AI Studio@RITS

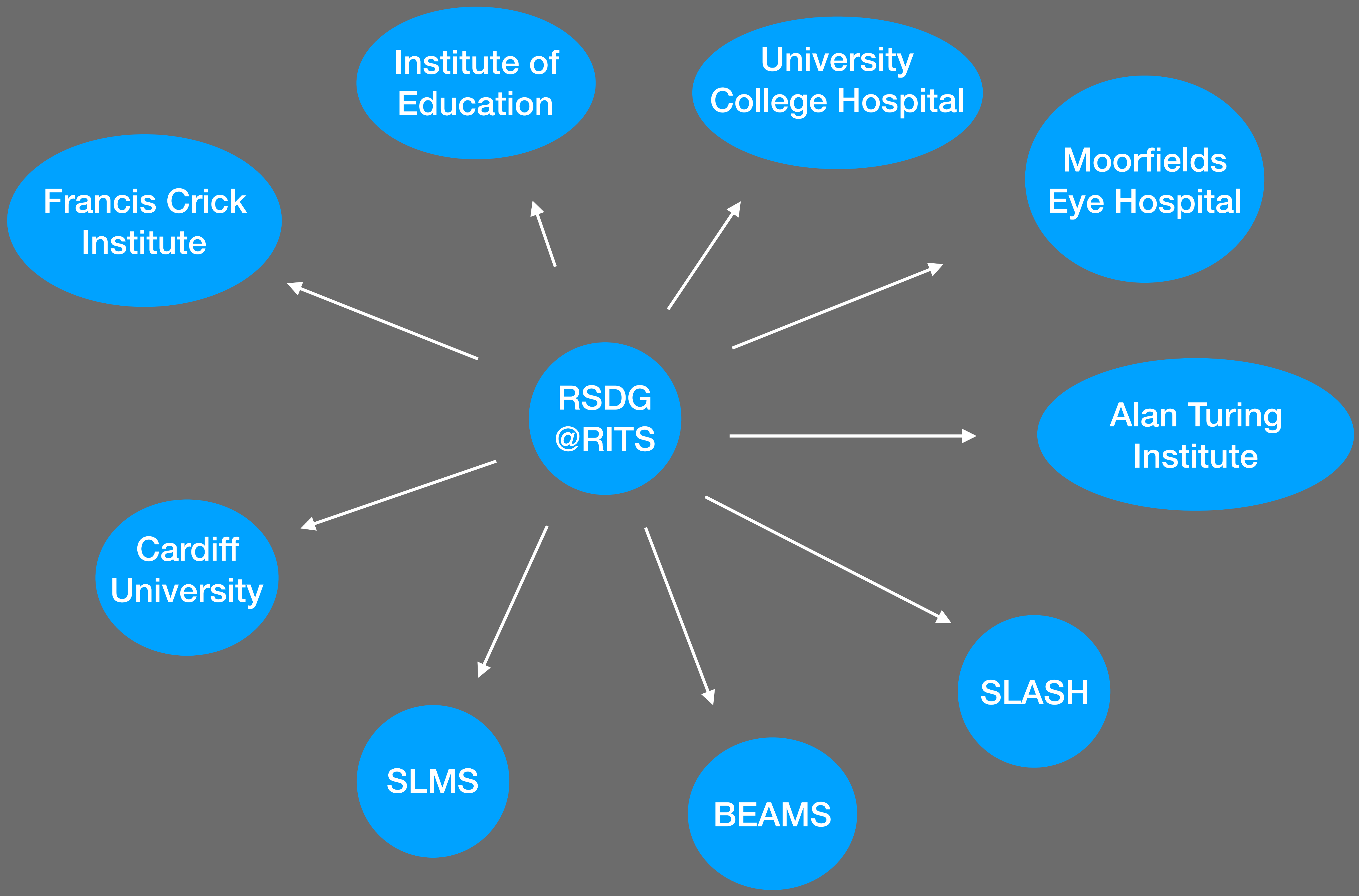
Tech Social-February 2019

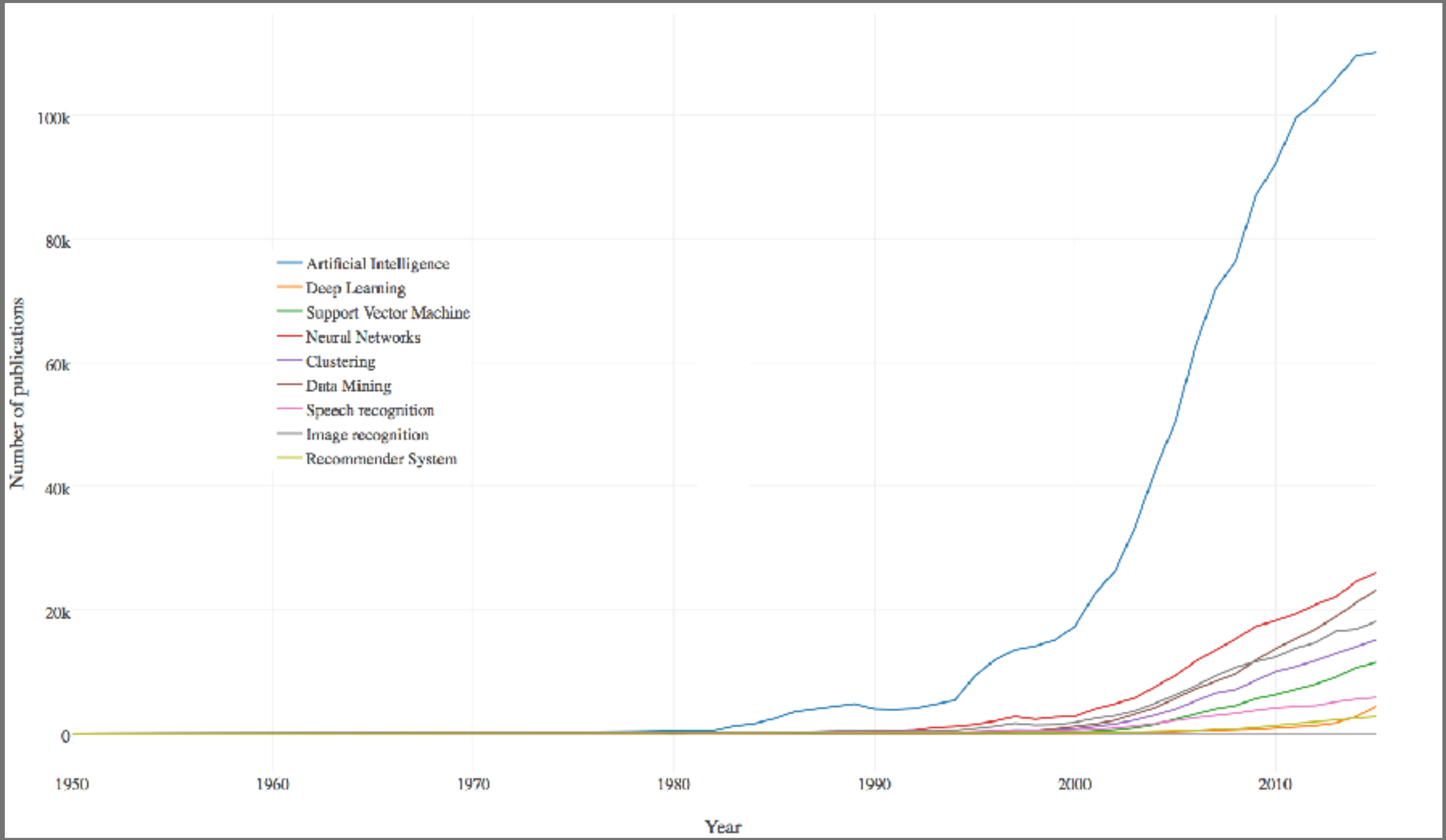
- My name is Sanaz Bayandor Jabbari
- Software engineering
  - Application of Reinforcement learning in Robocup
- Phd in computer science University of Sheffield
  - Statistical Natural language processing, Bayesian perspective
- Leading the R&D team at a tech start-up (Fizzback Ltd)
  - Sentiment analysis and classification of reviews across industries
- Applied computer scientist at Microsoft London
  - Bing, Outlook and Query formulation team
- AI Studio lead at UCL's Research IT Services

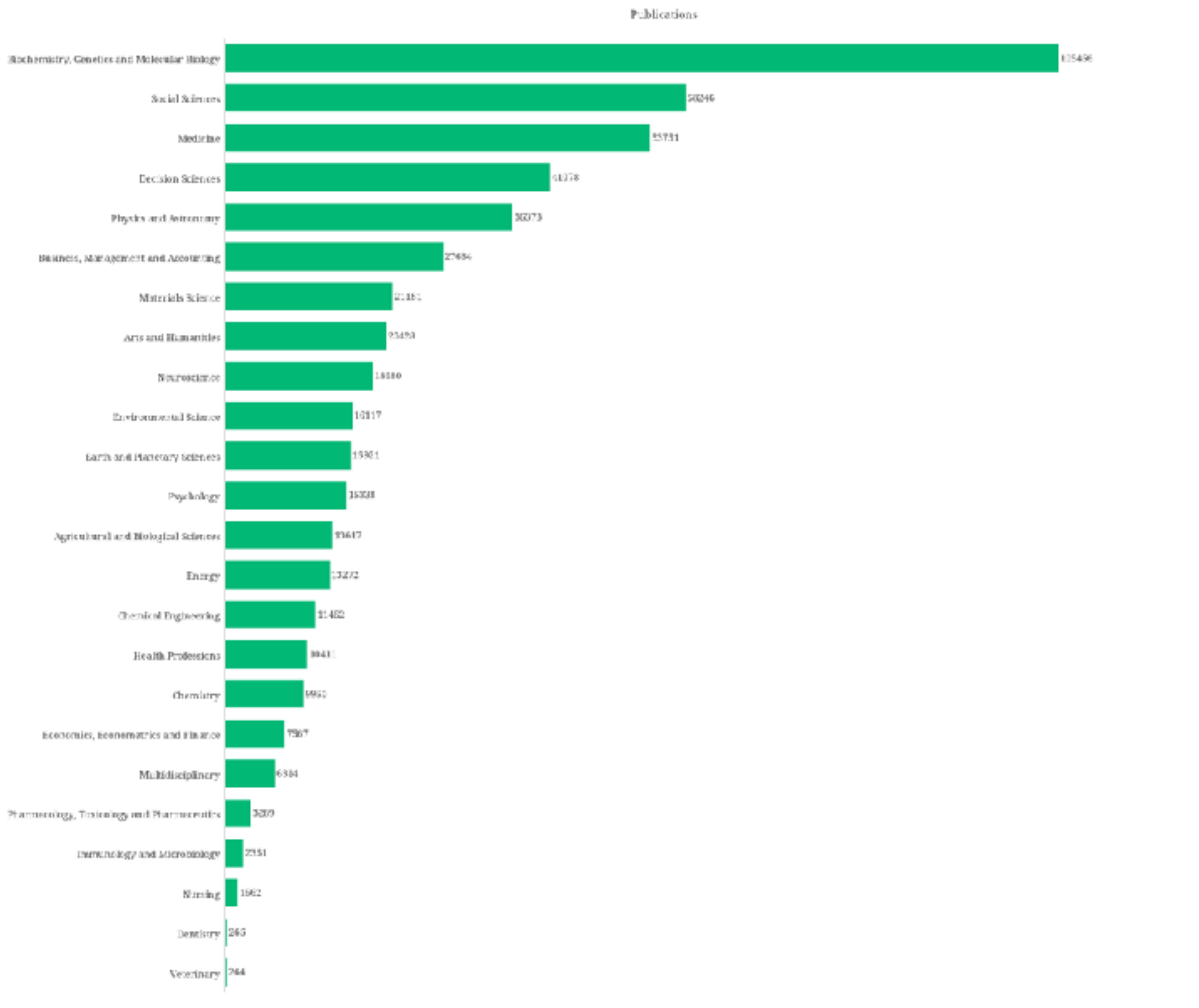
RSDG + AI Studio

- It's a part of **Research Software Development Group** at **RITS**
    - High quality research software
    - Combine academic research experience + software engineering
    - Producing software on behalf of researchers (Refactoring, Testing, Support their codes)
    - Work collaboratively
    - Providing the tools, advice and training researchers
- RSDG launched **AI Studio** in March 2018
    - To provide Data science/Machine Learning and AI services

RSDG collaborations:




100k
80k
60k
40k
20k
0
Number of publications
Artificial Intelligence
Deep Learning
Support Vector Machine
Neural Networks
Clustering
Data Mining
Speech recognition
Image recognition
Recommender System
1950
1960
1970
1980
1990
2000
2010
Year


Publications
Biochemistry, Genetics and Molecular Biology
Social Sciences
Medicine
Decision Sciences
Physics and Astronomy
Business, Management and Accounting
Materials Science
Arts and Humanities
Neuroscience
Environmental Science
Earth and Planetary Sciences
Psychology
Agricultural and Biological Sciences
Energy
13272
Chemical Engineering
11482
Health Professions
Chemistry
Economics, Econometrics and Finance
7567
Multidisciplinary
Pharmacology, Toxicology and Pharmaceutics
Immunology and Microbiology
Nursing
1562
Dentistry
265
Veterinary
264

AI in Healthcare

1. Efficient Diagnosis and Clinical Judgement
   - Incomplete medical histories and large case load
   - In 2015, misdiagnosing illness accounted for 10% of all US deaths.

2. AI-Assisted Robotics Surgery
   - Da vinci

3. Virtual Nursing assistants
   - Care Angel, 24/7, save billions for industry

4. Image analysis

5. Administrative tasks and improving workflow
   - A 2016 study in USA revealed 96% of patient complains over customer service, confusion over paperwork and negative front desk experience

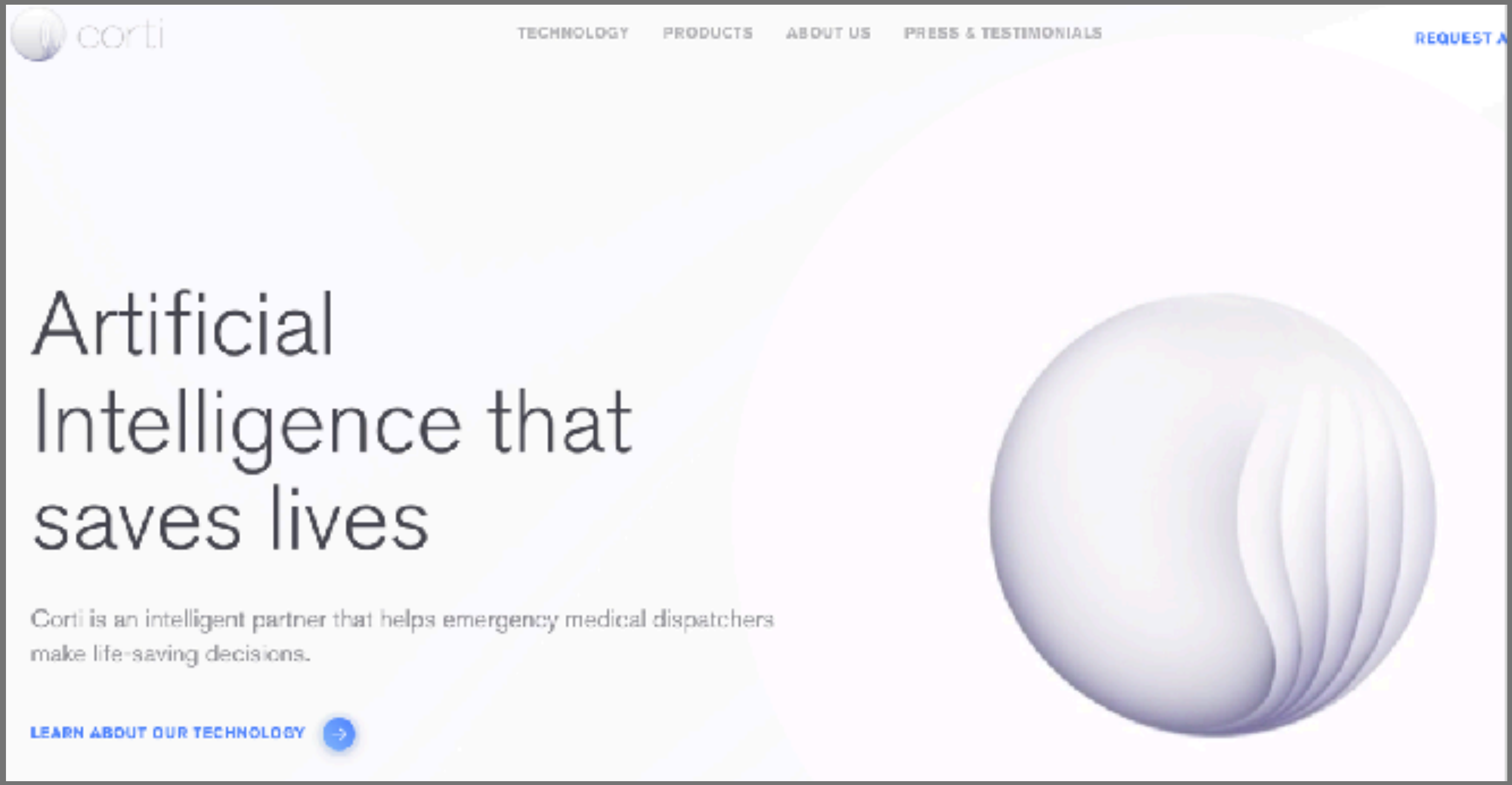
CAMELYON challenge:

Diagnostic Image Analysis Group + Radboud University Medical Centre

- A training data set of whole-slide images “with/without (110/160)”
- Evaluated by a panel of 11 pathologists
- Deep learning models diagnosed breast cancer at a better rate than 11 pathologists.

Table 2. Test Data Set Results of the 32 Submitted Algorithms vs Pathologists for Tasks 1 and 2 in the CAMELYON16 Challenge[a]

	Task 1: Metastasis Identification	Task 2: Metastases Classification		Algorithm Model		
Codename[b]	FROC Score (95% CI)[c]	AUC (95% CI)[c]	P Value for Comparison of the Algorithm vs Pathologists WTC[d]	Deep Learning	Architecture	Comments
HMS and MIT II	0.807 (0.732-0.889)	0.994 (0.983-0.999)	<.001	✓	GoogLeNet[24]	Ensemble of 2 networks; stain standardization; extensive data augmentation; hard negative mining
HMS and MGH III	0.760 (0.692-0.857)	0.976 (0.941-0.999	<.001	✓	ResNet[25]	Fine-tuned pretrained network; fully convolutional network
HMS and MGH I	0.596 (0.578-0.734)	0.964 (0.928-0.989)	<.001	✓	GoogLeNet[24]	Fine-tuned pretrained network
CULab III	0.703 (0.605-0.799)	0.940 (0.888-0.980)	<.001	✓	VGG-16[26]	Fine-tuned pretrained network; fully convolutional network
HMS and MIT I	0.693 (0.600-0.819)	0.923 (0.855-0.977)	.11	✓	GoogLeNet[24]	Ensemble of 2 networks; hard negative mining
ExB I	0.511 (0.363-0.620)	0.916 (0.858-0.962)	.02	✓	ResNet[25]	Varied class balance during training
CULab I	0.544 (0.467-0.629)	0.909 (0.851-0.954)	.04	✓	VGG-Net[26]	Fine-tuned pretrained network



- A Danish AI software company detects cardiac arrests
- By Analysing what a person says, the tone of voice and background noise
- keywords in the caller's description, faint background sounds
- Corti can identify patterns of anomalies or conditions of interest
- 93% success rate compared to 73% for human
- Reduce the number of undetected OHCAs by more than 50%

- Da Vinci Robotic Prostatectomy
- UCH team to do 600 prostate operations a year
- immersive 3D vision
- getting lots of information, little distraction
- no hand tremor, no limitation in terms of fatigue

- Asks simple questions in a conversational manner
- Gather information about physical and emotional well-being
- Measures that includes medication adherence and overall vitals
- The answers are converted to simple Dashboard Notifications, Alerts and reports

Atomwise

Better medicines faster.

- Two distinct problems in drug discovery: "biology and chemistry”
- Biology problem: decide which disease protein is the best one to target
- Chemistry problem: how to deliver a non-toxic molecule that can hit the chosen disease protein
- They developed a neural network called AtomNet, which helps in bioactivity prediction and identifying characteristics of patients for clinical trials.
- Tackles diseases like Ebola and multiple sclerosis.

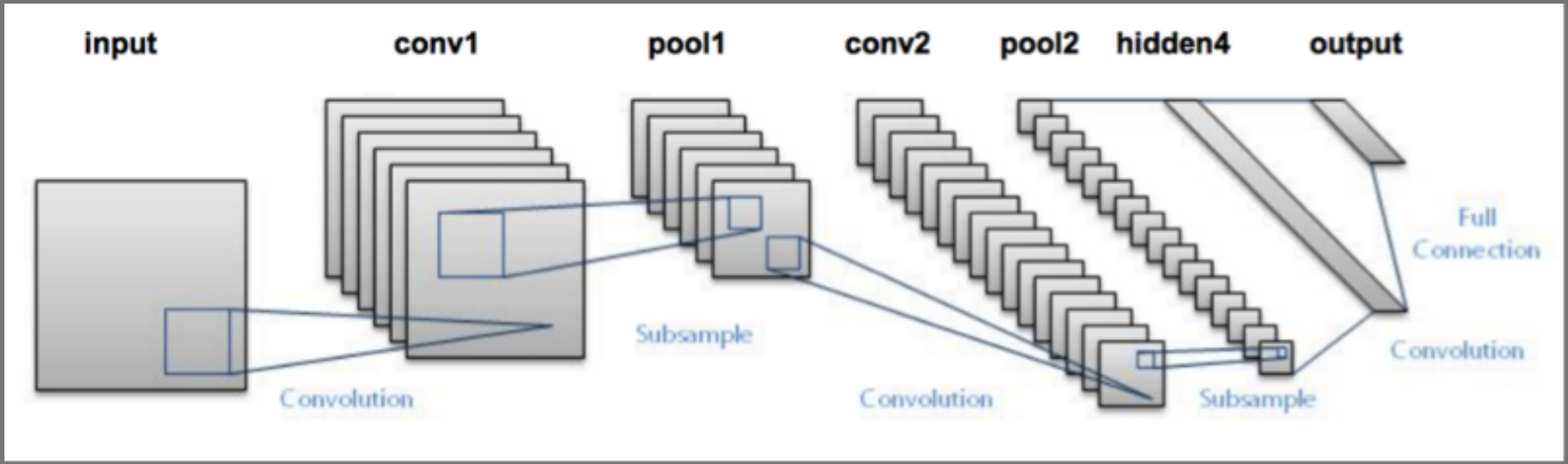
AI in Art

- AI composer that Learns from Mozart Bach, etc.
- Learned its own models of music theory
- Whether you are an amateur or a seasoned professional, AIVA assists you in creative process.
- Uses a combination of Deep learning models and Reinforcement Learning

Google Deep Dream

Generates Psychedelic looking images using CNN

- A base image is used
- Fed to the pre-trained CNN
- Forward pass is done till a particular layer
- Gaussian blurring to produce smooth images



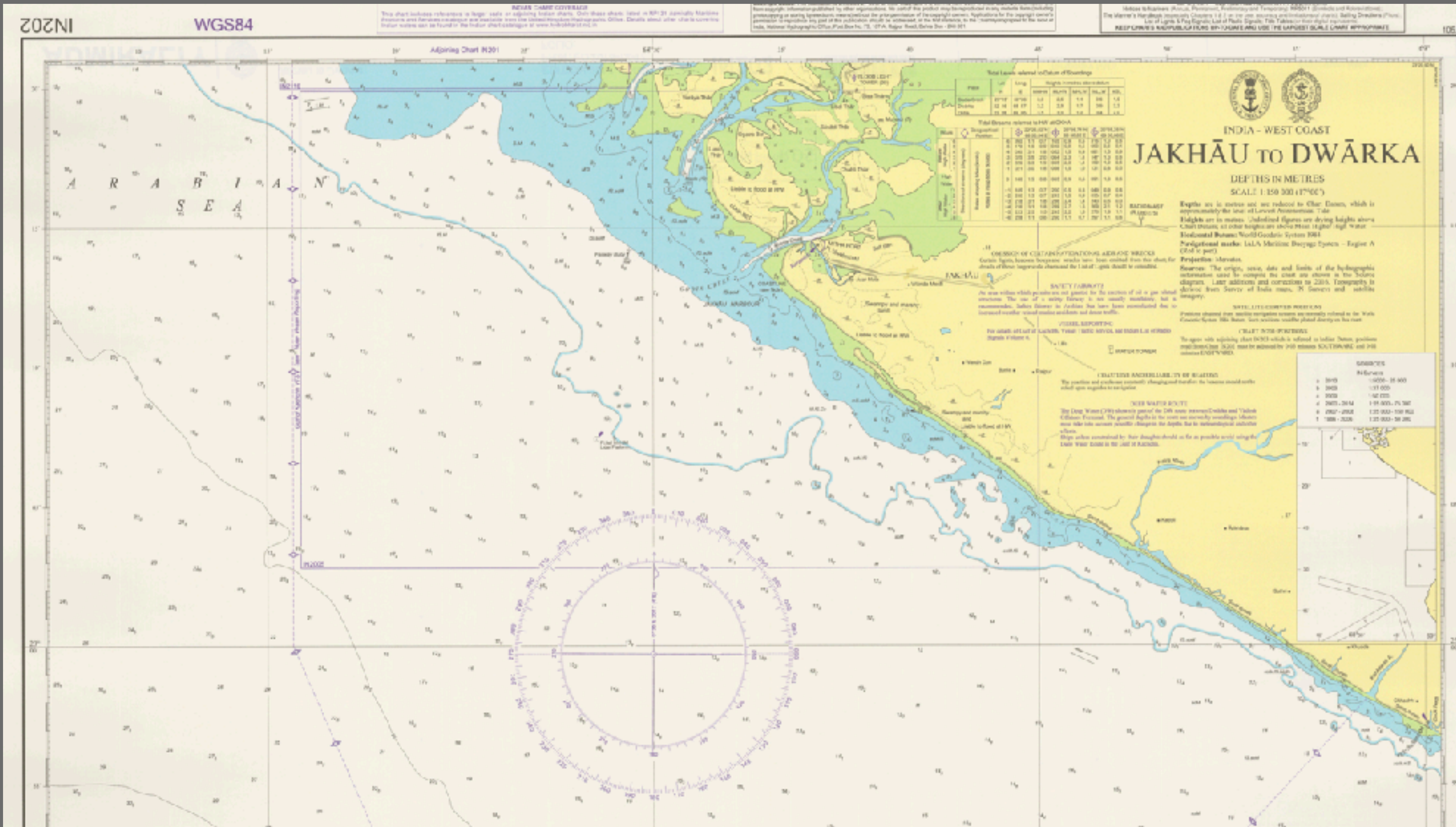
input
conv1
pool1
conv2
pool2
hidden4
output
Convolution
Subsample
Convolution
Subsample
Full Connection
Convolution

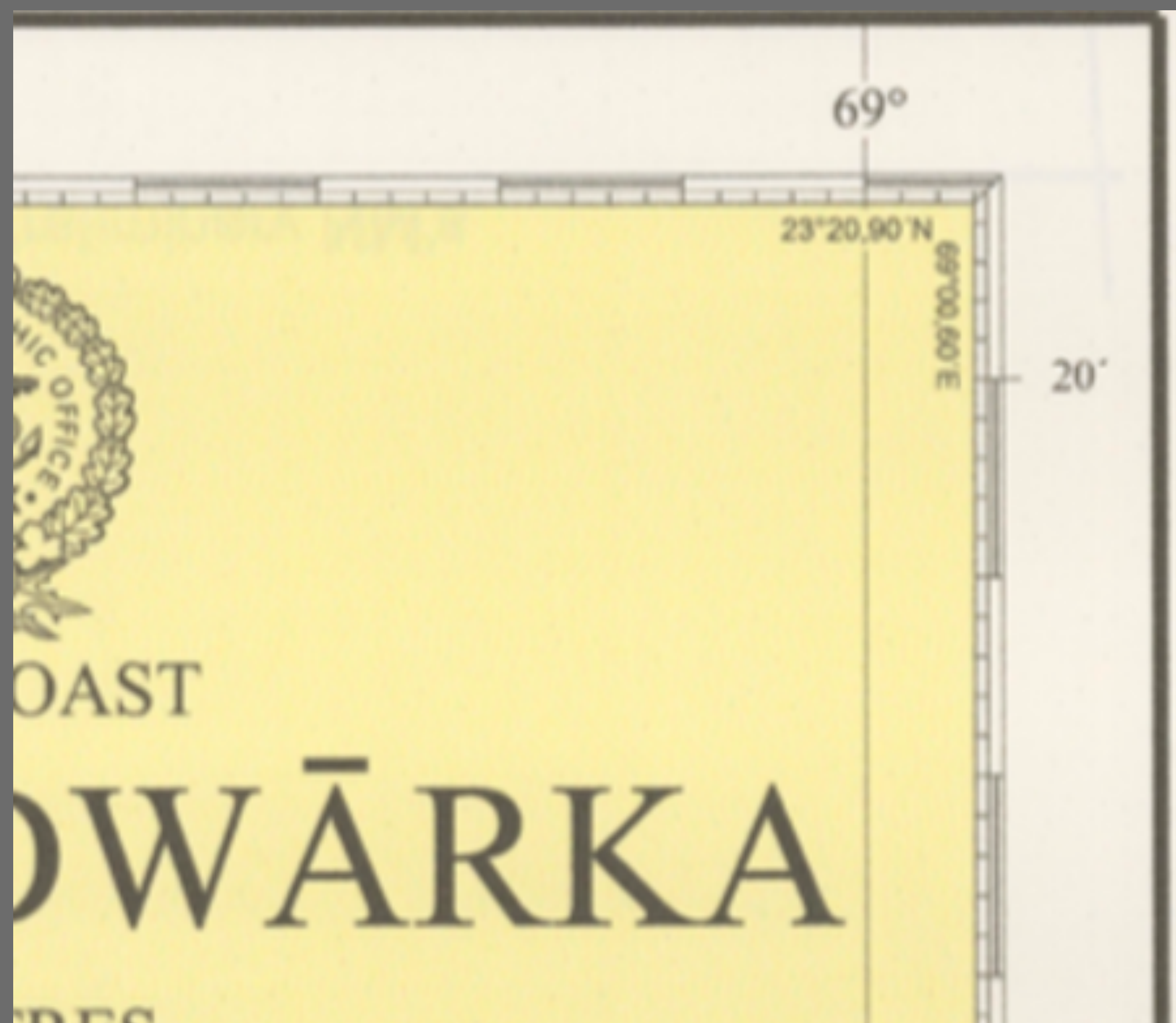
AI in Digital Humanities

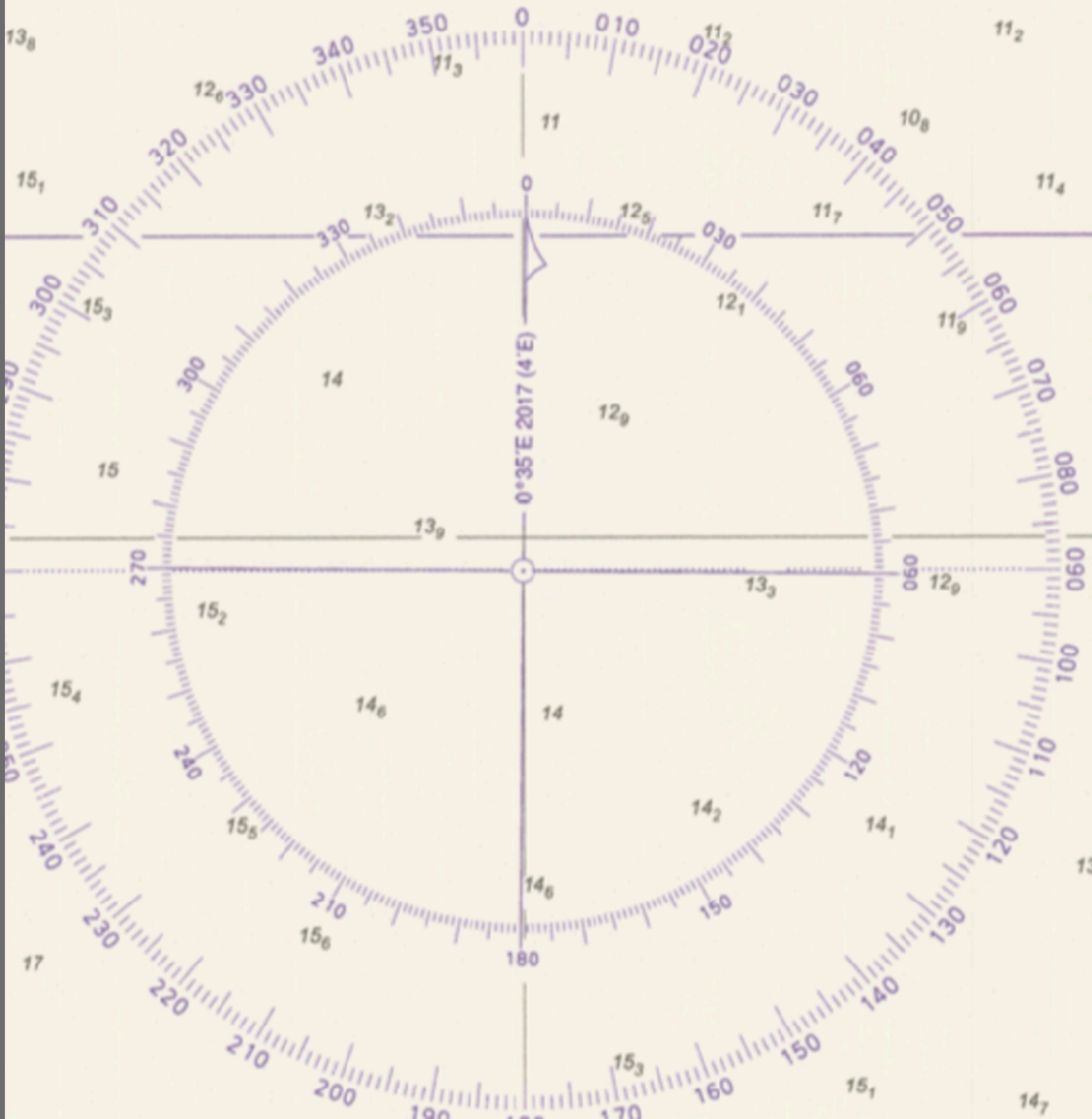
- Developing data-led approaches to answer research questions in the Humanities
- Analysis of our past and present societies
- Use of “Word Embeddings” in NLP:
  - Used in web search and machine translation, works by building up a mathematical representation of language
  - ability to reproduce complex and nuanced word associations
- Challenge: AI programs exhibit racial and gender biases
  - Machine learning algorithms are picking up deeply ingrained race and gender prejudices concealed within the patterns of language us

OCR for Maps

- Funded by Department of Statistics (PI: Serge Guilas)
- Develop a Model to forecast effects of Tsunamis in the coasts of India, and to some extent Pakistan and Iran
- They were using bathymetry data within very low resolution (about 1km), but have now access to scans of maps of 50-90m resolution.
- AI Studio: To extract Bathymetry information from scanned paper maps


WGS84
Adjoining Chart IN201
A R A B I A N
S E A
JAKHĀU
INDIA - WEST COAST
JAKHĀU TO DWĀRKA
DEPTHS IN METRES


69°
23°20,90'N
69°00,60'E
20'
OAST
WĀRKA


0°35'E 2017 (4'E)

JAKHAU
Asar Mata
Wānda Merāi
STLINE
Note)
Swampy and marshy
land
Liable to flood at HW
An area within w
structures. The
recommended. Sa
increased weather
For details of C
Signals Volum
Wandh Don
Budia
Swampy and marshy
land
Liable to flood at HW
fS
mS.soM
M.fS.Co
soM.fS
soM.fS

Automatic extraction of Bathymetry information

1. Localisation/Object(Bathymetry) Detection

2. Transform Normalised coordinates of the bounding box to pixel values

3. Digit detection (OCR)

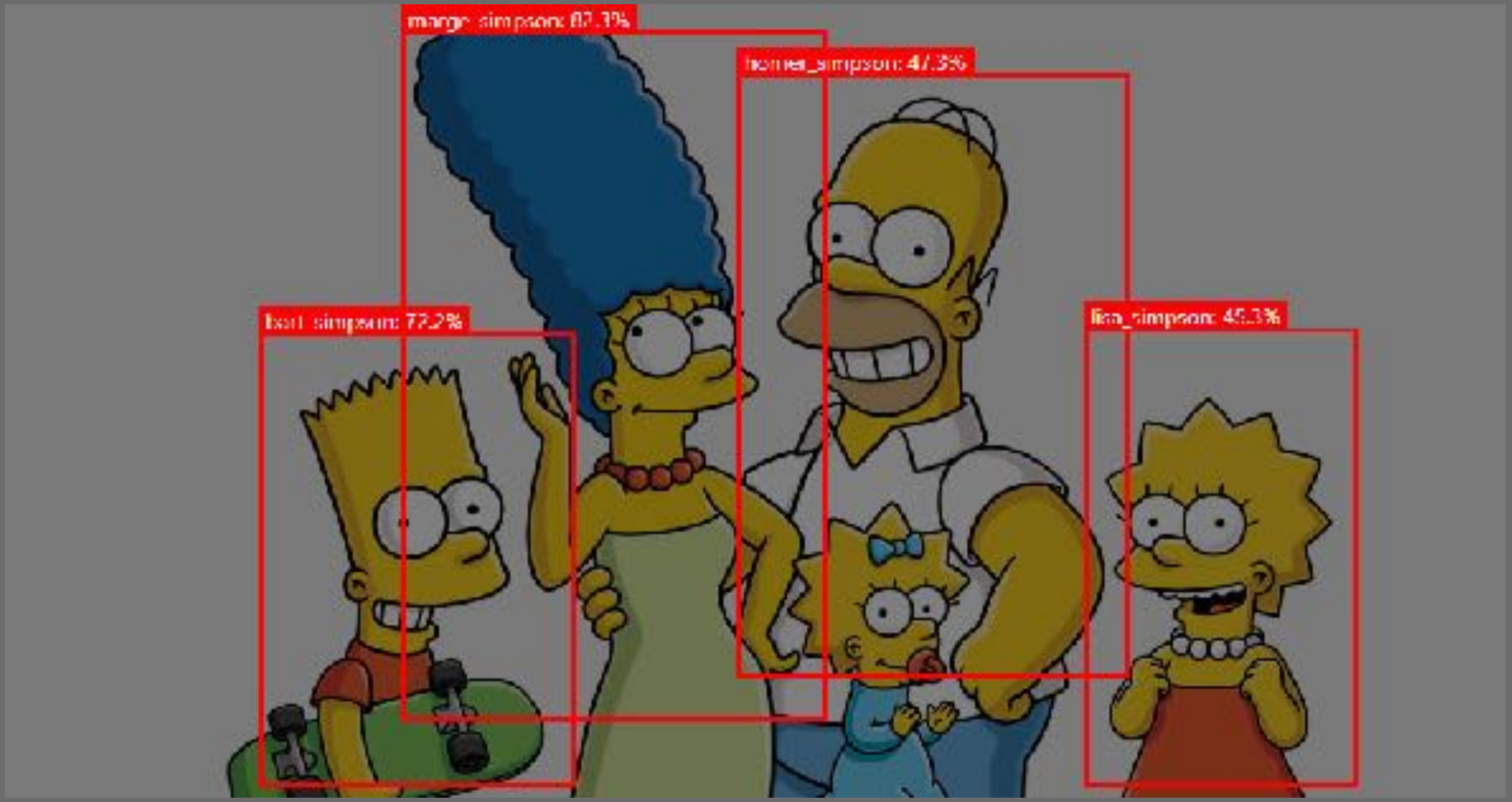
4. Final Mapping to Geographical Information System (GIS)

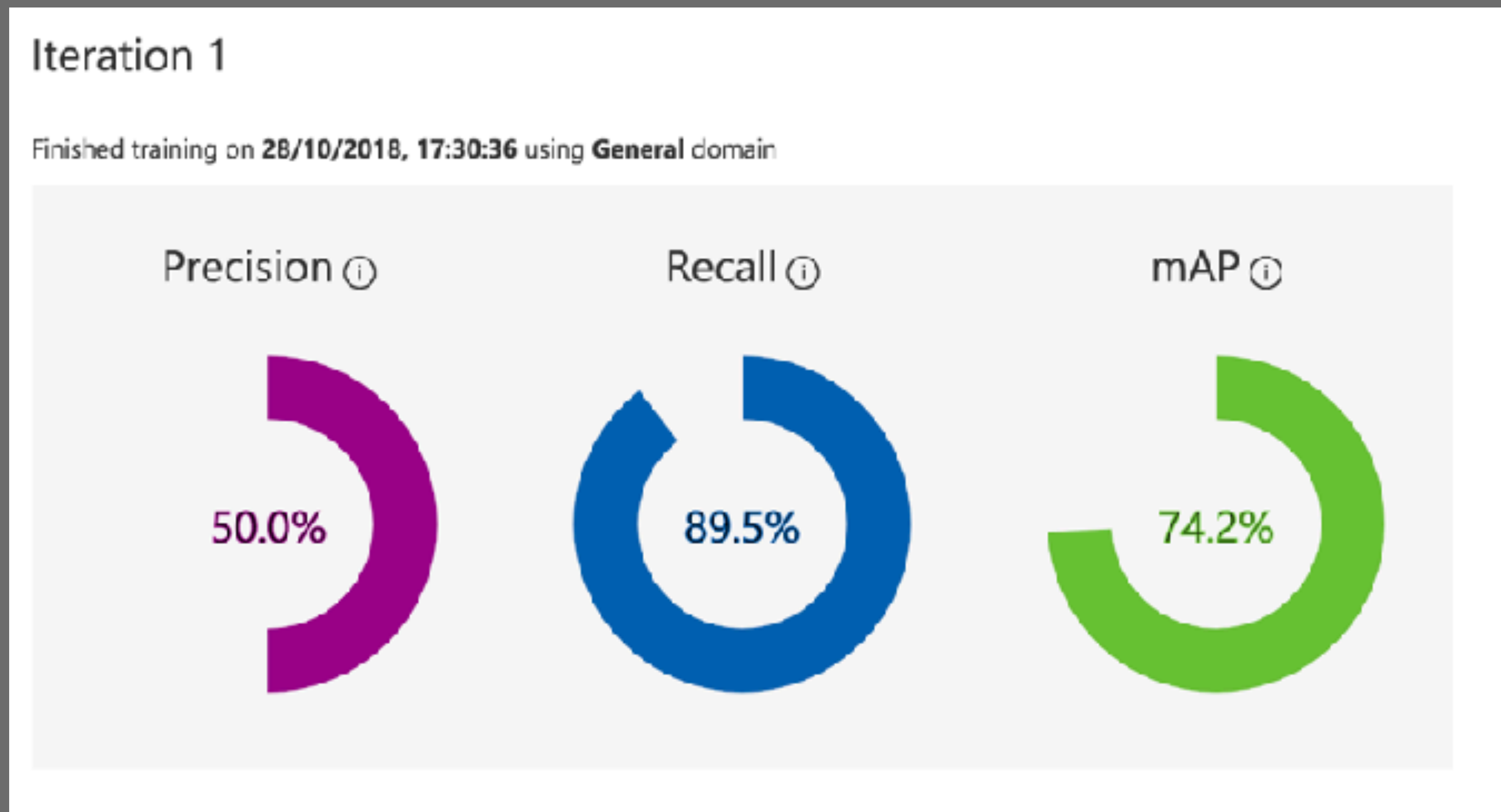
Microsoft Azure Custom Vision service API

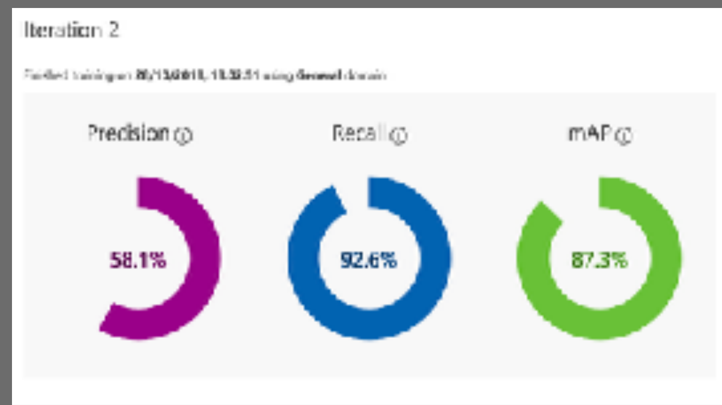
- The Azure Custom Vision API is a cognitive service that lets you build, deploy and improve custom image classifiers.
- It provides a REST API and a web interface to upload your images and train the classifier.
    - Step 1: Create a Custom Vision Service project
    - Step 2: Upload images to the project
    - Step 3: Train the project
    - Step 4: Get and use the default prediction endpoint

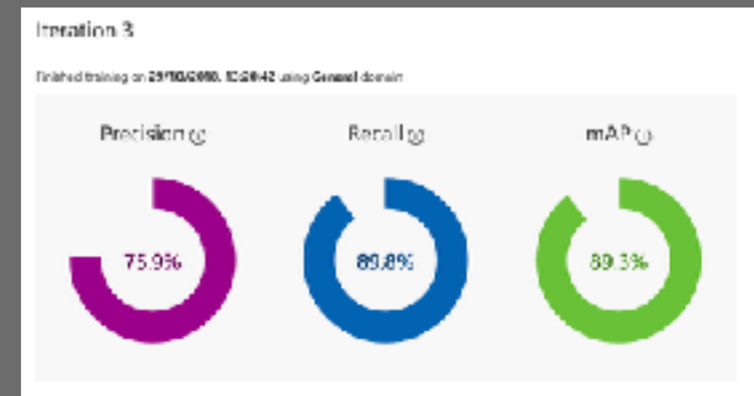
Object Detection Algorithms

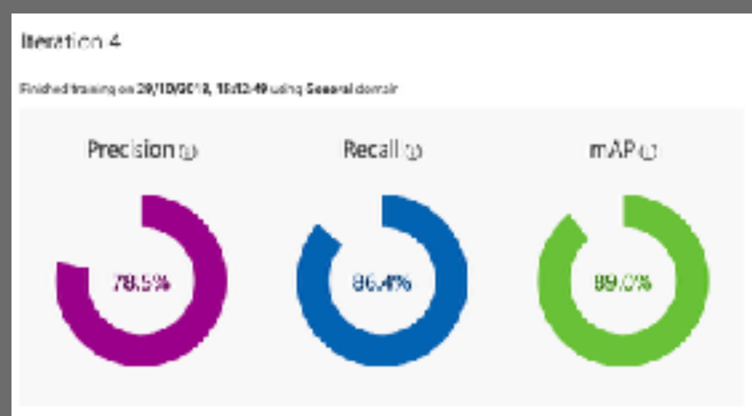
Algorithm	Features	Prediction time / image	Limitations
CNN	Divides the image into multiple regions and then classifies each region into various classes.	–	Needs a lot of regions to predict accurately and hence high computation time.
R-CNN	Uses selective search to generate regions. Extracts around 2000 regions from each image.	40-50 seconds	High computation time as each region is passed to the CNN separately. Also, it uses three different models for making predictions.
Fast R-CNN	Each image is passed only once to the CNN and feature maps are extracted. Selective search is used on these maps to generate predictions. Combines all the three models used in R-CNN together.	2 seconds	Selective search is slow and hence computation time is still high.
Faster R-CNN	Replaces the selective search method with region proposal network (RPN) which makes the algorithm much faster.	0.2 seconds	Object proposal takes time and as there are different systems working one after the other, the performance of systems depends on how the previous system has performed.

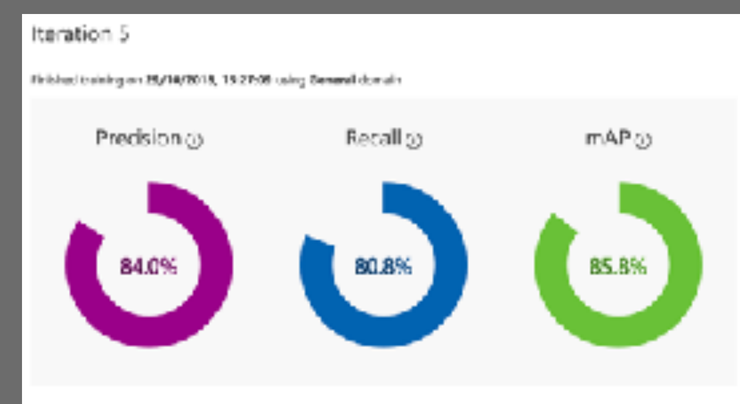

marge_simpson
homer_simpson: 47.3%
bart_simpson: 77.2%
lisa_simpson: 45.3%

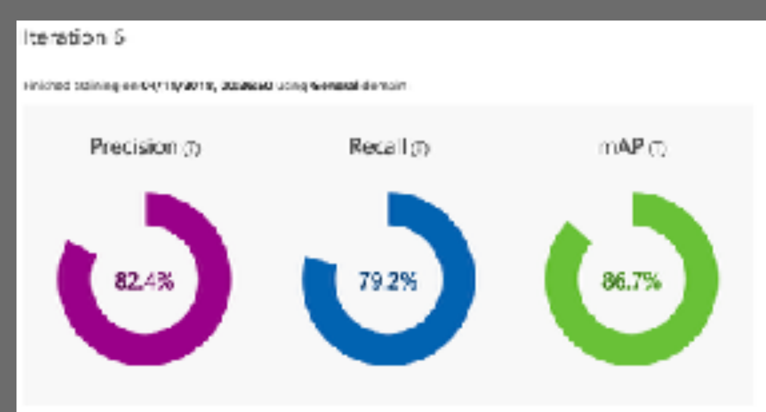

Iteration 1
Finished training on 28/10/2018, 17:30:36 using General domain
Precision
50.0%
Recall
89.5%
mAP
74.2%

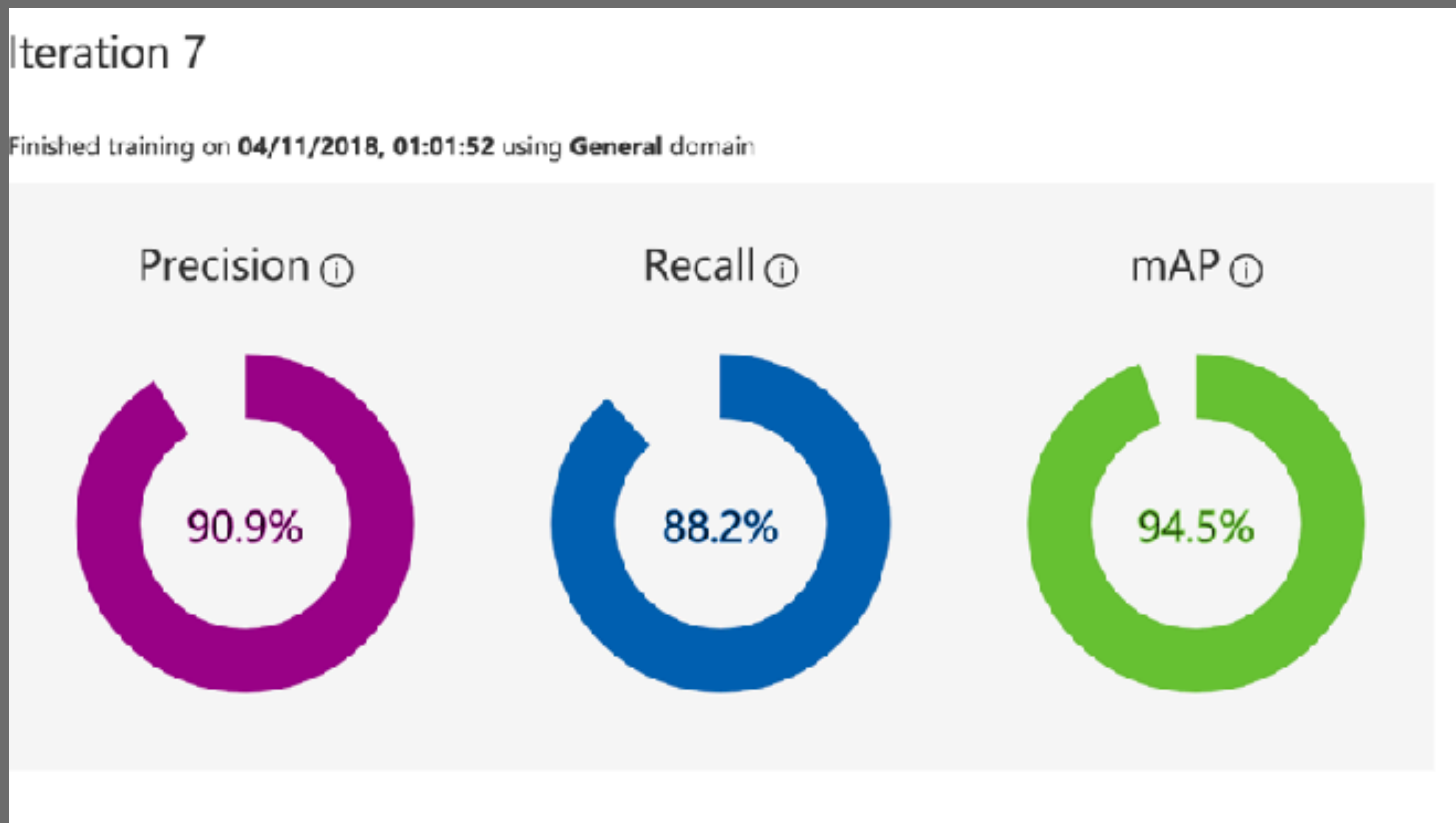

Iteration 2
Precision
Recall
mAP
58.1%
92.6%
87.3%


Iteration 3
Precision
Recall
mAP
75.9%
89.8%
89.3%


Iteration 4
Precision
Recall
mAP
78.5%
86.4%
89.0%


Iteration 5
Precision
Recall
mAP
84.0%
80.8%
85.8%


Iteration 6
Precision
Recall
mAP
82.4%
79.2%
86.7%


Iteration 7
Finished training on 04/11/2018, 01:01:52 using General domain
Precision
90.9%
Recall
88.2%
mAP
94.5%

```python
from azure.cognitiveservices.vision.customvision.training import training_api
from azure.cognitiveservices.vision.customvision.training.models import ImageUrlCreateEntry
```

```python
training_key = "0afd80975f7d4bcfbf41091026f63d18"
prediction_key = "494e7450f6c84f5cab9a19f15aacb8c7"
```

```python
trainer = training_api.TrainingApi(training_key)
```

```python
# Find the object detection domain
obj_detection_domain = next(domain for domain in trainer.get_domains() if domain.type == "ObjectDetection")
```

```python
from azure.cognitiveservices.vision.customvision.prediction import prediction_endpoint
from azure.cognitiveservices.vision.customvision.prediction.prediction_endpoint import models
```

```python
# Now there is a trained endpoint that can be used to make a prediction
predictor = prediction_endpoint.PredictionEndpoint(prediction_key)
```

```python
print(predictor)
```

```
<azure.cognitiveservices.vision.customvision.prediction.prediction_endpoint.PredictionEndpoint object at 0x106329c18>
```

```python
# Open the sample image and get back the prediction results.
with open("Desktop/Map-data/test.png", mode="rb") as test_data:
    results = predictor.predict_image("94cc37fa-2c33-46a6-bec0-4812ee6756e8", test_data, 9)
```

```python
# The results include a prediction for each tag, in descending order of probability - so we'll get the first one
prediction = results.predictions[0].tag_name + ": {0:.2f}%".format(results.predictions[0].probability * 100)
```

```python
prob_threshold = 30
for prediction in results.predictions:
    if(prediction.probability * 100 > prob_threshold):
        print ("\t" + prediction.tag_name + ": {0:.2f}%".format(prediction.probability * 100), prediction.bounding_box.
```

```
bathymetry: 84.47% 0.3687336 0.113785833 0.0427250564 0.06949571
bathymetry: 74.86% 0.893987834 0.05414215 0.03965074 0.07409763
bathymetry: 78.97% 0.05917628 0.18522799 0.0407806337 0.07265636
bathymetry: 51.88% 0.67333734 0.242517665 0.0381319523 0.0492717177
bathymetry: 77.72% 0.7625825 0.294527322 0.0403059721 0.05681169
bathymetry: 84.83% 0.200847253 0.390882343 0.0444975942 0.06986123
bathymetry: 62.79% 0.0 0.580588639 0.0334429145 0.06744242
bathymetry: 83.95% 0.312596381 0.66159004 0.0374214649 0.05937451
bathymetry: 49.13% 0.56536 0.7049216 0.04560536 0.0614088774
bathymetry: 64.97% 0.713270664 0.707325041 0.0405855775 0.0699597
bathymetry: 87.68% 0.146008015 0.878702 0.0421329439 0.06535214
bathymetry: 69.06% 0.4692494 0.8771325 0.0378058851 0.0607491136
bathymetry: 85.89% 0.8516862 0.865169942 0.04507923 0.06342441
```

Digit classification

- Azure’s Computer Vision's optical character recognition (OCR) feature detects text content in an image.
- MNIST dataset, train a deep learning model using tensor flow and keras
- Image processing with OpenCV library:
  - Template Matching!!

OpenCV and template matching

- To find objects in an image using Template Matching
- You will see these functions : cv2.matchTemplate()

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img_rgb = cv2.imread('mario.png')
img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
template = cv2.imread('mario_coin.png',0)
w, h = template.shape[::-1]

res = cv2.matchTemplate(img_gray,template,cv2.TM_CCOEFF_NORMED)
threshold = 0.8
loc = np.where( res >= threshold)
for pt in zip(*loc[::-1]):
    cv2.rectangle(img_rgb, pt, (pt[0] + w, pt[1] + h), (0,0,255), 2)

cv2.imwrite('res.png',img_rgb)
```

Enumerator	
TM_SQDIFF Python: cv.TM_SQDIFF	$R(x,y)=\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2$
TM_SQDIFF_NORMED Python: cv.TM_SQDIFF_NORMED	$R(x,y)=\frac{\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$
TM_CCORR Python: cv.TM_CCORR	$R(x,y)=\sum_{x',y'}(T(x',y')\cdot I(x+x',y+y'))$
TM_CCORR_NORMED Python: cv.TM_CCORR_NORMED	$R(x,y)=\frac{\sum_{x',y'}(T(x',y')\cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$
TM_CCOEFF Python: cv.TM_CCOEFF	$R(x,y)=\sum_{x',y'}(T'(x',y')\cdot I'(x+x',y+y'))$ where $T'(x',y')=T(x',y')-1/(w\cdot h)\cdot\sum_{x'',y''}T(x'',y'')$ $I'(x+x',y+y')=I(x+x',y+y')-1/(w\cdot h)\cdot\sum_{x'',y''}I(x+x'',y+y'')$
TM_CCOEFF_NORMED Python: cv.TM_CCOEFF_NORMED	$R(x,y)=\frac{\sum_{x',y'}(T'(x',y')\cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'}T'(x',y')^2\cdot\sum_{x',y'}I'(x+x',y+y')^2}}$

OCR Maps using opencv:

1. Make templates:

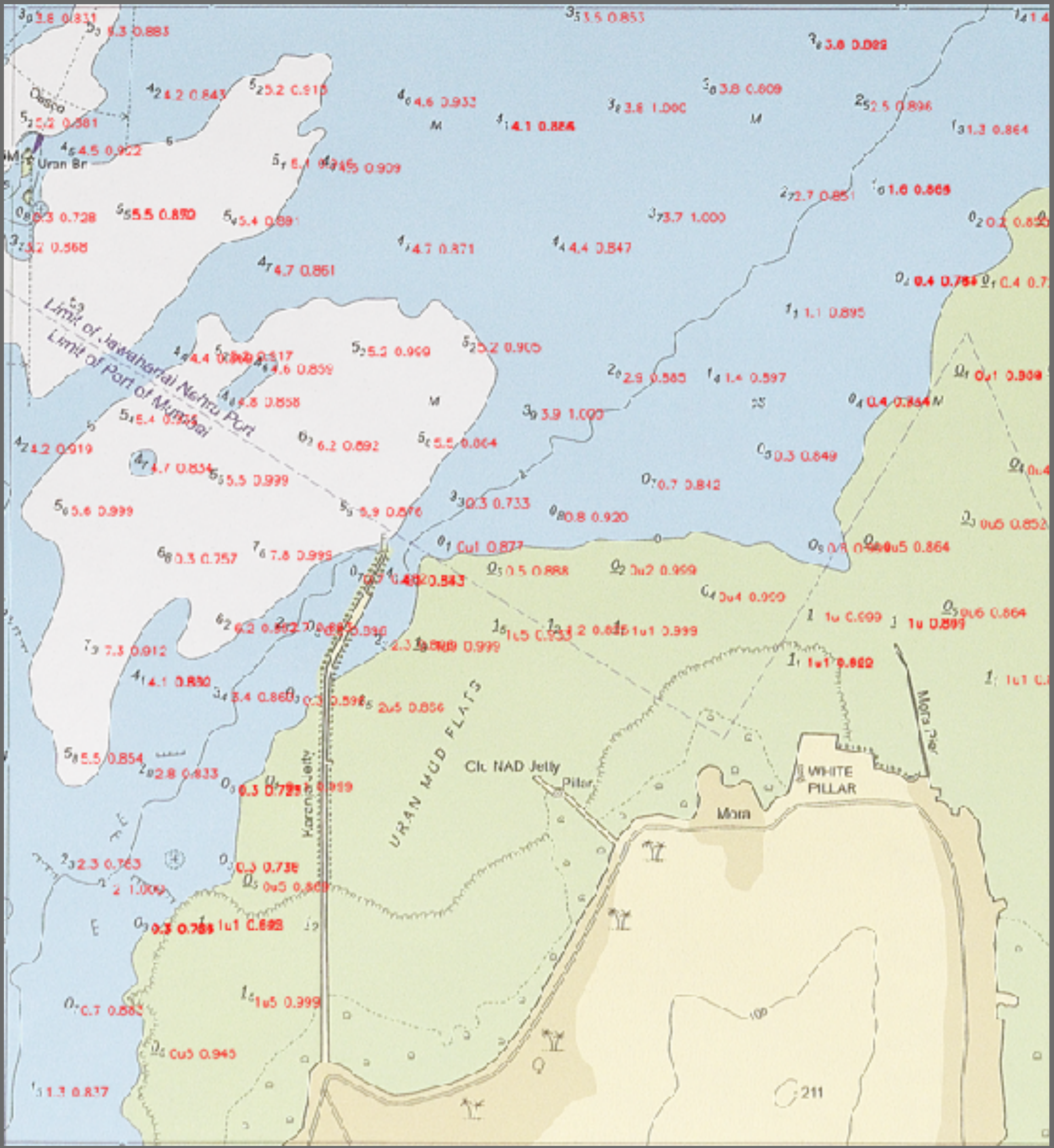
2. Make config file: Pick an optimal threshold for each template

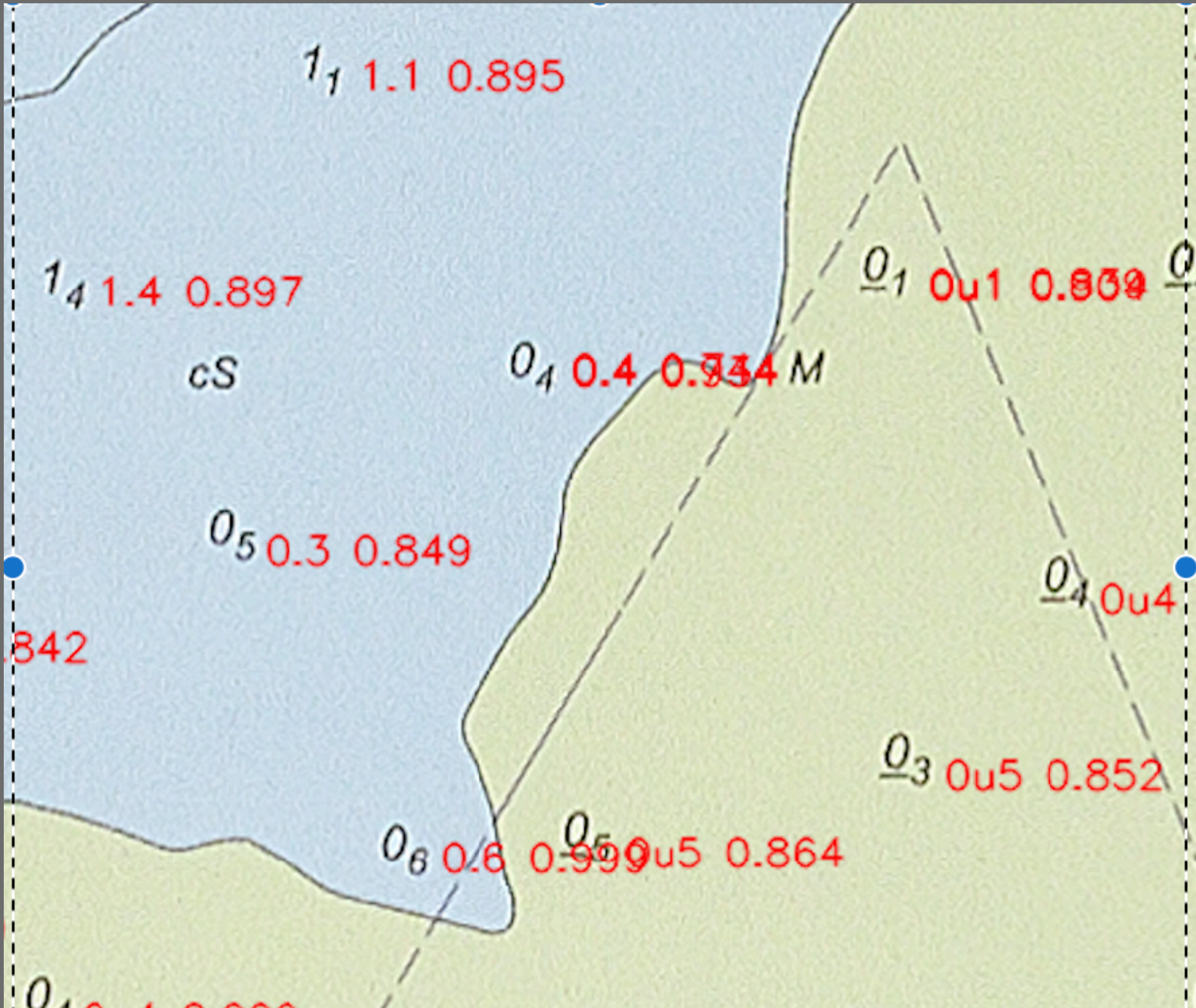
3. Match templates using the suitable method in opencv

4. Deal with the confusions

```
(13,)  points are detected as  0.1
(8,)  points are detected as  0.2
(12,)  points are detected as  0.3
(20,)  points are detected as  0.4
(18,)  points are detected as  0.5
(12,)  points are detected as  0.6
(17,)  points are detected as  0.7
(2,)  points are detected as  10.2
(1,)  points are detected a
(1,)  points are detected a
(8,)  points are detected a
(10,)  points are detected
```

```
(13,)  points are detected as  0.1
4049 , 2436
4194 , 2497
2688 , 2506
4072 , 2513
4098 , 2645
4121 , 2862
4360 , 3566
4546 , 3587
3313 , 3854
32 , 3895
4173 , 5132
4273 , 5200
877 , 6864
```


Uran Br
Limit of Jawaharlal Nehru Port
Limit of Port of Mumbai
URAN MUD FLATS
Pillar
WHITE
PILLAR
Mora
Mora Pier
211



1_1 1.1 0.895
1_4 1.4 0.897
cS
0_4 0.4
M
0_5 0.3 0.849
.842
0_1 0u1
0_4 0u4
0_3 0u5 0.852
0_6 0.6
0u5 0.864

Long/Lat conversion

```
0.1
0.88   (0, 0, 255)
(13,)  points are detected as  0.1
4049 , 2436
4194 , 2497
2688 , 2506
4072 , 2513
4098 , 2645
4121 , 2862
4360 , 3566
4546 , 3587
3313 , 3854
32 , 3895
4173 , 5132
4273 , 5200
877 , 6864
0.2
0.88   (0, 0, 255)
(8,)  points are detected as  0.2
```

```
    - botton_leff_lon: <9 56.40>
<IN2004_Kochi_Harbour.tif>:
    - width: <number of pixels>
    - height: <number of pixels>
    - top_left_lat: <76 18.00>
    - top_left_lon: <10 02.40>
    - bottom_left_lat: <76 04.60>
    - botton_leff_lon: <9 53.80>
 <40.png>:
    - width: 23810
    - height: 15164
    - top_right_lat: <67 00.21>
    - top_right_lon: <24 51.12>
    - bottom_left_lat: <66 55.45>
    - botton_leff_lon: <24 44.26>
```

DMS 67 00 13
DM.m 67 00.21
D.d 67.0035

The formulas are as follows:

Degrees Minutes Seconds to Degrees Minutes.m
Degrees = Degrees
Minutes.m = Minutes + (Seconds / 60)

Degrees Minutes.m to Decimal Degrees
.d = M.m / 60
Decimal Degrees = Degrees + .d

Find the number of degrees per image pixel size

Moorfields Eye hospital

- Categorisation and relation extraction in Glaucoma related letters from Moorfields Eye hospital
- About 650k Letter with mention of the word “Glaucoma”

A. Categorise Glaucoma related letters vs Non-Glaucoma letter

B. Extract the correct information for the right and left eye, ie. relation extraction from highly unstructured text

A glaucoma follow up appointment

Diagnosis:
High myopia both eyes
Normal pressure glaucoma suspect both eyes
Previous cataract surgery

Examination:
Visual acuity 6/9 glasses right 6/6 left
The intraocular pressure measured 18mmHg in both eyes

Impression:
Tilted myopic discs. Thinning of neuroretinal rims. Stable superior arcuate scotoma left and enlarged blind spot right. She is happy to continue on topical medications as normal tension glaucoma cannot be ruled out.

Treatment:
The treatment that has been prescribed: G. timolol BD both eyes until review

Follow up:
A follow up has been arranged in 1 year.

A glaucoma follow up who also has AMD

The above patient was reviewedin the glaucoma clinic today. She has had an FFA in June and has bilateral PEDs. She does not seem to be under MR follow up. She has narrow angles as well as OHT. Please would you review her.

Visual acuity: Right eye 6/36 left eye 6/36
IOP: Right eye 14mmHg, left eye 17mmHg
Optic disc: Right eye 0.2, left eye 0.2

The treatment has been changed to latanoprost nocte bothe eyes. IOP is well controlled.
Treatment: please continue
The patient will be reviewed in 6 months time.

A glaucoma follow up appointment

Diagnosis:
Primary open angle glaucoma
Right eye blind (old injury)

Pressures are much improved today at R13 and L15 on the current system of drops administered by a carer. The left eye has been stable for many years.

Investigations: Feids left eye stable stable 20 years with subtle arcuate defect
Medications prescribed: Continue latanoprost BE nocte
Follow up: review in 6 months with fields

NLP Toolkits and Libraries

- TextBlob
  - Sentiment Analysis, Noun phrase detection, translation using Google API, etc.
- Spacy
  - Industrial-strength Nlp, Name Entity Recognition, Dependency parsing
  - Built-in visualiser for syntax and NER
- NLTK
  - Interfaces to WordNet and other resources, parsers, tokenisers, semantic reasoning

Relation Extraction

- Extracting structured information from unstructured sources such as raw text.
- Chunking and noun phrase extraction, Part-of-speech tagging, parsing
- Steps required: Entity Linking (Co-reference analysis)
- DeepDive project@Stanford

Project Works: ML focus

- Natural language processing
  - Translation, relation extraction, name entity recognition, sentiment analysis
- Image processing
  - Digit recognition, segmentation, object detection
- AI in Healthcare
  - Medical imaging, Electronic record analysis, data-driven predictions
- Deep learning methods & tools for big datasets
  - Using state-of-the-art toolkits such as TensorFlow & Keras
- Reinforcement learning application

Training Activities

- Software and Data Carpentry

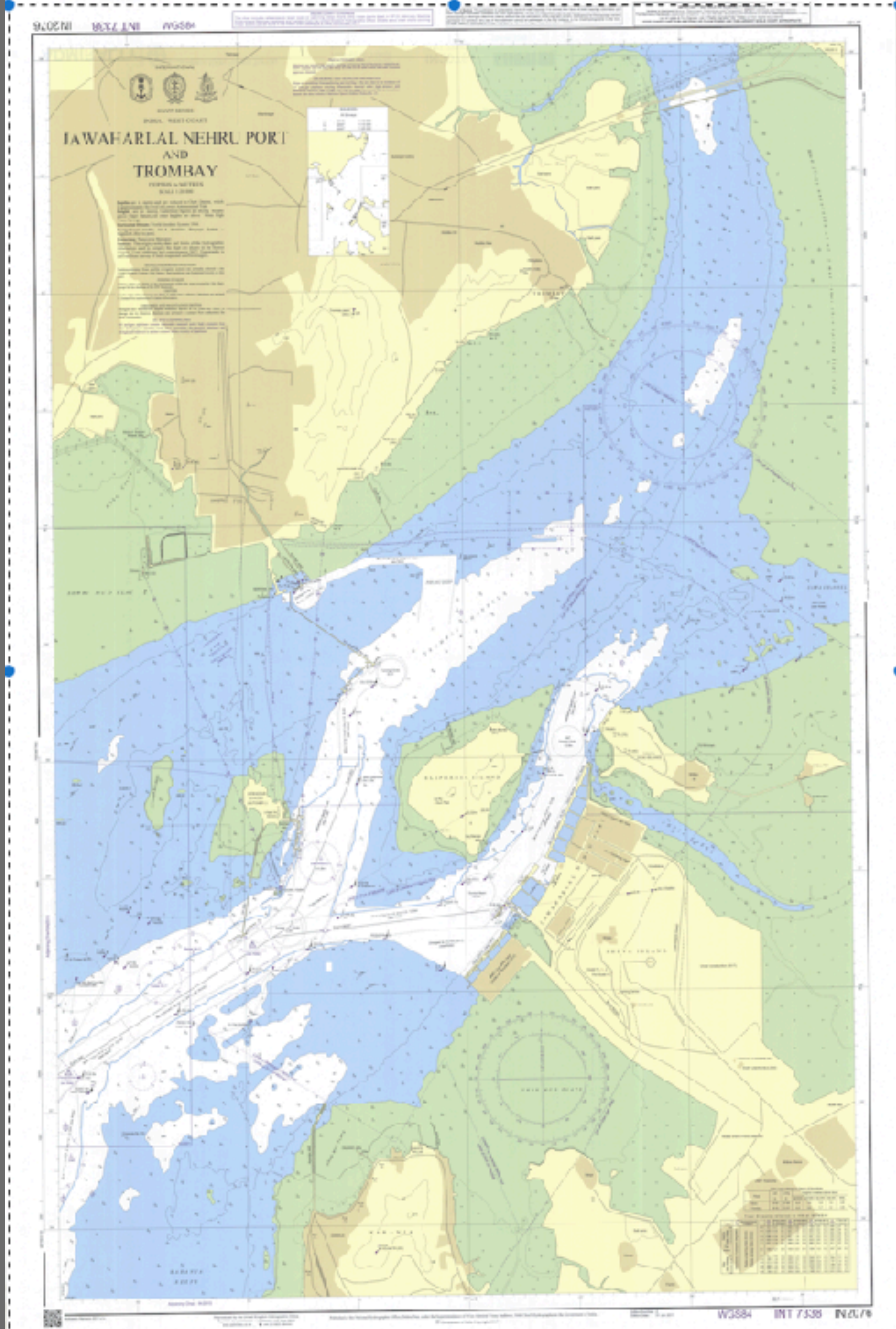
- Centre for Doctoral Training (CDT) - Data Intensive Science track
  - Intent modelling for ASOS Chat Data

- Crick Network Fund:
  "Building a Knowledge Exchange Hub for data-driven and machine learning-based large-scale biomedical image analysis", with Imperial and Kings College London.

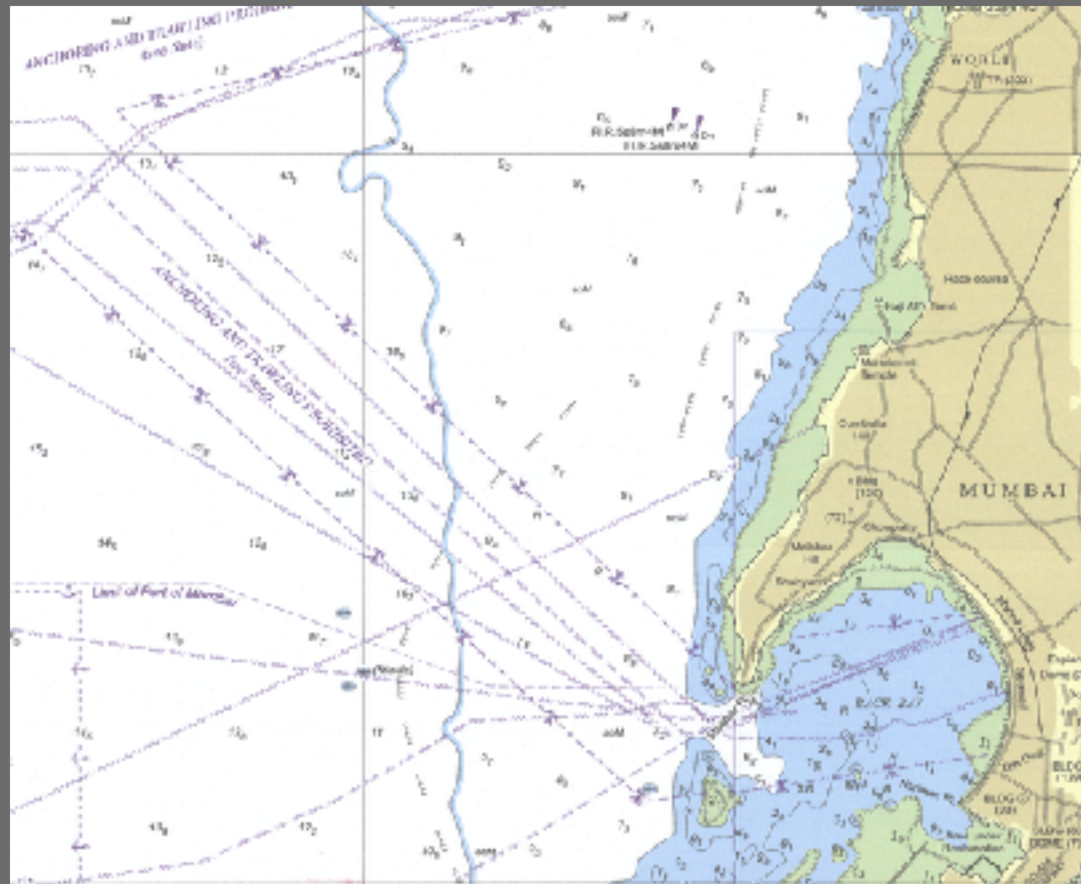
How to find us

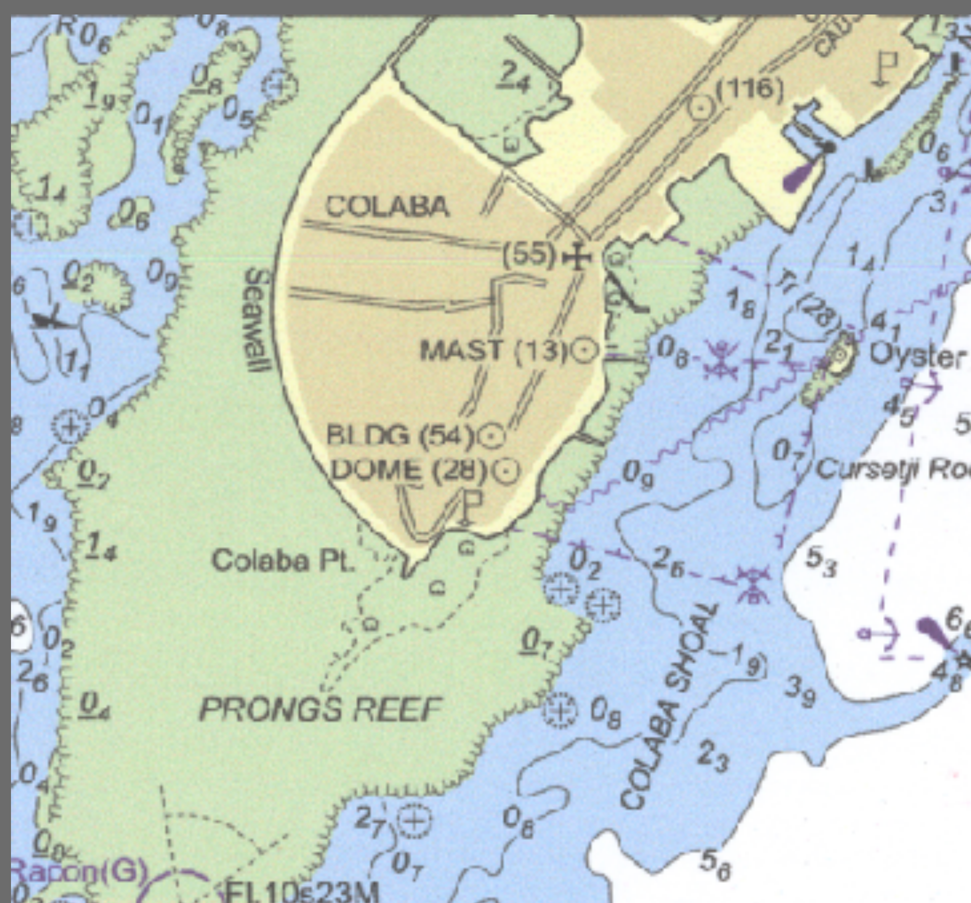
- Email me: “s.jabbari@ucl.ac.uk"
- Email: “rc-softdev@ucl.ac.uk"
- Level 6, 1-19 Torrington Pl, Fitzrovia, London WC1E 7HB

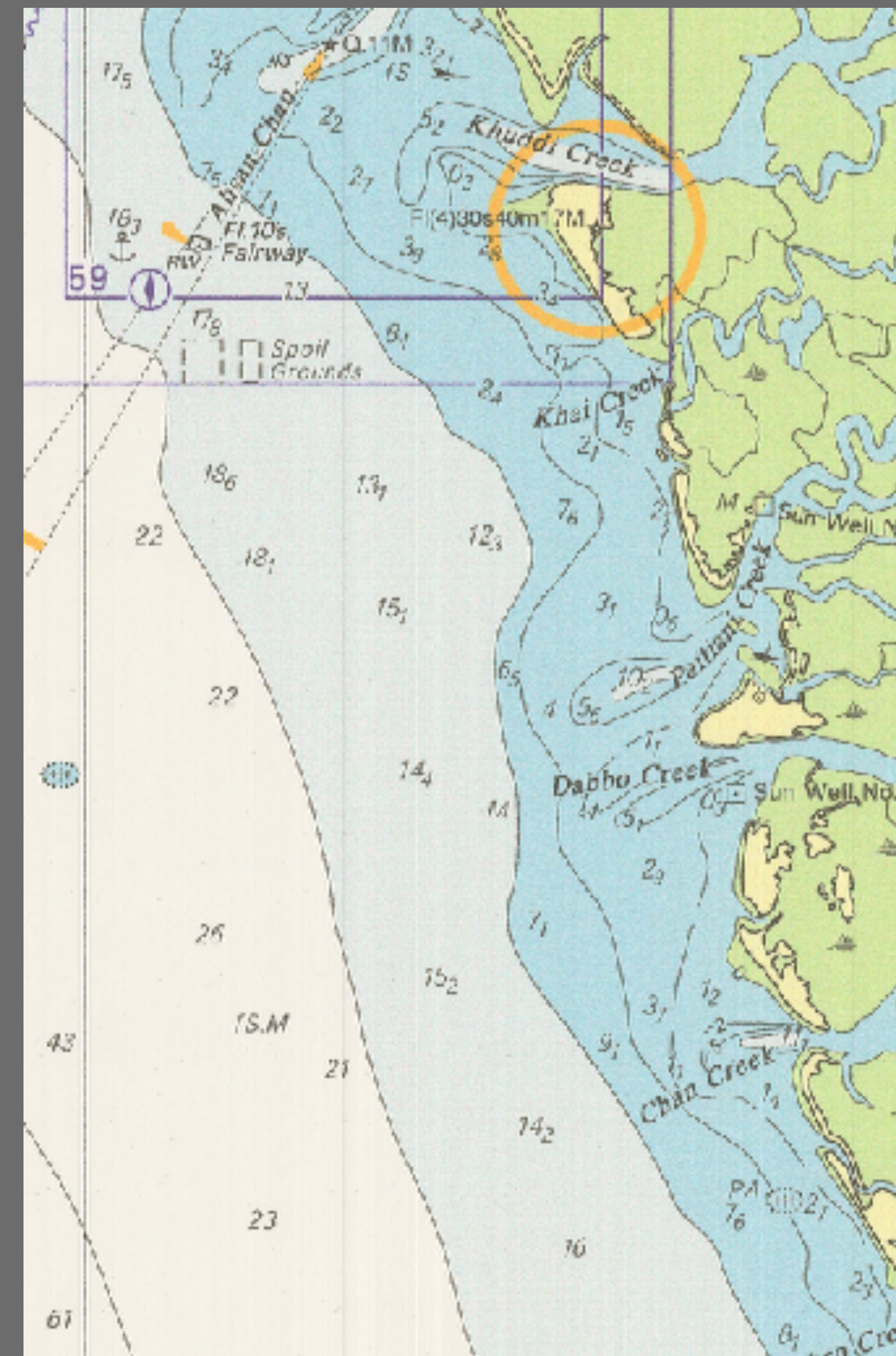
Enumerator	
TM_SQDIFF Python: cv.TM_SQDIFF	$R(x,y)=\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2$
TM_SQDIFF_NORMED Python: cv.TM_SQDIFF_NORMED	$R(x,y)=\frac{\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$
TM_CCORR Python: cv.TM_CCORR	$R(x,y)=\sum_{x',y'}(T(x',y')\cdot I(x+x',y+y'))$
TM_CCORR_NORMED Python: cv.TM_CCORR_NORMED	$R(x,y)=\frac{\sum_{x',y'}(T(x',y')\cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$
TM_CCOEFF Python: cv.TM_CCOEFF	$R(x,y)=\sum_{x',y'}(T'(x',y')\cdot I'(x+x',y+y'))$ where $T'(x',y')=T(x',y')-1/(w\cdot h)\cdot\sum_{x'',y''}T(x'',y'')$ $I'(x+x',y+y')=I(x+x',y+y')-1/(w\cdot h)\cdot\sum_{x'',y''}I(x+x'',y+y'')$
TM_CCOEFF_NORMED Python: cv.TM_CCOEFF_NORMED	$R(x,y)=\frac{\sum_{x',y'}(T'(x',y')\cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'}T'(x',y')^2\cdot\sum_{x',y'}I'(x+x',y+y')^2}}$



JAWAHARLAL NEHRU PORT
AND
TROMBAY
W358A
INT 7358
IN2076


MUMBAI


COLABA
(116)
(55)
Seawall
MAST (13)
BLDG (54)
DOME (28)
Colaba Pt.
PRONGS REEF
COLABA SHOAL
Racon(G)
Fl.10s23M


Khuddi Creek
Fl(4)30s40m17M
Fl.10s
Fairway
59
Spoil
Grounds
Khai Creek
Dabbo Creek
Sun Well No.3

AI in Physics

Particle Physics:

- One of the biggest physics discoveries, the Higgs boson particle or “God particle”, was discovered using the neural network.

- Large Hadron Collider (LHC) , largest particle accelerator in the world, in an hour it generates approximately as much data as Facebook handles in a year.

Objection Detection Evaluation Metric

- the Average Precision (AP) metric is a summary of the precision-recall curve
- Commonly used metric used for object detection challenges is called the mean Average Precision (mAP)
- The mAP metric avoids to have extreme specialisation in few classes and thus weak performances in others.
- The mAP score is usually computed for a fixed IoU (Intersection over Union)