CSCI 8980: Implementation of Generalized Grounding Graphs with Video Input

Yutong Li
University of Minnesota
lixx2874@umn.edu

December 2021

Abstract

More and more autonomous robots are showing up in real people's life and helping people perform tasks that are not easy to be achieved by human beings.[3] However, understanding human's natural language commands is not an easy task when people with different backgrounds may employ different vocabularies and grammars when assigning tasks to robots. Once the task is given to the robot, it's also smart if the robots can evaluate whether the command is executable based on its surroundings. In this way, robots can save more energy by not performing the tasks given by human beings that are not executable at all.

There are in total three sections in this project. Firstly, I implemented Generalized Grounding Graphs(G^3)[1] which defined a probabilistic graphic model dynamically according to linguistic parse structure of natural language command so that robots can better understand the natural language commands. Secondly, I added object detection feature to the model so that robots can use this feature to detect its surroundings. And at last, I implemented the feature to help robot calculate the probability of the natural language commands based surroundings detected in 2nd section.

Results showed that by using G^3, object detection features, and probability calculation feature together, robot can achieve two goals, including gained better understandings of whether the natural language commands are executable, and what is the best time to start executing the command.

1 Introduction

More and more robots autonomous robots are put into industrial and corporate with human beings nowadays. In first 9 months of 2021, orders of industrial robots climbed 37% as compared to the same period in 2020. Around 29,000 robots were ordered to work in factories [2] When more and more robots are now working with human beings, the communication between human beings and robots is becoming more and more important. Robots need to understand the natural language command accurately so that they can perform tasks given by human beings. To help robots achieve this goal, Generalized Grounding Graphs(G^3) which defined a probabilistic graphic model dynamically according to linguistic parse structure of natural language command was introduced by doctor Kollar and professor Tellex [1]. This model helps predict physical interpretations or

groundings for linguistic constituents. Specific groundings including objects, places, paths, or events are considered as nodes within the graph.

I used Spacy to analyze the dependencies of the input natural language command and built G^3 draft using the groundings mentioned in the natural language command. In addition, I implemented YOLO v3 in my model to help detect objects surrounded. Once there are groundings in G^3 detected by the agent in its surroundings, agent would update the G^3 and calculate the probability of commands execution. As a result, the robot can find an optimal time point that when the probability of the human's natural language command is maximized and start executing the task, or it can simply refuse performing the task when it finds that it's impossible to perform the task since there are groundings missing in its surrounded environment.

In section 2, some related works about Generalized Grounding Graph(G^3) are introduced. In section 3, specific methodology used in this project are fully explained. Experiment details and setups are provided in section 4. Results and evaluations are listed in section 5. Limitation of current model is listed in section 6. At last, conclusion and future work are explained in section 7.

2 Related Work

Generalized Grounding Graphs(G^3) is introduced by doctor Kollar and professor Tellex in 2017. It dynamically instantiates a probabilistic graphical model for a particular natural language command according to the command's hierarchical and compositional semantic structure[4]. Based on the contents within the commands, different components are initialized within Generalized Grounding Graph(G^3).

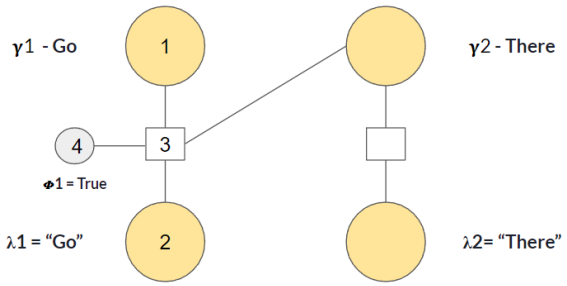

Figure 1: Structure of Generalized Grounding Graph(G^3)

There are several key components needed for building Generalized Grounding Graphs and they are showing below:

- λ - Words from commands (i.e. Node 2 in figure 1)
- γ - Groundings (i.e. Node 1 in figure 1)
- ϕ - Corresponding variance of words and groundings (i.e. Node 4 in figure 1)
- link nodes - nodes that represent the dependencies among ϕ, γ, and λ (i.e. Node 3 in figure 1)

By filling information into nodes in figure 1 using key information extracted from natural language commands, the relationship between natural language command and groundings desired by the robot can be fully represented in a Generalized Grounding Graph. Specific details of key information extraction algorithm is fully explained in section 3.2.1.

3 Framework & Methodology

3.1 Framework

There are in total 3 sections within this project and they are showing in figure 2 .

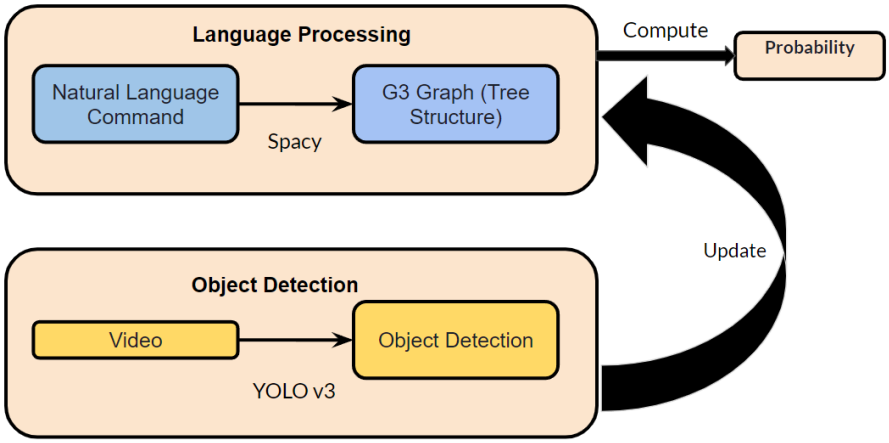

Figure 2: Framework of project

- Language Processing

  User's natural language command is entered into the model as input. Robot analyzes the dependencies of words in input commands and builds a Generalized Grounding Graphs draft based on the groundings robot is interested in.

- Object Detection

  Robot takes a video as input. Input video can be either entered by users manually, or sensed by robot using mono-camera from its surroundings. From the input video, the robot uses the model trained through convolutional neural network predict groundings in its surroundings.

- Probability Computation

  Once there are new groundings detected by the robot, robot updates Generalized Grounding graphs using the grounding it just detected. If there are any updates happening to the G^3, robot recalculates the probability of updated G^3 and the outcome is the probability of input natural language command.

- Summary

  Once the robot finds that the probability of the natural language command is equal to 1, which means that this task is executable by the robot, it will execute the task immediately and this time point is considered as the best time to start executing the task. However, if the probability of the natural language command is not good enough, which means that it's lower than the threshold defined by the designer or even worse, the robot informs users that given task is not executable.

3.2 Language Dependencies Analyze & Generalized Grounding Graphs Draft

3.2.1 Language Dependencies Analyze

In language processing section, language dependencies need to be analyzed at first. Robot only considers several groundings in G^3 including objects (e.g., a truck or a door), places (e.g., a particular location in the world), paths (e.g., a trajectory through the environment), or events (e.g., a sequence of actions taken by the robot)[1]. I used Spacy library to help me with this process. In specific, there are several kinds of words that should be considered as groundings in the graph.

- NOUN - Object or Place
- VERB - Event
- ADP - Path or Preposition
- DET or ADJ - Decorations of Objects and Places

Once words that our model is interested in are extracted from the natural language command, the dependencies can be analyzed. There are several relationship that should be considered and they are listed below.

- Event + Object/Place

  This is the most basic dependency in the graph. When an action is required for the robot to perform, there's always a place/object noun word after it which tells what's the object/place that this action should act on. A simple example of this pattern is "Go There".

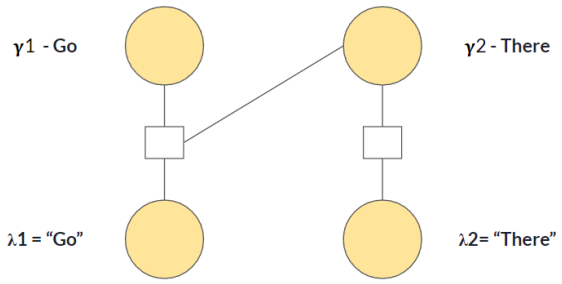

Figure 3: Event + Object/Place Dependency - Go There

- Event + Object & Path/Preposition

  A more complex dependency is consist of an event word that needs to act on two words. One word might be an object, when another one is a prepositional word/path. An example of this pattern is "Put Books On".

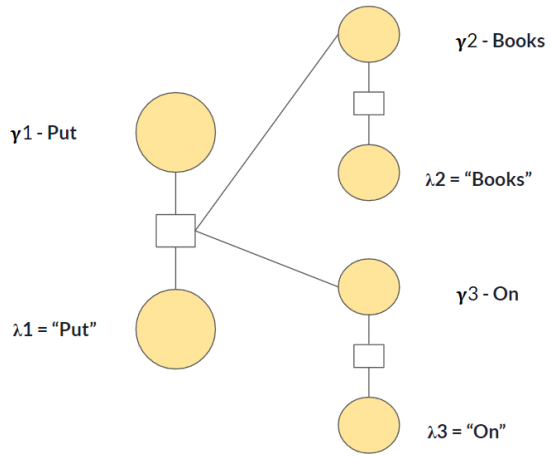

Figure 4: Event + Object & Path/Preposition Dependency - Put Books On

- Path/Preposition + Object/Place

  And the last dependency that the model can find out is the the relationship between path/prepositional word and object/place. An example of this pattern is "On the chair".

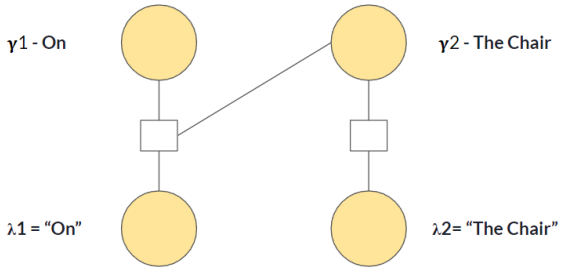

Figure 5: Path/Preposition + Object/Place Dependency - On The Chair

3.2.2 Grounding Graphs Draft

Once words robot is interested in are extracted and the dependencies among them are analyzed, the robot will build G^3 in a tree structure by linking all those sub dependency sections together. At the meantime, the corresponding variance ϕ that marks the relationship between λ and γ are initialized.

Below is an specific example of the Generalized Grounding Graph draft for natural language command "Put those books on the chair". In this example, I assumed that the robot is capable of the action "Put", and there's nothing on the chair. Thus, the corresponding variance ϕ between γ_1 and λ_1, as well γ_2 and λ_2 are both True.

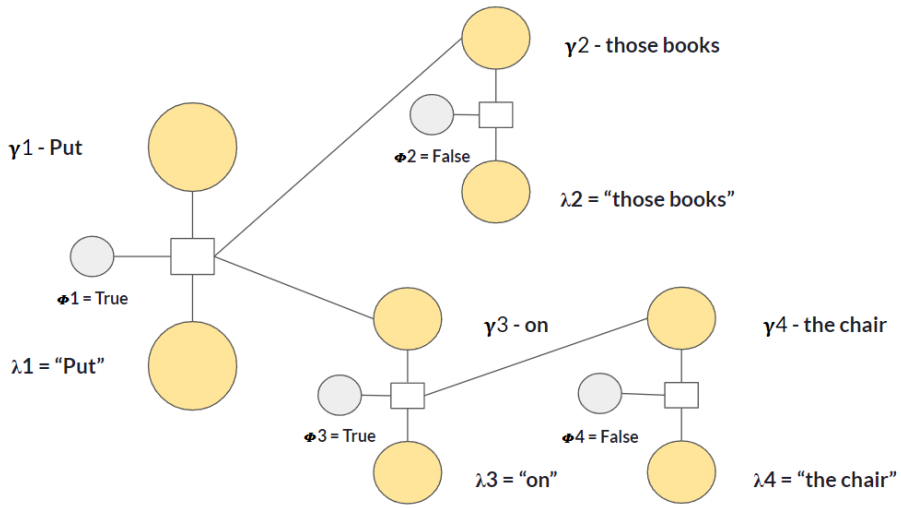

Figure 6: Generalized Grounding Graph Draft - Put those books on the chair

3.3 Object Detection & G^3 Graph Update

Once Generalized Grounding Graph draft is generated, robot is going to sense groundings in its surroundings and update G^3.

YOLOv3 is implemented to the model for object detection. YOLO is a real-time object detection algorithm that identifies specific objects in videos. It used features learned by a deep convolutional neural network to detect objects. It's widely used nowadays because that it's faster than other networks when it can still maintains accuracy of objects prediction.[5]

In this project, I used YOLOv3 pretrained on COCO dataset. This model is trained using more than 330,000 images and it can accurately predict 80 objects. The threshold of my object detection is set as 80% so that if and only if the model is more than 80% confident that object detected belongs to one of the 80 objects it learnt, it will label that object. What's more, I also implemented a cache in the model to help make the object detection more accurate. If and only if the detected object is the same in 5 frames, robot is sure that there's a new object detected in its surroundings and the G^3 should be updated.

Once there are objects that are initialized as groundings got detected by the robot, it will update the Generalized Groundings Graph that was built in previous section. More specifically, the corresponding variance of objects/places and λ of them will be updated.

3.4 Probability of Natural Language Command & Task Executing

The last step of the system is to recalculate the probability of the G^3 and find out whether the task given by the user is executable. The probability of natural language command is defined as the probability of the Generalized Grounding Graph the robot built. And the probability of natural language command is calculated by multiplying all link nodes (circle in red in figure 7) together.

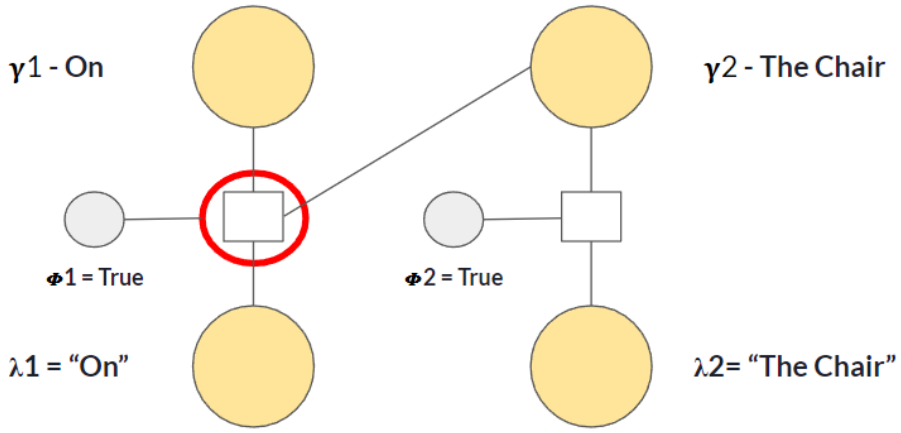

Figure 7: Link Node Example

Conditional probabilities need to be used because of the existence of dependencies. In figure 7, the first link node links γ_1, γ_2, ϕ_1, and λ_1. The equation of the probability of first link node is defined as

$$p(\phi_1|\gamma_1, \gamma_2, \lambda_1 = "Put") \tag{1}$$

A more concrete example can be seen from figure 6 when the probability of the natural language command can be computed as

$$\begin{aligned} &p(\phi_1|\gamma_1, \gamma_2, \gamma_3, \lambda_1 = "Put") \\ &\quad * p(\phi_2|\gamma_2, \lambda_2 = "thosebooks") * p(\phi_3|\gamma_3, \gamma_4, \lambda_3 = "on") * p(\phi_4|\gamma_4, \lambda_4 = "thechair") \end{aligned} \tag{2}$$

Sometimes, there might be a optimal solution for the graph and robot can find the probability of the whole structure is 1 which means that robot is 100% sure the task is executable. Once this result is concluded, the robot stops sense groundings in its surroundings and starts executing the tasks. However, in most cases, it's not possible. Thus, a threshold can be used and robot may start executing the task once the probability of the whole structure is above that threshold. In contrast, if robot cannot find the probability over the threshold after sensing the system for several rounds, it will stop sensing and let user know that the task is not executable.

4 Experiment

The experiment was taken in my apartment. A simple natural language command and a video taken by me using IPhone XR are entered into the model. Since this is only a simulation experiment rather

than testing it on the real robot, I simply give several basic actions that robot can perform including put, lift, move, etc. What's more, I assume that the corresponding variance ϕ for prepositional words are always true.

5 Results & Evaluation

5.1 Executable Task - Put those books on the chair

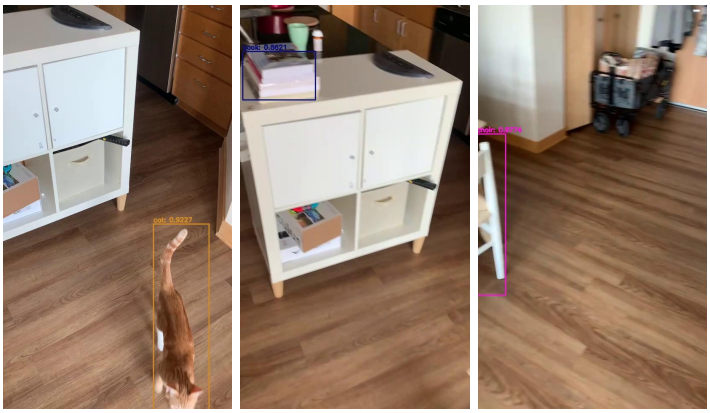

Figure 8: Robot's surroundings at start point, when grounding books were detected, and grounding chair was detected

```
Semantic Language G3 Map - Before Sensing
 └─ Put(EVENT) Corresponding Variance Phi:  1
    ├─ those books(OBJECT) Corresponding Variance Phi:  0
    └─ on(PLACES OR PATHS) Corresponding Variance Phi:  1
       └─ the chair(OBJECT) Corresponding Variance Phi:  0
Probability of language command:  0
 └─ Put(EVENT) Corresponding Variance Phi:  1
    ├─ those books(OBJECT) Corresponding Variance Phi:  1
    └─ on(PLACES OR PATHS) Corresponding Variance Phi:  1
       └─ the chair(OBJECT) Corresponding Variance Phi:  0
Probability of language command:  0
 └─ Put(EVENT) Corresponding Variance Phi:  1
    ├─ those books(OBJECT) Corresponding Variance Phi:  1
    └─ on(PLACES OR PATHS) Corresponding Variance Phi:  1
       └─ the chair(OBJECT) Corresponding Variance Phi:  1
Probability of language command:  1
```

Figure 9: Generalized Grounding Graph & probability at start point, when grounding books were detected, and grounding chair was detected

The natural language command of first experiment is "Put those books on the chair". Images in figure 8 shows robot's surroundings at start point, when grounding books were detected, and

grounding chairs were detected by YOLOv3. At the same time, the structure of the Generalized Grounding Graph and probability were updated and screenshots are showing in figure 9.

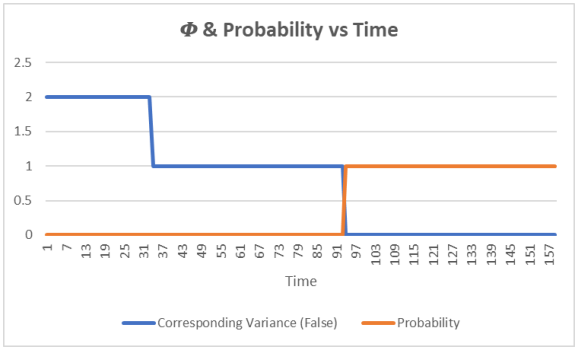

Figure 10: Corresponding Variance(False) & Probability vs Time

In figure 10, number of corresponding variance that are false and probability verses time plot of 1st experiment is provided. At time frame 93, all corresponding variances in Generalized Grounding Graph are updated to true and the probability of G^3 is updated to 1. As a result, robot knows that the command is executable at time frame 93 and that's the best time to start executing the task.

5.2 Non-executable Task - Jump on the chair

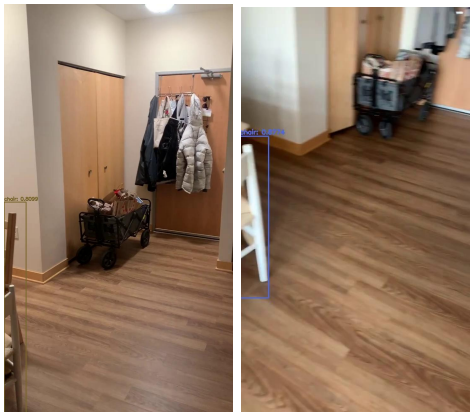

Figure 11: Robot's surroundings at start point, and when grounding chair was detected

The natural language command of second experiment is "Jump on the chair". Images in figure 11 shows robot's surroundings at start point, and when grounding chair is detected by YOLOv3. The structure of the Generalized Grounding Graph and probability of G^3 is showing in figure 12.

In figure 11, number of corresponding variance that are false and probability verses time plot of 2nd experiment is provided. Even though the corresponding variance of grounding chairs and

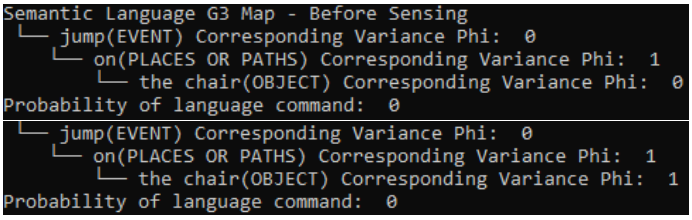

Figure 12: Generalized Grounding Graph & probability at start point, and when grounding chair was detected

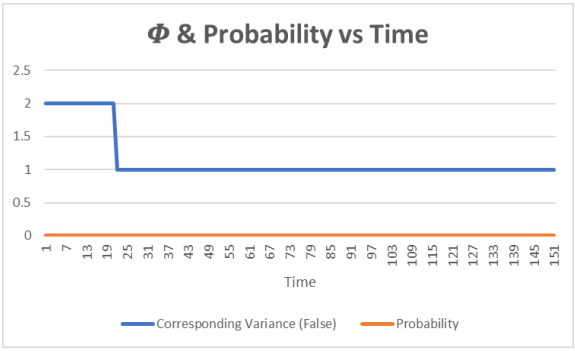

Figure 13: Corresponding Variance(False) & Probability vs Time

natural language chairs are updated to true at time frame 21, the corresponding variance of event jump and the natural language jump is always false since the robot cannot perform the action jump. As a result, the probability of G^3 is always 0 and robot knows that it needs to inform the user that this is not an executable task.

6 Discussion & Limitation

6.1 Ambiguity of event words

One limitation of my project is the ambiguity of the event words. Lots of event words involves more than one basic actions in real life. For example, when people are saying "Put those books on the chair", they want the robot to pick up those books first, and then drop those books on the chair. The event word "Put" actually means two separate event words - "Pick Up" and "Drop down". In simulations, this is not hard to achieve since I can simply define actions that robot can do.

6.2 Object Categories

Another limitation of my project is that this model can detect limited object categories since limited training is given and limited number of objects are labeled during training process. In real life,

tons of different objects exist and it's not possible for robot to learn features of all objects through convolutional neural networks. Even if there are enough data that can be used for training process, the training process may take more than years.

7 Conclusion & Future Work

In this project, I successfully replicated the simplified G^3 that defined a probabilistic graphic model dynamically according to linguistic parse structure of natural language command. Furthermore, I implemented object detection feature into the model so that robots can accurately predict the objects in its surroundings. At last, the model can calculate the probability of the input natural language command based on the updated G^3 and find out whether the command is executable.

There are still more challenges that I need to look at in this area. For example, the tree structure that can help dissemble ambiguous action words into basic action words that robots can perform will be necessary. What's more, robots under different working environments may be trained using objects that can be seen under such an environment so that the efficiency of training and accuracy of object detection can be guaranteed at the same time.

References

[1] Thomas Kollar et al. "Generalized Grounding Graphs: A Probabilistic Framework for Understanding Grounded Commands". In: *arXiv:1712.01097 [cs]* (Nov. 29, 2017). arXiv: 1712.01097. URL: http://arxiv.org/abs/1712.01097.

[2] *Record Number of Assembly Line Robots Ordered in 2021*. The Great Courses Daily. Nov. 19, 2021. URL: https://www.thegreatcoursesdaily.com/record-number-of-assembly-line-robots-ordered-in-2021/.

[3] *Rise of the Machines: The Future has Lots of Robots, Few Jobs for Humans — WIRED*. URL: https://www.wired.com/brandlab/2015/04/rise-machines-future-lots-robots-jobs-humans/.

[4] *Understanding natural language commands for robotic navigation and*. StuDocu. URL: https://www.studocu.com/en-us/document/university-of-pennsylvania/integrated-intelligence-for-robotics/understanding-natural-language-commands-for-robotic-navigation-and-mobile-manipulation/726650.

[5] *YOLOv3: Real-Time Object Detection Algorithm (What's New?)* viso.ai. Feb. 25, 2021. URL: https://viso.ai/deep-learning/yolov3-overview/.