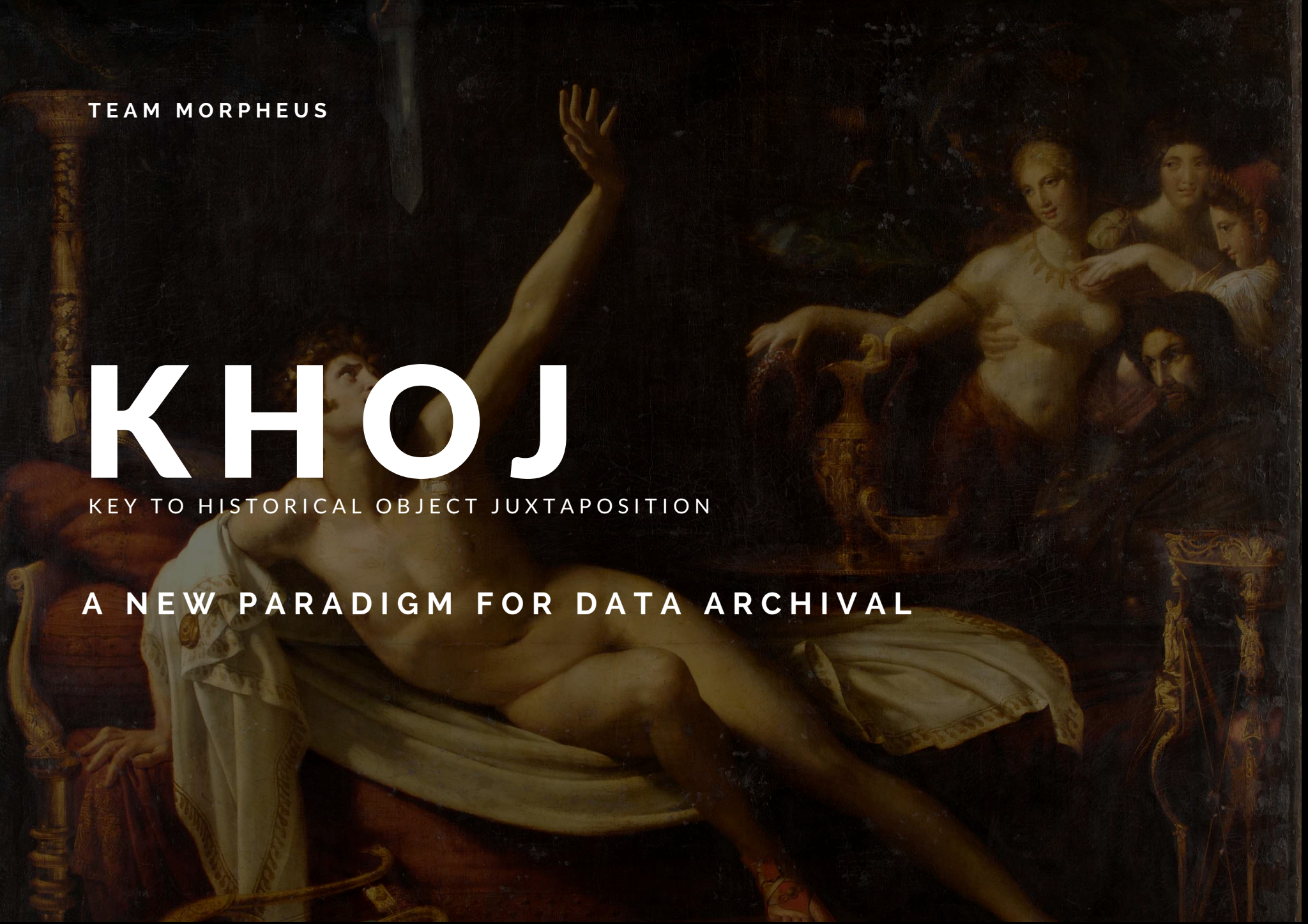


TEAM MORPHEUS

KHOJ

KEY TO HISTORICAL OBJECT JUXTAPOSITION

A NEW PARADIGM FOR DATA ARCHIVAL


WORKING WITH HISTORICAL DATA - AN "INTERESTING" EXPERIENCE

NO

- standard format
- labels or tags
- connections
- easy way to compare

LOTS OF

- errors
- hours spent
- curses hurled at MS Excel


LET'S FIX
THIS*
*IN A HIGHLY TECHNICAL AND COMPLICATED WAY

AUTOMATIC TAGGING

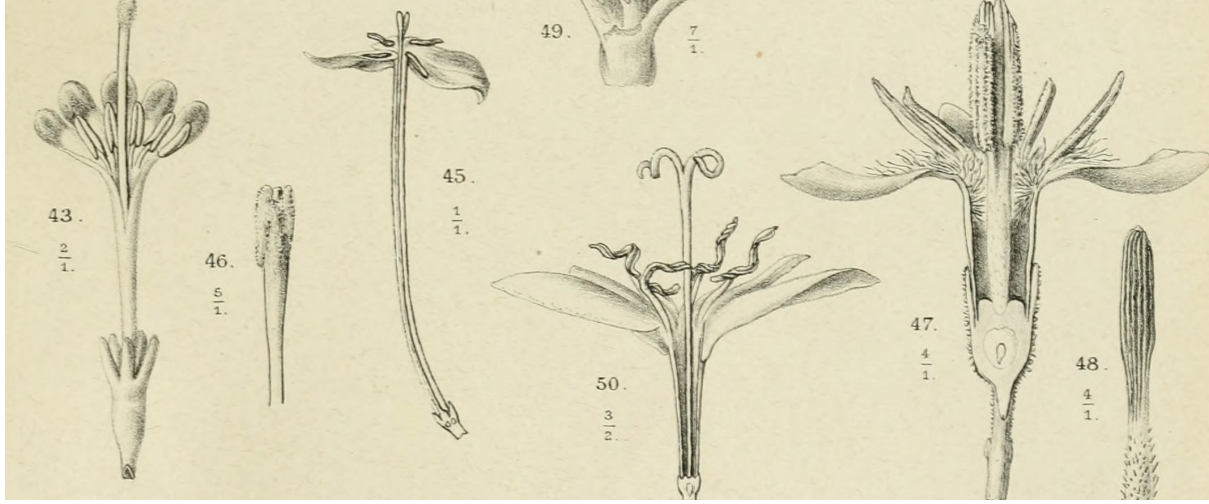
Computer recognises images and assigns text tags to each
e.g. 'terracotta' 'bust' 'coin' 'gold'

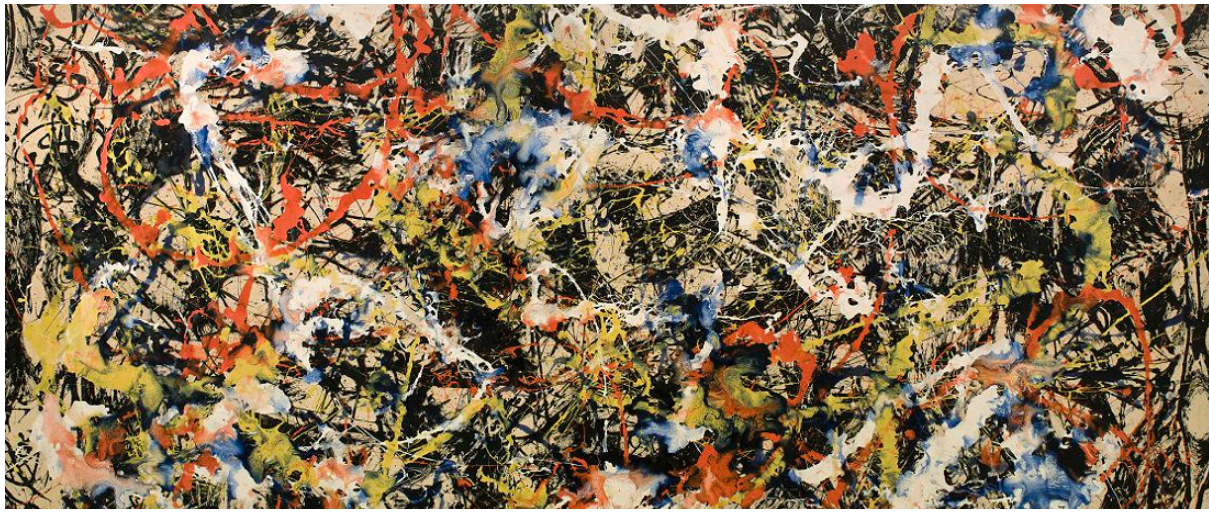
FLEXIBLE FORMAT

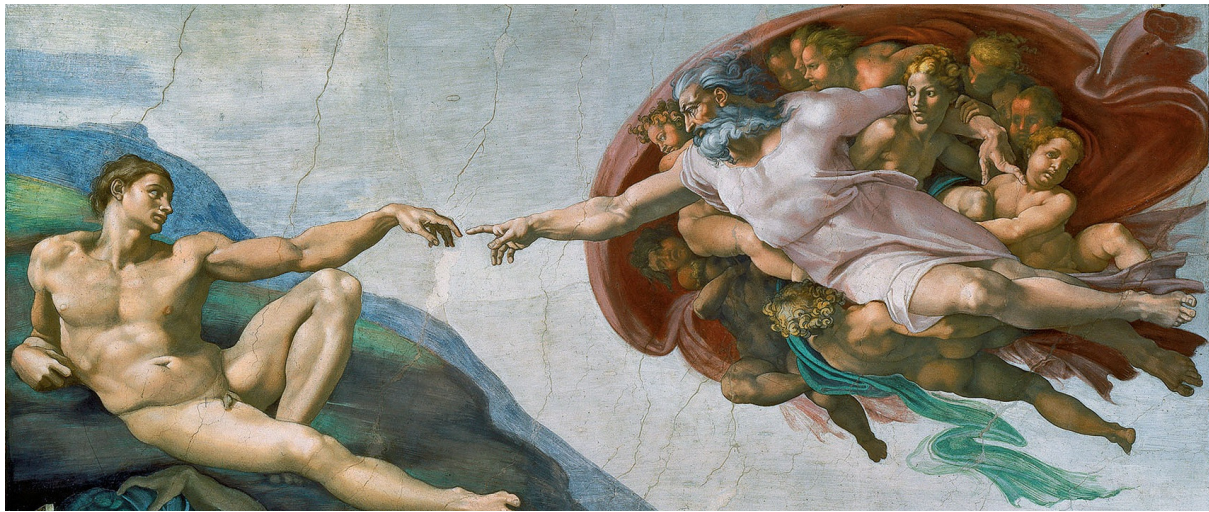
Each item can have any number of details

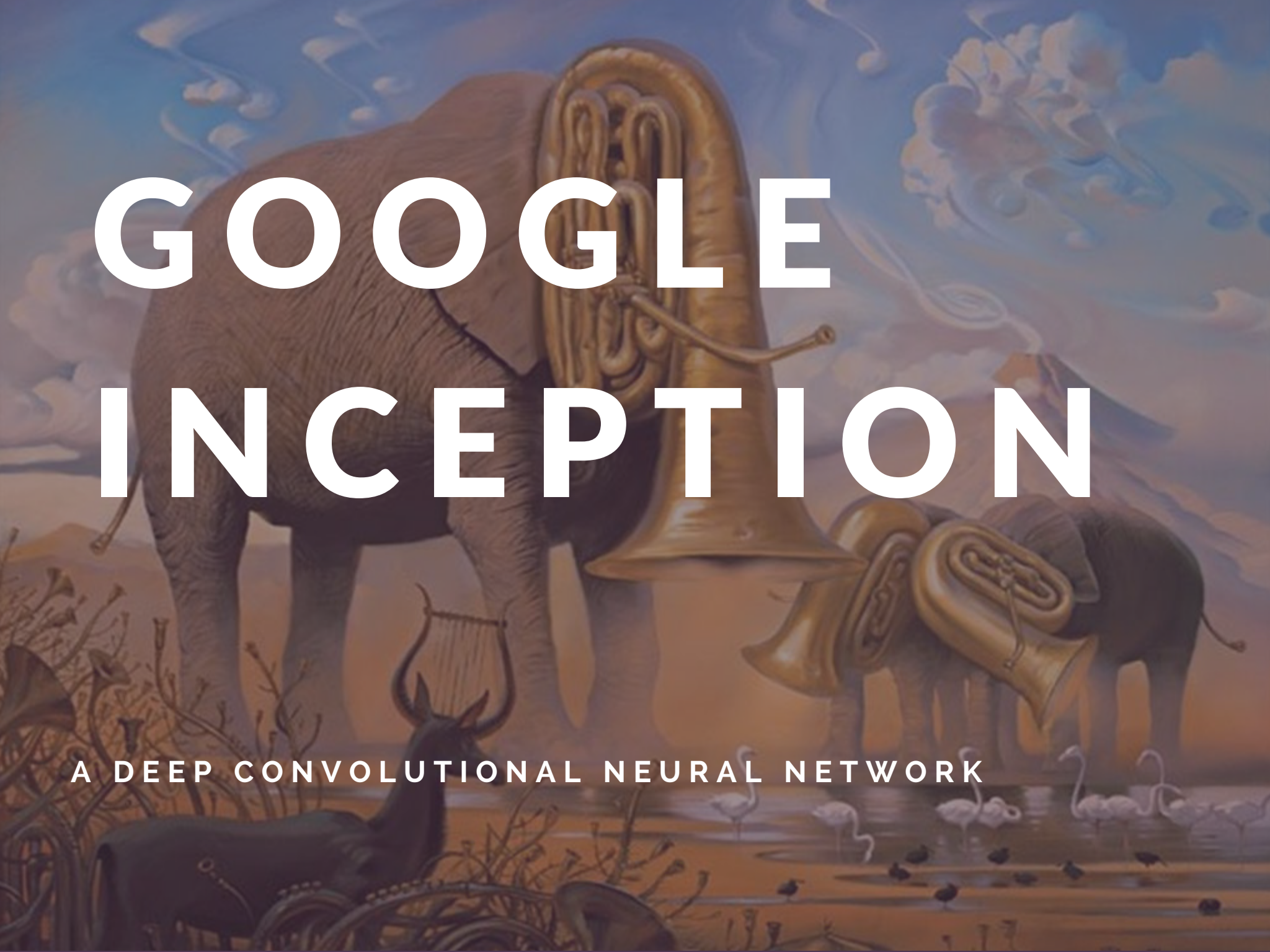
RELATIONSHIPS & COMPARISON

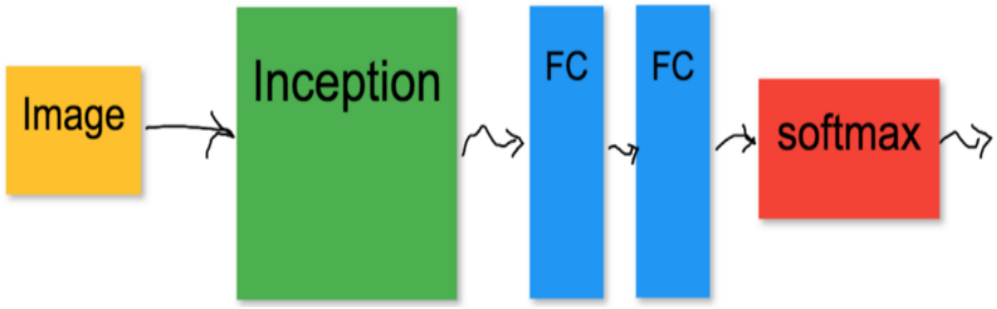
Allows for objects to be 'connected' in a searchable way
Stores data in a comparable format

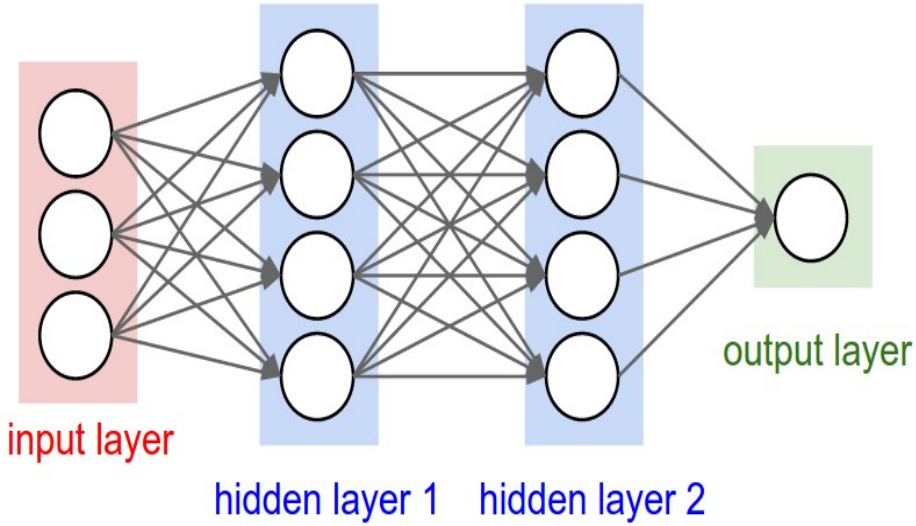






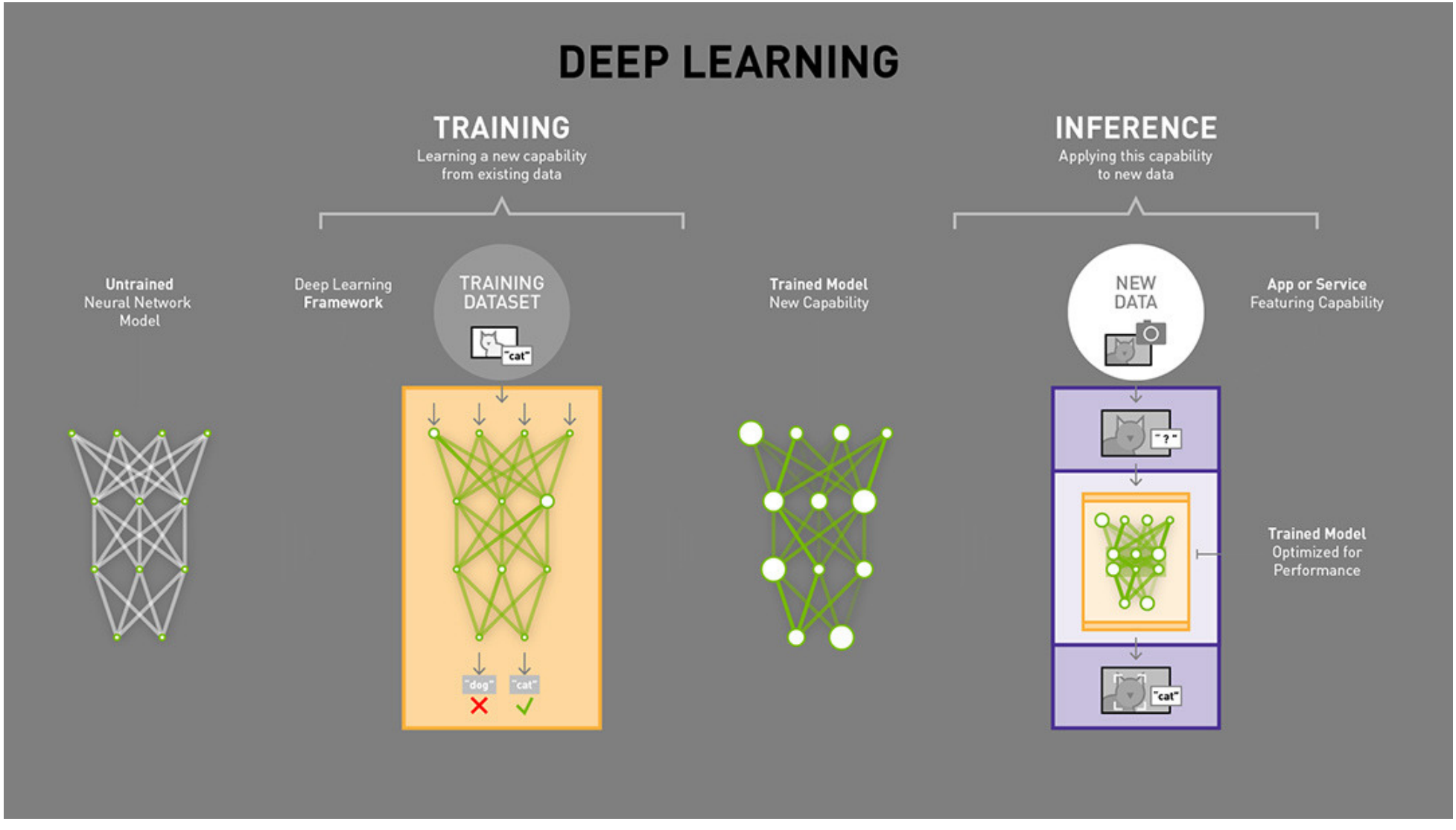

GOOGLE
INCEPTION
A DEEP CONVOLUTIONAL NEURAL NETWORK





Inception

A pre-trained model which can differentiate between 1,000 different classes, like “cat”, “dishwasher” or “plane”, "books".


DEEP LEARNING
TRAINING
Learning a new capability from existing data
INFERENCE
Applying this capability to new data
Untrained Neural Network Model
Deep Learning Framework
TRAINING DATASET
"cat"
"dog"
"cat"
Trained Model New Capability
NEW DATA
"?"
"cat"
App or Service Featuring Capability
Trained Model Optimized for Performance

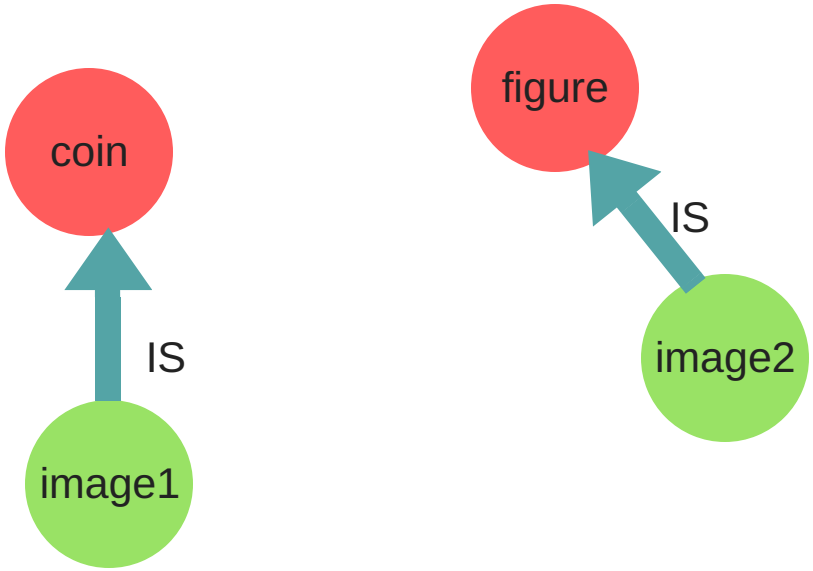
TRAINING INCEPTION

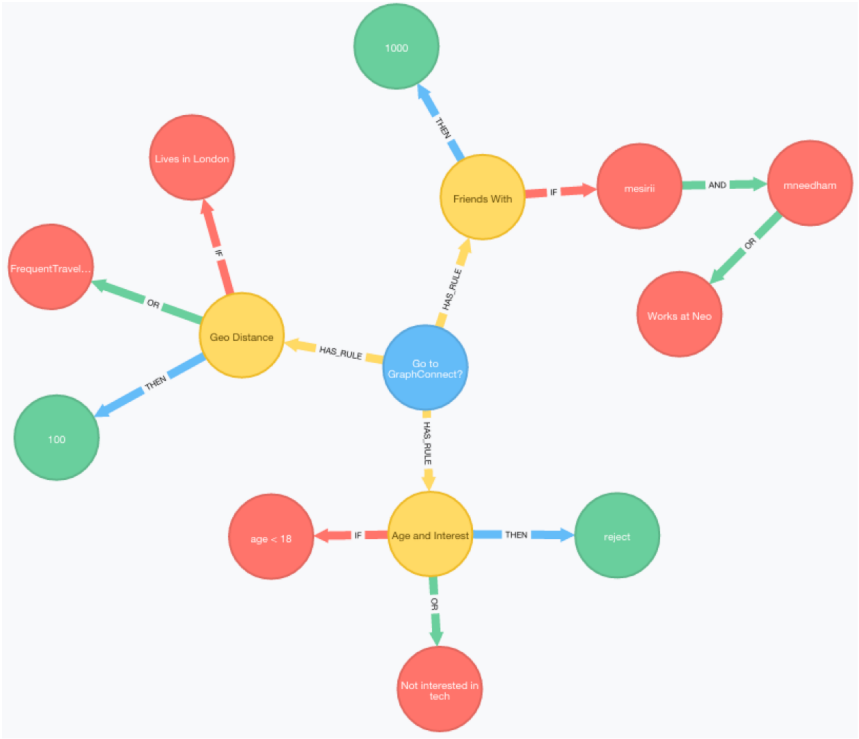
Pre-requisites: ~4500 non-redundant pre-labelled images stored in folders according to keywords.

Steps: 1. Modify the retraining script according to our sample images.

2. Run retrain.py which remove the old final output layer, and trains a new final output layer based on the museum object images.

Output: Training of the Inception model took ~30min with 70% accuracy.





Goodbye tables!

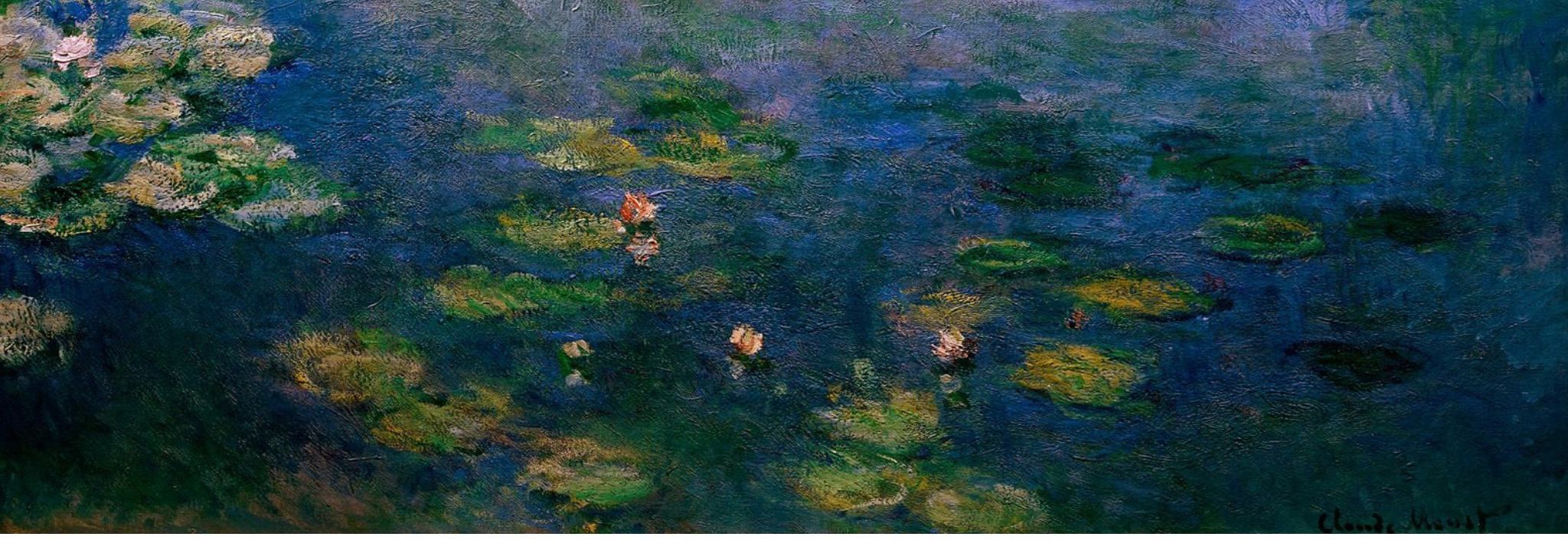
GRAPHS

Every object stored as a circle or node

Connections between them mapped as lines

Every 'node' can have qualities to describe it

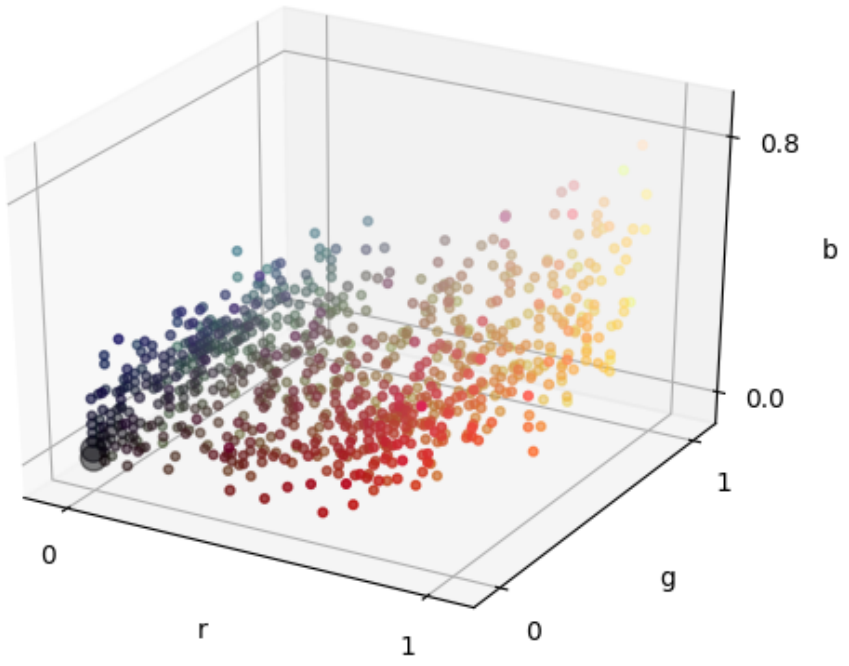
Every 'line' can have qualities to describe it



EXTRACTING NON-TEXT DATA

Every image is processed to get

- dominant colours
- a hash



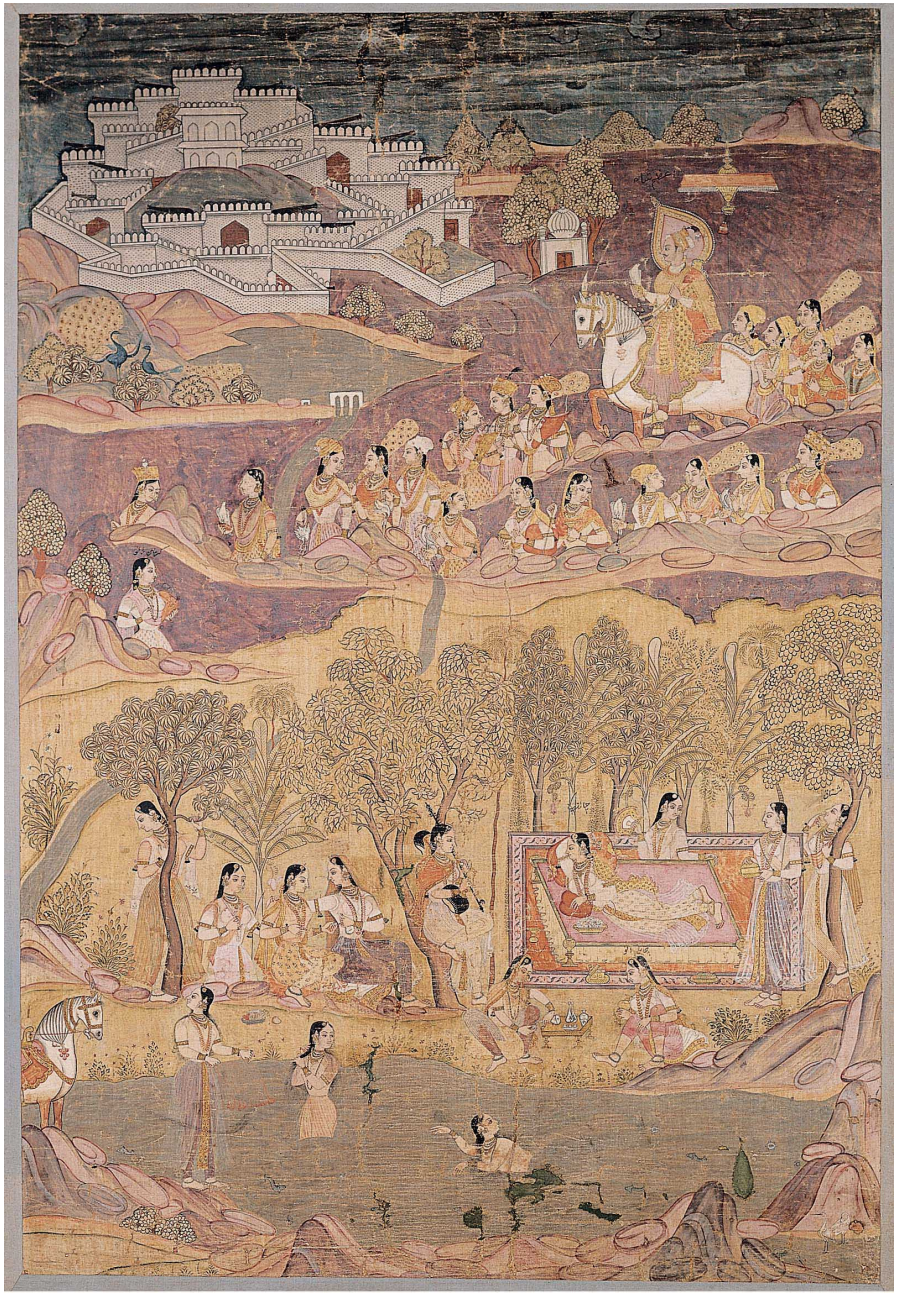
Colours

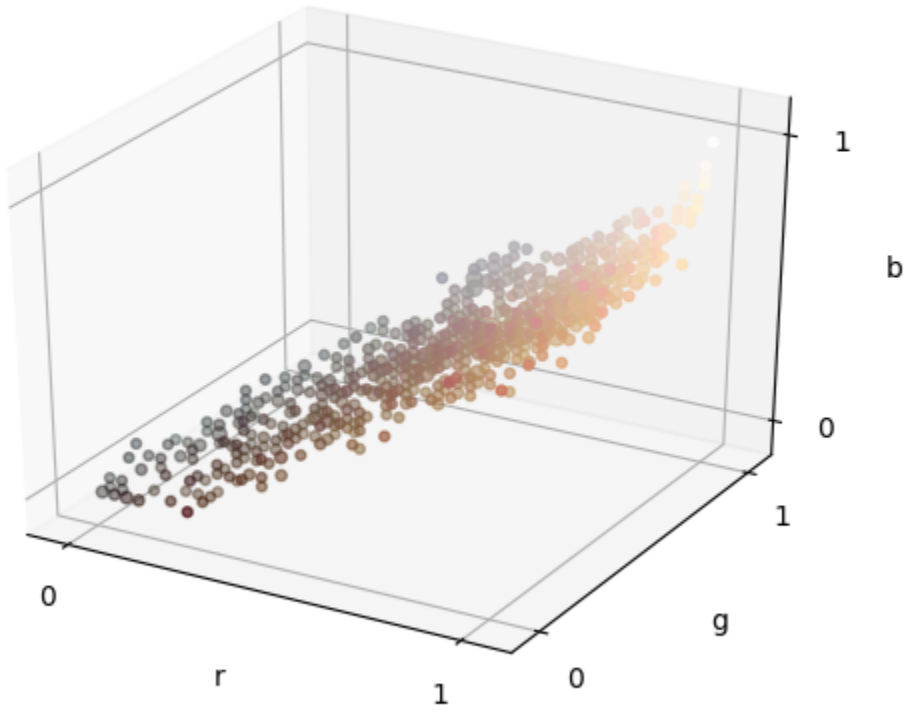
HOW DO WE FIND DOMINANT COLOURS?

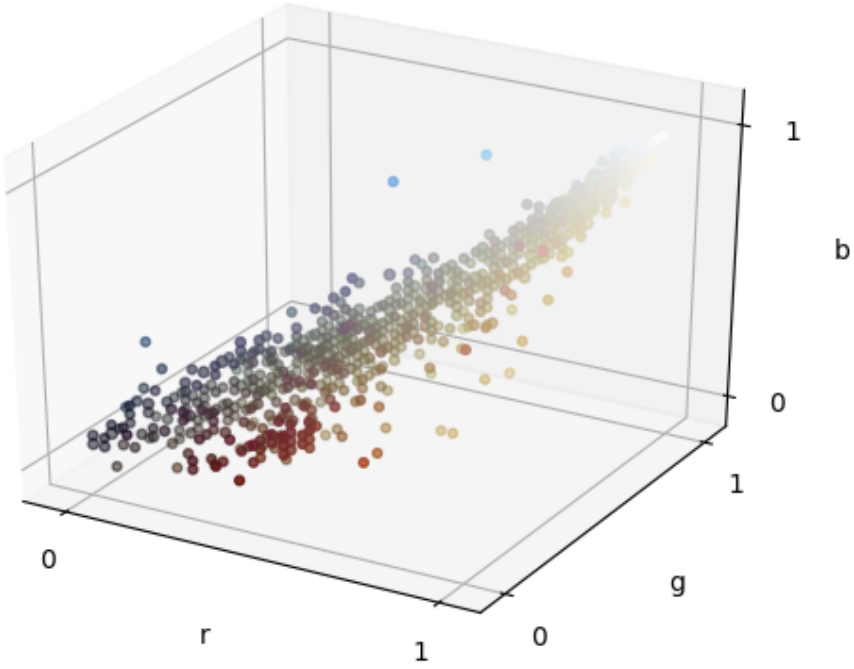
Every image pixel stored as (Red, Green, Blue) numerical value

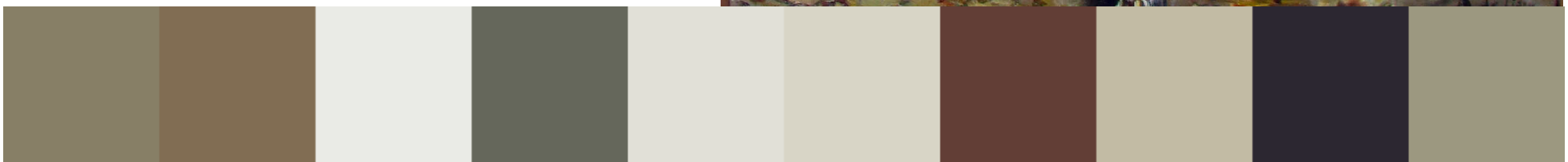
Show on a graph

Find centre of groups of similar pixels!




1
b
0
1
g
0
1
r
0

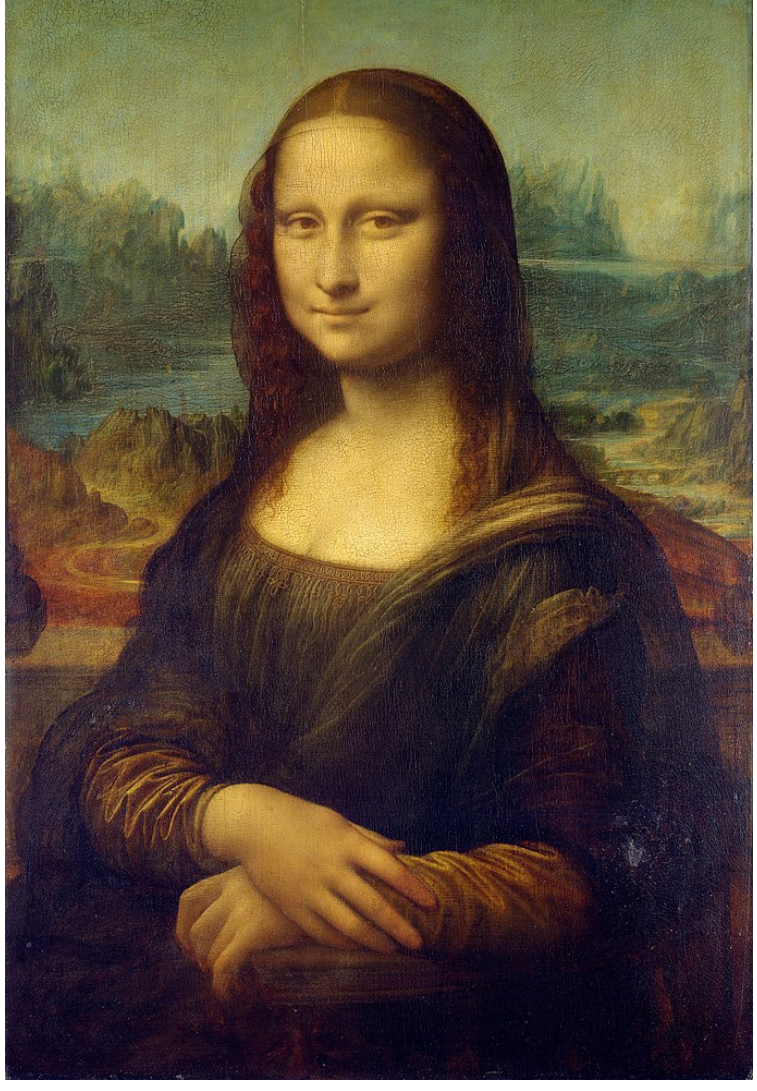

1
b
0
1
g
0
0
r
1



COMPARING PICTURES

Extract a 'hash value' of picture - apply a mathematical function to get a code that describes the picture - like a QR code!
This code is called a 'hash value'

Similar pictures will have similar codes
Hash values can be 'compared'



Similar with hamming distance = 14.000000

Similar with hamming distance = 17.000000

Similar with hamming distance = 24.000000

And now for something completely different!

Not similar with hamming distance = 33.000000

Future plans!

BETTER FRONT-END

EASIER WAY TO MAKE RELATIONSHIPS

IMPROVED TAGGING ACCURACY


THANK YOU