

Object extraction techniques and visual image search with Semantic web techniques

Aninda Maulik

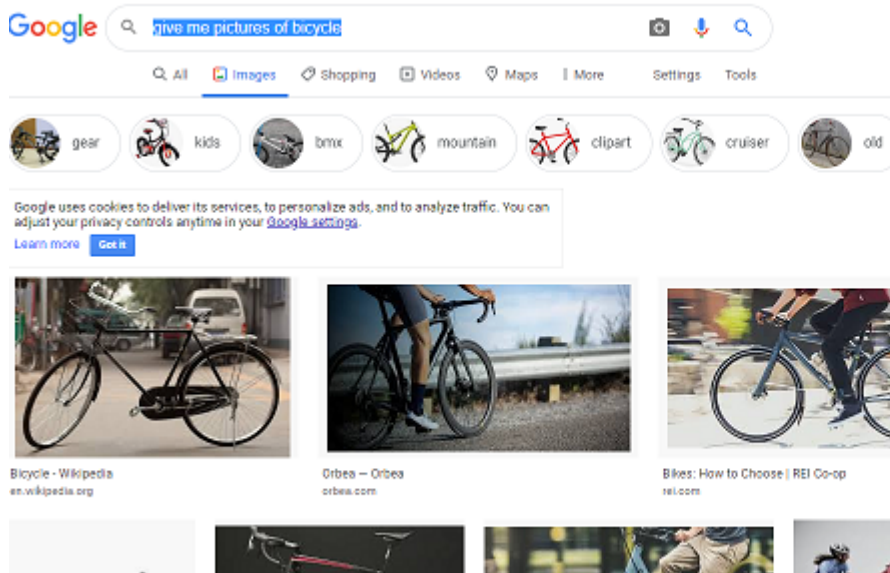
Supervisors: Prof. Pierre Maret
Dennis Diefenbach

Cyber Physical and Social Systems
University of Jean Monnet

July 2020

Introduction: Just Google

- images with bicycles



images with bicycles on the right



images with bicycles on the right



All

Images

News

Videos

Maps

More

Settings

Tools

Collections

road

size

riding

cycling

bike shops

rob english

saddle

road bike fit

mountain bike

lock

bike locks

frame

wh



How to pick the right size ...
pinterest.com



Bikes v. Cars: Who Has the Right of Way ...
lorenzoflorenz.com



Right Bike measures you up for the ...
newatlas.com



Road Bike Frame Sizes ...
pinterest.com



How to Pick the Best B...
dailyburn.com



6 Relatively Cheap Bikes You C...
forbes.com

Introduction:QAnswer

- QAnswer is knowledge (or ontology) based QA system.
- A knowledge base is a collection of facts that can be interpreted by a machine
- Such a fact can look like this:"bicycle" "in" "right".
- QAnswer translates from a natural language question to correct SPARQL queries.

give me pictures with bicycles on the right side of image

Go

/ has on the right / bicycle

Background SparQL



/ has on the right / bicycle

```
SELECT DISTINCT ?s1 WHERE {  
    ?s1 <http://qanswereu/data/datasets/objectPosition/has_on_the_right>  
<http://www.wikidata.org/entity/Q11442> .  
} LIMIT 1000
```



/ instance of / image

/ has in the center, has on the bottom, has on the left, has on the right / bicycle
/ instance of / image

```
SELECT DISTINCT ?s1 WHERE {  
    ?s1 ?p1 <http://www.wikidata.org/entity/Q478798> .  
    ?s1 ?p2 <http://www.wikidata.org/entity/Q11442> .  
    ?s1 ?p3 <http://www.wikidata.org/entity/Q478798> .  
} LIMIT 1000
```



QAnswer: images with bicycle in the right

bicycle in the right

Go

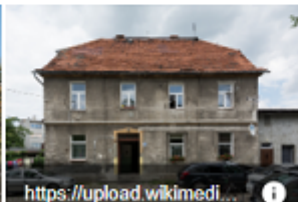
Confidence : 59 %

Is this the right answer? ☐ Yes ☐ No

/ has on the right / bicycle

LIST

IMAGES



set of bicycle images



2016_Strzelin%2C
ul_%C5%9Awi
%C4%99togo_Flo
riana_14_1.jpg



20080804_freight
_bicycle_Shangh
ai_2383.jpg



Barclays_Cycle_H
ire%2C_St._Mary
Axe%2C_Aldgate.
jpg



Chapelle_de_Bast
ide_vers_Lasbros_
DSC_0598.JPG



Cycling_Amsterd
am.jpg



Juist%2C_Altes
Warmbad_--201
4_--_3630.jpg



M%3%BCnster
%2C_Prinzpalma
rkt_--2014_--_46
89-93.jpg



Preah_Sihanouk_
National_Park_08
jpg



Puch_Olympian_
12_01.jpg



Restaurant_Laska
_c%3%B4%3
%A9_rue_Terraille
_%3%A0_Lyon...

- Implementation of an Algorithm for object extraction.
- Design of a semantic web modelling for extracted data.
- Implementation of a visual image search engine through Qanswer.

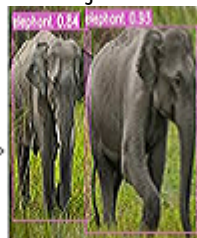
Implementation of an Algorithm for object extraction.

YOLO-(You Only Look Once)

- Implementation of an Algorithm for object extraction.



Computer
Vision API
with
PyTorch
Framework
and use
YOLO



84	34	289	170	20	0.934078
4	36	88	162	20	0.841598
X1 Y1 X2 Y2					
class number/class name					
confidence					

Class Number, Class Name, QID

0. person-Q215627	25.umbrella-Q41607	49.orange-Q39338
1.bicycle-Q11442	26.handbag-Q467505	50.broccoli-Q47722
2.car-Q1420	27.tie-Q44416	51.carrot-Q81
3.motorbike-Q34493	28.suitcase-Q200814	52.hot dog-Q181055
4.aeroplane-Q197	29.frisbee-Q131689	53.pizza-Q177
5.bus-Q5638	30.skis-Q172226	54.donut-Q192783
6.train-Q870	31.snowboard-Q178131	55.cake-Q13276
7.truck-Q43193	32.sports ball-Q63347096	56.chair-Q15026
8.boat-Q35872	33.kite-Q107061	57.sofa-Q131514
9.traffic light-Q8004	34.baseball bat-Q809910	58.pottedplant-Q203834
10.fire hydrant-Q634299	35.baseball glove-Q809894	59.bed-Q42177
11.stop sign-Q250429	36.skateboard-Q15783	60.diningtable-Q10578291
12.parking meter-Q953960	37.surfboard-Q457689	61.toilet-Q7857
13.bench-Q204776	38.tennis racket-Q153362	62.tvmonitor-Q289
14.bird-Q5113	39.bottle-Q80228	63.laptop-Q3962
15.cat-Q4167836	40.wine glass-Q1531435	64.mouse-Q7987
16.dog-Q144	41.cup-Q81727	65.remote-Q185091
17.horse-Q726	42.fork-Q81881	66.keyboard-Q250
18.sheep-Q7368	43.knife-Q32489	67.cell phone-Q17517
19.cow-Q830	44.spoon-Q81895	68.microwave-Q127956
20.elephant-Q7378	45.bowl-Q153988	69.oven-Q36539
21.bear-Q30090244	46.banana-Q503	70.toaster-Q14890
22.zebra-Q32789	47.apple-Q89	71.sink-Q140565
23.giraffe-Q862089	48.sandwich-Q28803	72.refrigerator-Q37828
24.backpack-Q5843		

```
https://github.com/pjreddie/darknet/blob/master/data/  
coco.names
```

```
73.book-Q571
```

```
74.clock-Q376
```

```
75.vase-Q191851
```

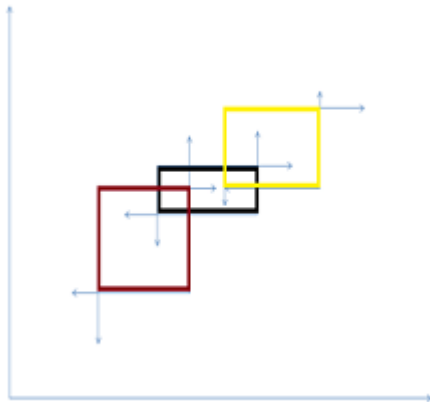
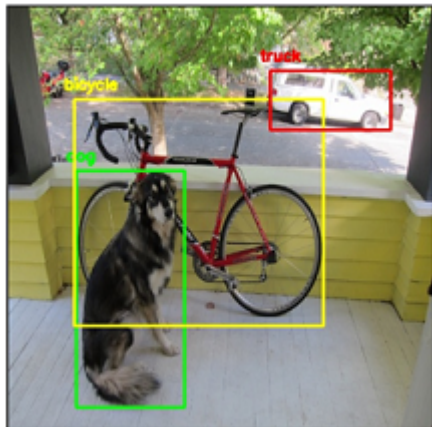
```
76.scissors-Q40847
```

```
77.teddy bear-Q213477
```

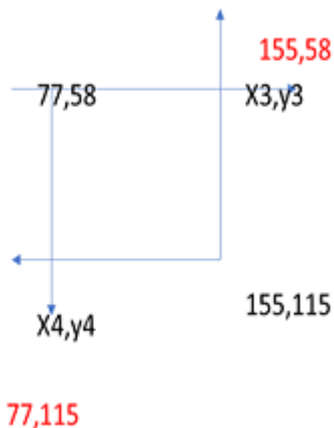
```
78.hair drier-Q15004
```

```
79.toothbrush-Q134205
```

Bounding Box



Co-ordinate representation of Bounding Box



$x_3 = x_2 = 155, y_3 = y_1 = 58;$
 $x_4 = x_1 = 77, y_4 = y_2 = 115$

Image-Object Relations

Image	relation	property value
	has on the left	
	has on the right	
	has on the top	
	has on the bottom	
	has in the center	

Algorithms for Image-Object Relations

Algorithm 1 has on the left and right

```
1: if  $X - \text{centre} \leq 0.3 * X - \text{ImageDimensions}$  then  
2:    $\text{hasontheleft} \leftarrow \text{object}$   
3: else  
4:   if  $X - \text{centre} \geq 0.6 * X - \text{ImageDimensions}$  then  
5:      $\text{hasontheright} \leftarrow \text{object}$   
6:   end if  
7: end if
```

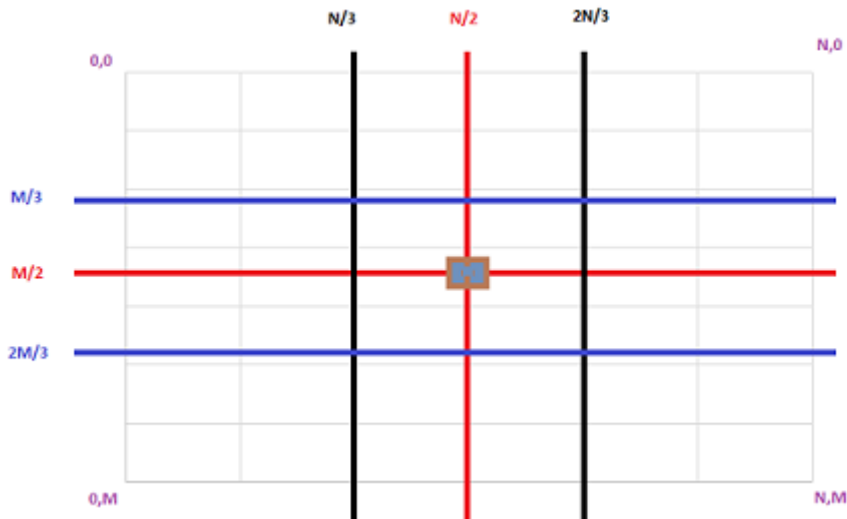
Algorithm 2 has on the top and bottom

```
1: if  $Y - \text{centre} \leq 0.3 * Y - \text{ImageDimensions}$  then  
2:    $\text{hasonthetop} \leftarrow \text{object}$   
3: else  
4:   if  $Y - \text{centre} \geq 0.6 * Y - \text{ImageDimensions}$  then  
5:      $\text{hasonthebottom} \leftarrow \text{object}$   
6:   end if  
7: end if
```

Algorithm 3 has in the center

```
1: if  $X - \text{centre} \geq 0.3 * X - \text{ImageDimensions},$   
    $X - \text{centre} \leq 0.66 * X - \text{ImageDimensions},$   
    $Y - \text{centre} \geq 0.3 * Y - \text{ImageDimensions},$   
    $Y - \text{centre} \leq 0.66 * Y - \text{ImageDimensions}$  then  
2:    $\text{hasinthecenter} \leftarrow \text{object}$ 
```

An attempt to show a graphical representation of tblrc



Algorithms for Image-Object Relations

Algorithm 1 has on the left and right

```
1: if  $X - \text{centre} \leq 0.3 * X - \text{ImageDimensions}$  then  
2:    $\text{hasontheleft} \leftarrow \text{object}$   
3: else  
4:   if  $X - \text{centre} \geq 0.6 * X - \text{ImageDimensions}$  then  
5:      $\text{hasontheright} \leftarrow \text{object}$   
6:   end if  
7: end if
```

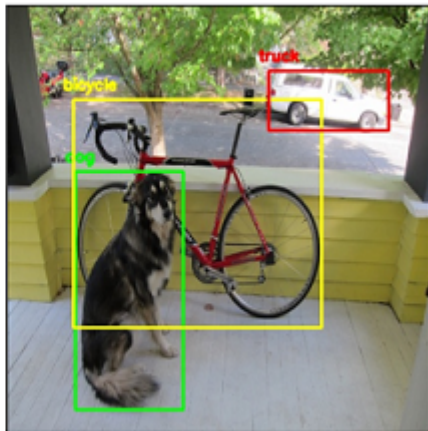
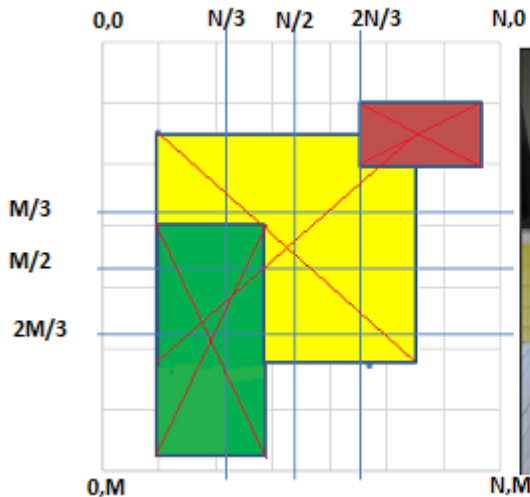
Algorithm 2 has on the top and bottom

```
1: if  $Y - \text{centre} \leq 0.3 * Y - \text{ImageDimensions}$  then  
2:    $\text{hasonthetop} \leftarrow \text{object}$   
3: else  
4:   if  $Y - \text{centre} \geq 0.6 * Y - \text{ImageDimensions}$  then  
5:      $\text{hasonthebottom} \leftarrow \text{object}$   
6:   end if  
7: end if
```

Algorithm 3 has in the center

```
1: if  $X - \text{centre} \geq 0.3 * X - \text{ImageDimensions}$ ,  
    $X - \text{centre} \leq 0.66 * X - \text{ImageDimensions}$ ,  
    $Y - \text{centre} \geq 0.3 * Y - \text{ImageDimensions}$ ,  
    $Y - \text{centre} \leq 0.66 * Y - \text{ImageDimensions}$  then  
2:    $\text{hasinthecenter} \leftarrow \text{object}$ 
```

An attempt to show a graphical representation of tblrc in comparison to an incoming photo



Using tblrc we create a csv file

X1	Y1	X2	Y2	object name	Image name	has on the left	has on the right	has on the top	has on the bottom	has in the center
190	813	3897	1932	airplane	Antonov_2	na	airplane	na	airplane	airplane
220	596	5021	1673	airplane	EBACE_20	na	airplane	na	airplane	airplane
742	1303	3933	2099	airplane	Embraer_1	na	airplane	na	airplane	airplane
889	1035	1378	1374	airplane	Kirchturn	na	airplane	na	airplane	airplane
172	278	4018	1722	airplane	Lufthansa	na	airplane	na	airplane	airplane
331	532	2362	1245	airplane	North_Am	na	airplane	na	airplane	airplane
1355	704	1444	813	person	North_Am	na	person	na	person	person
833	251	2202	1761	airplane	Paris_Air	na	airplane	na	airplane	airplane
1460	1207	1579	1738	person	Playing_ir	na	person	na	person	na
224	1063	466	1813	person	Playing_ir	person	na	na	person	na
756	1286	845	1562	person	Playing_ir	person	na	na	person	na
2374	990	2486	1359	person	Playing_ir	na	person	na	person	na
468	1194	546	1264	frisbee	Playing_ir	frisbee	na	na	frisbee	na
264	273	2284	1571	airplane	RUAG_Avina	na	airplane	na	airplane	airplane

Design of a semantic web modelling for extracted data.

Image-Object

- Design of a semantic web modelling based on the csv file.

Image	relation	property value
	has on the left	
	has on the right	
	has on the top	
	has on the bottom	
	has in the center	

Subject

URI

Blank Node

Predicate

URI

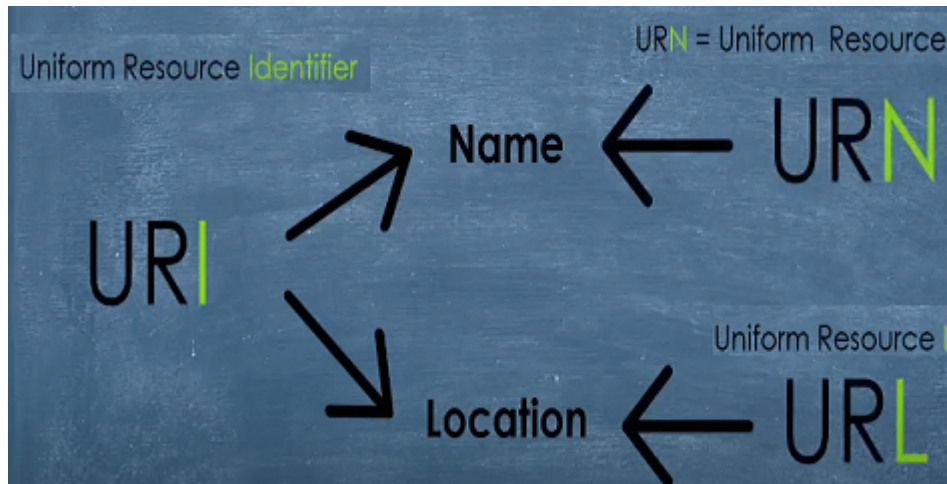
Object

URI

Blank Node

Literal

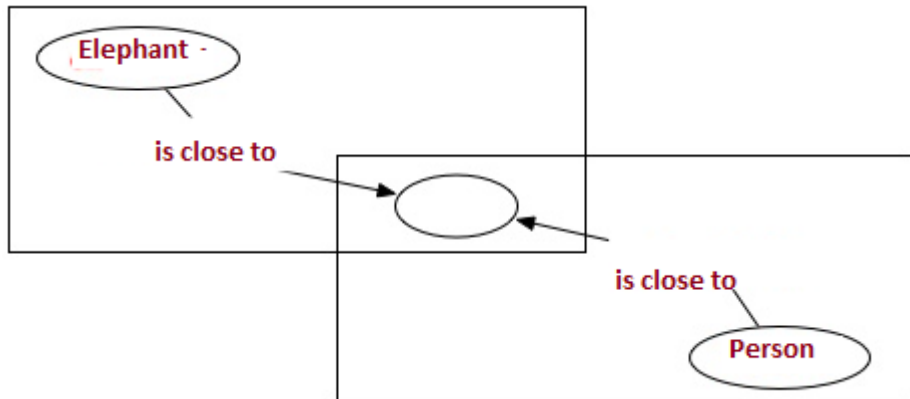
URIs



URLs, URIs, IRIs

- ▶ URL = Uniform Resource Locator
`http://www.learningsparql.com/resources/index.html`
- ▶ URN = Universal Resource Name
`urn:isbn:006251587X`
- ▶ URI = Universal Resource Identifier
 - encompasses both URLs and URNs
 - most URIs are URLs (sometimes the terms are used interchangeably)
 - `http://xmlns.com/foaf/0.1/Person`
- ▶ IETF released IRIs (Internationalized Resource Identifiers)

Blank Node



Object-Object Relation

Object	relation	property value
	left of	object
	right of	object
	top of	object
	bottom of	object
	close to	object
	far from	object
	overlapping with	object
	greater than	object
	smaller than	object
	% of image	value

OORelation algos ready to be used

Algorithm 4 is on the left and right of

```
1: if  $X - \text{centreOfObject}_1 \leq X - \text{centreOfObject}_2$  then  
2:    $\text{Object}_1\text{IsOnTheLeft} \leftarrow \text{Object}_2$   
3: else  
4:   if  $X - \text{centreOfObject}_1 \geq X - \text{centreOfObject}_2$  then  
5:      $\text{Object}_1\text{IsOnTheRight} \leftarrow \text{Object}_2$   
6:   end if  
7: end if
```

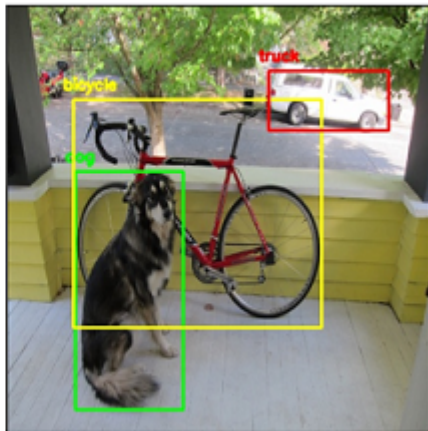
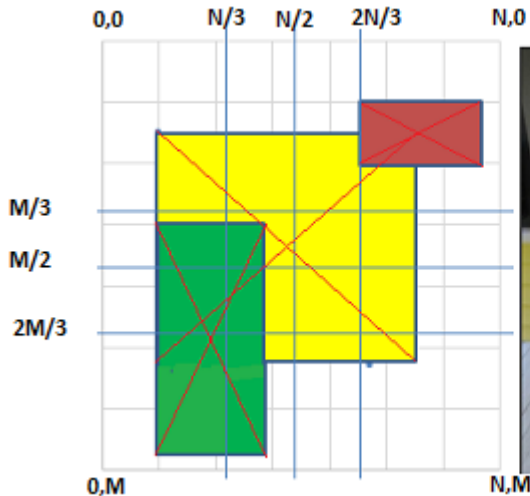
Algorithm 5 is on the top and bottom of

```
1: if  $Y - \text{centreOfObject}_1 \leq Y - \text{centreOfObject}_2$  then  
2:    $\text{Object}_1\text{IsOnTheTop} \leftarrow \text{Object}_2$   
3: else  
4:   if  $Y - \text{centreOfObject}_1 \geq Y - \text{centreOfObject}_2$  then  
5:      $\text{Object}_1\text{IsOnTheBottom} \leftarrow \text{Object}_2$   
6:   end if  
7: end if
```

Algorithm 6 close and far from

```
1: if  $\text{distance}(\text{Center1}, \text{Center2}) \leq \text{meansofdiagonalofthe2objects}$  then  
2:    $\text{Object}_1\text{IsCloseTo} \leftarrow \text{Object}_2$   
3: else  
4:   if  $\text{distance}(\text{Center1}, \text{Center2}) \geq \text{meansofdiagonalofthe2objects}$  then  
5:      $\text{Object}_1\text{IsFarFrom} \leftarrow \text{Object}_2$   
6:   end if
```

An attempt to show a graphical representation of O-O Relations in comparison to an incoming photo



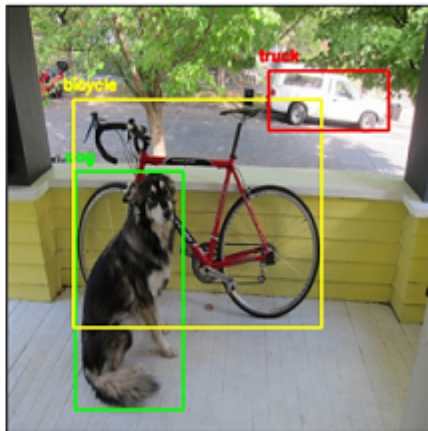
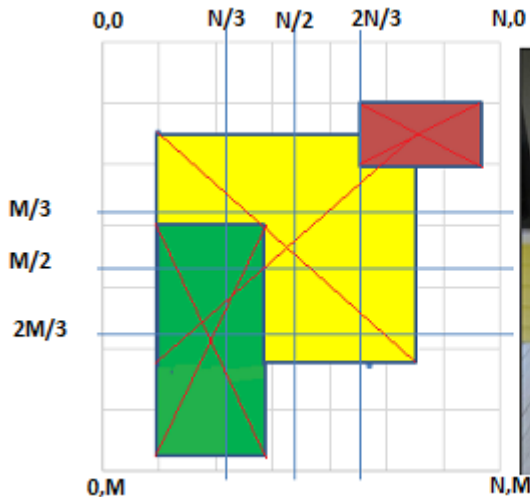
OORelation algos ready to be used, contd..

Greater and Smaller Than

Algorithm 7 greater and smaller than

```
1: if  $AreaOfObject_1 \leq AreaOfObject_2$  then  
2:    $Object_1IsSmallerThan \leftarrow Object_2$   
3: else  
4:   if  $AreaOfObject_1 \geq AreaOfObject_2$  then  
5:      $Object_1IsGreaterThen \leftarrow Object_2$   
6:   end if  
7: end if
```

An attempt to show a graphical representation of O-O Relations in comparison to an incoming photo



OORelation algos ready to be used, contd..

Overlapping With

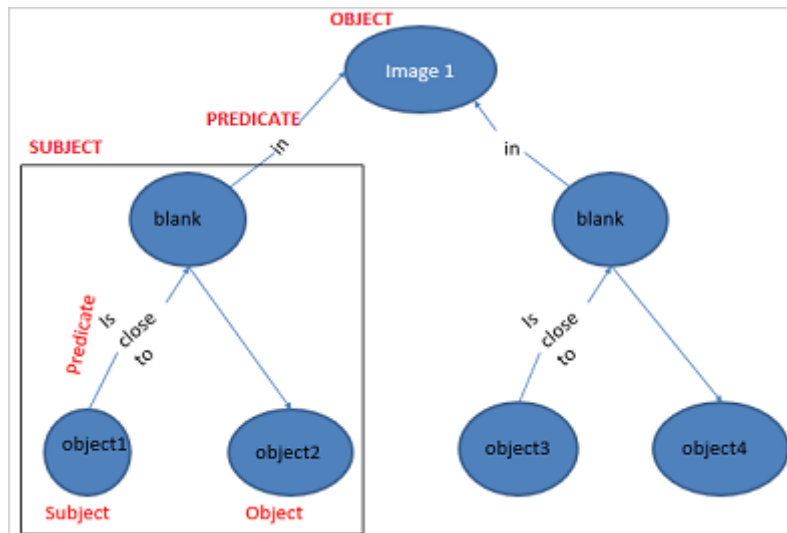
```
from shapely.geometry import Polygon
p1 = Polygon([(0,0), (1,1), (1,0)])
p2 = Polygon([(0,1), (1,0), (1,1)])
print(p1.intersects(p2))
```

True

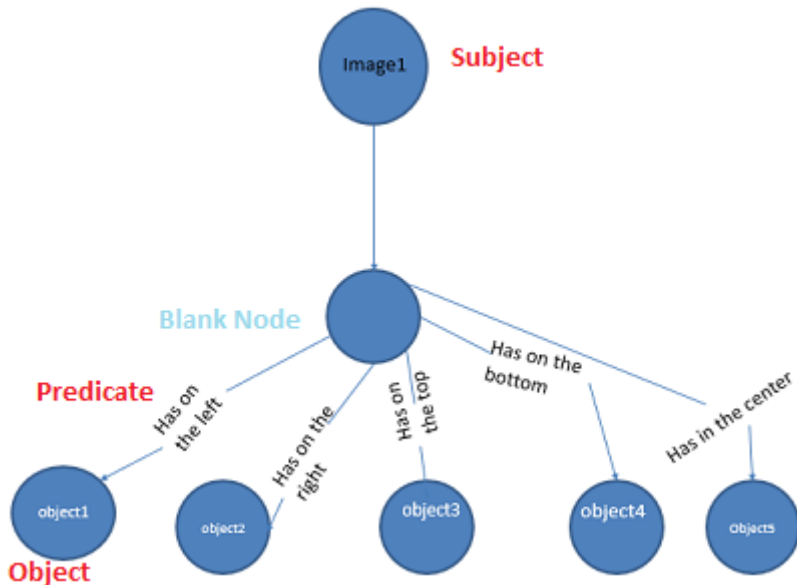
Object-Object Relation

Object	relation	property value
	left of	object
	right of	object
	top of	object
	bottom of	object
	close to	object
	far from	object
	overlapping with	object
	greater than	object
	smaller than	object
	% of image	value

Reified Triple



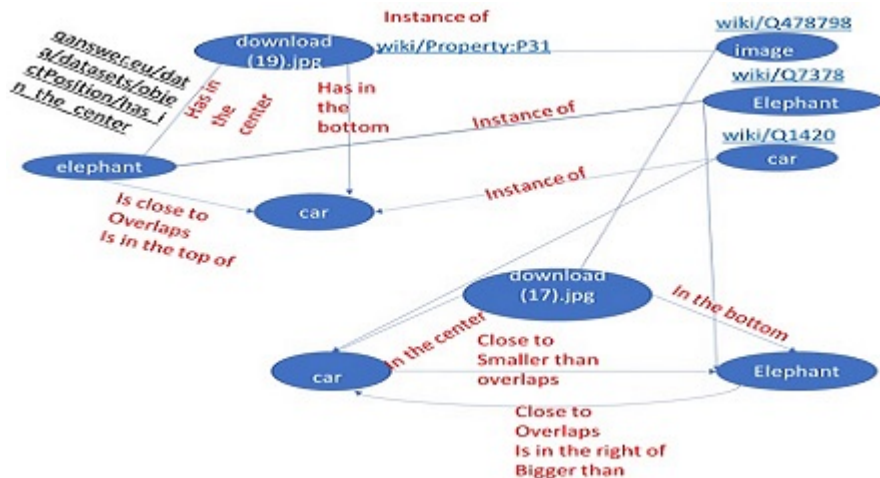
Regular Triple



Knowledge Graph

Image-Object Relation with a hint of Object-Object Relation

- Design of a semantic web modelling for extracted data.



Thereafter we use the CSV file in a Java program and convert it into a RDF file

Following this, we upload the file to QAnswer

- QAnswer: airplane in the center

Confidence :  55 %

Is this the right answer? ☐ Yes ☐ No

/ has in the center / airplane

LIST

IMAGES



images in the bottom

bench in the bottom

bench on the bottom

Go

Confidence :  55 %

SPARQL LIST

DIRECT ANSWER

Is this the right answer? ☐ Yes ☐ No

/ has on the bottom / bench

[https://upload.wikimedia.org/wikipedia/commons/7/78/Juist%20-2014 -- 3630.jpg](https://upload.wikimedia.org/wikipedia/commons/7/78/Juist%20-2014--3630.jpg)



images in the left

train in the left

```
https://qanswer-frontend.univ-st-etienne.fr/user/query?  
kb=onto&user=anindamaulik
```

Qanswer: Image-Object Relation-contains

- person and chair

person and chair

Confidence : 57 %

[SPARQL LIST](#)

[DID YOU MEAN](#)

[DIRECT ANSWER](#)

Is this the right answer? ☒ Yes ☐ No

/ has on the bottom, has on the left, has on the right / human

/ has on the bottom, has on the left, has on the right / chairperson

https://upload.wikimedia.org/wikipedia/commons/e/eb/CEE_Spi



special Wikimediacommons api give images

<https://commons.wikimedia.org/w/api.php?action=query&list=search&srsearch=haswbstatement:P180=Q7378&srnamespace=6&format=json>

```
{"batchcomplete":"","continue":  
{"sroffset":10,"continue":"-||"}, "query":{"searchinfo":  
{"totalhits":236}, "search":[{"ns":6,"title":"File:African  
Elephant.jpg", "pageid":15925090, "size":1529, "wordcount":8,  
lish African Bush elephant facing forward Urdu \u06c1\u062f\u06cc  
\u06cc", "timestamp":"2020-07-03T11:16:22Z"}, {"ns":6,"title":  
elefant.jpg", "pageid":8133970, "size":899, "wordcount":1, "sn  
h" "timestamp":"2020-06-05T21:24:05Z"}, {"ns":6,"title":"Fi  
Mammoth  
sculpture.JPG", "pageid":17924651, "size":883, "wordcount":15  
the copyright holder of this work, hereby publish it under
```

special Wikimediacommons api give human hand-annotated structured data

`https://commons.wikimedia.org/wiki/File:
African_elephants,_Lake_St_Lucia_06.jpg`

Items portrayed in this file **creator**
depicts

source of file

Search to add items

nature reserve

wildlife

Terrestrial animal

Indian elephant

mammal

elephant

safari

Search to add items

Some value without a Wikidata item

- URL: <https://commons.wikimedia.org/wiki/User:Bgag>
- author name string: Bernard Gagnon
- Wikimedia username: Bgag
- object has role: photographer

Search to add items

original creation by uploader


Inception

Enter time


10 November 2017

special Wikimediacommons api give human hand-annotated structured data like copyright details

copyright status

 Search to add items
copyrighted

copyright license

 Search to add items
GNU Free Documentation License, version 1.2 or later
Creative Commons Attribution-ShareAlike 4.0 International
Creative Commons Attribution-ShareAlike 3.0 Unported
Creative Commons Attribution-ShareAlike 2.5 Generic
Creative Commons Attribution-ShareAlike 2.0 Generic
Creative Commons Attribution-ShareAlike 1.0 Generic

`https://www.mediawiki.org/wiki/API:Search`

Limitations and Future Work

Query for images on the top

Issue with confidence of detection by YOLO

/ has on the top / clock

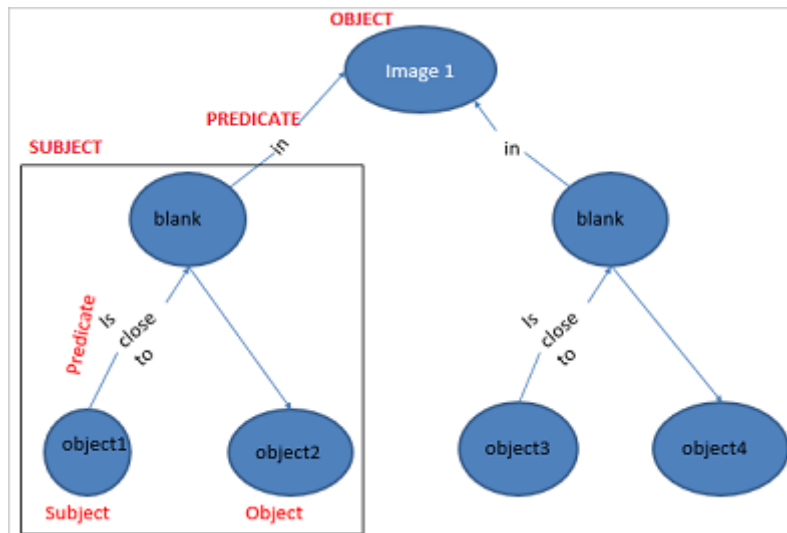
https://upload.wikimedia.org/wikipedia/commons/e/e7/Taipei_Rheinland-Office-Building-02.jpg



YOLO got overconfident



Reified Triple



Reified Triple not being generated by QAnswer

give me pictures with a person on right of a car

Go

Confidence :  39 %

SPARQL LIST

Is this the right answer? ☒ Yes ☐ No

/ has on the right / human

LIST

IMAGES



special Wikimediacommons api give images and structured hand annotated human data

Our time constraints

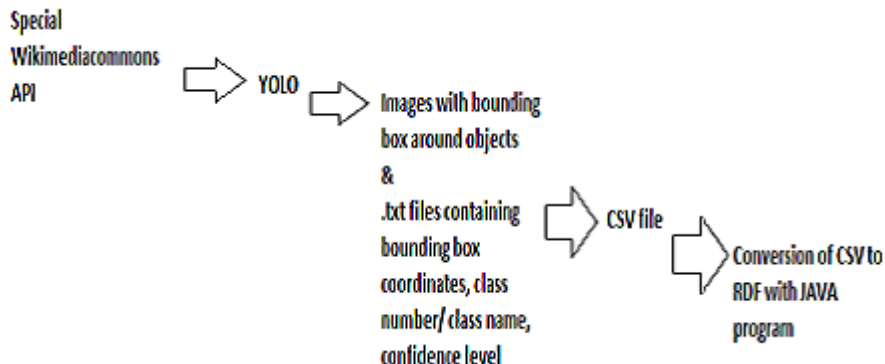
```
https://commons.wikimedia.org/w/api.php?action=query&list=
search&srsearch=haswbstatement:
P180=Q7378&srnamespace=6&format=json
```

- We are yet to handle the incoming RDF data from the api
- and merge with our RDF file,
- in order to be able to query
- "Give me pictures of bicycles from February 2019"

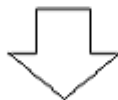
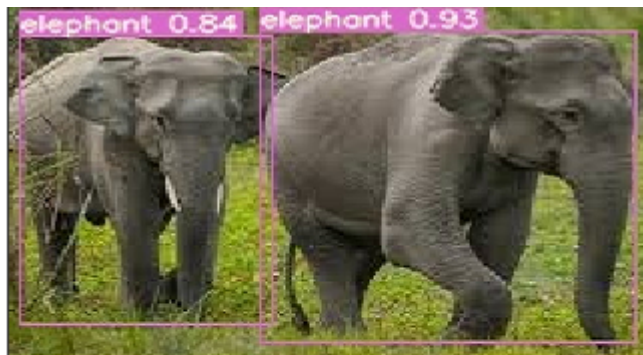
Conclusion

- ① We have worked on improving image search engines by combining
 - Computer Vision techniques
 - with Semantic Web
 - and Question Answering techniques.
- ② Computer Vision techniques is able to identify objects in images.
- ③ Semantic Web techniques give a
 - Semantic representation of the images
 - that can be queried with QA engine, namely QAnswer.
- ④ This work can be easily used by any search or query engine to give results
- ⑤ based on image-object relation
- ⑥ and in a near future on object-object relation.
- ⑦ We do not have a bench mark as of yet, since
 - Nobody has worked in developing this concept
 - Our future work also includes establishing a bench mark for such data to make proper evaluation
- ⑧ We have an automated Python program in place

Automation pipeline



Bounding Box around objects and text file



download (14).jpg.txt - Notepad

File Edit Format View Help

84 34 209 170 20 0.934078

4 36 88 162 20 0.841598

- Thank you for your attention, time and patience
- Please ask me any question that you have.
- Please provide your suggestion which can be used to make my work better