Visual Inference Using Homology of Human and Machine Vision Systems

Jacob W. Morosco JACOB.MOROSCO@COVENANT.EDU
Department of Computer Science, Covenant College, Lookout Mountain, GA 30750 USA

Yuchou Chang YCHANG1@UMASSD.EDU
Department of Computer and Information Science, University of Massachusetts Dartmouth, North Dartmouth, MA 02747 USA

Abstract

The homology of human and machine vision systems demonstrates that better machines could be designed with human assistance. Similar components can be mapped from neuroimaging data to visual features for recognizing an object. However, inferring object relationships between human and machine vision is unclear. To measure the similarity of human and machine visual inference, this work studies an inference method using the Microsoft COCO dataset. The input data is manually generated, and used for a java-based inference engine, which collects semantic data in a co-occurrence matrix, and writes the data to a knowledge graph in the DOT language. Unlike the black-box property of a deep neural network, the proposed method is transparent. When rendered by GraphViz tools, the visible results in the knowledge graph indicated that the COCO dataset-based machine inference is promising when compared to human inference, yielding an accuracy of 64% at best. This novel inference study on the COCO dataset reveals that the homology of human and machine vision systems is promising to be bridged. A bigger dataset and more concepts may increase the accuracy of future work.

1. Introduction

The machine vision system has been found to have homology with the human vision system through functional magnetic resonance imaging (fMRI) study (Horikawa & Kamitani, 2017). The connection of EEG signals of the human brain and the visual representation of an external machine is recently established, and it may bridge the gap of vision sharing between humans and agents. The research breakthrough of neuroscience demonstrates a homology between human and machine vision through the discovery of the joint brain-visual representation. Brain representation is extracted from fMRI (Horikawa & Kamitani, 2017) or Electroencephalography (EEG) signals (Palazzo et al., 2021), and visual representation is obtained by a convolutional neural network (CNN) or other hand-crafted features. The relationship between brain representations and hierarchical structures of layers of CNN makes it possible to not only predict the brain states in awake and anesthetized resting-state functional neuroimaging data (Grigis et al., 2020) but also transfer human knowledge to design better machines (Palazzo et al., 2021). Their representations of an object can be shared with each other on the corresponding layers of a deep neural network.

Based on accumulated knowledge, humans generally infer another object when one object is observed. For example, if there is a cup on the table, human infers that water may be put into the cup. This inference capability is related to both knowledge and inference from the human brain. However, this inference ability is not clear for an intelligence machine. Due to the homology of human and machine vision systems, visual inference of an external machine may also be consistent with the human visual system.

Inference, as defined by Merriam-Webster, is "a conclusion or opinion that is formed because of known facts or evidence." Human inference relies on background knowledge that humans already know that can be used to make certain assumptions or predictions about the world. Copernicus's theory indicates that the Earth orbited the Sun could not initially be confirmed because insufficient background knowledge was present to make accurate inferences. It was not until enough of this knowledge was collected and processed by some of the best scientific minds of the century that his theory was affirmed. This is an example of human inference in action. Likewise, the concept of artificial intelligence (AI) inference seeks to mimic and take advantage of the way the human brain makes inferences by using a combination of background knowledge and logical rules to make accurate predictions and decisions. This is also known as Rule-based AI because of the policies enacted upon the background knowledge to produce the desired results.

The more background knowledge there is to draw from, the more accurate the inferences from it will be. Fortunately for us, we are in the age of information--the AI revolution. We have more information available to us today than any civilization ever has, and the amount of information has caused our sum of knowledge to grow significantly. For the first time, we have both the technology, and the knowledge available to us to make significant improvements to AI and how it can be used, including the area of AI inference. However, in order to apply logical rules to this background knowledge, we first need a way to quantify the knowledge. This can be done using a knowledge graph, also known as a semantic network. Semantics, as a whole, is beyond the scope of this paper. We need only to understand how it plays into the quantification of the background knowledge used to make inferences. A knowledge graph is a theoretical graph structure in which the nodes are concepts, and the edges are relationships between the concepts, either with or without direction, hence the semantics. The bigger the knowledge graph, the more knowledge it contains, and the more accurate the inferences drawn from it can be. Fig. 1 shows an example of a small knowledge graph., which contains multiple knowledge concepts. A connection between two concepts can infer their relationship in real-world environments.

Since homology exists between human and machine vision systems, inference capability based on both vision systems is still unknown. We will study the commonalities and differences between human vision and machine vision-based inference systems. Inferences will be made from the Microsoft COCO dataset by visualizing a sample of the connections on the knowledge graph. While COCO has been used in other AI techniques such as object detection, there has yet to be a study of the inferences that can be made from the dataset using an inference engine. We will use an inference engine implemented in java to make the inferences. First, a portion of the COCO dataset is fed into the engine, which will determine the concepts in the dataset that are related to one another. Next, the engine will use this semantic data to write a knowledge graph in the DOT language. Finally, we will use GraphViz command line tools to render a knowledge graph and study the semantic clusters that appear in the graph visualization. These clusters are the inferences drawn from the COCO dataset and represent new discoveries that have not been studied yet. Unlike the black-box property in deep neural networks, the proposed method is transparent using a co-occurrence matrix, so a clear path of inference can be identified. Two human volunteers will also observe the objects and infer relationships among objects. Both human and machine inferences will be compared for evaluating the performance of the proposed method. In this paper, section 1 introduces the vision homology between humans and machines. Related works are given in the section part. The proposed methods are presented in the third section. Results are provided in the fourth part. Finally, a conclusion is given.

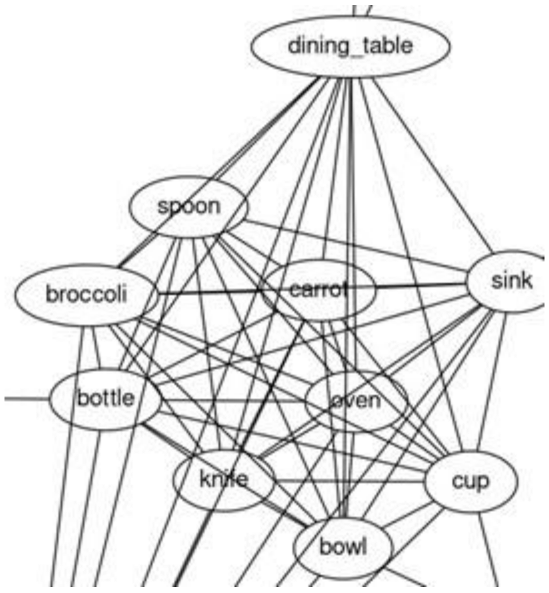


Fig. 1. An example of a small knowledge graph.

2. Related Work

2.1 Microsoft COCO Dataset

The Microsoft COCO Dataset (Common Objects in Context) is a large, well-annotated dataset meant for improving object recognition in computer vision (Lin et al., 2014). Unlike object detection and object localization, which focus on whether an image contains a certain concept, and where that concept is located in the image, respectively, the COCO dataset focuses on individual object segmentation and is used to train and evaluate machine learning models. One such model can be seen in the knowledge-aware object detection framework (Fang et al., 2017) where COCO is used to test an optimized object detection algorithm that uses semantic consistency. COCO was also used to evaluate a virtual semantic reasoning network (Li et al., 2017) which was a convolutional neural network (CNN) used to recognize the key objects and semantic concepts in images within a dataset to perform image-text matching of a semantic scene.

Fig. 2 shows a page from the COCO online dataset explorer. It is seen that multiple concepts (e.g. bird) have been identified and segmented in the objective image. Those concepts are corresponding to segmented objects in an image.

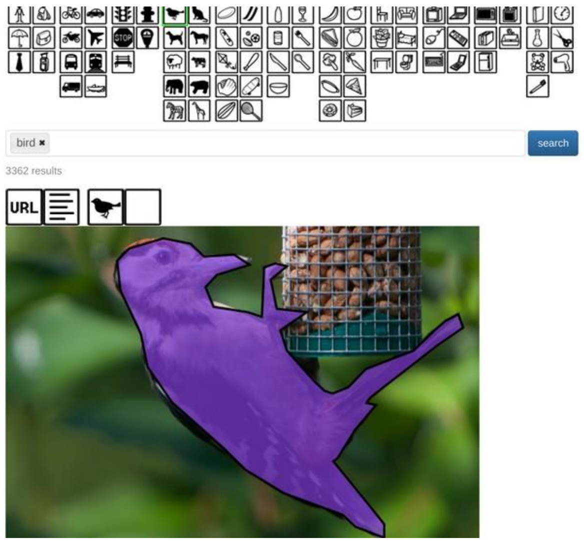


Fig.2. COCO online dataset explorer. A bird object is segmented in the sample image.

2.2 Inference and Inference Engines

Inference and its uses have been long-standing topics in many different fields. An in-depth manual for the migration from statistical analysis to causal analysis can be found in J Pearl's overview (Pearl, 2009), along with a number of mathematical solutions for causal effects, counterfactuals, and mediation. Inference engines have also been subject to a variety of uses. VIBES, the Variational Inference Engine for Bayesian Networks (Christopher, 2002), uses an algorithm to automatically generate and solve variational equations in probabilistic learning models. Another inference engine, Random Walk (Lao et al., 2011), uses a modified version of the Path Ranking Algorithm to make new inferences on a large-scale knowledge base and further improve learning accuracy.

2.3 Knowledge Graphs

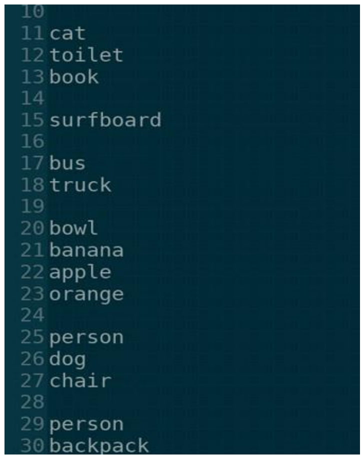
As new knowledge has continued to multiply, new knowledge graphs and knowledge bases have become available. One such knowledge base is ConceptNet (Liu et al., 2004), a semantic network and natural-language processing toolkit supporting many context-oriented inferences of commonsense knowledge. YAGO (Suchanek et al., 2007), another popular semantic knowledge base, uses rule-based methods to combine semantic knowledge from WordNet and Wikipedia, creating a stable and highly accurate model. YAGO is not alone in its quest to fuse existing knowledge bases together. Knowledge Vault (Dong et al., 2014) leverages supervised machine learning techniques to automatically construct large-scale knowledge bases from multiple existing knowledge sources, including Google's own Knowledge Graph. Knowledge vault also includes its own internal inference engine that uses probabilistic inference to determine the factual accuracy of a newly compiled knowledge vault, demonstrating the common cooperation of inference techniques and semantic knowledge vehicles.

3. Proposed Method

The inference engine is what bridges the gap between the semantic data in the COCO dataset and the knowledge graph we can produce from the data. This will be the focal point of this paper. The main two tasks of the inference engine are to gather the data from the input and to write it into an output format that can be visualized. The inference engine operates much like any other algorithm: it has an input, a sequence of instructions or operations, and an output. The java implementation of the inference engine has three main pieces: a Dictionary, a Co-occurrence Matrix, and a Knowledge Graph.

3.1 Input Generation

To generate the input, we need a pool of COCO images and a way to extract the concepts from the images so that the inference engine will understand how to use them. A concept is an object within an image, for instance, a cat, or a dining room table. The Python library FiftyOne was used in a small Python program to determine which images should be selected and used. FiftyOne selected a randomly generated set of 200 images from the 2017 training split of the COCO dataset and saved them to a local directory. Each image in this set was then manually located in the online COCO dataset explorer by using its unique image ID. The explorer provides a simplified view of the annotations for each image, including all the concept labels. As shown in Fig. 3, the concepts in each image were then recorded in a text file, with one concept label per line. A blank line was used to indicate the end of an image and the beginning of the next image in the set. Thus, all the concepts for each image are placed in their own group in the text file, and each group is separated by one blank line. This will help the engine to read through the file later. Our text file represented just under 250 images containing 85 unique concepts.



```

cat
toilet
book

surfboard

bus
truck

bowl
banana
apple
orange

person
dog
chair

person
backpack
```

Fig. 3. Part of one of the input text files. All the concepts for each image are placed in their own group in the text file.

3.2 Dictionary

Once our input file is constructed, we will read through it, putting each line as a String entry into two distinct hash tables, which are both fields in the Dictionary object. Each hash table entry will be identified by a positive integer, which is the key. The first hash table will contain every line in the file, including blank lines indicating separate images, labeled as "BLANK". The second hash table will contain one entry for every unique concept, even if this concept is found in many of the image representations in the text file. This dictionary object will serve as a miniature database from which we will draw semantic knowledge about COCO.

3.3 Co-Occurrence Matrix

The next step is to use the contents of the Dictionary object to produce a co-occurrence matrix. This matrix will be implemented by the COMatrix class, which uses a two-dimensional integer array to store the co-occurrence data found within Dictionary. The size of the matrix is determined by the total number of unique concepts in the second hash table. The purpose of this matrix is to record the number of times a given concept occurs in the same context as another given concept (appears in the same image). For example, let us suppose that there are 17 separate images in which both a person and a surfboard occur in the same image. In this case, the value at the intersection of the (person, surfboard) vertex in the co-occurrence matrix would be 17, and so also would be the (surfboard, person) vertex, because this is the co-occurrence value of these two concepts. This only applies to distinct objects, so if there are two surfboards in one image, this does not mean that we increase the count; the concepts are not self-reflexive.

3.4 Knowledge Graph

The third and final class in the implementation is the KnowledgeGraph object. The purpose of this object is to analyze the co-occurrence matrix and write an output file in the DOT language, which will be the actual knowledge graph. DOT is an open-source graph description language that can be used to write and define the nodes and edges in both undirected and directed graphs. Due to the time constraints on this project, only undirected graphs were used. DOT is commonly associated with the GraphViz package, which provides many tools for rendering and viewing graphs that were created using the DOT language. After the KnowledgeGraph object is initialized, two methods will be called on it: one to draw the nodes, which are the concepts, and another to draw the edges, which are the relationships between the concepts. Consideration can be given to the values found in the co-occurrence matrix. For example, if the value in the matrix at (cat, table) is very high, this means that there is a very strong relationship between these two concepts, and could be represented by a different color or size node. We can also program the method to draw only the nodes which have very strong relationships with each other, say, at least 3 co-occurrences. When the inference engine is run by the main, the result will be a DOT file (*.gv, *.dot) which can be rendered into a PNG image by the GraphViz tools. At this point, we are now able to see the inferences represented by the clusters of semantically related concepts. As shown in Fig. 4, a knowledge graph is produced by the proposed inference engine. A threshold of 3 is selected to remove some irrelevant items. It is clearly seen that some clusters exist for grouping related objects together. For example, “person” is the center of a big cluster in which the “person” object has strong relationships to other objects, so an inference can be obtained from “person” to those related objects.

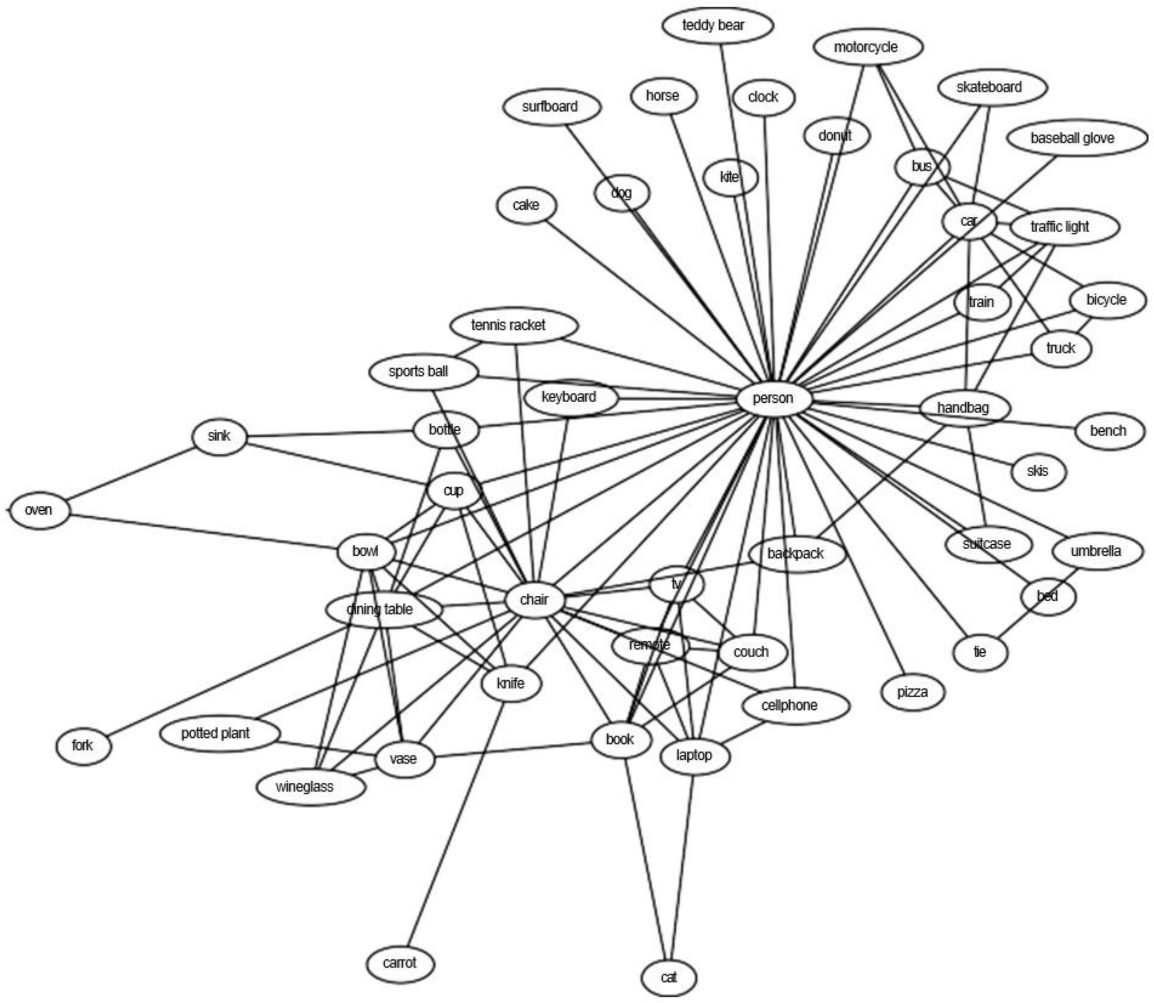


Fig. 4. A knowledge graph produced by the inference engine from our input. In this graph, all the related concepts have at least 3 co-occurrences. A quick look can see about 3-4 clusters of related concepts.

4. Results

We have categorized the most obvious concept clusters in the knowledge graph into three main categories, which are numbered respectively: Kitchen/Dining Room (1), Family/Living Room (2), and City Street (3). These three categories were then tested against two volunteers and their human knowledge of these given categories, so as to compare the output of the inference engine to the gold standard: human inference. The volunteers were asked to list what they thought to be ten common objects in these three given contexts. Tables I. and II. compare on two accounts the inferences found within a random sampling of over 240 images from the training split of the COCO 2017 dataset with that of human inferences. The column heads indicate human inferences with 'H', and COCO inferences with 'C,' starting with category 1, and ending with category 3.

TABLE I. COCO INFERENCES AGAINST VOLUNTEER #1

1 (H)	**1 (C)**	**2 (H)**	**2 (C)**	**3 (H)**	**3 (C)**
Oven	Yes	TV	Yes	Street Light	No
Fridge	Yes	Couch	Yes	Fire Hydrant	Yes
Pot	No	Coffee Table	No	Street Sign	Yes[b]
Pan	No	Photo	No	Building	No
Coffee Maker	No	Book	Yes	Church	No
Dishwasher	No	Board Game	No	Bus	Yes
Microwave	Yes	Dog	No	Car	Yes
Dish	Yes[a]	Lamp	No	Sidewalk	No
Dish Soap	No	Window	No	Person	Yes
Sink	Yes	Fan	No	Bird	No
Table	Yes	N/A	N/A	N/A	N/A
Chair	Yes	N/A	N/A	N/A	N/A

a. The dataset was more specific with different types of dishes, but had no concept of a general "dish," so we answered this question as yes.

b. The dataset did not have a general "street sign," but was more specific, and contained a "stop sign." We considered this inference accurate and marked it as yes.

The inference engine revealed that COCO had a 46.88% accuracy when compared with the human inference from volunteer 1, and a 46.67% accuracy when compared with volunteer 2. The average overall accuracy for both volunteers including all three context categories was 47.77%. When taking each category individually, the category in which COCO had the highest accuracy when tested against human inference was category 1, the Kitchen/Dining Room at 63.63% combined accuracy, and 64.17% average accuracy. Categories 2 and 3 were 30% and 45% combined accuracy, respectively. The limitations of the results are due to the small sampling size of the COCO dataset and the small sampling size and variety of human inference that it was tested against. The COCO dataset contains hundreds of thousands of images, and only 240 were used for the input to the inference engine. This means that the clusters of related concepts that were used to display the inferences could have been much larger and included more nodes and edges had the input been larger. An increase in the size of the input would likely result in more clusters that could be tested on human inference. Likewise, the amount of human inference that was used to test the main categories was very minimal— only ten concepts—and was only collected from two volunteers. An increase in the number of volunteers to test against would result in a more reliable evaluation of the artificial inferences.

TABLE II. COCO INFERENCES AGAINST VOLUNTEER #2

1 (H)	1 (C)	2 (H)	2 (C)	3 (H)	3 (C)
Table	Yes	Couch	Yes	Street Light	No
Chair	Yes	chair	Yes	Traffic Cone	No
Sink	Yes	TV	Yes	Street Sign	Yes[c]
Fridge	Yes	Coffee Table	No	Person	Yes
Microwave	Yes	End Table	No	Car	Yes
Oven	Yes	Blanket	No	Trash Can	No
Cabinet	No	Pillow	No	Cross Walk	No
Plate	No	Rug	No	Pot Hole	No
Table	Yes	Couch	Yes	Street Light	No
Fork	Yes	Fireplace	No	Dog	Yes
Dish washer	No	Shelf	No	Sidewalk	No

[c.] Please see table footnote ‘b’ about the stop sign.

5. Conclusion

5.1 Summary and Discussion

We proposed an inference engine based on the homology of human and machine vision systems. It is different from deep network architecture and has the transparency of the inference process. Our results are tested against the gold standard of human inference, with even the best cluster obtaining an accuracy of 64%. Human visual inference and machine vision inference still have gaps. Perhaps the accuracy would increase if many more images from the dataset were processed and there was more of a variety of concepts, but our input size did not have enough variety to yield greater accuracy. However, it is also worth noting that when the size of the input was increased from 100 to 200 images, the number of unique concepts found within those images increased only from 72 to 82. If the COCO dataset truly is more accurate against human inference than this study has shown, it would take a vastly larger input to verify, and a much stronger inference engine with more computing power. Furthermore, complicated inference by deep neural networks may improve the accuracy due to its larger capacity model. But transparency is sacrificed. Homology between human and machine vision provides a promising direction for co-inferring an object in an environment.

5.2 Future Work

One improvement to this work would be a way to generate the input file for the inference engine without having to construct the text file manually. This way, we could also generate a much larger input to feed to the engine and produce a more complex knowledge graph. Another improvement would be a better variety of human inference to test against, perhaps people from different cultures or ethnicities. Lastly, future work could take advantage of the wide range of capabilities of the DOT language in writing undirected graphs. This would make the renderings of the knowledge graph clearer, and the inferences would be easier to see and identify if the concept clusters could be displayed in several different ways by the GraphViz tools.

Acknowledgments

This work was supported by the National Science Foundation under grant number 2050972.

References

Christopher, B., Spiegelhalter, D., & Winn, J. (2002). *VIBES: A variational inference engine for Bayesian networks*. Advances in Neural Information Processing Systems, 15.

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W. (2014). *Knowledge vault: A web-scale approach to probabilistic knowledge fusion*. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Fang, Y., Kuan, K., Lin, J., Tan, C., & Chandrasekhar, V. (2017). *Object detection meets knowledge graphs*. International Joint Conferences on Artificial Intelligence.

Grigis, A., Gomez, C., Tasserie, J., Ambroise, C., Frouin, V., Jarraya, B., & Uhrig, L. (2020). *Predicting cortical signatures of consciousness using dynamic functional connectivity graph-convolutional neural networks*. bioRxiv - Neuroscience.

Horikawa, T., & Kamitani, Y. (2017). *Generic decoding of seen and imagined objects using hierarchical visual features*. Nature Communications, 8, 15037.

Lao, N., Mitchell, T., & Cohen, W. (2011). *Random walk inference and learning in a large scale knowledge base*. Proceedings of Conference on Empirical Methods in Natural Language Processing.

Li, K., Zhang, Y., Li, K., Li, Y., & Fu, Y. (2019). *Visual semantic reasoning for image-text matching*. Proceedings of the IEEE/CVF International Conference on Computer Vision.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014). *Microsoft coco: Common objects in context*. European conference on computer vision. Springer.

Liu, H. & Singh, P. (2004). *ConceptNet—a practical commonsense reasoning tool-kit*. BT technology journal, 22, 4, 211-226.

Palazzo, S., Spampinato, C., Kavasidis, I., Giordano, D., Schmidt, J., & Mubarak Shah, M. (2021). *Decoding brain representations by multimodal learning of neural activity and visual features*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43, 11, 3833 – 3849.

Pearl, J. (2009). *Causal inference in statistics: An overview*. Statistics surveys, 3, 96-146.

Suchanek, F.M., Kasneci, G., & Weikum, G. (2007). *Yago: a core of semantic knowledge*. Proceedings of the 16th international conference on World Wide Web.