LORISAL

Library Online Repository Image Scraper and Labeler

Daniel Helm -- @danielgileshelm
C.S. Junior at the University of Oklahoma
Member of OU Data Analytics Lab
Done in collaboration with Josh Imbriani
Under the advisement of Dr. Christan Grant

This Talk

- The First Presentation of the Software, but since this is a User's Group, I'll try to focus on the narrative of building the software.
- High-level, but brief discussion of how various python libraries were leveraged to simplify complex tasks
- My code isn't remarkably different than that available in online resources
- I'm not terribly experienced in databases, scraping, computer vision, ocr, machine learning, deep learning, full text search or web applications...
- which highlights how powerful and accessible the python ecosystem is!

Problem

A library’s collection holds a wealth of textual information that is well cataloged for finding writings on whatever topic a researcher is interested in. Images, diagrams, drawings and photographs inside those texts however are not nearly as accessible through computerized systems. As more library collections are being scanned, that vast amount of imagery is now available as digital data, but finding relevant materials as a researcher is still difficult when images are embedded in the text.

Attempted Solution: LORISAL

- Library Online Repository Image Scraper and Labeler
- A name that's successful in being pronounceable, but not so descriptive
- The goal is to create an online interface for researchers to quickly search the image content inside of a library's book collection
- The code should be highly modular. Each collection has their own types of books and imagery, along with unique data pipelines.
- This project is a WIP and proof-of-concept, showing the feasibility
- Future work will refine each module and increase its flexibility for reuse in various collections.

The High-Level Overview

- Scrape an online repository to build a database of books
- Scrape low resolution book scans for every page
- Identify pages with imagery (“figures”)
- Scrape high resolution page scans for those pages
- Run OCR on these page scans
- Extract figures from the scans
- Run Deep Learning models to tag and label the imagery
- Build a web interface for searching through this data

A note on modularity

- Ideally, each step of the process will be modular so that implementations can easily be replaced, refined and freely shared as different users build solutions to unique problems and needs.
- That being said, many of the current results are a starting point…

Step 1: Get the Data!

- Find a repository of scanned books: http://islandora.ca/islandora-installations
- If the institution is unaware, don’t publish their repo’s url...

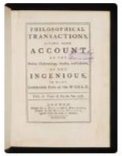
Islandora Installations

1. Adventist Digital Library
2. Albion College
3. American Philosophical Society
4. Andrews University
5. Atlantic Climate Adaptation Solutions Association
6. Barnard College
7. Berklee College of Music Archives
8. BESS Digital Archive
9. Biblioteca do Ministério da Fazenda no Rio de Janeiro (BMF/RJ)
10. Boston College
11. Botanical Research Institute of Texas
12. British Columbia Electronic Library Network
13. British Columbia Institute of Technology
14. Broughton Archipelago Monitoring Plan (BAMP)
15. Broward College
16. California Historical Society
17. California Institute of Technology (Caltech)
18. California Polytechnic State University
19. Camosun College
20. Canadian Writing Research Collaboratory - University of Alberta
21. Capilano University
22. Centre de recherches acadiennes de l'Île-du-Prince-Édouard
23. Chinese University of Hong Kong
24. City of Hope
25. CNR IPSP and CNR IRCrES - V2P2 Project
26. College of St. Scholastica
27. College of the Rockies
28. Colorado College
29. Colorado State Publications Library
30. Commission for Environmental Cooperation - Green Building Library
31. Commission for Environmental Cooperation - Virtual Library
32. Corvelis Desarrollos S.A de C.V
33. Danmarks Tekniske Informationscenter - Technical Information Center of Denmark
34. Davidson College
35. Delft University of Technology
36. Detroit Public Library
37. discoverygarden, Inc.
38. Douglas College

ISLANDORA REPOSITORY

1 2 3 4 5 6 7 8 9 10 ... next last

- Grid view
- List view



"A Description of the Plan of Peking, the Capital of China," Philosophical Transactions of the Royal Society of London, Vol. 50, Part 2


"Epistola circa Pythagoricorum, & Copernici opinionem de mobilitate terrae, et stabilitate solis..." in Galileo 1635 Systema cosmicum


"Sketch of the Analytical Engine Invented by Charles Babbage by L.F. Menabrea... with Notes," in Scientific Memoirs, vol. 3, pp. 666-731.


"The Systeme of the World in Four Dialogues" in Thomas Salusbury, Mathematical Discourses, 1661


A Prognostication euerlasting of right good effect...; Lately corrected and augmented by Thomas Digges his sonne

A Week's Conversation on the Plurality of Worlds [Entretiens sur la pluralite des mondes]


Admonitio ad astronomos


Almagestum novum [Volume 1, Part 1]


Almagestum novum [Volume 1, Part 2]

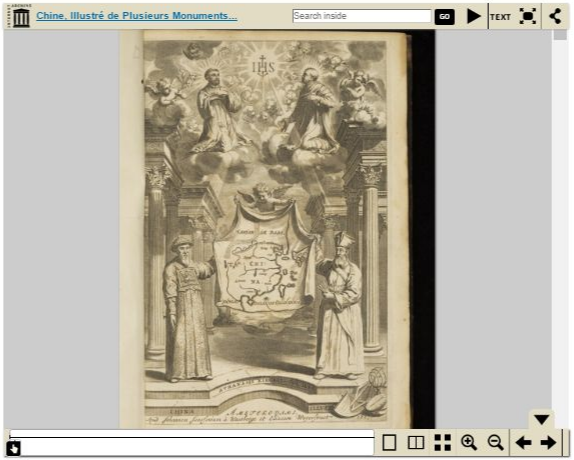

An Astronomical Catechism


An Inquiry into the Causes and Effects of the Variolae Vaccinae [The Cow Pox]


Anatomia

CHINE, ILLUSTRÉ DE PLUSIEURS MONUMENTS

View | Pages



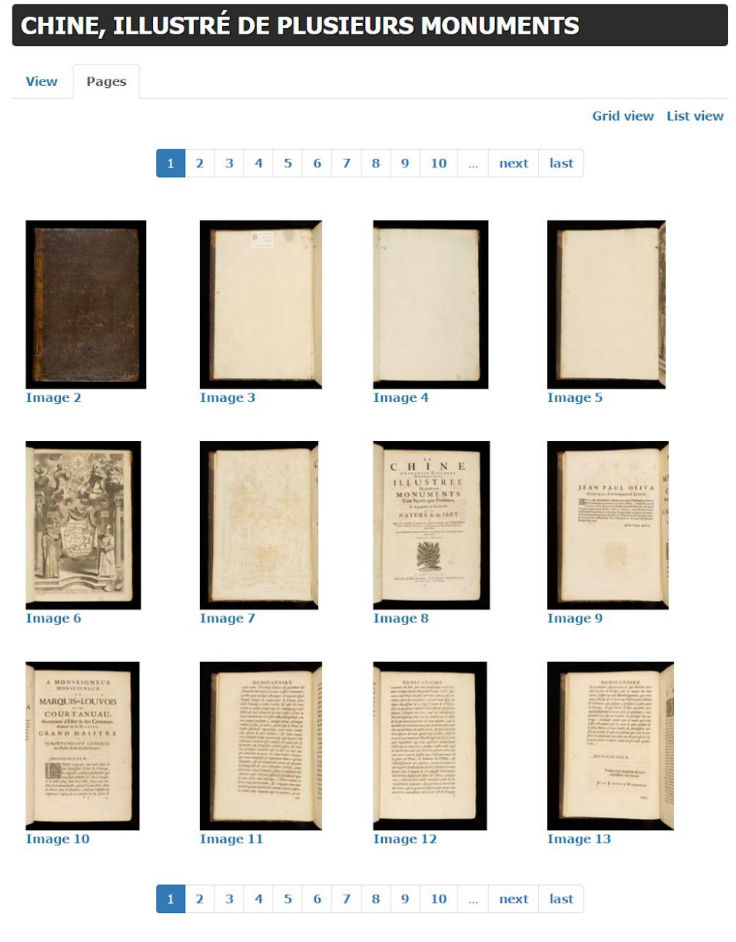
In collections

- History of Science Collection

Details

Title	Chine d'Athanase Kirchere de la Compagnie de Jesus
Contributors	Kircher, Athanasius 1602-1680 Grueber, Johann 1623-1680 Dalquié, F. S. Chardin, John 1643-1713
Alternative Title	Chine d'Athanase Kirchere de la Compagnie de Jesus
Type of Resource	text
Note	Added t.p., engraved, has imprint: Amstelodami, Apud Joannem Janssonium & Elizeum Weyerstraet, 1667. Translation of: Athanasii Kircheri e Soc. Jesu China monumentis ... illustrata. L'explication du monument d'un Syro-Chinois -- Des divers chemis qu'on a tenu pour aller à la Chine -- De l'idolatrie venue d'Occident -- La Chine illustrée des miracles de la nature & de l'art -- De l'architecture et des autres arts méchaniques des Chinois -- De l'escritures des Chinois -- La briefve & exacte responce du père Jean Grubere -- Dictionnaire chinois & françois. Includes index.
Statement of Responsibility	traduit par F. S. Dalquié.

Now how do we get all this data?



Beautiful Soup & Requests

- I first learned about this here!
- Beautiful Soup quickly parses and traverses HTML documents
- If you've ever dealt with CSS, the selectors work similarly
- Requests just makes code for dealing with HTTP requests even easier and more readable than the standard library's urllib

Beautiful Soup & Requests

- Beautiful Soup is used to for scraping, Requests to download files and images
- The site uses Universally Unique Identifiers (uuid) to label books and pages and uses them in the url
- Using <a href> tags, we can find the names and uuids of each book and add them to the database. Following the link we can download the book's metadata.
- Looking at the "Pages" area, we get the page number and uuid of each page.
- At this point, we can find a pattern in the urls to use as a template to generate all the urls needed to download page scan files.

Beautiful Soup Code Sample

```
def scrapeBookList(url):
    bookList = []
    pageNum = 0
    while True:
        soup = BeautifulSoup(urllib.request.urlopen(url + str(pageNum)), "lxml")
        for thumb in soup.select(".islandora-basic-collection-thumb > a"):
            uuid = thumb["href"]
            uuid = uuid[6:]
            book = {"full_title": thumb["title"], "uuid": uuid}
            bookList.append(book)

        if soup.find("a", text="next"):
            time.sleep(0.1)
            pageNum += 1
        else:
            return bookList
```

XML2Dict

- Each book has an XML file containing all of its metadata
- Parsing this is the worst!
- XML2Dict does exactly what we'd expect
- Takes an XML file and converts it to a python dictionary in a single function
- We can save this as plain text and return to it later
- How do we store this data once retrieved?

```
xml_meta = requests.get(urlToXMLFile)
metadata = xmltodict.parse(xml_meta.text)
# metadata is an OrderedDict object!
```

Peewee

- Databases are important, but writing SQL queries isn't why most of us use python
- Python has no shortage of great ORM (object relational mapping) tools that abstract database interaction
- Peewee does this simply, intuitively and pythonically.
- Built in support for sqlite, mysql and postgresql

```
class Book(BaseModel):
    repository = ForeignKeyField(Repository, related_name='books')
    full_title = CharField()
    title = CharField(null=True)
    uuid = CharField(unique=True, primary_key=True)
    mods_metadata = TextField(null=True)
    pages_scraped = BooleanField(default=False)

for book in booklist:
    Book.create(
        repository=repository,
        full_title=book['full_title'],
        uuid=book['uuid'],
        mods_metadata=getBookMODSmetadata(book['uuid']),
        pages_scraped=False
    )

query = Book.select().where(Book.pages_scraped == False)
for book in query:
    print(book.full_title)
```

What pages do we examine?


Image 133


Image 134


Image 135


Image 136


Image 137


Image 138


Image 139


Image 140


Image 141

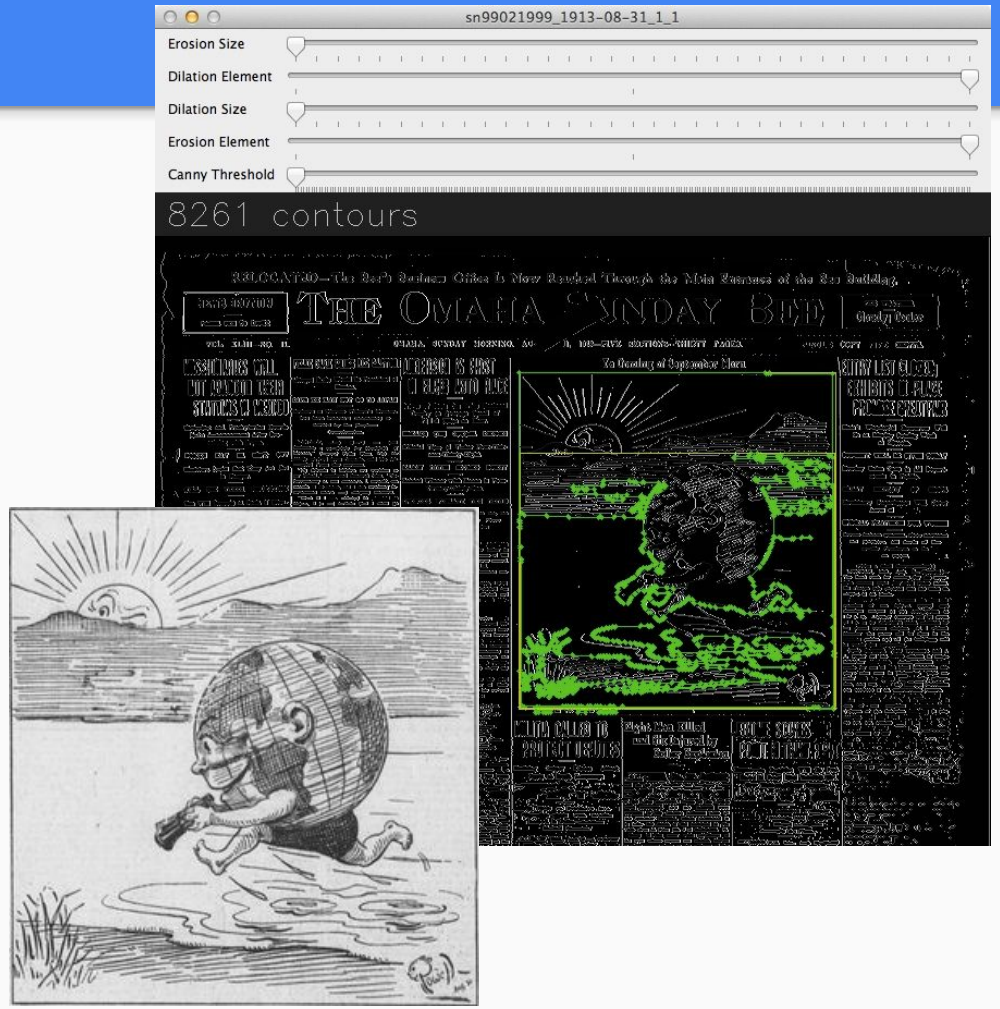

Image 142


Image 143


Image 144

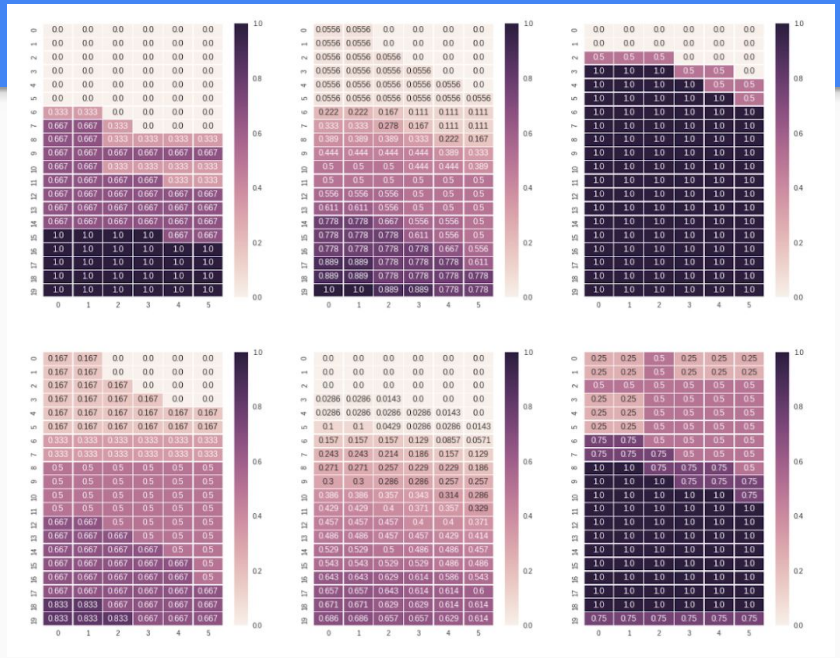
OpenCV and Image Mining

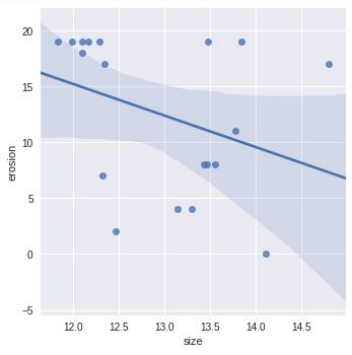
- OpenCV is a computer vision toolkit for doing image analysis
- Others have solved this problem, so we'll use their library!
- We can do a pass to first identify scans we suspect have images
- Then, download the full resolution versions and attempt image extraction
- But sliders mean fitting the algorithm to the source, hindering automation…

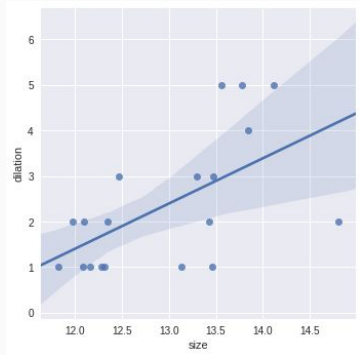


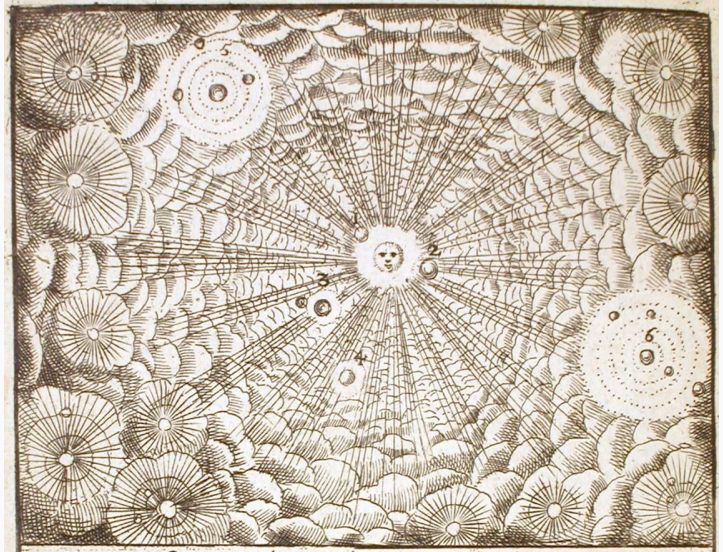
Determining Algorithm Parameters

- The “low resolution” imagery varies greatly in image dimensions, making one-size-fits-all parameter choices impossible.
- Taking a handful of books, we can determine which parameters work best for a given book (In above figure each axis is a parameter, success is the darkness of the color)
- From here, we can use linear regression to get a simple formula based on image dimensions.
- Some of the great Machine Learning tools in Python (Scikit-Learn, Pandas, Seaborn) were used to do this, but that code isn’t in the application so let’s keep moving…

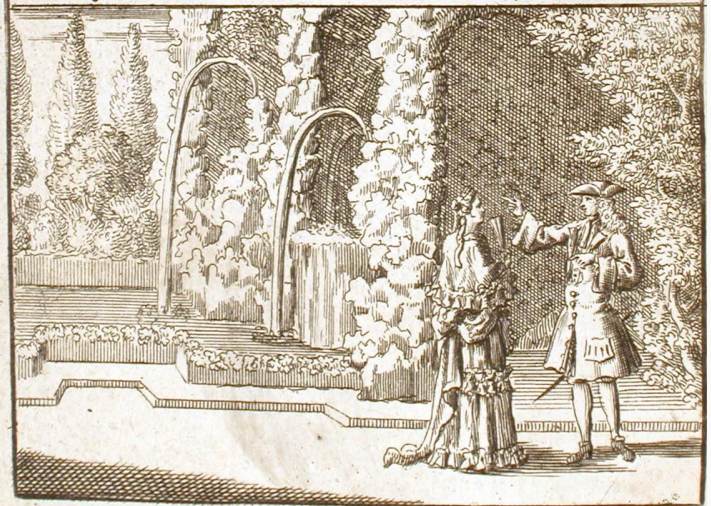


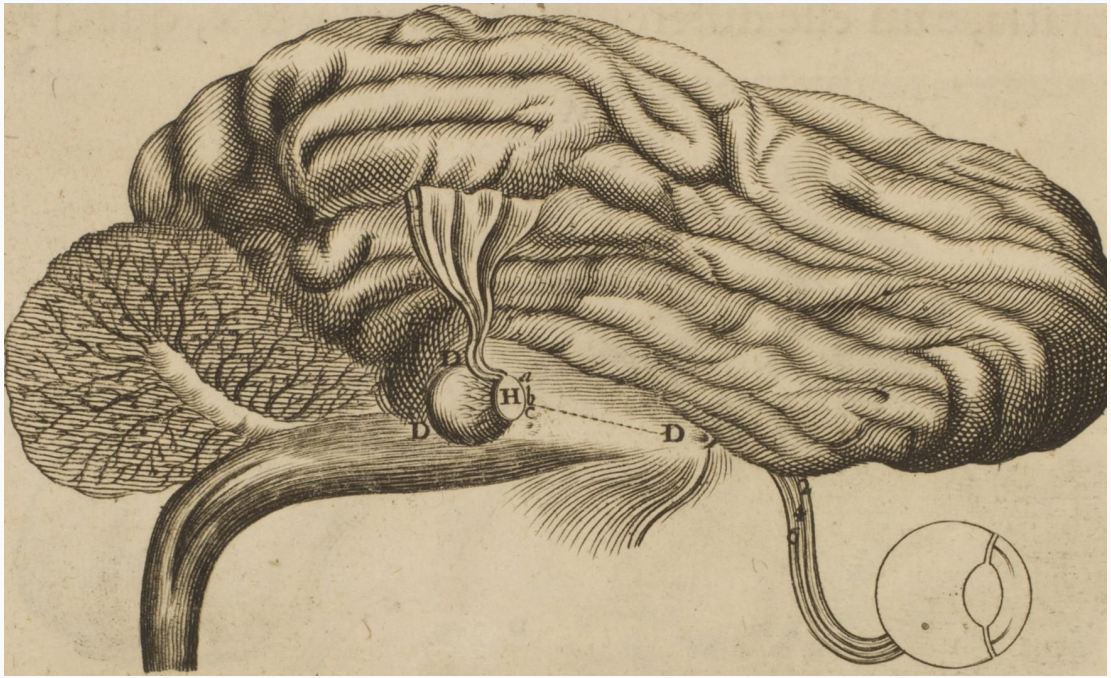


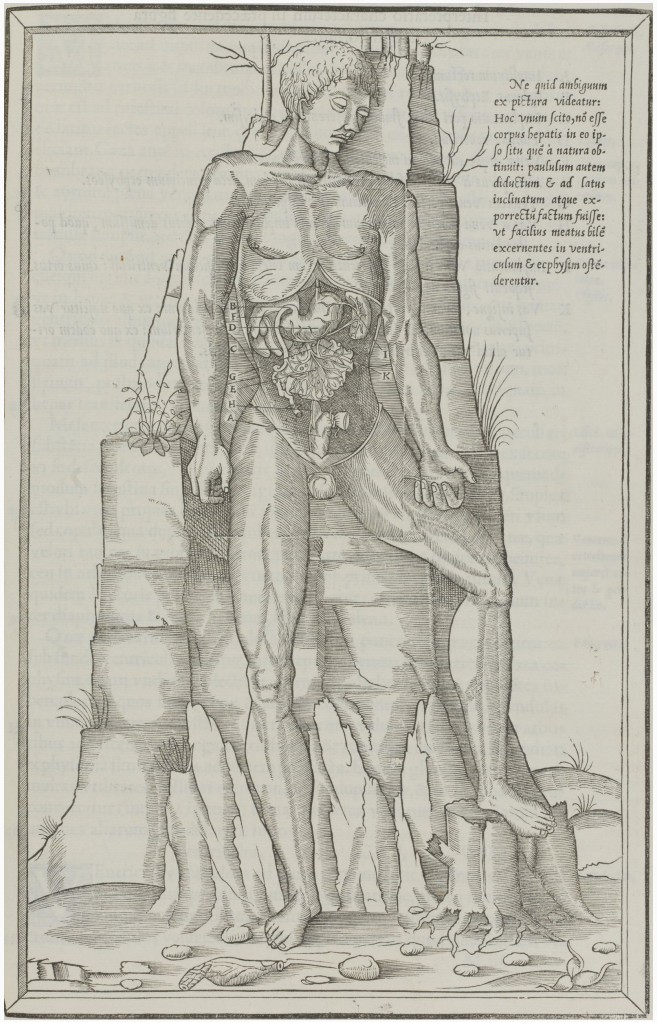




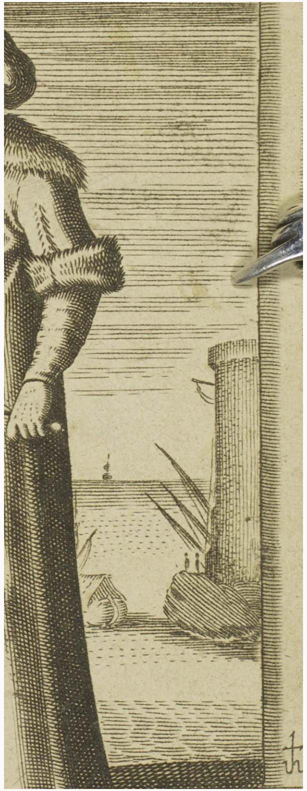
1 Mercury, 2. Venus, 3. the Earth, 4 Mars, 5. Jupiter, 6. Saturn.

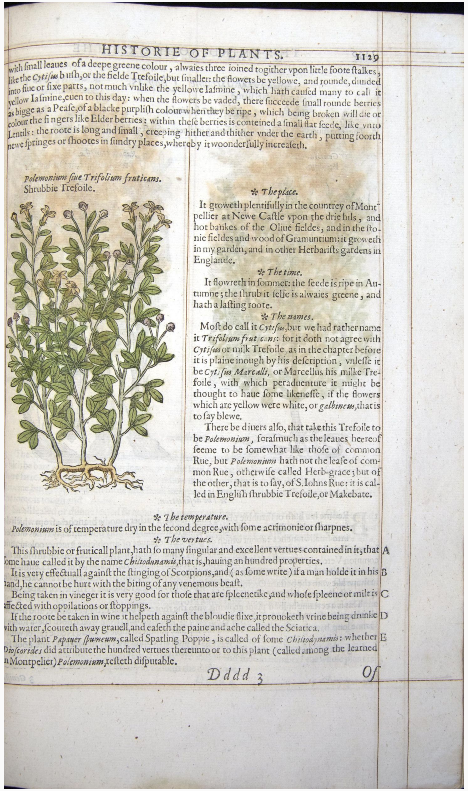



D
H
D


Ne quid ambiguum ex pictura videatur: Hoc vnum scito, nó esse corpus hepatis in eo ipso situ quẽ à natura obtinuit: paululum autem diductum & ad latus inclinatum atque exporrectũ factum fuisse: vt facilius meatus bilẽ excernentes in ventriculum & ecphysim ostẽderentur.
B
F
D
C
G
E
H
A
I
K

A Variety of Failures





HISTORIE OF PLANTS. 1129

with small leaues of a deepe greene colour, alwaies three ioined togither vpon little foote stalkes, like the *Cytisus* bush, or the fielde Trefoile, but smaller: the flowers be yellowe, and rounde, diuided into fiue or sixe parts, not much vnlike the yellowe Iasmine, which hath caused many to call it yellow Iasmine, euen to this day: when the flowers be vaded, there succeede small rounde berries as bigge as a Pease, of a blacke purplish colour when they be ripe, which being broken will die or colour the fingers like Elder berries: within these berries is conteined a small flat seede, like vnto Lentils: the roote is long and small, creeping hither and thither vnder the earth, putting foorth newe springes or shootes in sundry places, whereby it woonderfully increaseth.

Polemonium siue Trifolium fruticans.
Shrubbie Trefoile.

✲ *The place.*

It groweth plentifully in the countrey of Montpellier at Newe Castle vpon the drie hils, and hot bankes of the Oliue fieldes, and in the stonie fieldes and wood of Gramuntium: it groweth in my garden, and in other Herbarists gardens in Englande.

✲ *The time.*

It flowreth in sommer: the seede is ripe in Autumne; the shrub it selfe is alwaies greene, and hath a lasting roote.

✲ *The names.*

Most do call it *Cytisus*, but we had rather name it *Trifolium fruticans*: for it doth not agree with *Cytisus* or milk Trefoile, as in the chapter before it is plaine inough by his description, vnlesse it be *Cytisus Marcelli*, or Marcellus his milke Trefoile, with which peraduenture it might be thought to haue some likenesse, if the flowers which are yellow were white, or *galbineus*, that is to say blewe.

There be diuers also, that take this Trefoile to be *Polemonium*, forasmuch as the leaues heereof seeme to be somewhat like those of common Rue, but *Polemonium* hath not the leafe of common Rue, otherwise called Herb-grace; but of the other, that is to say, of S. Iohns Rue: it is called in English shrubbie Trefoile, or Makebate.

✲ *The temperature.*

Polemonium is of temperature dry in the second degree, with some acrimonie or sharpnes.

✲ *The vertues.*

This shrubbie or fruticall plant, hath so many singular and excellent vertues contained in it, that A
some haue called it by the name *Chiltodunamis*, that is, hauing an hundred properties.

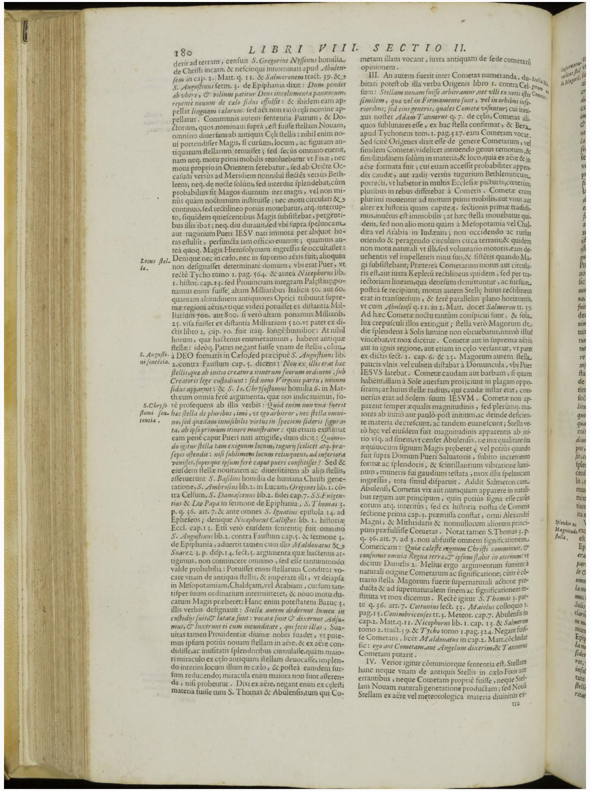
It is very effectuall against the stinging of Scorpions, and (as some write) if a man holde it in his B
hand, he cannot be hurt with the biting of any venemous beast.

Being taken in vineger it is very good for those that are spleenetike, and whose spleene or milt is C
affected with oppilations or stoppings.

If the roote be taken in wine it helpeth against the bloudie flixe, it prouoketh vrine being drunke D
with water, scoureth away grauell, and easeth the paine and ache called the Sciatica.

The plant *Papauer spumeum*, called Spatling Poppie, is called of some *Chiliodynamis*: whether E
Dioscorides did attribute the hundred vertues thereunto or to this plant (called among the learned in Montpelier) *Polemonium*, resteth disputable.

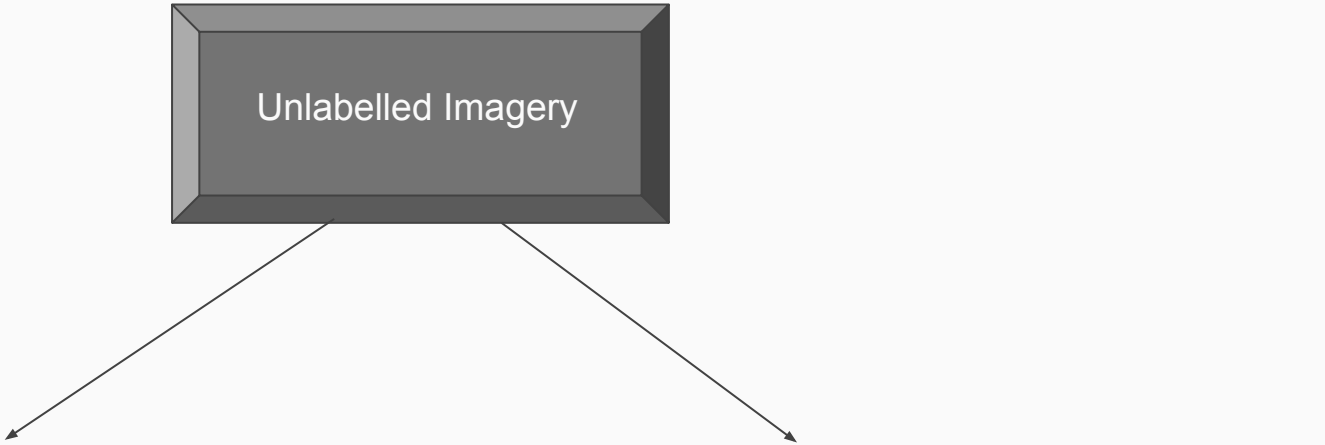
Dddd 3 Of



180 LIBRI VIII. SECTIO II.

Identifying Unlabeled Imagery

We get info from either the content of an image or the text surrounding it.



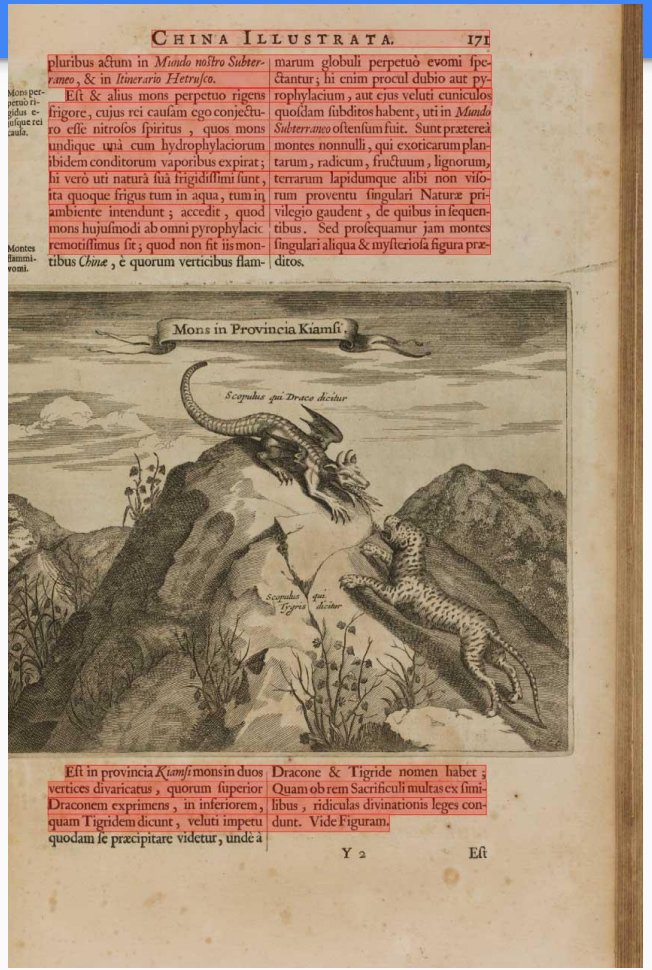
Analysis of the surrounding text, requiring Optical Character Recognition (OCR)

Object recognition and classification, requiring the magic of Neural Networks

pyOCR

- pyOCR is a python wrapper for multiple open OCR softwares (we use Tesseract)
- As simple as feeding the scan and its language (we have book metadata to tell us)
- This can also take text from diagrams and image captions.
- Libraries like unpaper help to clean scans to increase the success of OCR.
- Library tells us coordinates of text for later use in analysis

```
tools = pyocr.get_available_tools()
line_and_word_boxes = tool[0].image_to_string(
                Image.open(image_path), lang=lang,
                builder=tool[0]
        )
txt = ""
for line in line_and_word_boxes:
    txt += line.content
```



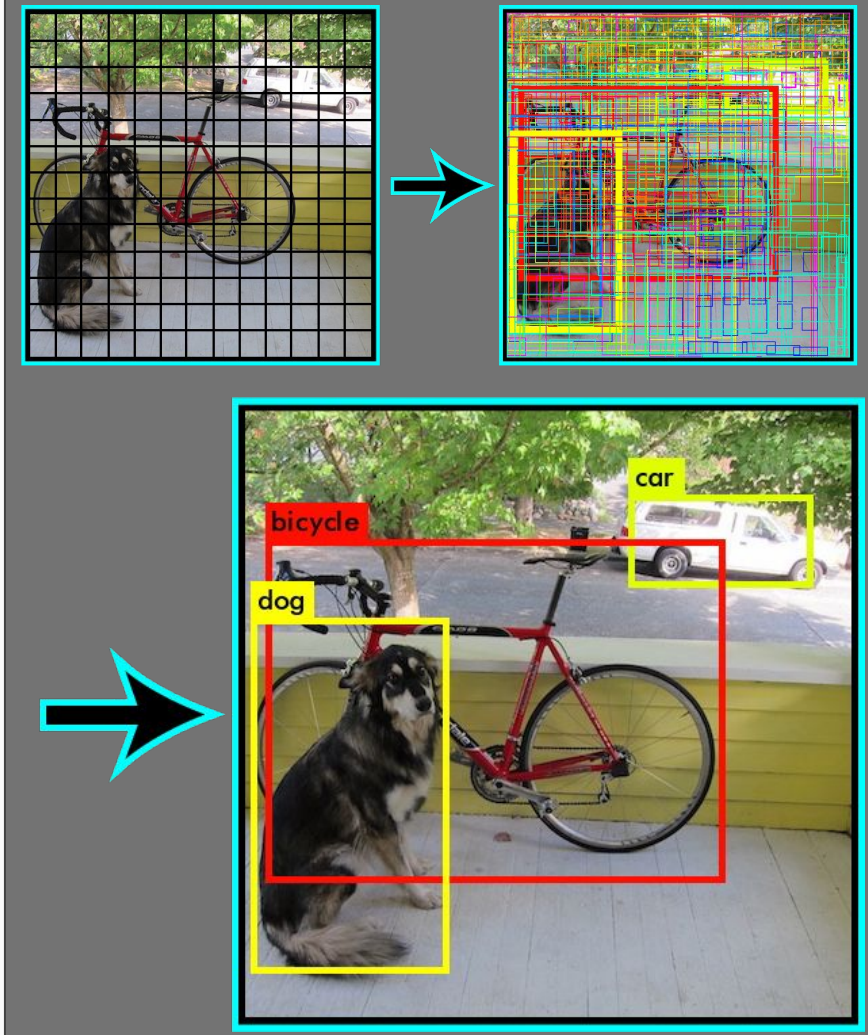
Tagging vs Captioning

- Explosion in deep learning gives us ways to identify objects in images
- Google’s Inception is well known example:

Tagging vs Captioning

- State of the art models are now being used to identify multiple objects in a scene (Here called “Tagging”)
- Natural Language Processing researchers are building on this to create human readable captions for images based on the relationship of these objects (Here called “Labelling”)

YOLO
Real-Time Object Detection



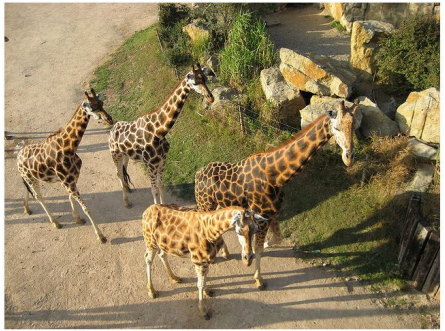
im2txt

A person on a beach flying a kite.

A black and white photo of a train on a train track.

A person skiing down a snow covered slope.

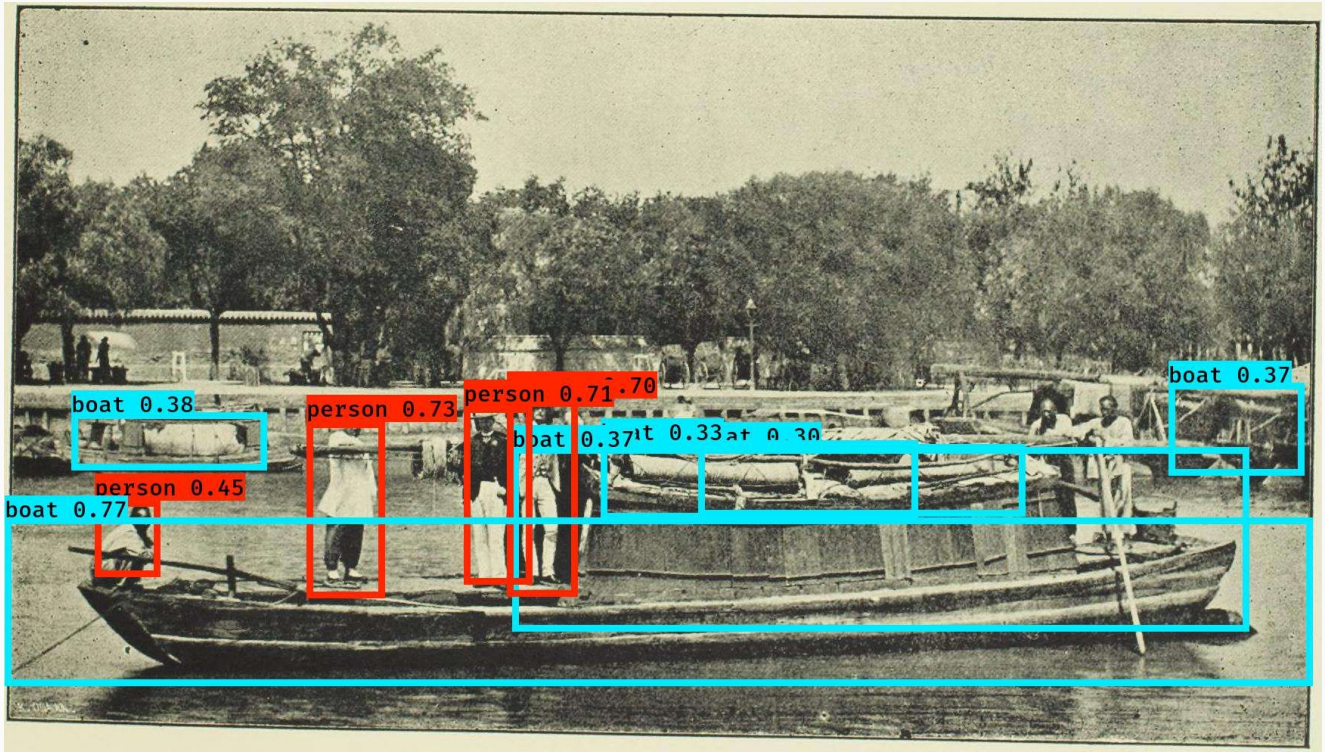
A group of giraffe standing next to each other.



Keras & TensorFlow

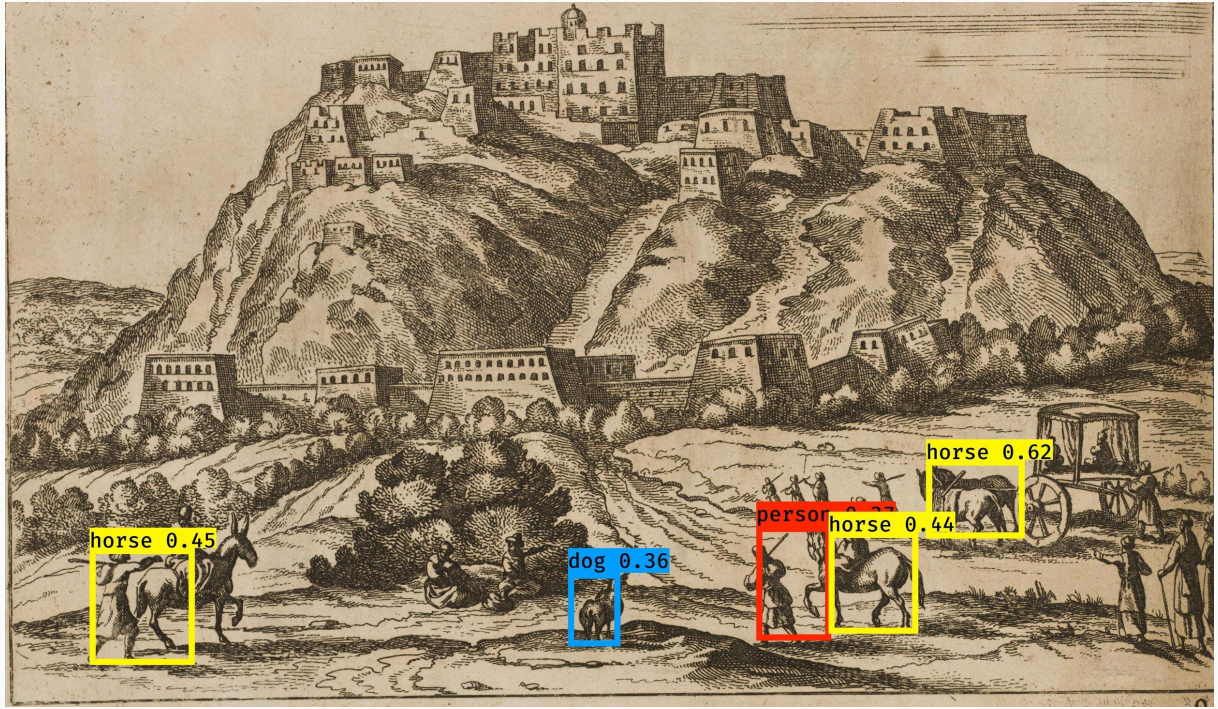
- Part of the explosion in Deep Learning is the democratization in who has access to it, hardware and software.
- TensorFlow makes it easy-ish, Keras simplifies model definition and the training process using TF (or other frameworks) as a backend.
- YOLO uses neither of these, but once a neural network architecture is developed and the model trained, it can be ported to any other framework!
- YAD2K is a port to Keras, making it simple to use in Python
- im2txt is a model included in the TensorFlow repo (also making use of NumPy and Natural Language Toolkit (NLTK)...)

A few examples…



['a black and white photo of people on a boat ', 'a black and white photo of people on a boat ', 'a black and white photo of people sitting on a boat ']

A few examples…

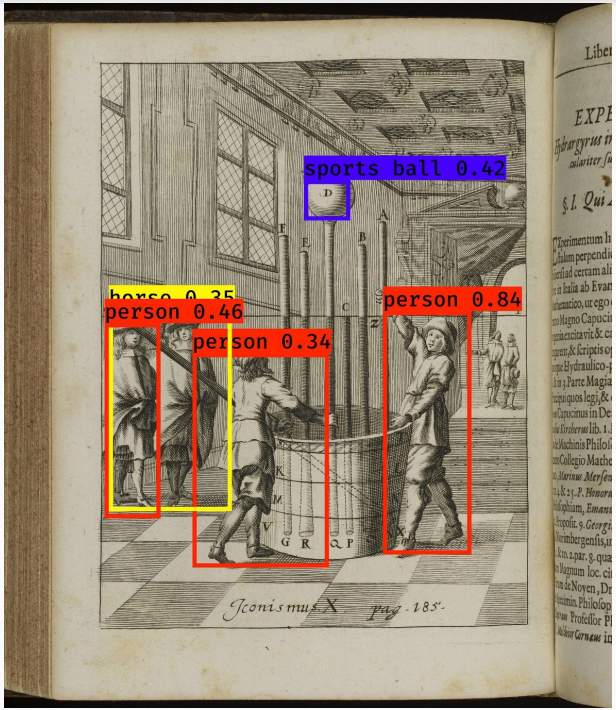


['a black and white photo of a group of people in a field ', 'a black and white photo of a group of people in a field ', 'a black and white photo of a group of people on a bench ']

A few examples...

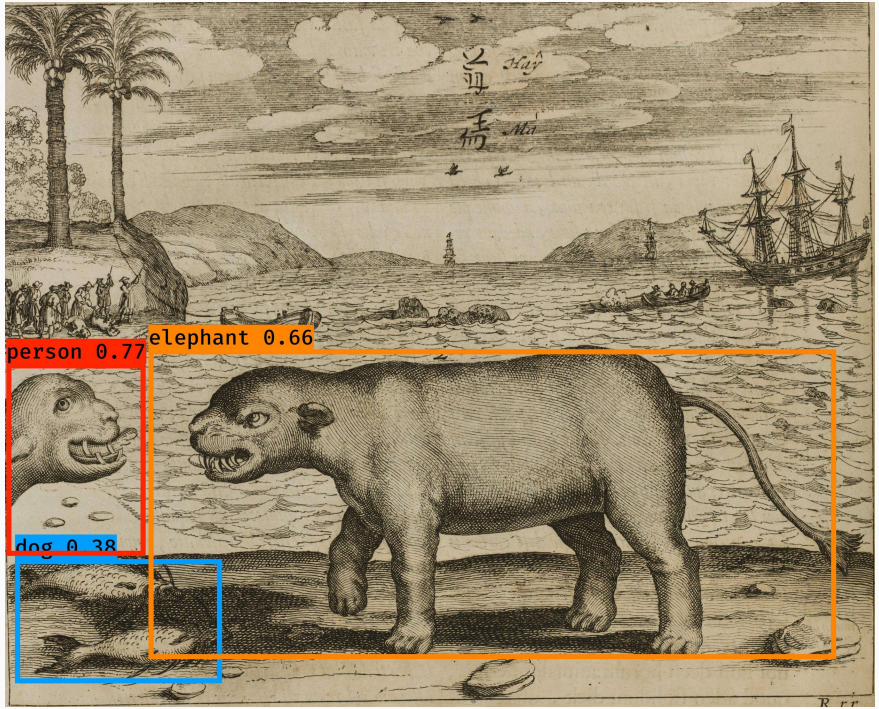
['a black and white cat sitting on top of a wooden table ', 'a black and white cat sitting on top of a wooden chair ', 'a black and white cat sitting on top of a wooden bench ']

A few examples…



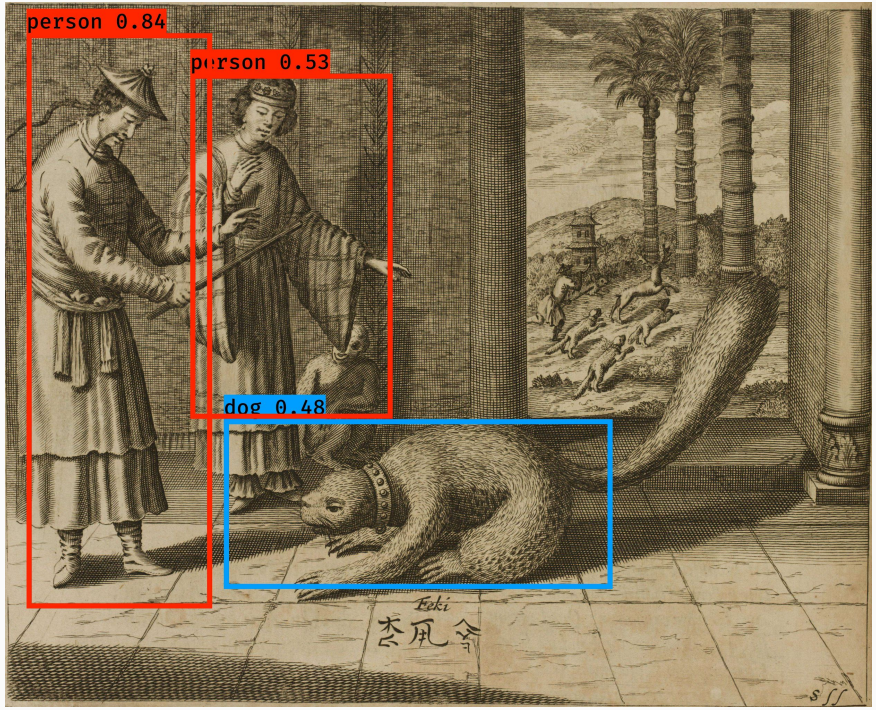
['a black and white photo of a clock on a wall ', 'a black and white photo of a clock on a wall ', 'a black and white photo of a clock on the wall ']

A few examples…



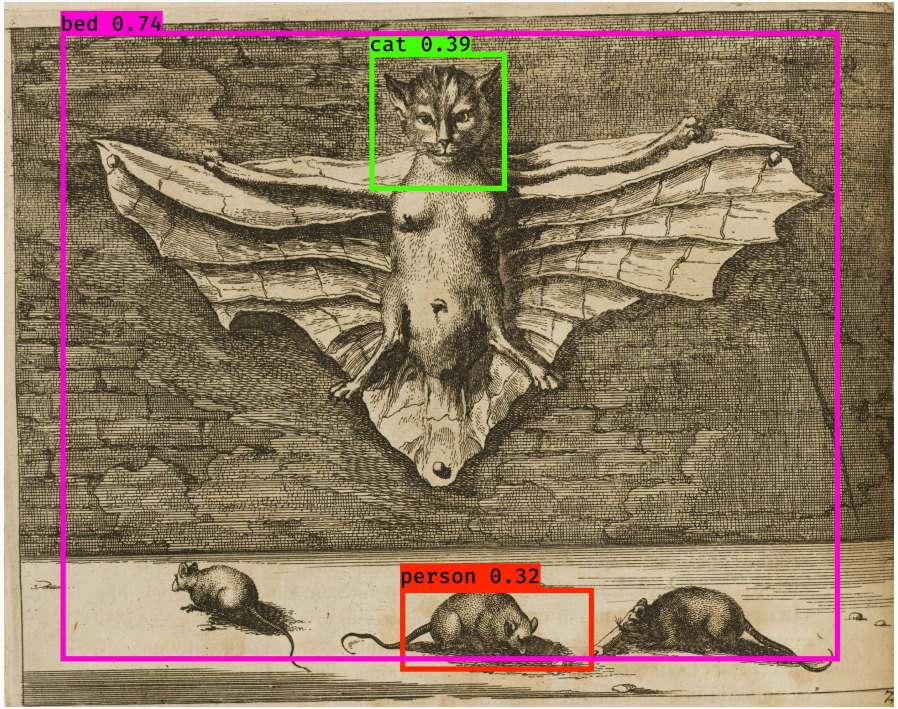
['a black and white photo of a horse in a field ', 'a black and white photo of a horse and a cow ', 'a black and white photo of a horse in a field ']

A few examples…



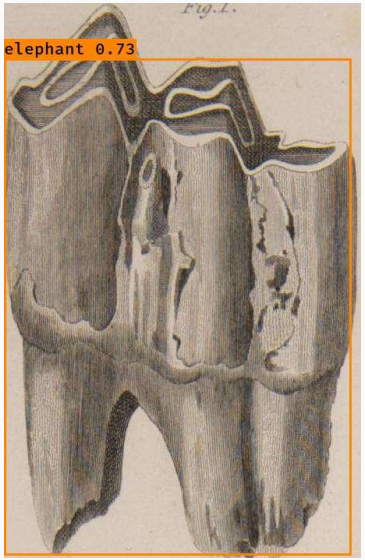
['a black and white photo of a tree in the woods ', 'a black and white photo of a tree in the woods ', 'a black and white photo of a tree in a forest ']

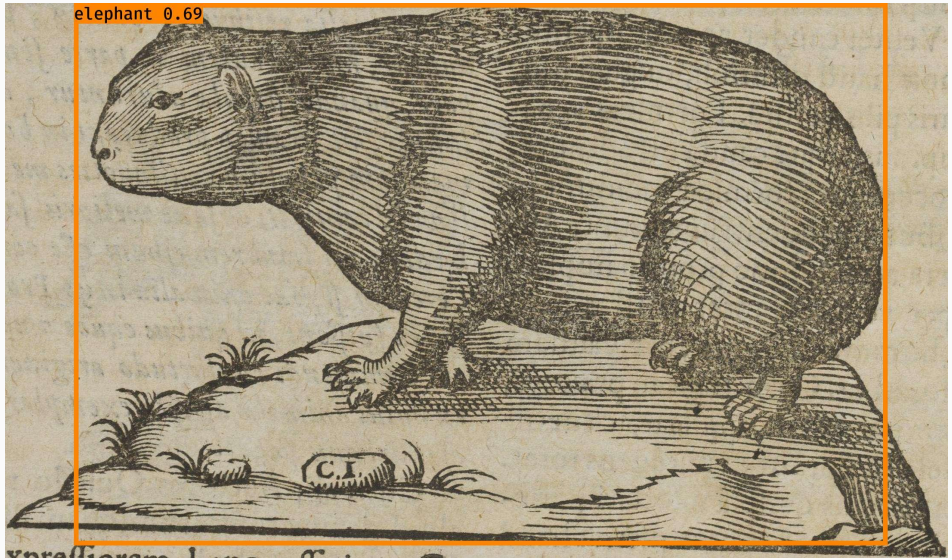
A few examples…



['a black and white photo of a clock on a wall ', 'a black and white photo of a clock on a wall ', 'a black and white photo of a clock on a table ']

A few examples…





But what if we want it to be useful?

- These results aren’t that great! But, this is expected.
- These models were trained on real-world photographs
- Our repository isn’t that!
- Future work will involve training models to determine a figure’s type (drawing, etching, photograph), and then training object recognition models for each type.

Building a User Interface

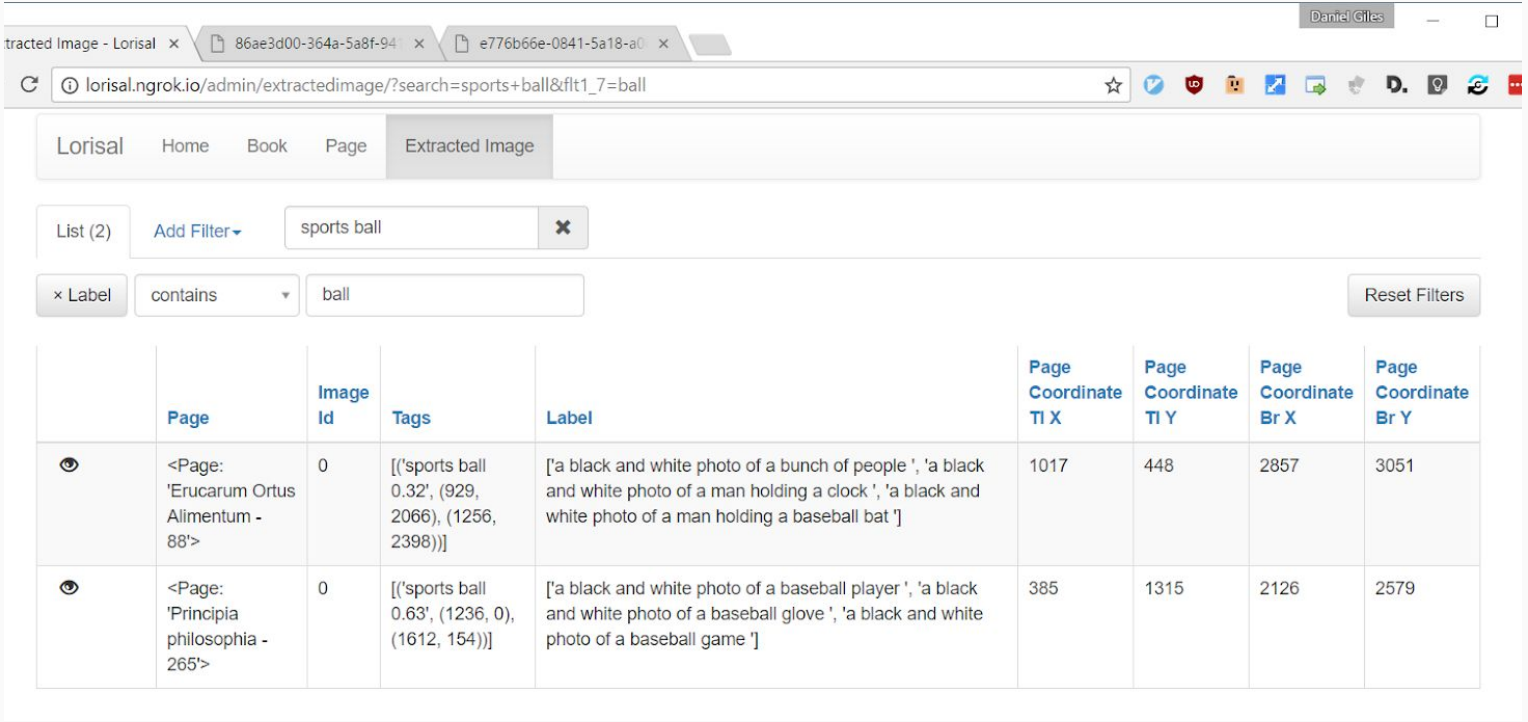
- We’ve got imagery and a database, but what about researchers?
- We need to build an intuitive web application to explore our progress.
- If given the choice to search out web developers or lean on Python’s wonderful ecosystem…

Flask & Flask-Admin

- Flask is a “microframework” that is really effective for quickly deploying webapps.
- Has a built in development server for quick testing
- Flask-Admin is an extension that automatically generates an “admin” backend for CRUD (Create, Read, Update, Delete) datastores
- Flask-Admin will read the model definitions we made from Peewee and automatically generate an easily customizable and expandable backend.
- This backend includes search & filtering!

```
from flask import Flask
import flask_admin as admin
from flask_admin.contrib.peewee import ModelView

class ExtractedImageView(ModelView):
    can_view_details = True
    details_template = 'extract_details.html'
    column_filters = ('tags',
                      'label',
                      'page_coordinate_TL_x',
                      'page_coordinate_TL_y',
                      'page_coordinate_BR_x',
                      'page_coordinate_BR_y',
                      models.Page.ocr_text,
                      models.Page.uuid,
                      )

app = Flask(__name__)

admin = admin.Admin(app, name='Lorisal', template_mode='bootstrap3')

admin.add_view(ExtractedImageView(models.ExtractedImage))
```

Demo


tracted Image - Lorisal
86ae3d00-364a-5a8f-94
e776b66e-0841-5a18-a0
Daniel Giles
lorisal.ngrok.io/admin/extractedimage/?search=sports+ball&flt1_7=ball
Lorisal
Home
Book
Page
Extracted Image
List (2)
Add Filter
sports ball
× Label
contains
ball
Reset Filters
Page
Image Id
Tags
Label
Page Coordinate Tl X
Page Coordinate Tl Y
Page Coordinate Br X
Page Coordinate Br Y
<Page: 'Erucarum Ortus Alimentum - 88'>
0
[('sports ball 0.32', (929, 2066), (1256, 2398))]
['a black and white photo of a bunch of people ', 'a black and white photo of a man holding a clock ', 'a black and white photo of a man holding a baseball bat ']
1017
448
2857
3051
<Page: 'Principia philosophia - 265'>
0
[('sports ball 0.63', (1236, 0), (1612, 154))]
['a black and white photo of a baseball player ', 'a black and white photo of a baseball glove ', 'a black and white photo of a baseball game ']
385
1315
2126
2579

Last Unsung Heros

- pip (and the open source community)
- Awesome Python
- Jupyter Notebooks for rapid prototyping, especially while waiting for multi-day processes!

Links

In no particular order…

https://www.crummy.com/software/BeautifulSoup/
https://keras.io/
https://pjreddie.com/darknet/yolo/
https://github.com/allanzelener/YAD2K
http://docs.peewee-orm.com/en/latest/
http://flask.pocoo.org/
https://flask-admin.readthedocs.io/en/latest/
https://github.com/acdha/image-mining
https://github.com/openpaperwork/pyocr
http://scikit-learn.org/stable/index.html
https://github.com/mcspring/XML2Dict
https://github.com/tensorflow/models/tree/master/im2txt
https://github.com/vinta/awesome-python

Thanks!