Interpreting attributes semantically.
Absolute attributes classification by different perspectives.

Sung Min Yang
Project report for Embodied and
Situated Language Processing course
University of Gothenburg
gusyansu@student.gu.se

Maryam Tashvir
Project report for Embodied and
Situated Language Processing course
University of Gothenburg
gustasma@student.gu.se

Abstract (SM)

In this paper, we designed a framework for detecting multiple objects and extracting attributes for each detected object. After extracting attributes from multiple objects, we can ask user questions to guide a system to get better interpretation of attributes. Briefly, we introduce a framework that user will be semi-supervising for training step. In this method, we can build more semantically abundant system. In this paper, we made a game as for training step.

1 Introduction (SM & MY)

According to (Mitchell, Van Deemter, & Reiter, 2013). object attributes can be split by two big categories which are Absolute attribute and Relative attributes. For example, color and shape of object are Absolute attributes while location is considered as Relative attributes, since it is easily changed by perspective. Since we found that these values can be various among people, we simplify all attributes are relative attributes and treat it equally. By our definition of attribute, we can make general framework for finding proper attributes value. In this paper, we will use several most probable values for attributes and ask users to check if it’s proper answer or not. For multiple object detections, we implemented Yolo [1] algorithms for speed which can be crucial for real-time object attribute detection.

Although we considered different scenarios, the goal is the same which is to find out the object with desired color that was chosen by user.

Moreover, this framework makes an interaction between system and user. In the first place, system shows a random image and user chooses an object in her/his mind. Then, system asks some questions to find desired bounding box for the object with specified color. Finally, corresponding object with desired color is shown by system. In scenario1, we focus on ‘yes/no’ answers.

System	User
S: <Considered object in relevant image>	-
S: Is Person your chosen object (yes/no)?	U: no
S: Is dog your chosen object (yes/no)?	U: no
S: Is horse your chosen object (yes/no)?	U: yes
S: Is it white (yes/no)?	U: no
S: Is this object you have chosen(yes/no)?	U: yes

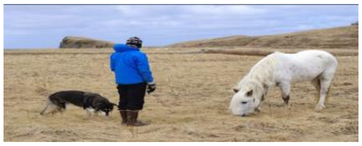
---

[1] https://pjreddie.com/darknet/yolo/

S: The system could find out your chosen object. Would you like to continue **the** game with another image (yes to continue/ no to end the game)?	U: yes … .

Table 1: Dialogue flow in scenario1

In addition, we add asking location part in order to narrow down the possible targets by each step. To simplify handle location question, we only consider perspective of viewer from the monitors. It can be naively calculated by comparing center of every objects to get location. We use 4 relative location left, right, below and above. Furthermore, we exclude most far located object rather choose closest one. For example, if user want to get “left”, system will not extract only one “left” object but rather exclude most far located one object from “left”. In other word, it excludes object located on most right.

System	**User**
S: <Considered object in relevant image> 	**-**
S: Is object color in [white, blue, black] (yes/no)?	U: no
S: Is object exist in [man, dog, horse] (yes/no)?	U: yes
S: Is object located in Left, Right, Above, Below (type left/right/above/below)?	U: right
S: (After few steps) Color seems wrong, from above. Is there any color in your mind? [possible 10 colors from white] (not exist/color name)?	U: not exist

S: Can you name color of object (color name)?	U: clean white
S: Save [horse_clean white].json visual features.	

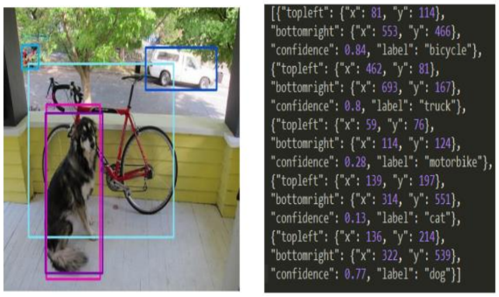
Table 2: Dialogue flow in scenario2

2 Related work (MY)

In visual question answering, which is written by Antol et al. [2](2015) proposed free form and open ended visual question answering task. It actually provides and accurate natural language answers to a natural language question which is about an image.

3 Implementation (MY)

We considered three main phases for this project: dataset building, object detection and attribute extraction. It should be stated that implementation of this project has been different since we had various scenarios. This will be more described in the following.



```
[{"topleft": {"x": 81, "y": 114},
"bottomright": {"x": 553, "y": 466},
"confidence": 0.84, "label": "bicycle"},
{"topleft": {"x": 462, "y": 81},
"bottomright": {"x": 693, "y": 167},
"confidence": 0.8, "label": "truck"},
{"topleft": {"x": 59, "y": 76},
"bottomright": {"x": 114, "y": 124},
"confidence": 0.28, "label": "motorbike"},
{"topleft": {"x": 139, "y": 197},
"bottomright": {"x": 314, "y": 551},
"confidence": 0.13, "label": "cat"},
{"topleft": {"x": 136, "y": 214},
"bottomright": {"x": 322, "y": 539},
"confidence": 0.77, "label": "dog"}]
```

Figure 1: Example of multiple objects detection and corresponding Bound Box information

---

[2] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C. and Parikh, D., 2015. Vqa: Visual question answering.

3.1 Image Dataset building (MY)

In this project, we used a model which is pre-trained on Coco dataset. Coco consists of around 200k labeled images. In addition, we considered 10 sample images for evaluation which can detect 20 different classes, person, bicycle and bird are typical examples.

3.2 Object Detection (SM & MY)

We used YOLO for object detection. It is pre-trained by Darknet with PASCAL 2007, 2012 datasets. Yolo algorithm is written by Darknet which is Neural Network libraries written in C language. Yolo is known as one of fastest object detection which is fast enough to be executable in real-time situation. With Yolo, we can show our framework for Semi-Supervised learning object's attributes in real-time situation.

YOLO uses a single CNN network for both classification and localizing the object using bounding boxes. Final layer consists of Bound Boxes information.

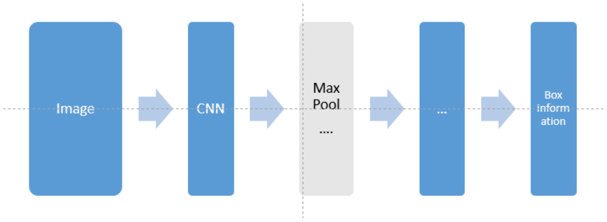


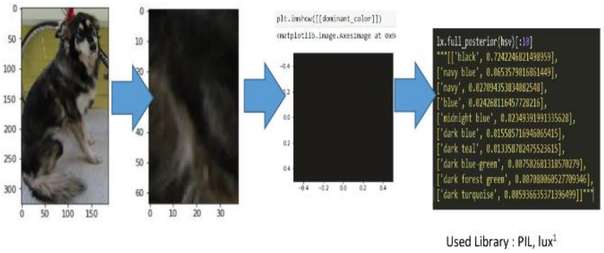
Figure 2. CNN structure in Yolo :
final layer consists of Bound Boxes information

In researchers' point of view, the difference between dog and cat in an image was almost impossible during the last ten years. In this regard, there is an effective approach called YOLO which can be lucrative to solve this problem. YOLO, it is applied a neural network to the full image into regions, predicts bounding boxes and class probabilities. The bounding boxes are weighted by predicted probabilities. Result of detections is based on model's confidence.

We used pre-trained YOLO weights and made training process faster. In output layer, it is chosen an object with the highest confidence and desired bounding box of object is demonstrated. There are two different kinds of models in YOLO called Tiny and Full. Although Tiny is faster than Full, it cannot be efficient since it has not sufficient accuracy. It should be considered that the "tiny" version of YOLO that we'll be using has only these 9 convolutional layers and 6 pooling layers. On the other hand, full YOLO has higher accuracy than its counterpart, but it takes more time in order to have more weights. In addition, full model uses three times as many layers and has a slightly more complex shape, but it's still just a regular Convolutional Neural Network. Since Full Yolo has more accurate detection, we decided to use it.

3.3 Extract Attributes (SM)

Attributes exist in different forms. According to Mitchell et al[3], there are two main properties are defined which are absolute property and relative property. Interestingly, we found that even absolute property such as color and shape are not fixed but rather changing by perspectives. For instance, color



```
plt.imshow([[dominant_color]])
<matplotlib.image.AxesImage at 0x
```

```
lx.full_posterior(hsv)[:10]
"""[['black', 0.7242246821498959],
['navy blue', 0.0653579816861449],
['navy', 0.027094353834882548],
['blue', 0.024268116457728216],
['midnight blue', 0.02349391991335628],
['dark blue', 0.015585716946065415],
['dark teal', 0.013358782475523615],
['dark blue-green', 0.007582681318570279],
['dark forest green', 0.007880060527709346],
['dark turquoise', 0.005936635371396499]]"""
```

Used Library : PIL, lux[1]

Figure 3: Example of centering and extracting dominant color and other possible colors by ground knowledge.

---

[3] Mitchell, M., Van Deemter, K. and Reiter, E., 2013. Generating Expressions that Refer to Visible Objects.

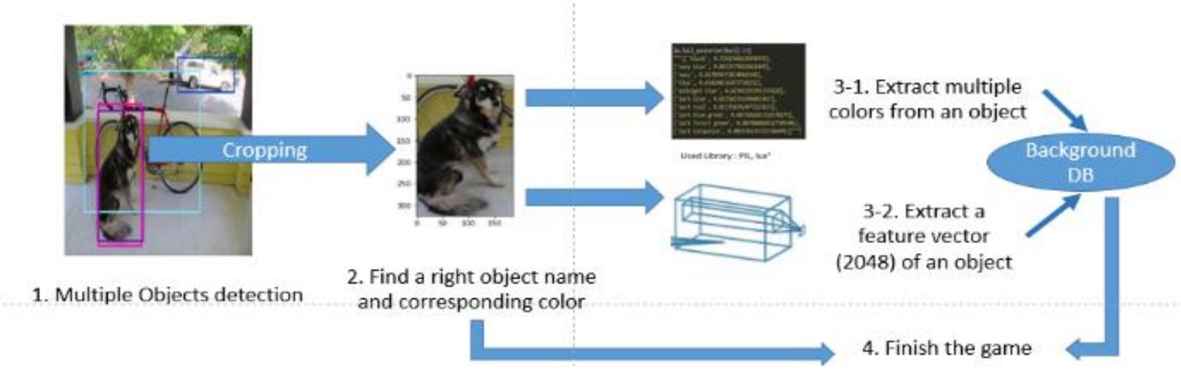

Cropping
1. Multiple Objects detection
2. Find a right object name and corresponding color
Used Library : PIL, lux
3-1. Extract multiple colors from an object
Background DB
3-2. Extract a feature vector (2048) of an object
4. Finish the game

Figure 4: Cropping object and focusing on center. Then, calculate possible colors by grounded color knowledge

of fire looks yellow to someone while other person sees its color as red. For this reason, we chose absolute property in the same way to treat relative property. Specifically, in this paper, we use ground color semantic knowledge for interpreting one color into possible colors. For simplification, we don't consider other perspective such as landmark or various perspective that Relative property can has, but rather we exclude all perspectives except a view of perceiving one color to other possible color.

In this part, we only focus on extracting one absolute attribute among many of them such as color, shape, etc. For simplification, here we only chose color for our task. Two different scenarios are used for implementation for color extraction.

I. Cropping and Focusing (SM)

From randomly chosen image by system and Bound Boxes data, we can crop objects area. Either we can use cropped image itself (MY) or focusing on 20% of center (SM) square of objects. First approach(MY) is to recognize object color based object itself. Second approach (SM_ is using 20% of center in objects. Both approach is based on finding dominant color in objects. However, if objects only exist in small part, system will fail to recognize color of object in both cases.

II. Extract dominant color and convert to multiple colors

For scenario 1 (MY), the first scenario was to recognize object color based on which color was exposed the most. Then, we used webcolor [4]Python library. We go through every pixel in each object box, recognize pixel color. Then, count each color by how many occurs in the object box. Finally, the color which was found the most is decided as the dominant object color.

For scenario 2 (SM), we used colorthief [5] library which is playing a same role as webcolor library does. After extract dominant color, we used lux [6] library for converting one color to multiple similar colors. This converting technique is based on color ground knowledge by experiments and pre-trained to classify one color to similar colors. In this paper, we chose most similar ten colors as candidates.

---

[4] webcolors.readthedocs.io
[5] github.com/fengsp/color-thief-py
[6] www.mcmahan.io/lux

3.4 Store Visual features (SM)

Image feature are extracted from ResNet50 for each object after users found matching object in their mind. At the same time, color histograms data will be stored into external storage together.

I. ResNet50 features
As figure4 describe, By ResNet50, we can obtain feature vector in 2048 dimensions. We used Keras for building Neural Network and used pre-trained weights model to get visual features. The reason why we chose to use ResNet50 is that in Yolo, we will get final layer which does not represent image feature but represent of bound boxes information and object classification probabilities. For example, output of final layer of Convolutional Neural Network in this paper has shape of 19×19×425, which is different from 19×19×1024. This is because Yolo algorithm was pre-trained with more layers on top of image feature layers. Furthermore, even if we access to use image features from each grid cell in 19×19, it is problematic to use grid-cell image feature, since object doesn't exist in one grid cell but rather shared by others.

II. Color histogram
Color is made of three components which are red, green and blue. Since we are observing color attribute, we decided to store color histogram for system so that it can use histogram history and return most similar color for other objects. When other objects appear similar color histogram, system will find proper color from color histogram history which is annotated by human ground knowledge. Library openCV is used for extracting color histograms.

4. Result (MY)

In most cases, our approach could recognize the objects in the image but it has some difficulties to realize the correct object's color. One reason was that most objects has two different dominant colors which makes it very difficult for our framework to detect the main object color. Because our framework has not learned with all different types of objects, predicting the unknown object (which has not been used in training process) results in wrong object detection.

As mentioned from above, it is considered 10 sample images and the System correctly recognizes 57 of 73 objects. In this regard, the sixteen of them have not been properly detected and it means that accuracy of this model is 78.08%. (image below is incorrectly recognized by YOLO).

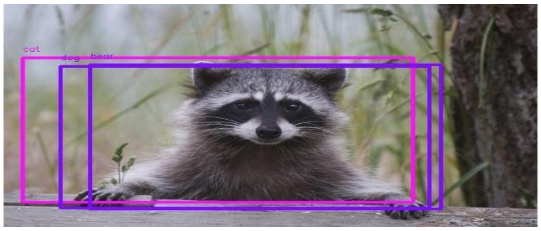
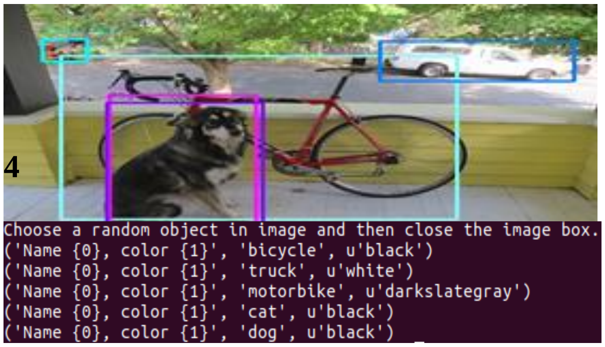


Figure 5: Raccoon is incorrectly detected by system.

Furthermore, the system is not able to recognize 30 of 57 the correct object's color. The value of accuracy is approximately 52.63% which refers to different dominant colors. (the following image illustrates incorrectly the main color detection).

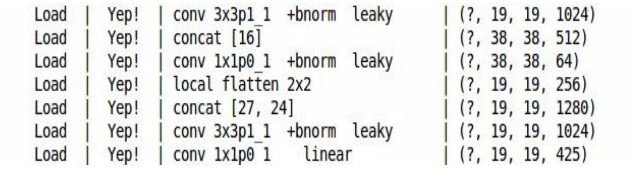


```
Choose a random object in image and then close the image box.
('Name {0}, color {1}', 'bicycle', u'black')
('Name {0}, color {1}', 'truck', u'white')
('Name {0}, color {1}', 'motorbike', u'darkslategray')
('Name {0}, color {1}', 'cat', u'black')
('Name {0}, color {1}', 'dog', u'black')
```

Figure 6: The bicycle consists of two main colors, red and black, the detection of system is black.

4. Discussion (SM)

As follow our framework, we can build a system which can learn more interpretation of attributes. However, there are few drawbacks exist in our system. First of all, errors in detecting multiple object by Yolo will be propagated to next levels.



```
Load | Yep! | conv 3x3p1_1  +bnorm  leaky      | (?, 19, 19, 1024)
Load | Yep! | concat [16]                     | (?, 38, 38, 512)
Load | Yep! | conv 1x1p0_1  +bnorm  leaky      | (?, 38, 38, 64)
Load | Yep! | local flatten 2x2               | (?, 19, 19, 256)
Load | Yep! | concat [27, 24]                 | (?, 19, 19, 1280)
Load | Yep! | conv 3x3p1_1  +bnorm  leaky      | (?, 19, 19, 1024)
Load | Yep! | conv 1x1p0 1    linear          | (?, 19, 19, 425)
```

Figure 7: Structure of last layers in Yolo besides of normal CNN layers (19×19×1024 is last layer)

This will lead system always get failed if Yolo is not able to detect correct object name and corresponding Bound Box at first stage. Second, we need more constructive additional models for semi-supervised learning. but there is not specific model to use stored visual features. For example, we can make more reliable loss functions to calculate similar visual features. However, in this paper, we only used RMSE (Root Mean Square Error) given by two images. Finally, evaluation should be little ambiguous, since we built a game that hard to give concise result.

Besides of above problems, it should be mentioned that why we chose ResNet50 for extracting image feature rather than just extracting existing Yolo structure. This is because ResNet50 last layer represents features of image, Yolo final layer contains information of information of Bound Boxes and classification. As we can see from figure7, Yolo final layer of CNN (19,19,425) contains 19×19 grid cell with x,y,width,height for BB, confidence score and 80 classes for classification. Thus, 5×(5+80) = 425 information will be stored in each grid cell. Nevertheless, we also get 19×19×1024 for visual feature for "Each grid cell" in the middle of CNN layers, it doesn't have any bound box information, but it only contains ‘visual feature of individual grid cell’. From our view, this cannot catch "Visual feature of object". For instance, if a 'dog" is located in grid number 13,14 and half of grid 15. What is the best way to get image features given two and a half cells? For these reason, we chose to use ResNet50 to extract image feature after cropping objects separately from Yolo.

Briefly, input of ResNet50 and Yolo is same as image, but output of each algorithm has different result. ReNet50 has image feature that can be used for classification while Yolo has information of each grid cells that has Bound Boxes information.

6. Future work (SM & MY)

First of all, we should find a way to detect more accurate colors of various objects in the image, particularly those objects which are composed of two main colors. In order to succeed in this process, it has to be more logical for the users to compare if the system recognize the objects that they had in mind. Furthermore, we should take much more time (a couple of weeks) to train our framework with a huge number of pictures. Then, the recognizing object detection process will be much more accurate.

Second, our framework must need a model for using image features such as ResNet50 and color histograms. Currently, our system only can handle storing these visual features but not reusing it in next training. In our future work, reusing stored visual features should be covered by system. This is crucial for semi-supervised learning system, since annotation is mainly collected by human.

References

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., & Parikh, D. (2015). Vqa: Visual question answering. *Proceedings of the IEEE International Conference on Computer Vision*, (pp. 2425-2433).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 770-778).

McMahan, B., & Stone, M. (2015). A Bayesian model of grounded color semantics. *Transactions of the Association for Computational Linguistics, 3*, 103-115.

Mitchell, M., Van Deemter, K., & Reiter, E. (2013). Generating Expressions that Refer to Visible Objects. *HLT-NAACL*, (pp. 1174-1184).

Redmon, J., & Farhadi, A. (2016). YOLO9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 779-788).

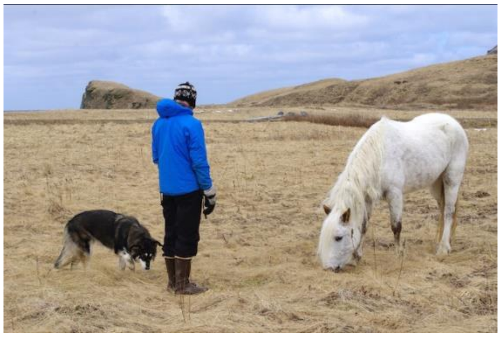
https://pjreddie.com/darknet/yolo/

https://github.com/thtrieu/darkflow

Appendix

Scenario1.1 (MY)

S: <Considered object in relevant image>

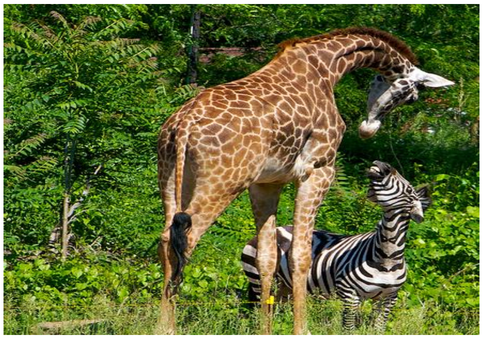


S: Is Person your chosen object (yes/no)?
U: No
S: Is dog your chosen object (yes/no)?
U: No
S: Is horse your chosen object (yes/no)?
U: Yes
S: Is it white (yes/no)?
U: Yes
S: Finally, the corresponding image is shown.

S: Is this object you have chosen(yes/no)?
U: Yes
S: The system could find out your chosen object.
S: Would you like to continue the game with another image (yes to continue/ no to end the game)?
U: Yes

Scenario1.2 (MY)

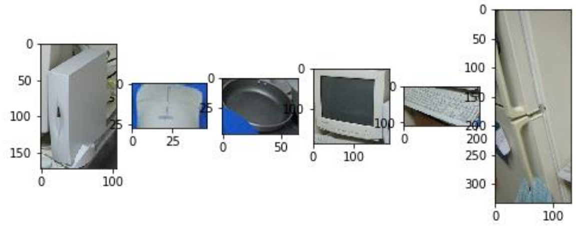
S: <Considered object in relevant image>



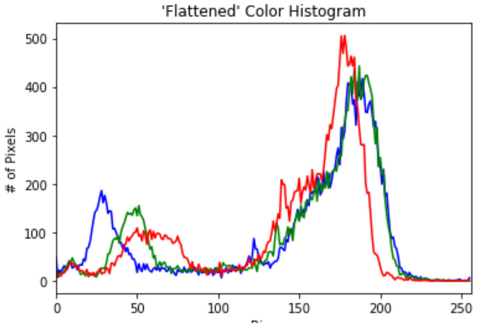
S: Is giraffe your chosen object(yes/no)?
U: No
S: Is zebra your chosen object(yes/no)?
U: No
S: Unfortunately, the system could not realize the object you have chosen!
S: Would you like to continue the game with another image (yes to continue/no to end the game)?
U: Yes
……… .

Scenario2.1 (SM)

S > is your target located in left/right...?
U > Left
S > Is it among [white, grey …] or doesn't exist?



U > grey



S > Is your object white monitor in the picture? Is this what you think?
U > No
S > Move to next object … Is your object grey keyboard in the picture? Is this what you think?
U > Yes

S > I got you! Ain't I good guesser or what?
S > Saving visual features: ResNet50 feature and color histogram annotated by "keyboard_grey".

S > The game is over.

Scenario2.2 (SM)

S > is your target located in left/right...?
U > Left
S > Is it among [white, brown] or doesn't exist?
U > no
S > Is there right name of the objects [bird]?
U > Yes.
S > Let's narrow it down, is your target located in left/right...?
U > left
S > I can see brown bird. But color is wrong! Are there any colors in your mind? Choose one!

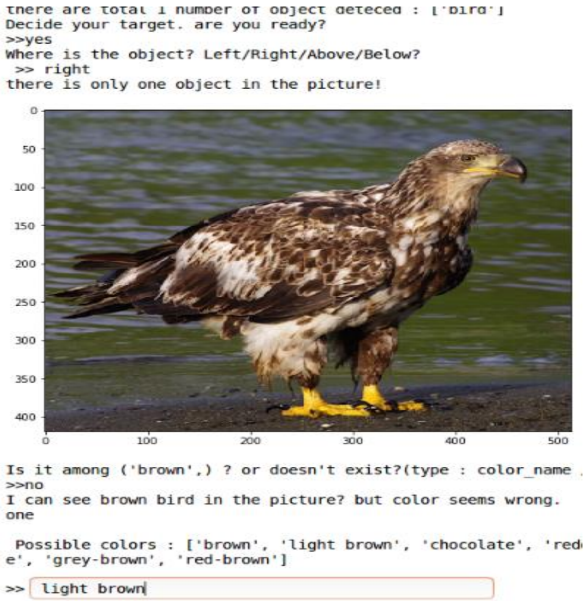
Possible colors:
['brown', 'dark brown', 'grey', 'light grey', 'dark grey', 'pale blue', 'baby blue', 'blue', 'very light blue', 'ice blue']

U > Yes. it's dark brown.

S > Anyway, I failed. I am sorry to disappoint you.

S > Saving visual features: ResNet50 feature and color histogram annotated by "bird_dark brown".

S > The game is over.



```
there are total 1 number of object deteced : ['bird']
Decide your target. are you ready?
>>yes
Where is the object? Left/Right/Above/Below?
 >> right
there is only one object in the picture!
```

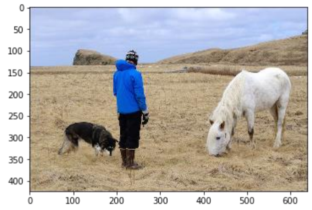
```
Is it among ('brown',) ? or doesn't exist?(type : color_name
>>no
I can see brown bird in the picture? but color seems wrong.
one

 Possible colors : ['brown', 'light brown', 'chocolate', 're
e', 'grey-brown', 'red-brown']

>> light brown
```

Example of choosing other grounded colors

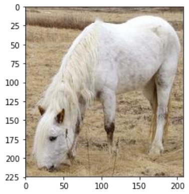
Scenario2.3 (SM)

Real examples from game: Reusing color histograms from external knowledge.

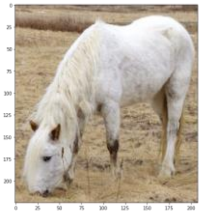
this is selected image : sample_person.jpg



Are you ready?
please type confidence level 0.0 - 1.0 :
>> 0.6
there are total 3 number of object deteced : ['person', 'dog', 'horse']
Decide your target. are you ready?
\>>yes
Where is the object? Left/Right/Above/Below?
>> right

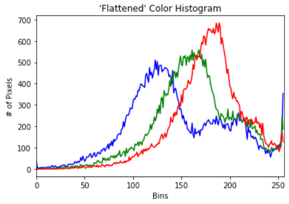


Is it among ('black', 'grey') ? or doesn't exist?(type : color_name / no)
\>>no
Let's narrow it down, Again, Where is the object? Left/Right/Above/Below?
>> right



Is it ['horse'] ? or doesn't exist?(type : y/n)
>>y

Extracting visual features : ResNet50 features and color histogram.



I can see grey horse in the picture, but label or color seems wrong.

which one is wrong? type: label/color/both
\>>color

I interpret this color as grey. if it is not what you think, type: not here

Possible colors : ['grey', 'tan', 'beige', 'light grey', 'light brown', 'taupe', 'brown', 'grey-brown', 'sand', 'warm grey']
\>>not here
Using External Knowledge Data

I am searching object color from external knowledge data, color histograms.
Please wait......
this is error list:

[('./storage/zebra_white with black stripes.json', 292.99091860570809), ('./storage/horse_white.json', 0.0), ('./storage/oven_rainbow.json', 651.72279203373728), ('./storage/oven_brown.json', 651.72279203373728)]

Is it color, white? y/n (trial:0)
>>n
Is it color, white with black stripes? y/n (trial:1)
>>n
Is it color, rainbow? y/n (trial:2)
>>n

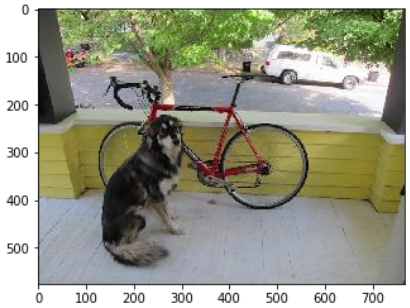
can you tell me color of object, please?
>> clear white

------ Saved file horse_clear white.json ------
the game is over

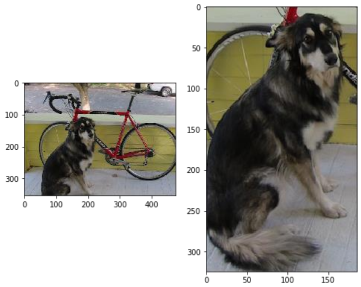
Scenario2.4 (SM)

Real examples from game: Reusing ResNet50 features from external knowledge.

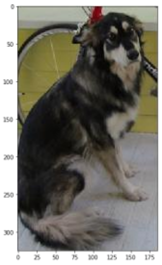
```
this is selected image :  sample_dog.jpg
```



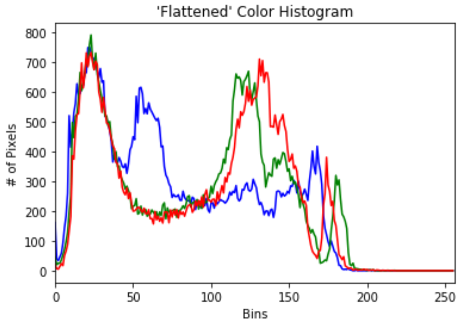
```
 Are you ready?
 please type confidence level 0.0 - 1.0 :
 >> 0.7
there are total 3 number of object deteced : ['bicycle', 'truck',
'dog']
Decide your target. are you ready?
>>yes
Where is the object? Left/Right/Above/Below?
 >> below
```



```
Is it among ('brown', 'black') ? or doesn't exist?(type :
color_name / no)
>>no
Let's narrow it down, Again, Where is the object?
Left/Right/Above/Below?
 >> below
```



```
Is it ['dog'] ? or doesn't exist?(type : y/n)
>>n
Is name wrong? or object not exist in list ['dog'], type
wrong/not exist
>>wrong
Okay, so object exist, but label is wrong!
Shall we narrow it with wrong label...
Extracting visual features : ResNet50 features and color
histogram.
```



```
['0.095388' '0.133069' '0.202854' '0.200689' '0.227916'
'0.597887'
 '0.020127' '0.005349' '0.111241' '0.227393'] 2048
I can see black dog in the picture, but label or color seems
wrong.
                    which one is wrong? type: label/color/both
>>label
I see object dog. but it seems wrong.
 Using External Knowledge Data

I am searching object name from external knowledge data.
Please wait......
[('./storage/tvmonitor_dark grey.json', 0.9229998251594793),
('./storage/person_blue and black.json',
1.0013692063492659), ('./storage/zebra_white with black
stripes.json', 0.9988250313385849),
('./storage/horse_white.json', 0.8472217235829329)]
Is it name, horse? y/n (trial:0)
>>n
Is it name, horse? y/n (trial:1)
>>n
Is it name, tvmonitor? y/n (trial:2)
>>n


can you name this object, please?
>> wolf


------ Saved file wolf_black.json ------
the game is over
```