



Autonomous Vehicle Applications with Common Sense Knowledge for Smart Cities

Priya Persaud and Aparna Varde

Technical Report

NSF GS-LSAMP

Montclair State University
Department of Computer Science

June 2017

Table of Contents

ABSTRACT.....	3
INTRODUCTION	4
OVERVIEW OF APPROACH.....	6
DETAILS OF METHODOLOGY.....	7
CSK and domain KB study.....	7
Transportation KB development.....	9
Programming a robot for image detection	10
EXPERIMENTS AND DISCUSSION.....	12
RELATED WORK	14
CONCLUSIONS AND FUTURE WORK	17
ACKNOWLEDGMENTS	18
REFERENCES	19

ABSTRACT

This research investigates the specifics behind the deployment of common sense knowledge (CSK) from worldwide data repositories. In particular, it focuses on building domain specific knowledge bases (domain KBs) encompassing CSK, Wikipedia, WordNet and other sources. Early work in this research entails studying the literature on robots including their potential for employment, e.g., a law firm recently hiring a humanoid robot, companies developing cars with robotic drivers etc. The study also includes delving into current data repositories with CSK, in particular, WebChild that we find potentially helpful in the design of humanoid robots. The findings from our study have gauged the current cognitive capacity of robots and indicated areas for improvement. An important aspect of the implementation is to build and augment domain KBs based on CSK that would help robots acquire useful data to grow closer to thresholds of human cognition. The research presented here is conducted within the context of the smart mobility characteristic of smart cities. The resulting domain KB programmed into a robot used along with existing smart city data is aimed to enhance a robot's decision-making capabilities by enabling it to make data-driven decisions pursuant to CSK analogous to humans.

INTRODUCTION

Machine learning is focused on the idea of creating computers capable of achieving cognitive capacities pursuant to those of humans. To achieve this, machines must have the ability to think in a manner similar to humans. With perseverance, computers can be programmed to absorb much data about the real world. Learning is attainable for computers because they behave in accordance with the information hardwired into them. Reliance on pre-programmed knowledge is sufficient until computers are faced with novel tasks. In such scenarios, computers must learn to make their own decisions. This is difficult for them due to an important aspect: they lack what humans possess, namely, common sense, which is often intuitive [15]. Common sense allows humans to logically approach situations in which they have no prior knowledge. If computers had common sense knowledge (CSK) and could apply that to specific domains, they would have the necessary data and tools to make intuitive and logical decisions analogous to a human [19]. This would augment decision-making.

Recent research in autonomous vehicles [5, 11] has shown that even if a robot is equipped with modern tools, it has trouble identifying and appropriately reacting to the objects in their path. For example, in May 2016, a Tesla Model S crashed into a truck since the Tesla thought the truck was an overpass because of the truck's height. Humans possess common sense knowledge to easily distinguish between a truck and an overpass given the fact that the overpass is stationary while the truck can move. We hypothesize that CSK aware robots and automated vehicles are more capable of properly distinguishing between objects such as a truck and an overpass; and can thus drive better. This paper is an initial step towards bringing this common sense reasoning to robots and automated vehicles.

With the advancements of artificial intelligence, such as large-scale common sense knowledge acquisition and reasoning techniques, CSK aware machines are no longer a fantasy. Companies such as Google, Microsoft and Amazon are already developing digital assistants such as Alexa, Cortana and Echo that provide context-aware assistance. CSK has also been used in smart city applications. A framework with CSK, text mining and ontology has been developed [7] to automate the generation of implicit requirements (as opposed to explicit ones given by users) in requirements specifications for software development, especially useful for smart city tools. However, we are not aware of such work in the context of autonomous vehicles. This paper fills this gap with the design and implementation of a specific application, namely, autonomous vehicles to enhance smart cities.

The objectives of this research project are threefold. First, it is important to address the issue of CSK in the software design and production of humanoid robots and indicate areas for improvement within

their cognitive capacity. The second objective is to develop a domain-specific knowledge base (domain KB) that can be programmed into humanoid robots to enhance their decision-making skills pertinent to driving autonomous vehicles. The final objective is to perform a series of experiments using the developed domain KB along with XNOR-NET / YOLO and either Raspberry Pi or an online simulation of robots.

As a potential source of common sense knowledge specific to transportation domain, we find that WebChild [15], a large-scale common sense KB, can be used to build and enhance domain-specific knowledge bases [19]. This can be useful to simulate the manner in which an individual would perform decision-making. For example, in earlier work [6] CSK was used to guide data extraction from Twitter to gather public opinion on air quality from a health standpoint. When programmed with CSK, the computer could extract and categorize the Twitter data analogous to a human. This was mined and then used in decision-making for urban planning. Our work in this paper falls in the same general category of deploying CSK for simulating human judgment, this time with a focus on autonomous vehicles that would be potentially useful in smart cities.

OVERVIEW OF APPROACH

The first step of this research process involves a thorough investigation of existing scholarship pertaining to CSK and domain-specific knowledge bases referred to as domain KBs. This research study is particularly interested in the automation of information within the transportation sector of smart cities. Thus, it fits into the smart mobility characteristic of smart cities [20, 22]. We conduct a thorough investigation of WebChild [15, 16, 23], in the context of transportation domain in smart cities.

As a second step, it is necessary to create a transportation-specific domain KB by augmenting the CSK repository provided by WebChild and its related smart city KB [15, 19]. Once the transportation sector of WebChild is extracted, it is modified to include images and concepts of obstacles commonly found on the roadways while driving. Such images are provided with high resolution over distributed sources and are then processed using suitable algorithms [13].

In the third step, this domain KB for transportation [10] is programmed into a small robot which is allowed to traverse a miniature driving course featuring obstacles similar to those within the KB. The robot's actions are monitored to analyze the robot's decision-making capabilities. Once programmed with CSK, it is expected that the robot would be able to make better decisions analogous to a human.

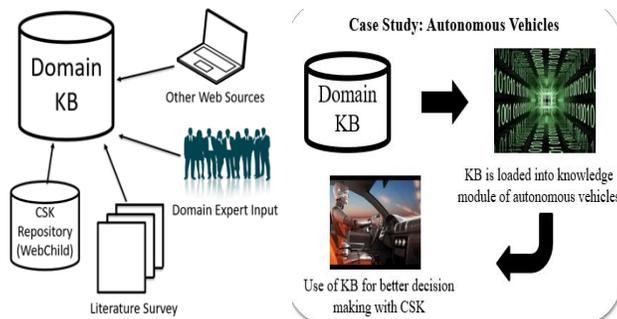


Figure 1. Creation and implementation of transportation KB.

The approach in our work is illustrated in Figure 1. We capture knowledge from the CSK repository WebChild, related domain KBs, other Web sources and domain expert input. We further build and enhance a domain KB for transportation with particular reference to autonomous vehicles. We incorporate this KB into the knowledge module of sample autonomous vehicles, in this case, a small-scale robot. We then demonstrate the use of this CSK-enhanced autonomous vehicle on a small scale with the claim that similar setups can be used in full-fledged autonomous vehicles.

DETAILS OF METHODOLOGY

CSK and Domain KB Study

We thoroughly investigate WebChild along with the details of its development. WebChild contains fine-grained common sense knowledge on several everyday concepts. It is noteworthy that WebChild has an alignment to the lexical database WordNet: this allows distinguishing the different senses of concepts, e.g., knowing the different meanings of a *green plant* (in context of the color “green” and an energy efficient power plant). As the concepts in WordNet are aligned to nearly 200 different domains, WebChild’s alignment to it enables WebChild’s slicing for different domains, such as the transportation domain. Apart from this alignment with WordNet, multimodal content and fine-grained relations make WebChild different from other common sense KBs such as ConceptNet [14] or Cyc [8].

Table 1 summarizes the WebChild statistics [15]. Overall, WebChild contains close to 18 million CSK facts comprising 3 million concepts and activities, connected by thousands of relations. It can be seen from precision values therein that this gives quite an accurate notion of CSK concepts.

Relation	#Sub-relations	#Assertions	Precision
Properties	19	4.3 M	0.82
Comparatives	6,331	1.1 M	0.90
Locations	1	0.1 M	0.86
Part-whole	3	6.7 M	0.88
Activities	7	6.1 M	0.85

Table 1. WebChild Statistics, over 3M concepts and activities

Figure 2 provides a snapshot of the WebChild browser [23]. It depicts an example of defining the term “vehicle” using WebChild data characterization. The common sense knowledge stored in WebChild has been gathered from Web content using data mining techniques, such as mining Web corpora, movie scripts and visual contents, for example, images.

The screenshot shows the 'WEBCHILD Commonsense Browser' interface. At the top, the search term 'vehicle' is entered. Below the search bar, there is a section titled 'Guess the concept' with a small image of a blue car and the text 'a conveyance that transports people or objects'. To the left of the main content area is a sidebar with several expandable categories: Domain, Comparable, Physical Part, Activity, Property, and Location, along with an 'Ask me!' button. The main content area displays a structured list of semantic relationships for the term 'vehicle':

- TYPE OF**: conveyance
- Related to **artifact**, under the category of **vehicles**
- COMPARABLES**: vehicle.car, vehicle.gasoline, vehicle.automobile, vehicle.elephant, vehicle.petrol, More
- ACTIVITIES**: become vehicle, drive vehicle, operate vehicle, provide vehicle, use vehicle
- HAS PHYSICAL PARTS**: splashboard, library, location, system, bridge, More
- IS PHYSICAL PART OF**: museum, street
- HAS SUBSTANCE**: air, fuel, water, oxygen, hydrogen, More
- IN SPATIAL PROXIMITY WITH**: ground, driver, site, road, person, More
- PHYSICAL PROPERTIES**: strong, small, light, fast, slow, More
- ABSTRACT PROPERTIES**: old, limited, efficient, safe, young, More

Figure 2. Vehicle Definition by WebChild data characterization

In addition to WebChild, we study related domain KBs. In particular, we refer to a smart city KB developed mainly with the goal of catering to urban policy [19]. It contains knowledge in domains such as *environment, transport, buildings, vehicles and town planning*. These domains are also relevant from the perspective of autonomous vehicles. With reference to this KB, a partial snapshot of an entry in this KB relevant to “transport” appears in Figure 3 describing a simple common sense concept “bridge” [19]. This shows properties along with their values, in common sense terminology, e.g., its “location” is “over road” etc.

Domain : transport
Concept : bridge

CSK

LOCATION	▼	over road / waterway / viaduct	▼
LENGTH	▼	long	▼
SIZE	▼	wide	▼
WEIGHT	▼	heavy	▼
SHAPE	▼	horizontal rectangle with vertical	▼
COLOR	▼	typically grey, could be green	▼
ACTIVITY	▼	allows land vehicles to pass	▼
PART OF	▼	city, town, wilderness	▼
MOTION	▼	stationary except drawbridge	▼
COMPARABLE	▼	tunnel	▼

Figure 3. Simple example of CSK concept in domain KB

As we can see in the “motion” property here, a bridge is typically “stationary”. Incorporating such knowledge in an autonomous vehicle would help it to distinguish between a bridge and a huge moving truck, thus helping to avoid accidents. However, notice that the “motion” property also includes an exception, namely, a drawbridge. In reality, we know that a drawbridge moves very slowly compared to a truck on the road. However, this needs to be explicitly entered in a KB to be used by for autonomous vehicles. Given these and other issues, we create a transportation KB to suit the general purpose and particularly, autonomous vehicle applications in a smart city context. This is described next.

Transportation KB Development

The development of a transportation knowledge base follows a three-stage approach. We start by slicing WebChild for the transportation domain (stage 1: filtering); we then use assertions relevant to the context of autonomous vehicles for our task (stage 2: ranking); and finally, we augment and curate this knowledge (stage 3: augmentation). These stages are elaborated as follows.

Stage 1: Filtering - At the filtering stage, we manually identify 17 domains (approximately 10% of the 171 domains present in WebChild) that are relevant to *transportation*, namely: *animals, law, person, geography, buildings, color, town_planning, transport, number, furniture, home, environment, school, architecture, vehicles* and *railway*. We discard the obvious irrelevant domains such as biology, sports, etc.

Stage 2: Ranking - After the filtering stage, we rank the content in WebChild for the concepts in these 17 domains, such that the resulting knowledge is relevant for our task of autonomous vehicles and vision. It has been shown that properties, part-whole common sense, spatial common sense, activities, and multimodal content can be useful to help vision processing techniques (e.g., to tell apart an overpass from a truck) and their common properties (such as: overpass is static, while a truck typically moves on roads).

Stage 3: Augmentation - In the augmentation stage, we supply knowledge and curate the ranked knowledge to remove the noise and to increase the coverage. We follow the overall methodology described in [19], in a similar setup as shown in Figure 3.

Programming a Robot for Image Detection

There is little work on comparative semantics for making decisions at traversal time. This is a combination of better object detection and draw signals for reasoning (as will be elaborated in our related work section). Based on the current literature and our research, we have conducted implementation as described next.

The implementation of this research project has been divided into two parts. In the first part, Scribbler, a simple, over the shelf robotic platform has been employed. The Scribbler robot [24] is a popular platform in introductory robotics courses often paired with Fluke 2, an interface/controller card developed by a third party group [1]. The combination results a low cost platform equipped with a variety of sensors (including light infrared and imaging), wireless connectivity (Bluetooth), and multi-core computing capability [2]. This particular robot is easy to use, especially through the Calico Myro programming environment [2]. Apart from classroom use, the robot has also been employed in the development of road sign recognition systems [3] and enhanced with the LIDAR capability in order to improve autonomous navigation [9].

Prior to introducing the robot to the image collection task, we have connected the robot to the desktop Calico application via Bluetooth. After initialization, we have used simple Python commands to manoeuver the robot towards images of items that are commonly associated with driving and the roadways, such as a stop sign, an overpass and a construction site. As the robot has approached the images, it has been instructed to conduct activities such as: “stop moving”, “take a picture”, “display the picture on the laptop screen”, and “save the picture in the Calico folder”. Figure 4 is a partial snapshot of relevant Python commands used in Calico for the task of robotic programming.

```

turnLeft (1,10)
pic=takePicture ()
show (pic)
savePicture (pic, "construction.jpg")
forward (5)
turnRight (2,10)
pic=takePicture ()
show (pic)
savePicture (pic, "stopsign.jpg")

```

Figure 4. Python commands in Calico for Robotic Programming

The second part of our design and implementation will include XNOR-NET / YOLO for faster image processing as shown in Figure 5. The images taken by the Scribbler robot will be loaded into XNOR-NET / YOLO such that the system can learn the images and prepare for image detection tasks to come. Once the system has learned the images, the results of the image processing will be programmed into the Scribbler robot using the Calico programming environment. The Scribbler robot will then be introduced to various obstacles commonly found on a roadway and its decision-making skills will be recorded. In order to achieve this in the long run, we will first test the detection accuracy of software such as YOLO and make recommendations for augmenting it with CSK. The same experiments will also be conducted on the Scribbler robot without the use of CSK. These results will help determine the degree to which the CSK enhances decision making.

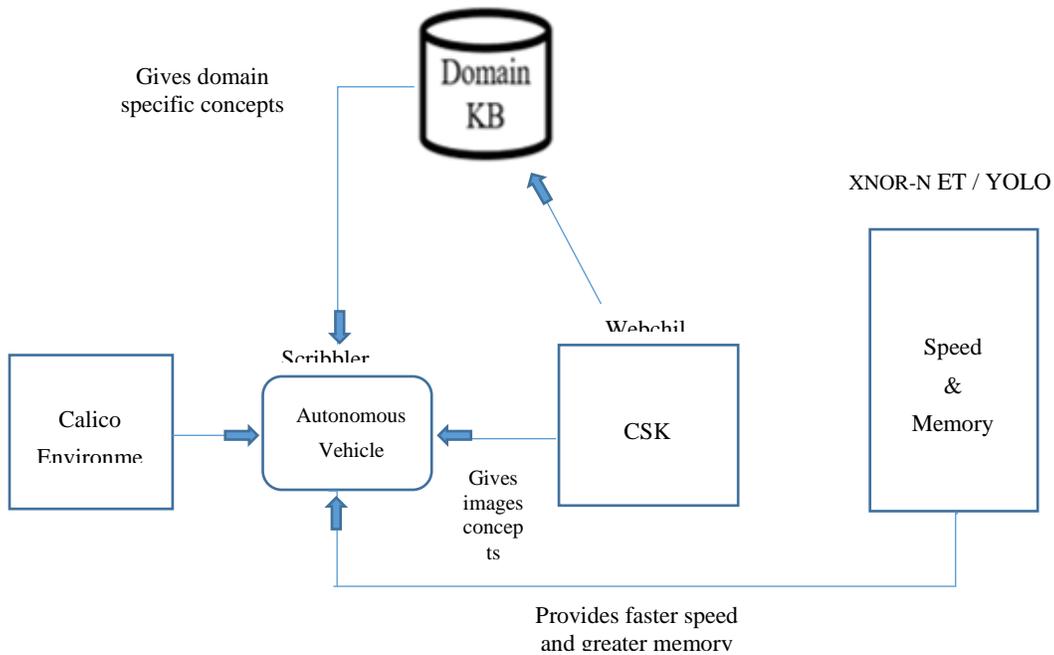


Figure 5. Test-bed for fast image processing with CSK

EXPERIMENTS AND DISCUSSION

Implementation of this research project has been conducted using a working Scribbler robot [24]. The first part of the implementation process has solely involved image collection. This part of the implementation has been successful as we have currently produced a program with the capabilities of performing image analysis using CSK. Figure 6 shows a sample visualization of our robot driving on a small scale. This illustrates the navigation of the robot on a floor and depicts how it deals with obstacles.



Figure 6. Visualization of small scale robot driving

Due to the rather minute processing capabilities of the Scribbler, we have been able to progress with these tasks to some extent as a good initial experimental step. However, our experiments so far have revealed some computational challenges in simultaneously gathering and processing images in conjunction with reasoning based on the KB.

An obvious solution is to incorporate more powerful robots with higher computation power. There is a high computation complexity of deep neural model based architectures typically employed in vision tasks. Therefore, the costs involved in acquiring a powerful robot that can cope with these challenges can be very high. While acquiring powerful robots is on our roadmap for larger scale experiments (subject to budgetary issues), we are also considering adopting newer technologies for image processing that are faster and space efficient. Faster detectors such as the YOLO object detector [26] offer 100x speedup at nearly comparable accuracies to the state-of-the-art Fast-RCNN models. Space efficient solutions such as the object detector by XNOR.ai [25] allows sophisticated neural models for object detection to run in real-

time, even on a smart phone. By performing approximations in CNN calculations, they nearly match the accuracy of powerful GPUs, while using an order of magnitude less computation power.

Due to fiscal limitations, we are unable to purchase a more advanced robot. As such, we have decided to implement the experiments using online simulations with an image classification tool, YOLO-v2. We have downloaded all of the YOLO-v2 files from and have executed some image classification commands to ensure the proper functioning of the software.

```

pritya@pritya-VirtualBox:~/darknet
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
pritya@pritya-VirtualBox:~/darknet$ mkdir -p imgs && tar xf 15VRC2812_img_val.tar -C imgs
tar: 15VRC2812_img_val.tar: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
pritya@pritya-VirtualBox:~/darknet$ git clone https://github.com/pjreddie/darknet.git
Fatal: destination path 'darknet' already exists and is not an empty directory.
pritya@pritya-VirtualBox:~/darknet$ cd darknet
bash: cd: darknet: Not a directory
pritya@pritya-VirtualBox:~/darknet$ make
make: Nothing to be done for 'all'.
pritya@pritya-VirtualBox:~/darknet$ curl -O http://pjreddie.com/media/files/extraction.weights
% Total % Received % Xferd Average Speed Time Time Current
100 184 100 184 0 0 649 0 --:--:-- --:--:-- --:--:-- 658
pritya@pritya-VirtualBox:~/darknet$ ./darknet classifier predict cfg/imagenetik.data cfg/extraction.cfg extraction.weights data/dog.jpg
layer filters size input output
0 conv 64 7 x 7 / 2 224 x 224 x 3 -> 112 x 112 x 64
1 max 2 x 2 / 2 112 x 112 x 64 -> 56 x 56 x 64
2 conv 192 3 x 3 / 1 56 x 56 x 64 -> 56 x 56 x 192
3 max 2 x 2 / 2 56 x 56 x 192 -> 28 x 28 x 192
4 conv 128 1 x 1 / 1 28 x 28 x 192 -> 28 x 28 x 128
5 conv 256 3 x 3 / 1 28 x 28 x 128 -> 28 x 28 x 256
6 conv 256 1 x 1 / 1 28 x 28 x 256 -> 28 x 28 x 256
7 conv 512 3 x 3 / 1 28 x 28 x 256 -> 28 x 28 x 512
8 max 2 x 2 / 2 28 x 28 x 512 -> 14 x 14 x 512
9 conv 256 1 x 1 / 1 14 x 14 x 512 -> 14 x 14 x 256
10 conv 512 3 x 3 / 1 14 x 14 x 256 -> 14 x 14 x 512
11 conv 256 1 x 1 / 1 14 x 14 x 512 -> 14 x 14 x 256
12 conv 512 3 x 3 / 1 14 x 14 x 256 -> 14 x 14 x 512
13 conv 256 1 x 1 / 1 14 x 14 x 512 -> 14 x 14 x 256
14 conv 512 3 x 3 / 1 14 x 14 x 256 -> 14 x 14 x 512
15 conv 256 1 x 1 / 1 14 x 14 x 512 -> 14 x 14 x 256
16 conv 512 3 x 3 / 1 14 x 14 x 256 -> 14 x 14 x 512
17 conv 512 1 x 1 / 1 14 x 14 x 512 -> 14 x 14 x 512
18 conv 1024 3 x 3 / 1 14 x 14 x 512 -> 14 x 14 x 1024
19 max 2 x 2 / 2 14 x 14 x 1024 -> 7 x 7 x 1024
20 conv 512 1 x 1 / 1 7 x 7 x 1024 -> 7 x 7 x 512
21 conv 1024 3 x 3 / 1 7 x 7 x 512 -> 7 x 7 x 1024
22 conv 512 1 x 1 / 1 7 x 7 x 1024 -> 7 x 7 x 512
23 conv 1024 3 x 3 / 1 7 x 7 x 512 -> 7 x 7 x 1024
24 conv 1000 1 x 1 / 1 7 x 7 x 1024 -> 7 x 7 x 1000
25 avg 1000 -> 1000
26 softmax 1000
27 cost 1000
Loading weights from extraction.weights... Done!
data/dog.jpg: Predicted in 4.857865 seconds.
-nanx: kit fox
-nanx: English setter
-nanx: Siberian husky
-nanx: Australian terrier
-nanx: English springer
pritya@pritya-VirtualBox:~/darknet$

```

Figure 7. Implementation of image classification using a pre-trained model

In our experiments, once YOLO-v2 has been properly installed, we have trained the software to perform image classification on the specific images we provided. To do so, we have provided a dataset of images from which the software can learn. The required dataset, called ImageNet, requires a subscription prior to download. After waiting one week, we have been granted access to the dataset, seen in Figure 7. Further experimentation continues in order to assess the image detection capabilities of YOLO-v2. We aim to find anomalies in object identification within the images, that present CSK-hard challenges. This sets the stage for further work to enhance the image detection based on CSK-enabled systems. Common sense knowledge would serve to augment the image detection and hence increase its accuracy. This would be useful when programmed into applications such as autonomous vehicles for smart cities.

RELATED WORK

We overview related work in the context of CSK for three major applications: smart cities, object detection and autonomous vehicles. In particular, we review improving vision decisions in robotic or autonomous vehicles with relevance to our work.

CSK for better smart city applications: A framework called COTIR comprising common sense knowledge, text mining and ontology has been developed [7] to automate the generation of implicit requirements (as opposed to explicit ones given by users) in the requirements specifications phase of software development. The authors claim that this would be especially useful in artificial intelligence applications, e.g., it can be used in the development of smart city tools. This is an interesting deployment of CSK with a vision for enhancing smart cities. Our work in this paper goes a step further in actually using CSK for the design and implementation of a specific application, namely, autonomous vehicles to enhance smart cities.

Concepts pertaining to smart cities and related terms are discussed in [4]. The authors consider twelve categories here, e.g., “green city”, “sustainable city”, eco-city” and “information city” often used interchangeably by urban policy makers, developers, planners etc. They address the issue of the categories having distinct conceptual perspectives with a focus on their application in practice. Their research design entails a Scopus database of relevant articles with criteria on recognized terms in academic literature. They count co-occurring keywords among these twelve categories in the literature, present the evolution of categories over time and build a network structure of author keywords associated with the categories. Their findings provide an insight into conceptual relationships at work among the twelve categories and hence their use in urban development. We find that common sense knowledge is implicitly used here and therefore such work could potentially benefit from CSK repositories and related domain KBs with respect to efficiency, accuracy and the actual deployment of appropriate concepts in urbanization.

Smart cities are attracting attention all over the world today, e.g., Melbourne in Australia is ranked as one of the finest “knowledge cities” [4, 20]. In particular, a huge emphasis on smart city development is found in Europe which by far has the maximum number of smart cities today as described in the literature, e.g., [20]. For instance, in Barcelona, buses operate on routes designed to minimize the number of signals and maximize energy efficiency. In Amsterdam, street and canal lights are dimmed and brightened based on pedestrian usage, thus making them operate in an intelligent manner [20]. Applications such as these surely embody common sense concepts. We envisage that programming and maintaining such systems would be further facilitated by common sense knowledge bases (e.g., WebChild) and pertinent domain KBs beyond the databases already being used in existing smart city tools.

Thus, the CSK repositories can have prospective applications in enhancing the current systems in smart cities.

CSK for better object detection: Object detection has dramatically improved over the last few years, especially with the advent of recent neural models. Faster R-CNNs [12] is an object detection system that works very well on data sets such as MS COCO with 100 categories such as *human, chair*, etc. This introduces a Region Proposal Network with full image convolution features and uses neural networks with attention mechanisms. It is found to achieve state-of-the-art object detection accuracy.

To further improve object detection performance, various types of contextual information has been employed to improve object detection. This contextual information can be in the form of global scene context, geometric context in the form of 3D surface orientations, relative location, 3D layout or geographic information [17]. Of these spatial context, global scene context, and part-whole knowledge in particular is inspired by CSK. Torralba et al. [18] define relevant co-occurrence of objects, and penalize the presence of objects in irrelevant scenes. However, this work assumes object detection is accurate, and only handles missing detected objects, e.g., updating the object detection of two or zero chairs instead of actual three or one chair in an image. We do not make the assumption that object detection is highly accurate. Unlike our work, they do not rely on context from background knowledge, or from text.

While these methods typically use common sense knowledge jointly with object detection, other methods inject CSK into object detection as post processing. Consider an example of a tennis player with a racket, and object detection system confuses the *ball* with a *lemon*. CSK has been used to understand that a *lemon* is not suitable in the context of a tennis scene. Researchers [17] have developed a model that distinguishes semantic relatedness from visual relatedness. Such CSK is relevant to our scenarios. Yet, these models would neither penalize the presence of an *overpass*, nor a *truck*, in the context of a *car*. They do not use the detailed context relevant CSK that an *overpass* cannot move, or that there is a risk of hitting a truck if it is misclassified as an overpass.

CSK for better autonomous vehicles: There has been a tremendous amount of work in path planning in the context of AI, that intensively makes use of common sense. However, these systems have typically operated over toy sized KBs, and their primary focus is on reasoning.

A system called RoboBrain [21] has been developed at Cornell that learns concepts from the Web, robot trials and computational simulations. It interprets text, pictures and videos and stores this knowledge in a comprehensive manner. This serves to provide a fully integrated KB. The WebChild knowledge base with a broader spectrum of CSK relations can be considered as a larger alternative to RoboBrain.

Our work in this paper can benefit from the research advances described herein. However, to the best of our knowledge, we would still need decently sized domain KBs to serve as a type of cache and retrieve useful data in specific applications, as opposed to searching vast repositories created by self-learning robots or using a generic object detection system. This would provide an efficient process to get only relevant data for decision-making. In the future, we could consider expanding our work by incorporating knowledge from such systems, and by running benchmark studies with our CSK enabled automated vehicles using more advanced robots.

CONCLUSIONS AND FUTURE WORK

This paper addresses the issue of autonomous vehicles augmented with common sense knowledge to enable better decision-making analogous to a human. It is potentially useful in smart cities, especially with reference to their smart mobility characteristic.

The approach in this paper entails deploying a worldwide repository called WebChild with a multitude of common sense concepts; building and harnessing related domain specific knowledge bases; and programming a small scale robot. Our design, implementation and preliminary experiments reveal that common sense knowledge is useful in this application and that faster image processing and greater memory are needed for a full illustration of CSK enhanced decision-making. This is part of our current work.

Once enhanced by programming with CSK, it is envisaged that humanoid robots would advance the field of Computer Science and encourage greater participation of scholars and scientists in new areas of research. This research also furthers the study of data science as it is heavily reliant upon the development, maintenance and adequate deployment of domain-specific knowledge bases.

As time goes on, the knowledge bases will undoubtedly require updates. Such updates can be achieved with automated data mining techniques, such as those used in earlier WebChild work. Not only does this research present smart cities with a new method of data automation, but it also allows for great strides in artificial intelligence research to be conducted in conjunction with data management. This paper that outlines a vision with implementation and initial experiments, would thus be useful to data scientists and AI scientists as well as developers of related applications, especially in smart cities.

ACKNOWLEDGMENTS

This research is funded by the National Science Foundation (NSF) that supports Priya Persaud through the program LSAMP (Louis Stokes Alliance for Minority Participation) since August 2016, led by Dr. Aihua Li of the Mathematical Sciences Department at Montclair State University. We would like to thank Dr. Stefan Robila of the Montclair State University Computer Science Department for his contributions in working with the Scribbler robot. We sincerely acknowledge Niket Tandon from the Allen Institute for Artificial Intelligence (AI2), Seattle, Washington, for his contributions on reuse of the WebChild code, valuable interactions during our research and useful inputs on this paper. The early work on this project started when Aparna Varde was a visiting researcher at the Max Planck Institute for Informatics (MPII), Saarbruecken, Germany in August 2015 in the Databases & Information Systems Group of Gerhard Weikum.

REFERENCES

- [1] Balch T. et al., “Designing personal robots for education: Hardware, software, and curriculum,” *IEEE Pervasive Computing.*, 2008, 7(2): 5-9. (very long author list)
- [2] Blank D., Kay J. S., Marshall J.B., O’Hara K. and Russo M., “Calico: A multi-programming-language, multi-context framework designed for computer science education”, *ACM SIGCSE*, Raleigh, NC, 2012, pp. 63–68.
- [3] Bilgin E. and Robila S.A., “Road sign recognition system on Raspberry Pi”, *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, 2016, pp. 1-5.
- [4] De Jong M., Joss S., Schraven D., Zhan C. and Weijnen M., “Sustainable-Smart-Resilient-Low Carbon Eco-Knowledge Cities; Making sense of a multitude of concepts promoting sustainable urbanization”, *Journal of Cleaner Production*, Elsevier, Dec 2015, 109:25-38.
- [5] Driggs-Campbell K., Shia V. and Bajcsy R., “Decisions for Autonomous Vehicles: Integrating Sensors, Communication, and Control”, *ACM HiCoNS*, Berlin, Germany, Apr 2014, pp. 59-60.
- [6] Du X., Emebo O., Varde A., Tandon N., Nag Chowdhury, S. and Weikum G., “Air Quality Assessment from Social Media and Structured Data”, *IEEE ICDM Workshops*, Helsinki, Finland, May 2016, pp. 54-59.
- [7] Emebo O., Varde A. and Olawande D., “Common Sense Knowledge, Text Mining and Ontology for Implicit Requirements”, *WorldComp DMIN, IEEE*, Las Vegas, NV, Jul 2016, pp. 146 – 152.
- [8] Lenat D.B., “CYC: A Large-Scale Investment in Knowledge Infrastructure”, *Communications of ACM*, 1995, 38: 32-38.
- [9] Miller K.S. and Robila S.A., “Road sign recognition system on Raspberry Pi”, *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, 2017, 6 pgs., in press.
- [10] Persaud P., “Domain Specific Knowledge Bases with Common Sense Knowledge for Autonomous Vehicles in Smart Cities”, Presented at *NSF GS-LSAMP Undergrad Research Conference*, New Brunswick, NJ, Oct 2016.
- [11] Persaud P., Varde A. and Drake I., “Common Sense Knowledge, Humanoid Robots and Human Rights”, Presented at *IEEE MIT Undergrad Research Technology Conf.* Cambridge, MA, Nov 2016.
- [12] Ren S., He K., Girshick R. and Sun J., “Faster R-CNN: Towards Real Time Object Detection with Region Proposal Networks”, *ARXIV e-prints: 1506.01497*, Jan 2015.
- [13] Robila S.A. and Ricart S., “Distributed algorithms for unmixing hyperspectral data using non-negative matrix factorization with sparsity constraints”, *IEEE IGARSS*, 2013, pp. 2156-2159.
- [14] Speer R. and Havasi C., “Representing General Relational Knowledge in ConceptNet 5”, *LREC*, 2012.
- [15] Tandon N., de Melo G., Suchanek F. and Weikum G., “WebChild: Harvesting and Organizing Commonsense Knowledge from the Web”, *ACM WSDM*, New York, NY, Feb 2014, pp. 523-532.
- [16] Tandon N., de Melo G. and Weikum, G., “Acquiring Comparative Commonsense Knowledge from the Web”, *AAAI*, Quebec City, Canada, Jul 2014, pp. 166-172.
- [17] Torralba, A., Murphy, K., Freeman, W.T., “Using the Forest to See the Trees: Object Recognition in Context”, *Communications of the ACM*, Mar 2010, 53(3):107-114.
- [18] Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A., “Context-Based Vision System for Place and Object Recognition”, *ICCV*, Oct 2003, 3:273-280.
- [19] Varde A., Tandon N., Nag Chowdhury S. and Weikum, G., *Common Sense Knowledge in Domain-Specific Knowledge Bases*, Technical Report, Max Planck Institute for Informatics (MPII), Saarbruecken, Germany, Aug 2015.
- [20] Vienna University of Technology (TU Wien), *European Smart Cities*, Technical Report, Vienna, Austria, 2015.
- [21] RoboBrain <http://robobrain.me/#/>
- [22] Smart Cities <http://smartcitiescouncil.com/>
- [23] WebChild <https://gate.d5.mpi-inf.mpg.de/webchild/>
- [24] Working Scribbler Robot Implementation Manual http://wiki.roboteducation.org/Learning_Computing_With_Robots
- [25] XNOR.ai object detection: <https://xnor.ai/index.html>
- [26] YOLO Real time object detection, <https://pjreddie.com/darknet/yolo/>