https://deepoch.io/>

Generative Adversarial Networks (GANs)



<https://arxiv.org/abs/1812.04948>

GANSynth: Adversarial Neural Audio Synthesis



<https://storage.googleapis.com/magentadata/papers/gansynth/index.html#overview>

Introduction to Python Programming

How to learn to program

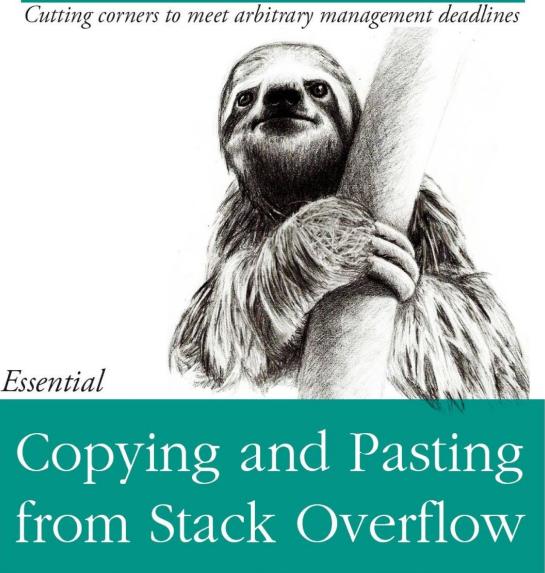
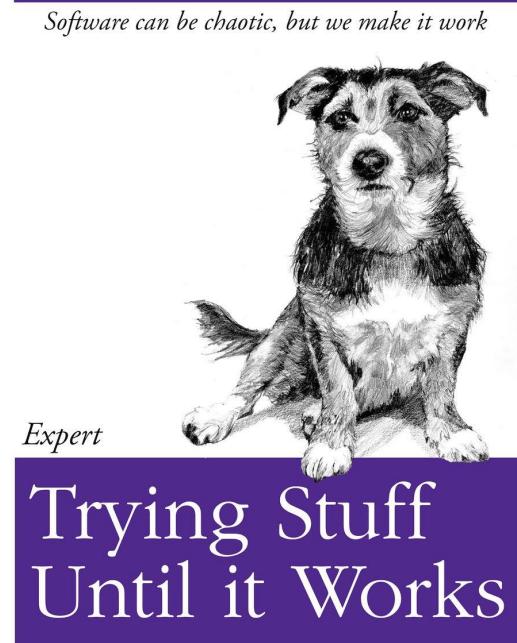
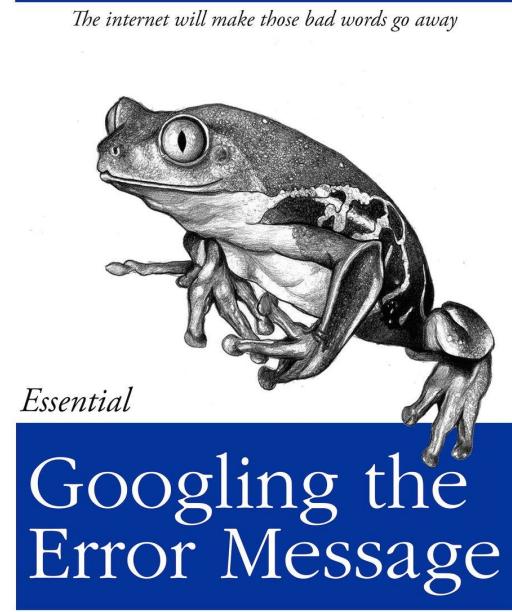
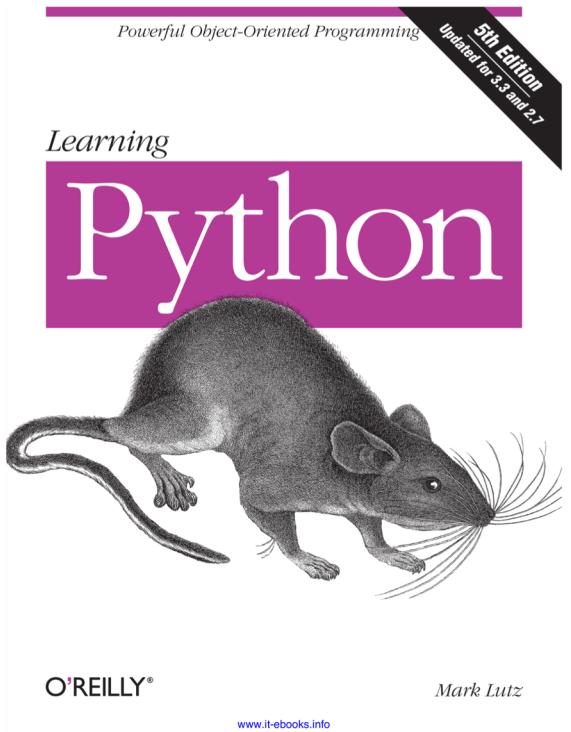
The ‘Hacking’ method:

- Find a compelling reason to learn (“I want to make generative art”)
- Read/watch basic introduction to the language (< 5 hours), then:
 1. Try to complete a small task (“Load and reshape an image”)
 2. Find a (related) method on the web/book
 3. Debug/investigate method
 4. Consult reference to understand why your solution does/doesn’t work
 5. Go to step 1 and increase the complexity of your task (“Add a transparency layer”)

Skills necessary

- Curiosity → Always be asking: “Why have they done that?”
- Creativity → Use imagination to create solutions to problems
- Pattern recognition → See patterns in code and adapt them
- Ability to experiment → Try different approaches to understand the language
- Ability to search for solutions → Learn search functionality on Stack Overflow
- Ability to ask for questions → See resources on the Github page

Essential reading



O'REILLY®

www.it-ebooks.info

Mark Lutz

O RLY?

The Practical Developer
@ThePracticalDev

O RLY?

The Practical Developer
@ThePracticalDev

O'REILLY®
The Practical Developer
@ThePracticalDev

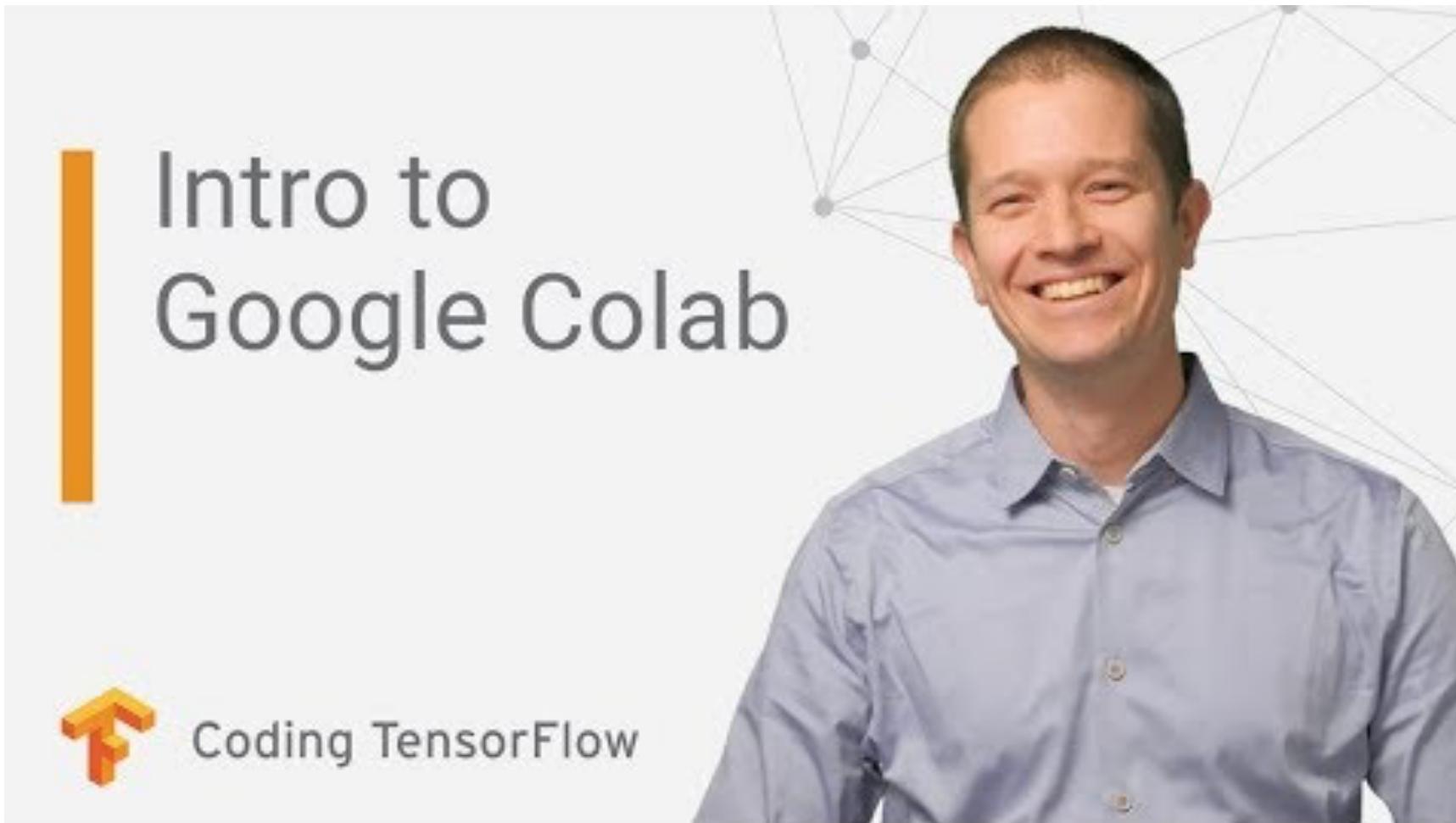
“What have you tried?”

Always be prepared to answer this question when asking for help.

My usual workflow:

1. Get stuck.
2. Write out question on SO/email/slack
3. 1/3 of time realise my mistake. Don't send message. Good times.
4. 1/3 of time write out list of things I've tried. Realise my mistake. Don't send message. Good times.
5. 1/3 of time send message. Get response. Hopefully realise my mistake. Maybe good times.

Your ecosystem: Google Colab



<https://www.youtube.com/watch?v=inN8seMm7UI>

Exercises

Notebooks 1, 2: exercises

Notebook 3: deep dreaming annotated and explained

Notebooks 4 - 7: other notebooks pulled from Colab seedbank for your enjoyment

~Fin~

