1-2023

# Object Detection and Image Categorization by Transferring Commonsense Knowledge with Premises and Quantifiers

Irina Chernyavsky

# ABSTRACT

Domestic, or household robots, are autonomous robots designed to make our home-life easier by performing chores and mundane tasks such as cleaning, or cooking. Currently domestic robots are specialized to complete a specific task and, therefore, are confined by factors such as mobility, size, and complexity. With the fast development of computer vision and robotics, the need for more compact, advanced and multi-task robots has emerged. Therefore, the robot needs to be multi-functional, able to discern the environment and the tasks. The aim of this paper is to categorize images in domestic robots as relevant to the culinary, laundry, vacuum class or non-relevant at all. The traditional approach in computer vision involves manual annotation of the large number of images and training the model. The most widespread model training techniques comprise of methods such as convolutional neural networks, regression and support vector machine algorithms. We propose an approach that takes a different route by incorporating commonsense knowledge into the algorithm. Our approach, a Commonsense Knowledge-Detector, or a CSK-Detector, performs the basic object detection on a few household objects via Mask R-CNN, and then utilizes commonsense knowledge clauses obtained from a state-of-the-art Knowledge Base "Dice" for large scale image categorization. The live web-camera object detection is also implemented into the model, allowing the CSK-Detector to classify room environment in real time. In addition, our model is a white-box algorithm that returns explainable results in the form of a decision tree. Moreover, it reaches accuracy scores higher than 90% on the whole, which is similar to the black-box core deep learning models in literature. The CSK-Detector Model refinement and expansion to other, non-domestic domains can potentially aid human-robot collaboration and next-generation robotics.

**Keywords:** Artificial Intelligence, Big Data, Robotics, Commonsense Knowledge, Categorization, Mathematical Modeling, Robotics

MONTCLAIR STATE UNIVERSITY

# Object Detection and Image Categorization by Transferring Commonsense Knowledge with Premises and Quantifiers

by

Irina Chernyavsky

A Master's Thesis Submitted to the Faculty of

Montclair State University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

December 2022

College of Science and Mathematics

Department of Mathematics

Thesis Committee:

[redacted]

______________________

Dr. Aparna Varde
Thesis Sponsor
(Advisor)

[redacted]

Dr. Weitian Wang
Committee Member

[redacted] ___

Dr. Deepak Bal
Committee Member

[redacted]_

Dr. Ashwin Vaidya
Chair of Mathematics Department

# OBJECT DETECTION AND IMAGE CATEGORIZATION BY TRANSFERRING COMMOSENSE KNOWLEDGE WITH PREMISES AND QUANTIFIERS

A THESIS

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science

by

Irina Chernyavsky

Montclair State University

Montclair, NJ

2022

## ACKNOWLEDGMENTS

I would like to thank my thesis advisor Dr. Aparna Varde for her support and guidance.

I would like to thank Dr. Deepak Bal and Dr. Weitian Wang for serving as my committee member.

I would like to thank Dr. Simon Razniewski from Max Planck Institute for Informatics, Germany, for serving as a co-author for the paper.

I would like to thank the Mathematics Department for enrolling me as a graduate student.

I would like to thank the Mathematics Department Chair Dr. Ashwin Vaidya and the CS Department Chair Dr. Constantine Coutras.

The thesis advisor acknowledges NSF grants 2117308 (MRI: Acquisition of a Multimodal Collaborative Robot System (MCROS) to Support Cross-Disciplinary Human-Centered Research and Education at Montclair State University) & 2018575 (MRI: Acquisition of a High-Performance GPU Cluster for Research and Education), both related to this MS thesis.

# DEDICATION

I would like to thank my family for their support.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

| | |
|---|---|
| CSK | Commonsense Knowledge |
| KB | Knowledge Base |
| MCS | Machine Common Sense |
| BOF | Bag of Features |
| MIL | Multiple Instance Learning |
| SVM | Support Vector Machine |
| DT | Decision Tree |
| CART | Classification and Regression Tree |
| GI | Gini Index |
| CNN | Convolutional Neural Networks |
| R-CNN | Region-Based Convolutional Neural Networks |

# LIST OF SYMBOLS

| | |
|---|---|
| PI | Plausible |
| RI | Related |
| Ty | Typical |
| Sa | Salient |
| s, s1, s2 | Semantic entities |
| Rm | Room |
| A score | Aggregate score |
| P | Commonsense premises |
| Q | Quantifier |
| L | Limit |
| C | Class |

# 1 Introduction

## *1.1 Background and Motivation*

Advances in artificial intelligence enable developing computer vision models to classify images and videos. Although, this aids robotics, misclassification on real-world images is yet a problem. As humans, we possess subtle knowledge of concepts, properties and relationships, i.e. *commonsense knowledge (CSK).* It helps us intuitively categorize images, even at first sight. Robots do not have natural CSK, and thus, classify images based on prior training. Many algorithms thrive on neural networks and deep learning, i.e. a black-box without CSK, possibly erring in first-time tasks, requiring huge training data, and lacking explainability [1]. This motivates object detection and image categorization infused with CSK to address the following:

- Relevance of the detected object to a category
- Explainability of the categorization algorithm
- Easy adaption to other autonomous systems
- Automated data-set preparation for image learning

The solution should not require annotating an overwhelmingly large number of full images to create prior labeled training data. Deep learning models require such huge labeled data-sets; the more the data is, the better the learning. If image annotations have errors, the learning can be flawed. Furthermore, they use general domains, which are hard to apply to specific tasks [2].

## *1.2. Contributions of This Study*

We propose an approach called *CSK-Detector:* ***C****ommon* ***S****ense* ***K****nowledge based object* **Detector**, with the following contributions:

1. It does not need prior annotation of each full image
2. It uses a visualizable, explainable classifier
3. It can generalize/specialize to other autonomous tasks.

### *1.3 Thesis Structure*

The thesis is structured as follows. First, prerequisite information about commonsense knowledge, human-robot collaboration, domestic robotics and image categorization is presented in Section 2. Afterwards, the proposed approach, the details of the methodology and the study's limitations are outlined in Section 3. From there, the experiments and their results are presented in Section 4. Lastly, conclusions and future work are presented in Section 5. The rest of this thesis is organized into these sections.

# 2 Related Work

## *2.1 Commonsense Knowledge*

Commonsense is the knowledge that we all human beings have. We all gradually acquire it through our lifetime without even being aware of it. For instance, “our parents are older than us”, or “dogs do not speak English.” This knowledge is often used by experts to solve domain-specific problems. We often acquire this knowledge through our curiosity and experience, and usually take it for granted. Artificial intelligent systems do not have commonsense knowledge and acquiring it has become an important effort. Moreover, many years of technological progress showed that building commonsense reasoning system is a work-intensive and costly task. Therefore, commonsense reasoning has become an important task and many approaches have being developed to build the knowledge-base systems.

The importance of commonsense knowledge was first noted by the farther of AI, John McCarthy. In his paper, “Programs with Common Sense,” he was the first to propose to use logic to represent information in a computer [5]. It was apparent that computers were very competent at solving specific problems, but they were useless in reasoning with unusual data. For instance, a machine learning model trained for medical diagnosis can perform well on the specific medical data, but, it won’t realize if the data is substituted with non-medical data. Humans, on the other hand, have intuitive knowledge that helps them to catch mistakes and avoid meaningless results. Moreover, human experts can reason more intuitively in a very specific field, compared to AI. A medical specialist might rule out a life-threatening surgery for the elderly with the short life expectancy, compared to the AI surgical robot which might not even realize that. Thus, it is essential that more human-level AI is developed.

Delivering human-level AI presents even more difficulties with the advancement and ubiquitous use of the deep learning systems. Those systems are keen at pattern recognition, very precise and fast

[3]. However, when the slightest change is performed, the system has to be re-trained on a huge amount of data, which is time consuming and computationally expensive [3-4]. Humans, in contrast, can adapt quickly when encounter slightest changes through reasoning and commonsense in the domain of their knowledge, or a new domain that has some familiarity. Deep reasoning models, on the other hand, are not capable of doing so. Therefore, building a knowledge-base systems for the use of AI is the key for the human-level AI development.

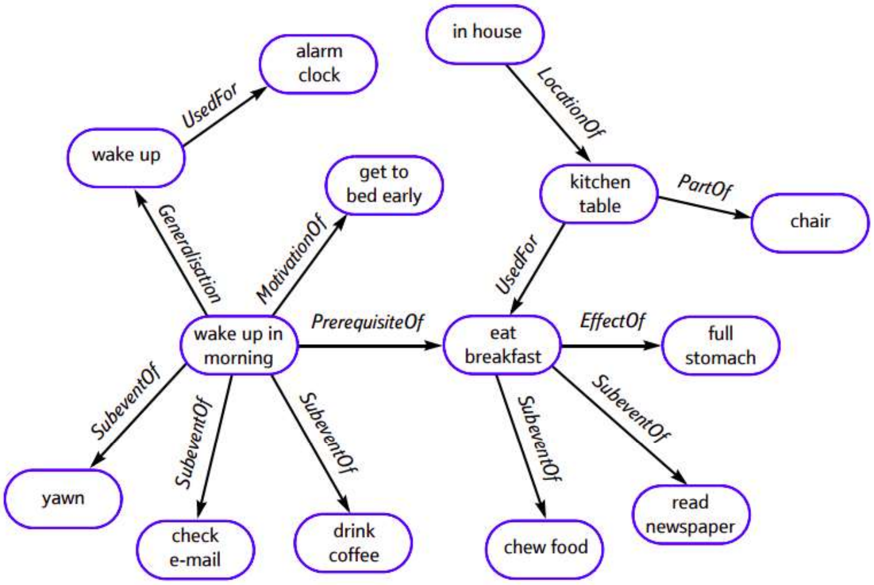
Various knowledge-base systems were developed in the past 30 years. The first system was CYC, which began its development in July 1984 by Douglas Lenat. It was a bold attempt to assemble a massive knowledge base spanning human knowledge [6]. It consists of a knowledge-base with hand-coded common sense and an inference engine that can deduce further facts. It uses a declarative language called Cycl which is based on the first-order logic and was written in Lisp. The main lexical components are constants. A constant is a set of concepts, and can be a specific object, or a relationship between objects. Each constant begins with #$. For instance, #$isa #$MontclairState #$University, means "Montclair State is a University [7]. This was an example of a fact in the CYC knowledge base system (KB). The Cycl syntax can also contain prefixes, nested parenthesis and variables that are called "rules." The inference engine converts the sentences in the first-order logic into a natural English language. Even though inference language can perform deductive logic on the facts in the database, it cannot take new data and form new relationships. For that, the Cyc project received a lot of criticism among AI researchers [8-9]. Even though it was criticized, it contains the world's broadest and deepest common sense knowledge base, by orders of magnitude, and it is easily transferable into other AI projects.

Another, less costly project, which uses an open-source model for capturing data on the web, is ConceptNet. An open-source model means that any internet user can construct the KB. The ConceptNet was launched in 1999 at the MIT Media Lab and was called an Open Mind Common

Sense. The KB uses semantic networks to represent knowledge, which are graphical methods that describe the relationship between concepts and events.

**Figure 1**

*ConceptNet*



*Note*: An excerpt from ConceptNet's semantic network of commonsense knowledge. Compound (as opposed to simple) concepts are represented in semi-structured English by composing a verb ('drink') with a noun phrase ('coffee') or a prepositional phrase ('in morning') [10].

As of today, ConceptNet contains over 1.6 million assertions of commonsense knowledge such as physical, psychological or social aspects of everyday life, and with each release the system is refined and re-tuned. Even though the ConceptNet is a huge KB, some of its characteristics make the relation classification difficult such as several distinct relation types can be true for the same concept pair, and many concepts are expressed through a multi-word arguments [11]. Despite those difficulties, the ConceptNet is successfully used in chatbots and some natural language support.

Beyond chatbots and language support, projects that learn how to read web-sites were developed. One such project is NELL: Never-Ending Language Learning that began in 2011 at Carnegie Mellon

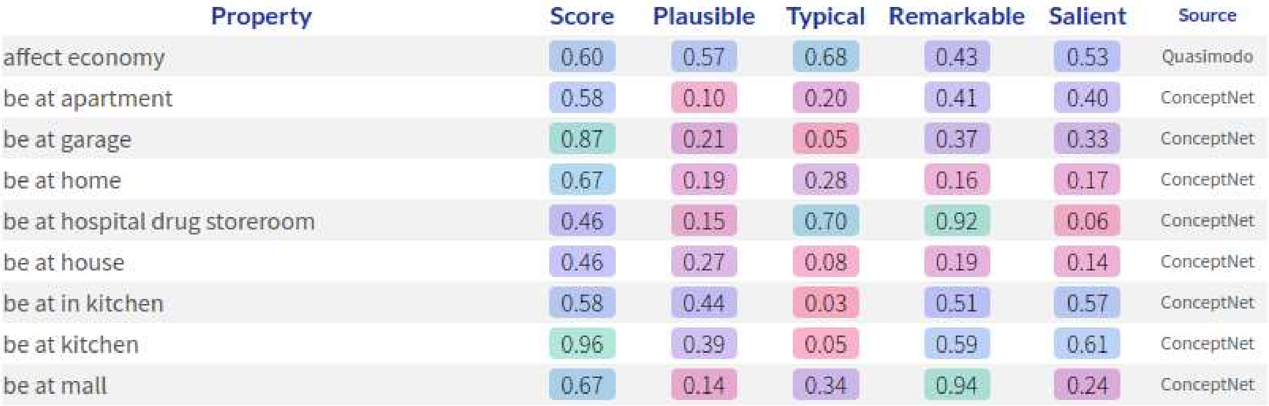
University. The main idea behind NELL is that it is programmed to search the web and identify linguistic patterns to deduce its meaning. So far, NELL has accumulated over 80 million candidate beliefs with high confidence. Additionally, it has learned how to reason over these beliefs to infer new beliefs so that its ontology can be expanded [12].

The ability to infer new beliefs is important in AI and an interesting common sense knowledge base has been developed by the US defense department's research agency, DARPA. The KB is called Machine Common Sense, MCS, and it constructs computational models that mimic the core domains of child cognition for objects, places and agents. Just as human infants must learn from experiencing the environment, same the MCS learns. Through simulated training environments, the MCS system demonstrated the improved understanding on how to grasp an object, adapt to obstacles and change speed for various goals. The system definitely moves us closer to building a robust human-level robotics and represents a revolutionary reasoning framework.

Another reasoning framework is Dice. This framework derives refined commonsense knowledge from existing commonsense collections (CSK). It mainly explores two CSK collections which are ConceptNet and Quasimodo. For each subject the user enters, the interface will show parents and siblings selected for the joint reasoning, along with their weight [13].

**Figure 2**

*Dice Interface*



| Property | Score | Plausible | Typical | Remarkable | Salient | Source |
|---|---|---|---|---|---|---|
| affect economy | 0.60 | 0.57 | 0.68 | 0.43 | 0.53 | Quasimodo |
| be at apartment | 0.58 | 0.10 | 0.20 | 0.41 | 0.40 | ConceptNet |
| be at garage | 0.87 | 0.21 | 0.05 | 0.37 | 0.33 | ConceptNet |
| be at home | 0.67 | 0.19 | 0.28 | 0.16 | 0.17 | ConceptNet |
| be at hospital drug storeroom | 0.46 | 0.15 | 0.70 | 0.92 | 0.06 | ConceptNet |
| be at house | 0.46 | 0.27 | 0.08 | 0.19 | 0.14 | ConceptNet |
| be at in kitchen | 0.58 | 0.44 | 0.03 | 0.51 | 0.57 | ConceptNet |
| be at kitchen | 0.96 | 0.39 | 0.05 | 0.59 | 0.61 | ConceptNet |
| be at mall | 0.67 | 0.14 | 0.34 | 0.94 | 0.24 | ConceptNet |

*Note:* An example of the facts returned by the Dice interface when user enters the subject "refrigerator."

As seen from figure 2, each fact has four weighted dimensions, or scores: plausibility, typicality, remarkability and salience. The plausibility score indicates whether the statement makes sense at all [13]. Typicality means whether a property holds true for most instances of the concept. Remarkability indicates whether the property distinguishes the concept from other related concepts and salience expresses whether the property is the key trait of the concept. Each weighted score represents the degree of relatedness of the concepts. Experimental results showed that Dice is able to capture CSK along the four dimensions better than any other, single-dimensional baseline [13]. The system is thus able to consolidate existing large CSK collections into one, more cleaner knowledge base.

Knowledge base construction proved to be a difficult task and many approaches and techniques were devised in the past 30 years. One of the difficulties seems to arise from the lack of precise definition of what exactly is commonsense is and what parts constitute it. Therefore, the progress in commonsense knowledge base development and application is crucial for the human-level AI advancement.

### *2.2. Domestic Robots*

A domestic robot is a type of autonomous robot that is used mainly for the housework, but can also be used for the therapy, entertainment, emotional support, security or education. Some domestic robots are very simple, others are sophisticated and are highly autonomous. By 2006, more than 3,540,000 domestic robots were in household use [14]. To be classified as an autonomous, the robot has to comply to specific criteria and contain certain components. In this section, the criteria for autonomous robots will be briefly discussed, followed by the short history of domestic robots and the recent development in the field.

Some of the components that autonomous robots should have include self-maintenance, sensing the environment, task performance and autonomous navigation [14]. As the name autonomous implies, an autonomous is the robot that can perform tasks autonomously. A fully autonomous robots are capable to learn their environment, work without human intervention and adapt new methods in completing a task. Even a fully autonomous robot requires a routine maintenance. Self-maintenance is the first requirement for the fully autonomous robots. They need to have "proprioception", or the ability to sense its own internal environment, such as low battery or overheating. Many battery operated robots that are on the market today, have self-maintenance ability, and can locate and connect to the outlet when the battery is low.

Ability to locate the outlet comes from the exteroception, or the capacity to sense things in the immediate environment. Some examples of exteroception include sound, chemical smell, touch, temperature or pressure change, electromagnetic vibrations. Vacuum robots, for instance, can sense the amount of dirt picked up which signals them to stay longer in the area if needed. Thus, they are able to complete the task, which is another criteria for autonomous robots. Most vacuum robots employ simple propriety algorithm that help them cover the floor area beyond random bouncing.

Security robots contain conditional algorithms that help them detect intruders and respond in a specific way.

Before exhibiting a specific behavior, a robot must first be able to navigate in the environment. There are two types of environment: outdoor and indoor. Since obstacles are rare in the air, the outdoor navigation for air robots such as drones is the easiest. Ground robot, such as a lawn-mowing robot, would be the most difficult to navigate due to terrain and weather instability. The indoor robot navigation is of medium difficulty. In 1970, the first robots had wire-guided indoor navigation which further progressed to beacon-based triangulation. In 2000, robots were designed to navigate through the building by using manually created CAD floor plans and sensing natural features such as walls. Sonar sensing was in use that time. Nowadays, mobile robots rely on multiple laser sensors that create laser-based floor map. Information from multiple sensors is constantly fed into the algorithm that re-maps the environment as the robot moves. Therefore, a vacuum robot is able to safely navigate through the entire floor without exhibiting danger to people and its surrounding.

Vacuum robots were not always a commodity. The first robots were designed during the Industrial Revolution, around 1760, and their purpose was to process building materials and other commercial products. The robots were not classified as domestic and people did not consider robots as household helpers. Only when the people’s living standards drastically improved after Industrial revolution, people began to consider household robots.

The first domestic robot was sold in 1980 and was called a “HERO.” It was designed and manufactured by Heathkit. The robot was educational and had a great success. The Heathkit created four models of HERO: HERO1, HERO JR, HERO 2000, and the Arm Trainer. The HERO1 model was purely educational, while HERO JR was an improved model of HERO1. HERO 2000 was the latest generation of HERO1 and had advanced programmable features, low-cost, had vibrant personality [14]. The Arm Trainer was an industrial robot.

Another domestic robot was “Topo,” released in 1983 by Androbot Inc. The robot lacked sensors, and, therefore, was unable to respond to orders correctly. The issue was resolved with the next generations of “Topo” which were equipped with sensors. With the release of HERO and Topo, domestic robots became more affordable and widespread, which fueled the development of specific type of domestic robots that perform house chores both outside and inside the house.

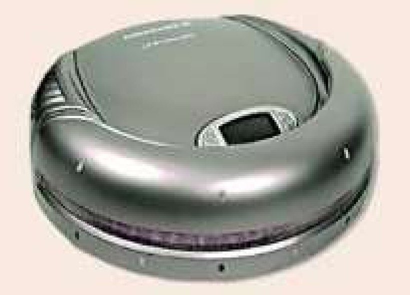
Nowadays, domestic robots that perform house chores include vacuum robots, floor-washing robots, dressman that is used to dry and iron shirts, laundroid, cat litter robot, FoldiMate that organizes and folds clothes.  Atlas is an example of a domestic robot that can do sweeping, and opening doors. Knightscope is a security robot, equipped with night-vision camera and detects movements. Lawn-mowing robots are examples of outdoor domestic robots that memorize the distance and angles traveled. This enables the repetition of the operation without being re-programmed. Some of these robots are capable to mowing complex lawns with uneven terrain. In general, all domestic robots can be roughly divided into the following categories: cleaning robots, lawn-mowing robots, smart appliances and smart homes.

Cleaning robots are designed to clean the space. The space, however, ranges in dimension and complexity. Therefore, the robots that clean are greatly varied. Pool-cleaning, window-cleaning and floor-cleaning robots are some examples. Pool-cleaning robots are the most established robots on the market. They are not even called robots. Since the shape of most pools is rectangular, and there are no obstacles, all that robot has to do is bounce between walls until the battery is dead. Floor-washing robots, on the other hand, have to learn to navigate in the 3D environment, with many rooms, and stairs without collision and in within specific time-frame. Carpet, apparently, has to be treated differently than the floor, which makes the design of a house cleaning robot a much more difficult task. The following represent technical challenges of the cleaning robots: absolute positioning, dynamic environment, unknown area coverage, error recovery, safety, power supply, multi-robot coordination, human-robot interaction [15]. Absolute positioning is the knowing its current position, which is an absolute for the robot. A robot that loses its position definitely not able to execute the task at hand. The solutions to the absolute positioning include landmark-based position estimation [15]. Some of the examples of the floor-cleaning robots include Trilobite 2.0, Robocleaner and Orazio.

Trilobite was launched by Sweden company AB Electrolux. The robot uses sophisticated sonar system for navigation. After the robot is unplugged from the charging station, it follows the walls of the workspace and returns to its starting point. Thus, it creates a map that helps the robot in navigation and performance. It is also equipped with infrared sensors that detect stairs.

**Figure 3**

*Trilobite 2.0*



Robocleaner was launched in 2003 by Karcher GmbH (Germany). It randomly moves from wall to wall and gradually covers the whole area. It is also equipped with sensors that detect air pollution to know which areas need specific cleaning. It will increase its suction level in such area. The robot does not have a sophisticated sonar system as Trilobite; it only has tactile sensors. Even with this deficiencies, the Robocleaner is able to clean the floor on its own, return to the docking station once the battery is low, dump the dust and recharge the battery. Therefore, the robot is very reliable and much appreciated by the customers. Unlike Trilobite and Robocleaner, Ozario has additional wet cleaning functions. Ozario was manufactured by Zuccetti, an Italian company. It has the following functions: continuous vacuuming, wet cloth, dry cloth and vet cloth with vacuuming. Ozario uses only tactile senses and turns at a random angle. Therefore, it cannot achieve systemic coverage.

Ozario is an example of a floor-cleaning robot. Aquabot, pproduced by Aqua Products, USA, is an example of a pool-cleaning robot. It has two alternate moving patterns. One is a zigzag motion, meaning once it hits the wall, it will reverse the direction and move towards the pool center at a certain angle. Another is a rectangular pattern, where it uses the wall as its navigation [15]. Aquabot became an established cleaning robot on the market due to its efficiency and price.

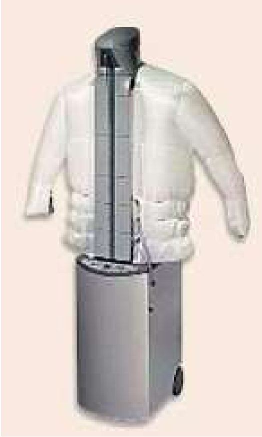
Lawn-mowing robots are another well-established robots on the market. In fact, they were released long before domestic floor-cleaning robots. This may be due to the fact that performance for the lawn-mowing robot is not as critical as for the floor-cleaning robots. Most lawn-mowing robots have castor wheels and differential drive system and are equipped with similar sensors as floor-cleaning robots. Since these robots also employ bang-and-bounce strategy, they can run away as there are no obstacles preventing that. The solution is to bury the wires in the ground that emit electromagnetic field and keep the robot enclosed. All lawn-mowing robots have a lift-off mechanism, meaning they will shut off once lifted or turned upside down. AutoMower, realesed by Sweden Company, is an example of a lawn-mower. It is charged by nickel-hybrid batteries and returns to its charging station once the batteries run low.  Given that the majority of people live in cities, lawn-mowing robots are not as popular as floor-cleaning robots or smart appliances.

When asked what smart appliances are, people think of appliances that communicate with each other through internet. In this work, smart appliances means appliances which use robotic technology, such as actors, sensors or control system. Therefore, smart appliances can be divided into ironing robots, intelligent refrigerators and smart wardrobes.

Dressman is an example of an ironing robot released by Siemens, Germany.

**Figure 4**

*Robot*



*Note:* Siemens: Ironing robot Dressman.

The robot is equipped with a heater inside, and some resistors. It stores the heat in such a way that, when the shirt is positioned and start button is pressed, the whole ironing dummy fills with hot air and presses the shirt. After that, the robot blows cold air for 1 minute to stabilize the cloth. Since the dummy is inflatable, it can adjust to any size of a shirt.

Ironing shirts, floor-cleaners, lawn-mowers obviously simplified our lives; but current technology goes significantly beyond simple robots. Smart home technologies represent the advancement in home automation. JEITA House, Japan, was the house project carried out from 1999 to 2001 [15]. Several companies participated in the project, including Japan Electronics and Information Technology Industries Association. The project was implemented in a two-story Japanese house and included features such as opening curtains, turning on and off the lights, feeding pets, watering plants, and key less system. They even had a robot dog AIBO that would greet the persons. The house was also equipped with sensors that monitored person's health, such as heart rate and blood pressure. If the health indicators were within dangerous range, the system would notify relatives. Another interesting aspect of the project was that the house would adapt to the habits of the family. Overall, the JEITA demonstrated that robotic system can be successfully integrated into homes.

Integration of robotic system into houses is no longer a dream, but a reality. Clearly the sales market for the domestic robots exploded in the past 20 years. Roomba sells their domestic robots five times more than industrial. Even with the promising market, domestic robots still remain unpopular. Apparently, the reason is in the cost of a robot. For instance, would an average household family invest $5,000 into the floor-cleaning robot that promises to systematically clean the floor, or would they rather buy a $100 bouncing from wall-to-wall robot that randomly vacuums the floor? Even if the family decides to hire a professional cleaning service biweekly, the cost is estimated to be below $6,000 per year (as for New Jersey in 2022). Therefore, the robot maintenance, and its costs must be below $6,000. However, including sensors and algorithms into the robot that would make it far more advanced than the bouncing cleaner makes the robot costly and out of reach of the average family. Therefore, the advancement of the robotics and AI should not be fueled just by enthusiasm of engineers, but also by finding a cost effective solutions.

### *2.3. Image Classification*

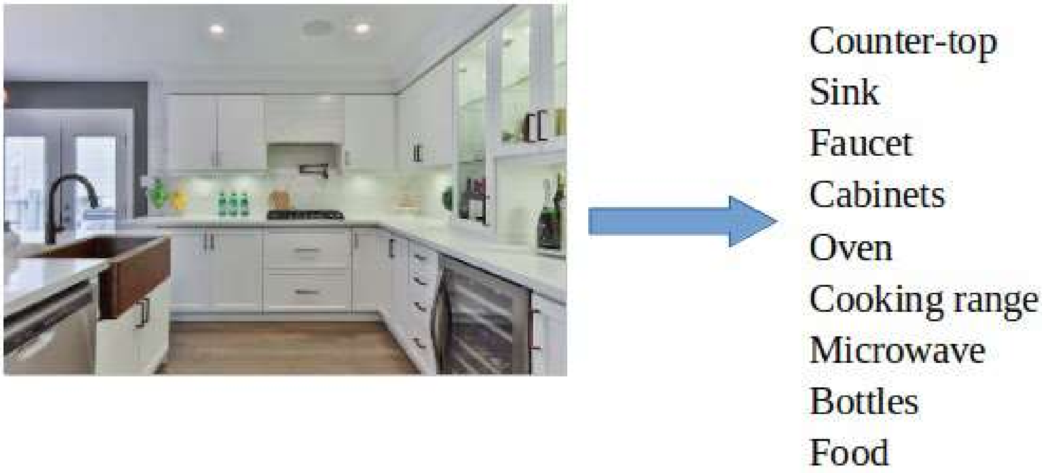
With the rapid development of the internet huge amounts of images are produced. As an example, just by looking through VRBO or NJMLS web-sites, it is easy to see the vast amount of images. These images would be useless if we cannot classify them or put them into a specific category, such as an image of a kitchen, or a bathroom, or a house.  Therefore, image classification plays an essential role in organizing digital information.

Despite the great development of computer vision, image classification is still in its early stage of development, and more research needs to be put into the field. Image classification is to organize images into various classes based on the features of the image [15]. Image annotation is to label images with different semantic classes such as table, chair, lamp, etc. Image annotation annotates an image with multiple labels while image classification classifies an image into multiple classes. The two concepts are closely related since if the image annotation is done correctly, then image is classified into a proper class. Image annotation can be done manually with the available online tools such as VGG annotator or labelme. It can also be done with Multiple Instance Learning (MIL). In MIL, an image is represented as a Bag of Features (BOF), and image is labeled positive if any of the features in the bag are positive [15]. In the end, the goal of image annotation is to extract certain numeric features that can be further used for image classification.

Given an image of a kitchen as in Figure 5, and a numeric sequence of extracted features, we want to classify it into one of the semantic classes, such as “kitchen”, “dining room”, “living room”, or a “bathroom.” One way to classify an image would be to learn from experience or prior knowledge. Given our understanding of a kitchen or a bathroom, we can compare the features we have with that of a kitchen, and classify an unknown image into a kitchen. We can use this method to classify all images as a kitchen in our data-base. However, this approach is not ideal since one image is not a good representation of all kitchen types, and some images of the kitchen might not be even retrieved from the data-base. Another approach would be to identify all the images of a kitchen in the data-base and then train the classifier to identify kitchen. This approach requires annotation of the huge amount of images and then training the classifier.

**Figure 5**

*An Image*

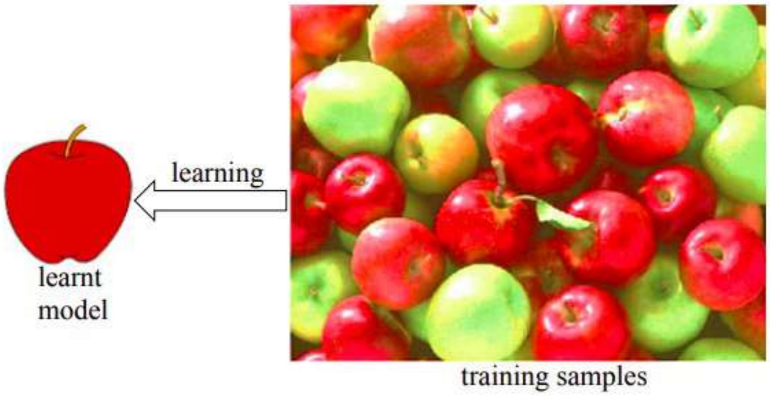


*Note*: An image to be classified into one of the classes.

Classifier can be trained in two approaches: generative and discriminative. The generative approach assumes there is a general, abstract model underlying all objects [15], such as there is a model behind all trees, humans, apples, etc. When people try to recognize an object, they are actually comparing an object with this model. One way to create general model is to take an average.

**Figure 6**

*Generative approach*



*Note*: Generative approach [15].

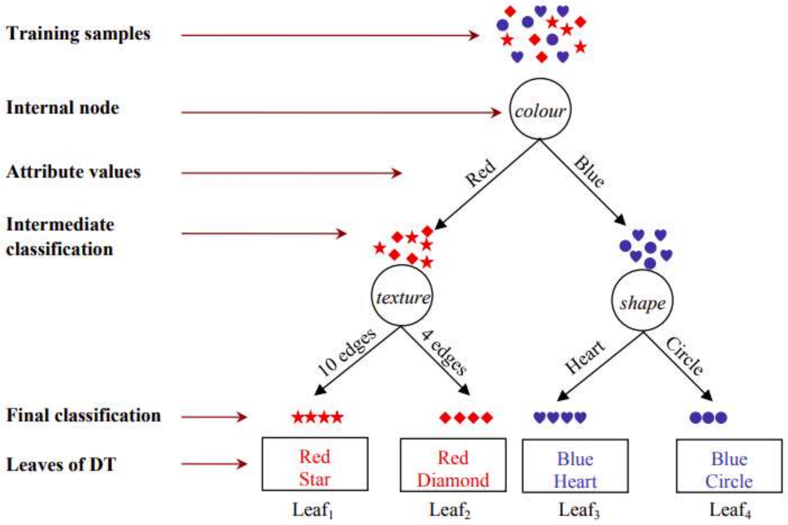
As seen in Figure 6, the model takes an average of all apples and creates a general model of an apple. This approach, however, can only distinguish an apple from non-fruit. When it comes to classifying an apple from other fruits, such as peach, the approach has difficulties. The approach usually exploits probability distributions such as Gaussian and Bayesian. Discriminative approach has no underlying abstract model. It does classification based on similarity or difference of objects in the image. This approach requires large sample data for training, and then, a hyperplane, called a classifier, is fitted between classes in a high-dimensional feature space. The optimal hyperplane is found through trial and error. An example of a discriminative approach would be Support Vector Machine (SVM) classifier.

SVM is a linear classifier that divides a data-set into two classes with the hyperplane. The data points that make up the hyperplane are called the support vectors. The SVM works by trial and error, fitting hyperplanes until an optimal hyperplane is obtained. Recent progress in SVM is the development of a kernel-based SVM that transforms data into higher space so that it can be easily separated. In general, SVM is a binary, non-probabilistic classifier, which makes it less robust than Bayesian classifier or a Decision Tree (DT) classifier.

DT classifier is an intuitive, hierarchical and step-by-step analysis. It has a “divide-and-conquer” approach in learning classification from a data-set. DT method is transparent and is easy to understand. It starts at the root node, and continues until all instances of the subset have the same class or no data-set is left. The DT has an upside down model. An example of  a DT is shown in Figure 7. On the DT, an internal node is labeled with the attribute, and the branches coming out of the node are labeled with the possible values of the attribute. The leaf node has no outgoing branches and it is labeled with the class or a probability distribution over the classes. Depending on the data-set, a DT can be either a classification tree or a regression tree. In a classification tree, the predicted outcomes are the class labels and the input attributes are discrete values. Ina regression tree, the input values are continuous and the predicted outcomes are real numbers. The first DT algorithm was called ID3 and it accepted only discrete features [15]. The algorithm was later developed into C4.5 that accepted both discrete and continuous values [15].

**Figure 7**

*Classifier*



*Note*: An example of DT.

All DT algorithm have a splitting point which is critical to the success. Therefore, a variety of splitting criteria have been developed. The idea is to select an attribute that will give the minimum amount of uncertainty. Each type of DT has its own classification splitting criterion. For example, Classification and Regression Tree (CART) has a two-ing criteria which measures the difference between the two split nodes, maximizes this difference and then splits. Gini impurity is another splitting criteria that is used in DT. First, the Gini Index (GI) is computed for the left and right split nodes and then the Gini Impurity is calculated.

In summary, a DT is a powerful image classification tool which is in between generative and discriminative approaches. It is simplistic, transparent compared to other machine classification models such as Convolution Neural Networks (CNN). DT can handle both numeric and categorical data, it does not require complex computation, had intuitive approach and a step-by-step analysis which is based on selected attributes.

### *2.4. Mask R-CNN*

Mask R-CNN is a variation of a Deep Neural Network, a Convolutional Neural Network (CNN). It detects objects in an image and generates segmentation masks for each instance. Since Mask R-CNN is the variant of CNN, the CNN and R-CNN will be briefly discussed before detailing Mask-R-CNN.

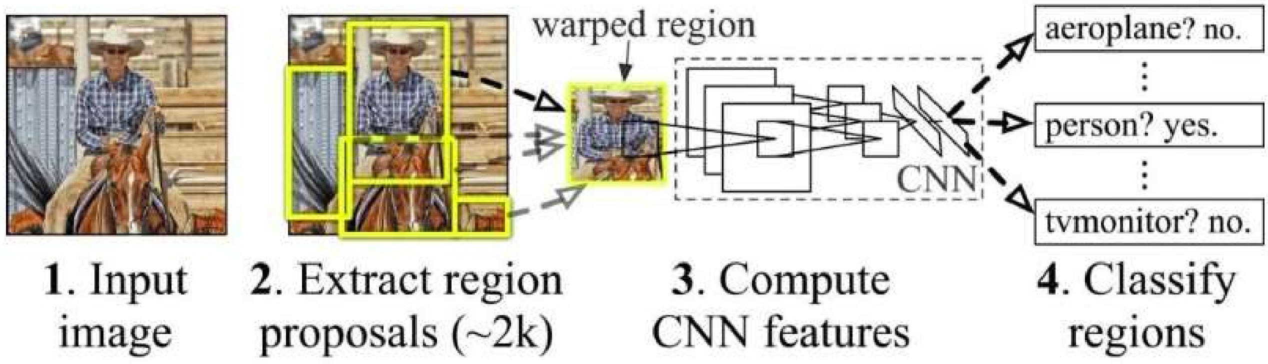
CNN is a type artificial neural network that is used for image recognition. It consists of the three main layers [18]:

- convolutional layer which uses filters and kernels to abstract an input image
- pooling layer which helps to summarize the features into the feature map
- fully connected layer which connects every neuron in one layer to every neuron in another layer.

CNN is the basic building block in object detection. A betterment of the CNN would be Region-Based Convolutional Neural Network (R-CNN). R-CNN utilizes bounding boxes across object regions, and then CNN evaluates each region independently to classify multiple image regions into a specific class [19].

**Figure 8**:

*R-CNN architecture*



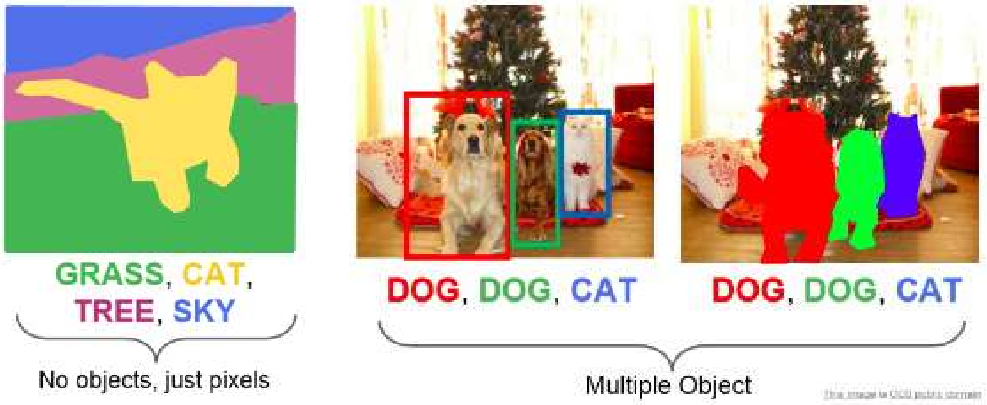
An even better version of R-CNN would be Faster R-CNN that has the following stages:

- Regional Proposal Network (RPN) which proposes multiple objects available within an image
- Faster R-CNN which extracts features using Region of Interest Pooling (RoIPool) from each bounding box and performs classification. It extracts a small feature map from each RoIPool.

Faster R-CNN is an advancement of R-CNN as it is built to optimize computational speed [20]. Mask R-CNN is built on top of Faster R-CNN. It partitions an image into multiple segments, known as image objects. This segmentation helps to locate objects and their boundaries in the image. Mask R-CNN employs two types of image segmentation which are semantic and instance segmentation. In semantic segmentation similar objects in a single class are classified from the pixel level. This kind of segmentation is used for the background segmentation.

**Figure 9**

*Segmentation*

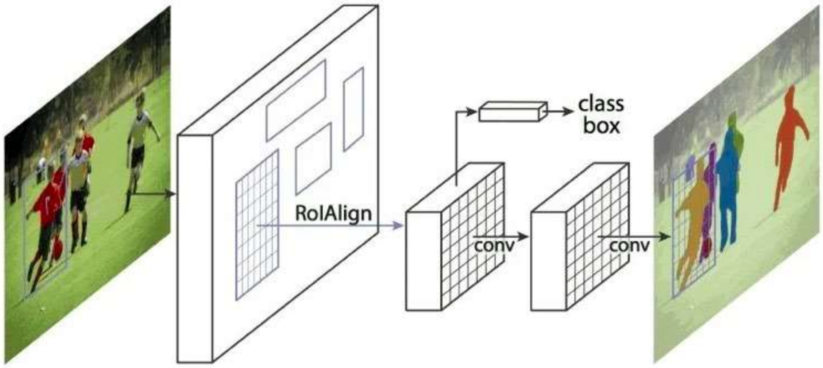


*Note*: Semantic segmentation versus instance segmentation.

Instance segmentation, on the other hand, is able to segment an image into various instances and detect objects as can be seen in Figure 9. Mask R-CNN has three outputs for each candidate object, a class label, a bounding-box offset and an object's mask which provides finer spatial layout of an object [17].

**Figure 10**

*Mask R-CNN*



*Note*: Mask R-CNN architecture.

The advantages of Mask R-CNN to other object detection algorithms include simplicity, solid performance, efficiency and flexibility. The algorithm is easy to generalize to other tasks, simple to implement and train. For these reasons, Mask R-CNN was chosen as the basic object detection mechanism for this project. Even though the thesis is based on the use of the explainable classifier and commonsense premises, the basic object detection was performed with the Mask R-CNN, the black-box algorithm. It might be feasible in the future to incorporate commonsense in basic object detection which might improve the detection algorithms.

# 3 Proposed Approach

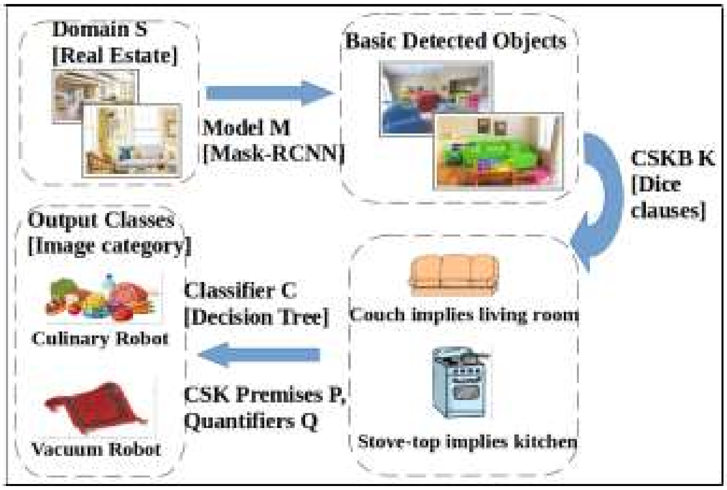
## *3.1 Approach Overview*

Commonsense is universal and essential to how humans understand and learn the world around them. Despite being an innate and ubiquitous, we have no single definition that precisely nails what commonsense is. At least, most definitions agree it is naturally taught human ability that helps them navigate throughout life. The concept is unusually broad and includes social abilities and naive sense of the world. Naive sense indicates we know not to place a hard rock on a hanging piece of plastic as it can break. People know these things without learning physics equations or higher mathematics. Humans also have a good sense of time and space that helps them plan and organize without being too exact. Intriguingly, commonsense has been a challenge for the artificial intelligence. Modern AI is designed to tackle very specific problems but all the models fall flat when dealing with vague, abstract rules. As an example, so many AI tools were built to help detect Covid, but none of them have really stood the test of time or proliferated largely among the masses for Covid detection, nor have they been considered truly indispensable by the healthcare professionals [21]. Another example, when given the following text to GPT-3 text generator, "You poured yourself a glass of cranberry, but then absentmindedly, you poured about a teaspoon of grape juice into it. It looks OK. You try sniffing it, but you have a bad cold, so you can't smell anything. You are very thirsty. So you," the AI response was "drink it, you are dead" [22]. Therefore, it is clear that the results in incorporating commonsense into AI are diminishing. It takes enormous amount of time, resources and data to create commonsense data-bases and implement them into the computer vision. Yet, with all the research, AI has still proven unable to grasp nuances of human commonsense. Our paper is an attempt to incorporate commonsense knowledge into the image classification algorithm with the use of the Dice CSKB (Commonsense Knowledge Base) [13].

The  Dice CSKB framework was chosen due to the fact that it derives refined knowledge from the existing CSK collections such as ConceptNet and Quasimodo. Our approach does not require a prior image annotations since the classifier is already trained. We use a Decision Tree (DT) classifier due to it being transparent and explainable. Figure 11 depicts an overview of the approach.

**Figure 11**

*CSK-Detector*

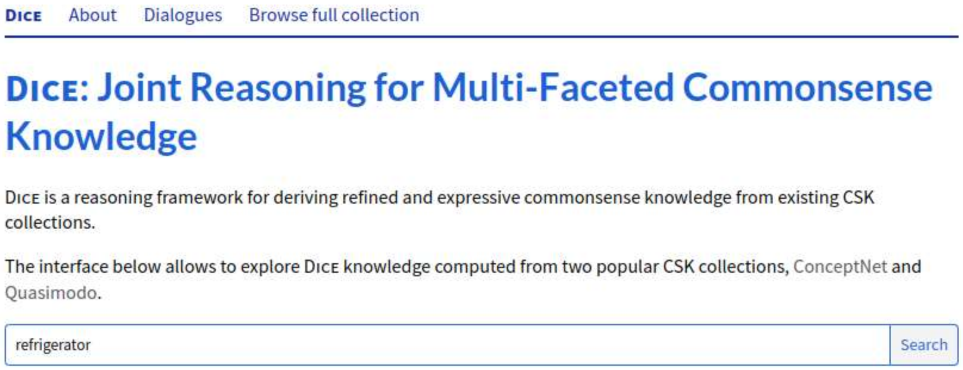


*Note*: Overview of CSK-Detector approach executed in domestic robotics.

We collected more than a thousand of room images that included kitchen, living room, bathroom, and bedroom. The images were resized and annotated using labelme software. The basic object detection was performed using Detectron2 which utilizes Mask R-CNN algorithm. Detectron2 is a Facebook AI Researcher's library that provides state-of-the-art detection and segmentation algorithms. It also includes trained models and a large set of baselines. Once the objects were detected, such as couch, lamp, microwave, refrigerator, etc, the images were passed through the quantifiers that determined the relevance of the image. For example, it makes sense to us that normal household kitchen would have no more than two refrigerators. If there are more, then, probably, we see an image of a showroom such as Home Depot, but not the real kitchen. After that, relevant images were collected and passed through the Dice reasoning clauses. The Dice framework interface is presented in Figure 12. If the user enters the word refrigerator, the Dice system will return the clauses of the form “be at home,” or “be at in the kitchen,” etc., along with the source of the knowledge base. Each clause has 5 scores: total score, plausible, typical, remarkable and salient.

**Figure 12**

*Dice framework*



Dice: Joint Reasoning for Multi-Faceted Commonsense Knowledge

Dice is a reasoning framework for deriving refined and expressive commonsense knowledge from existing CSK collections.

The interface below allows to explore Dice knowledge computed from two popular CSK collections, ConceptNet and Quasimodo.

*Note:* Dice reasoning framework interface.

The details on how scores were obtained are discussed in the next section. Based on this scores, the training data for the DT classifier was formed where rooms were assigned a semantic class, such as a kitchen, living room, bathroom or a bedroom. Once the room classes were determined, the robot categories were assigned. The robots were of the three categories: culinary, vacuum and laundry. Since vacuum robot cleans the floor and the carpet, virtually any room can belong to this category. Culinary robot performs the job in the kitchen and, thus, the kitchen room category indicates a job for the culinary robot. The laundry robot should be able to collect clothes and clean them. As such, images with clothes on the floor and of the laundry should indicate the laundry robot chores. Thus, the training data-set contained images with the semantic class and robotic label attached to them. After that, a DT classifier was trained. The classifier learned with the precision close to 91% in the kitchen category due to the large amount of CSK clauses in the Dice data-base. Other image categories were close to 80% and the lowest 72% was for the laundry room due to scarcity of clauses. This indicates the precision of the classifier depends from the refinement and the amount of clauses. Therefore, further model improvement involves addition of reasoning clauses.

### *3.2. Details of Approach*

Our CSK-Detector approach has the following parts:

1. Source domain S with basic detection model M
2. CSKB K to find object relationships in images
3. Explainable classifier C such as a decision tree
4. Simple CSK premises P with quantifiers Q

It outputs image category based on relevant object detection. 1. For S, we use real estate, employing M as Mask R-CNN. It resizes images, and detects basic objects, e.g. couch, TV etc. Note that there are just a few basic objects in homes (100s), vs. many growing real-world images (millions). 2. For K, we harness the multifaceted Dice CSKB (Commonsense Knowledge Base) with joint reasoning on sets of interrelated statements. It has 4 facets: plausibility, typicality, remarkability, salience [23]; each has a score, relying on soft constraints, with logical clauses, e.g. Eq. 1, 2.

$$Pl(s_1,p) \wedge Rl(s_1,s_2) \wedge \neg Pl(s_2,p) = Rm(s_1,p) \qquad (1)$$

$$Ty(s,p) \wedge Rm(s,p) = Sa(s,p) \qquad (2)$$

Here, P l means plausible, Rl is related, Rm is remarkable y is typical, Sa is salient, s, s1, s2 are semantic entities or concepts (e.g. rooms, chores), and p is a property (e.g. Objects in rooms). If it is plausible to find a bed in a bedroom and not plausible to find it in a related entity kitchen, a bed is remarkable for a bedroom and gets a high remarkability score there. Using scores for each facet, aggregated scores are learned via a regression model in Dice. We reuse these, calling them A scores in CSK-Detector. Each detected object per image receives an A score. It is used to assign classes on a uniform scale of 4: [0.0–0.24] : no, [0.25–0.49] : low, [0.5–0.74] : medium, [0.75–1.0] : high, e.g. if TV has A = 0.8 for bedroom, it has a high chance of being there. This logic extends to object-combinations via average aggregate scores, calculated as A avg for n distinct objects as in Eq.3.

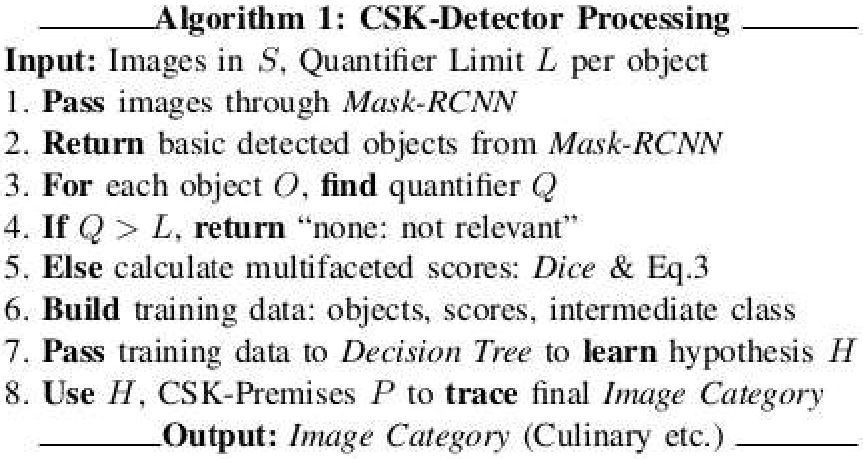
$$A_{avg} = (A_1 + A_2 + ... + A_n)/n \qquad (3)$$

For instance, if spices, sink are detected in an image with A scores 0.91, 0.58 respectively for the concept kitchen, then A avg = 0.75 (high chance that the image is a kitchen). 3. Information is fed to C, a decision tree classifier by creating a data-set of all the scores and an intermediate class (e.g. room=kitchen: no, low, medium, high). This forms training data for C to learn a hypothesis H. 4. We propose commonsense premises P in CSK-Detector as per concepts in K (Dice). These, along with the learned hypothesis H help to gauge the final image categories (e.g. culinary). Premises P entail positive and negative clauses for added emphasis. Examples are in Eq. 4, 5.

$$Room = KitchenClass = Culinary \text{ (4)}$$

$$Chore = WashClothesClass = \neg Vacuum \text{ (5)}$$

We propose quantifiers Q for objects. By our own commonsense, it is not feasible to have more than 2 ovens in a kitchen, so if that occurs in an image, it can be something else, e.g. showroom. Various objects have different Q values with upper limits L. We pre-define these using basic CSK and store them for □100 potential objects in S (e.g. oven, bed). They are used by CSK-Detector for filtering, e.g. “o1=oven, q1=3: class=none” (q1 is quantifier for object o1). Given all this discussion, the algorithm for CSK-Detector appears as Algorithm 1, based on our execution.



```
Algorithm 1: CSK-Detector Processing
Input: Images in $S$, Quantifier Limit $L$ per object
1. Pass images through Mask-RCNN
2. Return basic detected objects from Mask-RCNN
3. For each object $O$, find quantifier $Q$
4. If $Q > L$, return “none: not relevant”
5. Else calculate multifaceted scores: Dice & Eq.3
6. Build training data: objects, scores, intermediate class
7. Pass training data to Decision Tree to learn hypothesis $H$
8. Use $H$, CSK-Premises $P$ to trace final Image Category
Output: Image Category (Culinary etc.)
```

Since the algorithm is based on the decision tree classifier, the time complexity would be$O(k)$, where k represents the depth of the tree. The corresponding time complexity for the training a tree is$O(nlog(n)d)$, where n is the number of training points and d is the number of features. The latter formula is based on the fact that sorting numerical features takes $O(nlogn)$ time and sorting all features is $O(nlog(n)d)$. Even though the information gained at each splitting node takes $O(nd)$ time for all features and the total time complexity is $O(nlognd) + O(nd)$ , in the big O notation the formula simplifies to$O(nlog(n)d)$. The space complexity is$O(p)$, where p represents the number of nodes.

As it is known, decision trees tend to over-fit the training data. To avoid overfitting, the use of random forest was applied. Random forest is an ensemble method which uses multiple decision trees. The starting point in determine the number of trees is ten times the number of predictors. However, the improvement of the method decreases as the number of trees increases, which means at the certain point the benefit in prediction performance from learning more trees will be lower than the cost in computation time for learning from additional trees. This point will be seen as the expected variance will decrease, and the proper number of decision trees can be obtained from it.

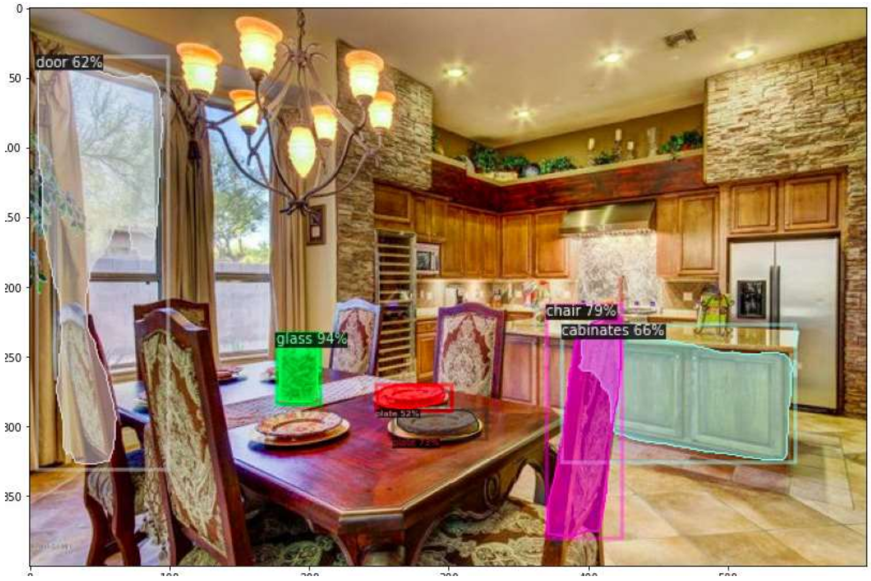
### *3.3 Implementation in Domestic Robotics*

CSK-Detector is implemented in the context of domestic robotics in this paper. We obtain images online from the real estate domain. These images are resized to the height, width of 430 pixels, and 640 pixels respectively. They are preprocessed and passed into Detectron2 (PyTorch-based modular object detection library), which uses Mask-RCNN for object detection. It detects basic objects, e.g. refrigerator, couch, table, chair, sofa, counter-top (see Fig.13). Each object is assigned a cutoff as a quantifier upper limit L based on commonsense knowledge, e.g. a few houses may have 2 refrigerators but it is odd to see 3 refrigerators in a household. Hence, 3 is a quantifier limit L for refrigerator w.r.t kitchen (so if 3+ refrigerators are detected in the image, it is considered irrelevant to a household, and hence to domestic robotics, e.g. it may be an image of a retail store

Showroom such as Home Depot). See Table I for examples of L for some basic objects.

**Figure 13-A**

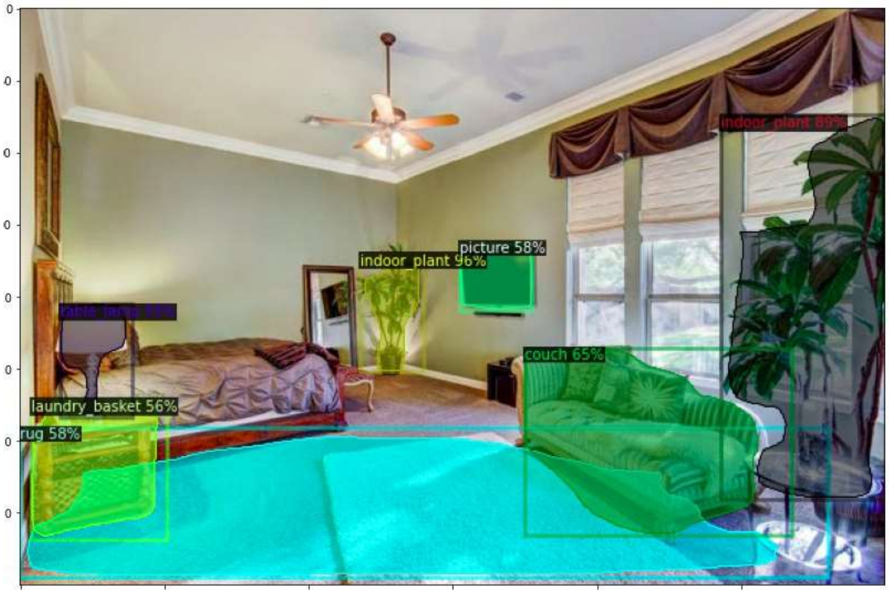
*Detected objects*



*Note*: Examples of basic detected objects in an image.

**Figure 13-B**

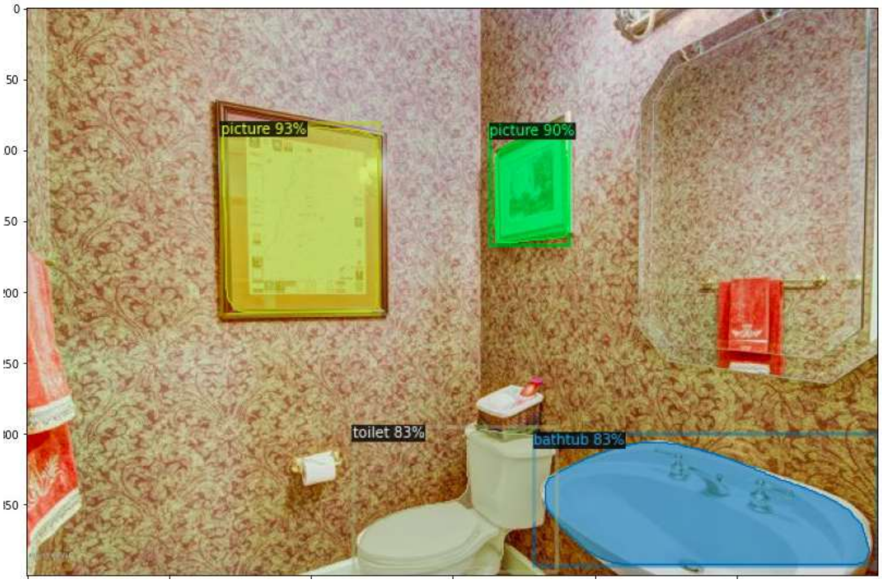
*Detected objects*



*Note:* Examples of basic detected objects in an image.

**Figure 13-C**

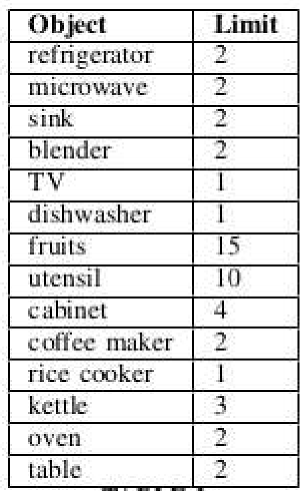
*Detected objects*



*Note*: Examples of basic detected objects in an image.

**Table 1**

*Quantifiers*



| Object | Limit |
|---|---|
| refrigerator | 2 |
| microwave | 2 |
| sink | 2 |
| blender | 2 |
| TV | 1 |
| dishwasher | 1 |
| fruits | 15 |
| utensil | 10 |
| cabinet | 4 |
| coffee maker | 2 |
| rice cooker | 1 |
| kettle | 3 |
| oven | 2 |
| table | 2 |

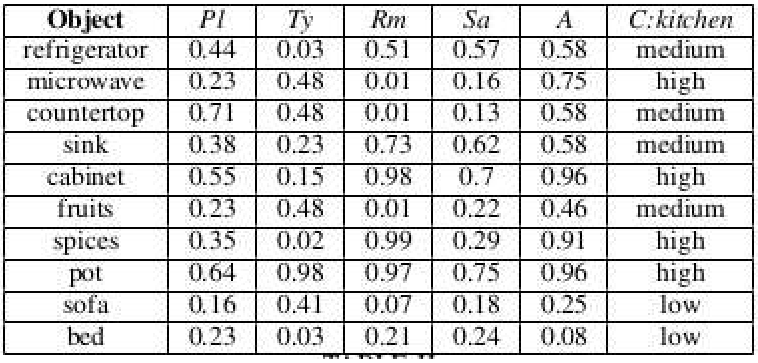
*Note*: Example of quantifier upper limits for a few objects.

After selecting relevant images based on quantifiers and limits, detected objects are passed through relational clauses derived from the multifaceted CSKB Dice [23]. Its statements are simple, e.g. “refrigerator in a kitchen”, “couch in a living room”. They carry weighted soft constraints for reasoning on 4 facets: plausibility, typicality, remarkability, and salience. Plausibility indicates if the statement makes sense w.r.t. a concept and its properties, e.g. does it make sense to see a bathtub in the kitchen? (No). Typicality means that a property holds for most instances of the concept, e.g. utensils in the kitchen. Remarkability is a property that distinguishes it from its siblings (i.e. other similar concepts), e.g. spices are remarkable in the kitchen because it is odd to find them in other rooms of a house. Salience is a property truly characteristic of a concept (salient feature), so it must be remarkable & typical.

Each facet carries a score. Scores are fed into a regression model in Dice to learn an aggregate score. Aggregate scores derived from Dice, denoted as A scores in CSK-Detector, are used as guiding scores to assign classes. For instance, given a concept (room) “kitchen” and a property (object) “sink”, there can be many statements in Dice about both of them: “sink is found in kitchen”, “sink is located in kitchen”, “sink is near kitchen” and so on. They have their respective aggregate scores based on plausibility, typicality etc. Thus, A scores in CSK-Detector are the average aggregate scores of all such statements. Likewise, A scores for combinations of 2 or more objects (e.g. sink, oven) are calculated as their average A avg as in Eq. 3, where n is the number of distinct objects. For instance, w.r.t. kitchen, the A score for “spices” can be combined with that for “sink” to obtain an average A score for both, and hence classify an image. As per our own commonsense knowledge, we can gauge that if an image has spices and a sink, it is likely to be a kitchen, but if it has a bathtub and a sink, it is likely to be a bathroom. Such combinations can be addressed via average A scores. Hence, these combinations help to better distinguish concepts in images and thereby classify them into intermediate classes (kitchen etc.) which are then fed to the decision tree classifier. See Table II for a partial snapshot of the data used for learning a hypothesis H in the classifier. (Note: Here Pl is Plausible, Ty is Typical, Rm is Remarkable, Sa is salience, A denotes aggregate i.e. A score, and C:kitchen stands for class=“kitchen”)

**Table 2**

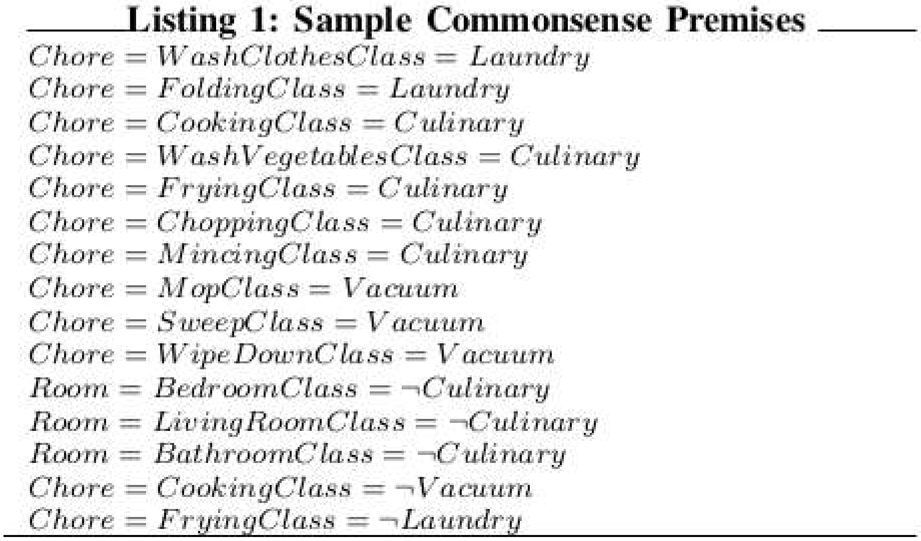
*Sample data*



| **Object** | *Pl* | *Ty* | *Rm* | *Sa* | *A* | *C:kitchen* |
|---|---|---|---|---|---|---|
| refrigerator | 0.44 | 0.03 | 0.51 | 0.57 | 0.58 | medium |
| microwave | 0.23 | 0.48 | 0.01 | 0.16 | 0.75 | high |
| countertop | 0.71 | 0.48 | 0.01 | 0.13 | 0.58 | medium |
| sink | 0.38 | 0.23 | 0.73 | 0.62 | 0.58 | medium |
| cabinet | 0.55 | 0.15 | 0.98 | 0.7 | 0.96 | high |
| fruits | 0.23 | 0.48 | 0.01 | 0.22 | 0.46 | medium |
| spices | 0.35 | 0.02 | 0.99 | 0.29 | 0.91 | high |
| pot | 0.64 | 0.98 | 0.97 | 0.75 | 0.96 | high |
| sofa | 0.16 | 0.41 | 0.07 | 0.18 | 0.25 | low |
| bed | 0.23 | 0.03 | 0.21 | 0.24 | 0.08 | low |

*Note*: Partial sample of data fed into Decision Tree Classifier.

As mentioned earlier, these intermediate classes are uniformly assigned on a scale of four (25, 50, 75, 100), based on the A score for each object (property) as per that concept. Thus, if ”refrigerator” has A score as 0.58 “to be in a kitchen”, it indicates a medium chance of an image being a “kitchen” based on that object alone. If that along with another object gives an average aggregate score of 0.75 or higher, the image with both these objects can have a higher chance of being a kitchen. The same logic extends to other combinations of 3 objects or more. The dataset is created based on the 4 facets scores (plausibility, typicality, remarkability, salience), the aggregate score and the intermediate class. Once these are fed to the decision tree classifier, it learns a hypothesis H. This serves as the basis to categorize images into final classes (culinary / laundry etc.), using our commonsense premises P. For example, if the intermediate class is a “kitchen” (medium/ high), the final class or image category is assigned as “culinary”, based on the commonsense premise in Eq. 4.



**Listing 1: Sample Commonsense Premises**

$$\begin{array}{l}
Chore = WashClothesClass = Laundry \\
Chore = FoldingClass = Laundry \\
Chore = CookingClass = Culinary \\
Chore = WashVegetablesClass = Culinary \\
Chore = FryingClass = Culinary \\
Chore = ChoppingClass = Culinary \\
Chore = MincingClass = Culinary \\
Chore = MopClass = Vacuum \\
Chore = SweepClass = Vacuum \\
Chore = WipeDownClass = Vacuum \\
Room = BedroomClass = \neg Culinary \\
Room = LivingRoomClass = \neg Culinary \\
Room = BathroomClass = \neg Culinary \\
Chore = CookingClass = \neg Vacuum \\
Chore = FryingClass = \neg Laundry
\end{array}$$

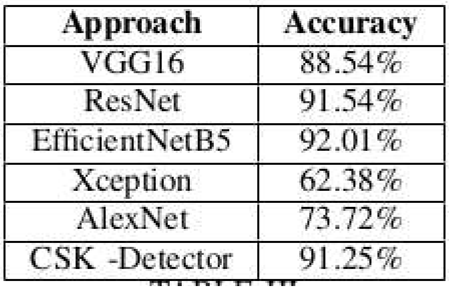
Likewise, we list a few more commonsense premises in Listing 1. These are used to detect final image categories (Algorithm 1). Hence, CSK-Detector can produce its output.

## 4 Experimental Evaluation

We summarize our evaluation of CSK-Detector as per domestic robotics. In our experiments, data sets have 2140 real estate images from Kaggle (indoor, outdoor). Abstract and retail store images are added, e.g. icons of smart cities, Home Depot showrooms etc. Ground truth is mainly based on existing image captions. Training data for CSK-Detector (classifier C) is as explained the previous section; n-fold cross-validation is used for testing (n=4,10). It gives single-class outputs (each image is in its most likely category). We compare CSK-Detector with other approaches, e.g. VGG16 [24], AlexNet [25]. Evaluation is summarized in Table 3.

**Table 3**

*Accuracy*



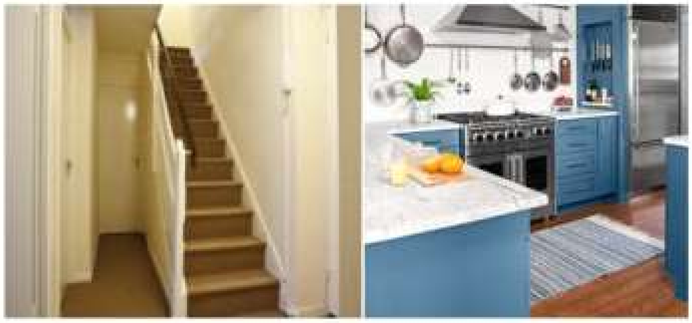
| Approach | Accuracy |
|---|---|
| VGG16 | 88.54% |
| ResNet | 91.54% |
| EfficientNetB5 | 92.01% |
| Xception | 62.38% |
| AlexNet | 73.72% |
| CSK -Detector | 91.25% |

*Note*: Comparative evaluation of CSK-Detector with % accuracy.

It is observed that CSK-Detector performs well in comparative studies (with the same data set for all approaches), While evaluation is performed with a sample of real data, we have designed CSK-Detector such that it can easily scale well to larger data sets. We briefly discuss our experimental results, highlight their merits, and point out the scope for improvement. Examples of correctly and incorrectly classified images from CSK-Detector appear in Figs 14, 15, 16. For simplicity, we show an example of an image in each category (culinary/vacuum/laundry) with misclassification & correct classification, respectively.

**Figure 14**

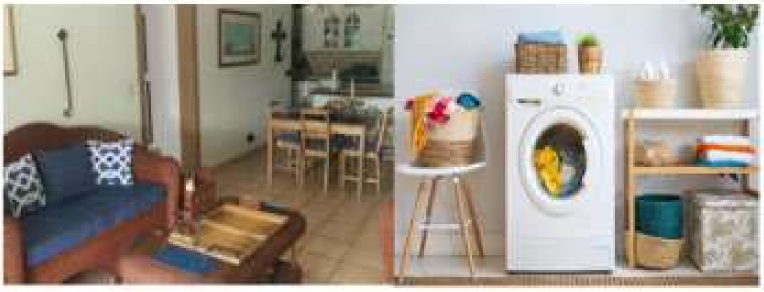
*Classified images*



*Note*: Misclassified (left) and correctly classified (right) culinary category.

**Figure 15**

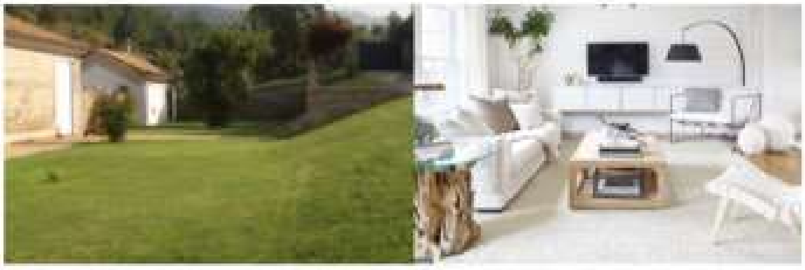
*Classified images*



*Note*: Misclassified (left) and correctly classified (right) laundry category.

**Figure 16**

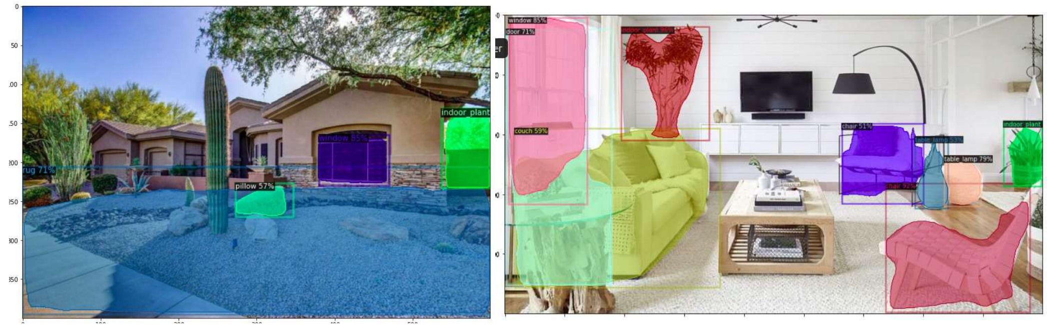
*Classified images*



*Note*: Misclassified (left) and correctly classified (right) vacuum category.

**Figure 17**

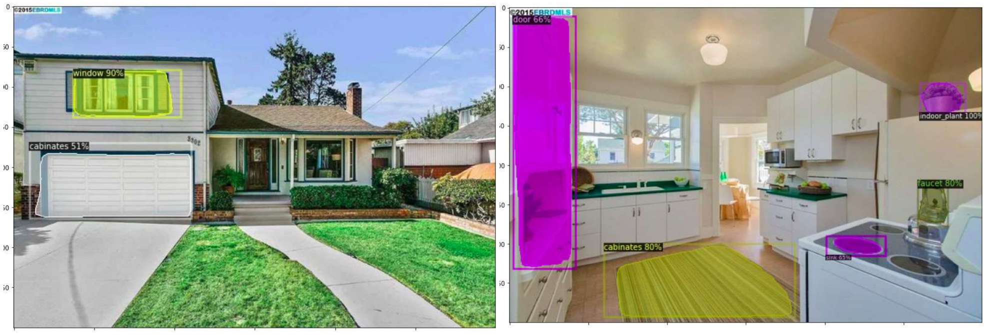
*Classified images*



*Note*: Misclassified (left) and correctly classified (right) as living room.

**Figure 18**

*Classified images*



*Note:* Misclassified (left) and correctly classified (right) as kitchen.

As observed in these examples, the misclassification can happen due to lack of commonsense knowledge derived from the logical clauses within the concerned CSKB utilized or due to incorrect basic object detection. For example, in Fig. 16, the misclassified image looks like a carpet (hence is detected thus by the Mask R-CNN), and is mistaken for the carpet by the classifier within CSK-Detector, hence considering the image to be in the vacuum category. In Fig. 4 the misclassification results from a smaller data set for the laundry category, as found in the CSKB Dice. Hence, inclusion of more logical clauses, and increasing the CSKB size as well as versatility can provide a better classification. This motivates development of domain-specific CSKBs, e.g. a domestic robotics CSKB, so as to enhance image classification and other activities in AI and robotics. Additionally, we can harness spatial collocations via related systems [26] to enhance basic object detection.

# 5 Conclusions and Future Work

## *5.1 Conclusions*

This work proposes an approach CSK-Detector to detect objects and categorize images for automated learning, and is executed in domestic robotics. That is, the approach is object-based rather than the traditional pixel-based approaches in machine learning. It is explainable and includes transferring commonsense knowledge, with premises and quantifiers; easy to fathom and modify for specific purposes. In addition, it can easily generalize to other autonomous systems. CSK-Detector has similar or higher accuracy vs. approaches with deep learning black-box models. The main contributions of this work include:

- the model that does not require image annotations from the user
- the algorithm which is easy to understand and the explainable results
- use of the CSK that reduces misclassification
- the model that can be adapted to any image category

Our work opens further research: (1) refine methods to raise accuracy; (2) add images, target classes, multiclass outputs, and spatial collocations; (3) extend to other domains. CSK-Detector can potentially aid human-robot collaboration [35] as well as next-generation multipurpose robots [36] for smart living and related pillars of smart cities.

On an important concluding note, while trying to ascertain tractability, we classify the research problem in this work as P and NP, but not as NP-Hard or NP-Complete. This can be justified as follows. The proposed solution CSK-Detector has a poly-time algorithm, so it can be classified as P. Since P is a subset of NP, as well known in fundamentals of algorithms and computation, this can be reasoanbly classified as NP. Another manner in which this can be interpreted is via a poly-time certifier, which in this work entails running the CSK-Detector algorithm for a given dataset (e.g. real estate dataset as per domestic robotics). NP-Hard problems must be intractable to the degree that no efficient algorithms exist to solve these problems. In our work, since there exists quite clearly an efficient algorithm to solve the object detection and image classification problem with respect to the data, it is thus feasible not to classify it as NP-Hard. Finally, NP-Complete problems must be part of both NP as well as NP-Hard problems, therefore it is justifiable not to classify this as NP-Complete. In summary, it can be logically justified that the problem of object detection and image classification addressed in this work, for which we propose the CSK-Detector approach as a solution by transferring commonsense knowledge with premises and quantifiers, is both P as well as NP, but not NP-Hard or NP-Complete. Hence, we conclude with a note based on the tractability of this work.

### *5.2 Limitation of Study*

The limitations of this work is due to the lack of reasoning clauses for all robotic categories. Kitchen or culinary category had the most amount of clauses, while other categories, especially laundry, lacked the data. From the DT classifier accuracy, it was obvious that the more clauses were included, the better the accuracy was. Another limitation is that the classifier was tested on Real Estate images which are images of the perfect houses. They are made to look excellent for people to buy the house. Even though real images with people, house dirt and images of showrooms were included, those images constitute a small part of all image collection. Since the more data the classifier has, the better it learns, it is possible to improve the accuracy if a higher proportion of real images are included. However, it is difficult to achieve since internet is limited to only certain amount of images.

### *5.3 Future Work*

Future work includes refinement of the model by increasing its accuracy and enlarging the application domain. The model's accuracy was highest for the culinary category due to the large pool of reasoning clauses in the CSK base Dice. Expanding the CSK base with more clauses in other categories will improve the model. The model can also be adapted to other domains within computer vision that require image categorization. For instance, real estate images can be easily classified with the model instead of manually classifying them. Other applications include general purpose robotics that are capable of handling a variety of domestic chores, and not limited to one category. Those robots have to deal with many uncertainties within their environment, such as obstacles or handling objects of different shapes. Therefore, understanding the environment they are in is of critical point; proper image categorization and object detection will help faster development of these kind of robotics. The model can also be applied across a wide range of other industries, such as health care, industrial manufacturing, smart city and many more.

# Works Cited

[1] M. Najafabadi, F. Villanustre, T. Khoshgoftaar, “Deep learning applications and challenges in big data analytics,” in *Journal of Big Data,* 2015.

[2] I. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” in *SN Computer Science,* 2021.

[3] S.R. Nandakumar, M.L Gallo, C. Piveteaus, “Mixed-Precision Deep Learning Based on Computational Memory,” in *Front. Neuroscience*, 2020.

[4] L. Alzubaidi, J. Zhang, A. Humaidi, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” in *Journal of Big Data*, 2021.

[5] J. McCarthy, “Programs with Common Sense,” Computer Science Department, Stanford University, 1959.

[6] K. Pitmann, D. Pratt, “CYC: Toward programs with common sense,” in *Communications of the ACM*, 1990.

[7] D. B. Lenat, M. Prakash, M. Shepherd, “CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks,” in *AI Magazine,* 1986.

[8] D. Lenat, G. Miller, T. Yokoi, “CYC, WordNet, and EDR: Critiques and Responses,” in *Communications of the ACM,* 1995.

[9] F. Jared, “The Sole Contender for AI,” in *Harvard Science Review*, 2003.

[10] H. Liu, P. Singh, “ConceptNet – a practical commonsense reasoning tool-kit,” in *BT Technology Journal*, 2004.

[11] M. Becker, M. Staniek, V. Nastase, “Assessing the Difficulty of Classifying ConceptNet Relations in a Multi-Label Classification Setting,”  in *ACL Anthology,* 2019.

[12] T. Mitchell, W. Cohen, E. Hruschka, “Never-Ending Learning,” in *Communications of the ACM,* 2018.

[13] Y. Chalier, S. Razniewski, G. Weikum, “Joint Reasoning for Multi-Faceted Commonsense Knowledge,” in *Automated Knowledge Base Construction,* 2020.

[14] F. Sabry, “Autonomous Robotics,” by *One Billion Knowledgeable,* 2021.

[15] S. Khatib, “Handbook of Robotics,” *Springer,* 2008.

[16] D. Zhang, “Fundamentals of Image Data Mining,” *Springer,* 2019.

[17] K. He, G. Gkioxari, “Mask R-CNN,” in *Facebook AI Research*, 2018

[18] S.Indolia, A. Goswami, S.P. Mishra, “Conceptual Understanding of Convolutional Neural Network – A Deep Learning Approach,” in *Science Direct,* 2018.

[19] R. Abushahma, M. Ali, O. Isma, “Region-based Convolutional Neural Network as Object Detection in Images,” in *IEEE Xplore,* 2019.

[20] T. Bai, Y. Pang, J. Wang, “An Optimized Faster R-CNN Method Based on DRNet and RoI Align for Building Detection in Remote Sensing Images,” in *Research Gate,* 2020.

[21] W. Heaven, “Hundreds of AI tools have being built to catch the Covid, but none of them helped ,” in *MIT Technology Review*, 2021.

[22] G. Marcus, E. Davis, “Experiments testing GPT-3’S ability at commonsense reasoning: results,” in *Technology Review,* 2020.

[23] Y. Chalier, S. Razniewski, and G. Weikum, “Joint reasoning for multi-faceted commonsense knowledge,” in *AKBC*, 2019.

[24] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[26] A. Garg, N. Tandon, and A. Varde, “CSK-Sniffer: Commonsense Knowledge for Sniffing Object Detection Errors,” in ACM EDBT BigVis, 2022.

[27] A. Pronobis, O. Martinez Mozos, B. Caputo, and P. Jensfelt, “Multimodal semantic place classification,” in *The International Journal of Robotics Research,* vol. 29, no. 2-3, pp. 298–320, 2010.

[28] D. Chaves, J.-R. Ruiz-Sarmiento, N. Petkov, and J. Gonzalez-Jimenez, “Integration of CNN into a robotic architecture to build semantic maps of indoor environments,” in *Intl. Conf.* on ANN, pp. 313–324, 2019.

[29] P. Espinace Ronda, T. Kollar, A. Soto, and N. Roy, “Indoor scene recognition through object detection,” in *ICRA*, pp. 1406–1413, 2010.

[30] A. Quattoni and A. Torralba, “Recognizing indoor scenes,” in *IEEE CVPR,* pp. 413–420, 2009.

[31] N. Tandon, A. S. Varde, and G. de Melo, “Commonsense knowledge in machine intelligence,” in *SIGMOD Record*, vol. 46, pp. 49–52, 2017.

[32] A. Pandey, M. Puri, and A. Varde, “Object detection with neural models, deep learning and common sense to aid smart mobility,” in *2018 IEEE 30th international conference on tools with artificial intelligence (ICTAI),* pp. 859–863, IEEE, 2018.

[33] A. Garg, N. Tandon, and A. S. Varde, “I am guessing you can’t recognize this: generating adversarial images for object detection using spatial commonsense (student abstract),” in *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 34, pp. 13789–13790, 2020.

[34] S. Razniewski, N. Tandon, and A. S. Varde, “Information to wisdom: Commonsense knowledge extraction and compilation,” in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 1143–1146, 2021.

[35] C. J. Conti, A. S. Varde, and W. Wang, “Human-robot collaboration with commonsense reasoning in smart manufacturing contexts,” in *IEEE Transactions on Automation Science and Engineering,* 2022.

[36] Mira-Robotics-Japan, “Ugo - the multi-purpose household robot of the future.” www.dw.com/en/ugo-the-multi-purpose-household-robot-of-the-future/video-55585607, 2021.

[37] Y. Chalier, S. Razniewski, and G. Weikum, “Joint reasoning for multifaceted commonsense knowledge,” in AKBC, 2019.

[38] Y. Chalier, S. Razniewski, and G. Weikum, “Joint reasoning for multifaceted commonsense knowledge,” in AKBC, 2019.

[39] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no.6, pp. 84–90, 2017.

[41] A. Garg, N. Tandon, and A. Varde, “CSK-Sniffer: Commonsense Knowledge for Sniffing Object Detection Errors,” in ACM EDBT BigVis, 2022.

# APPENDIX A. THE COMMONSENSE KNOWLEDGE BASE DICE

Parts of Culinary Data

| item | plausible | typical | remarkable | salient | score | culinary |
|---|---|---|---|---|---|---|
| refrigerator | 0.44 | 0.03 | 0.51 | 0.57 | 0.58 | 2 |
| microwave | 0.23 | 0.48 | 0.01 | 0.16 | 0.75 | 3 |
| countertop | 0.71 | 0.48 | 0.01 | 0.13 | 0.58 | 2 |
| sink | 0.38 | 0.23 | 0.73 | 0.62 | 0.58 | 2 |
| cabinet | 0.55 | 0.15 | 0.98 | 0.7 | 0.96 | 3 |
| food | 0.7 | 0.18 | 0.43 | 0.68 | 0.58 | 2 |
| fruits | 0.23 | 0.48 | 0.01 | 0.22 | 0.46 | 2 |
| spices | 0.35 | 0.02 | 0.99 | 0.29 | 0.91 | 3 |
| rice_cooker | 0.69 | 0.37 | 0.57 | 0.52 | 0.58 | 2 |
| kettle | 0.11 | 0.99 | 0.9 | 0.29 | 0.46 | 2 |
| blender | 0.16 | 0.02 | 0.91 | 0.65 | 0.87 | 3 |
| dishwasher | 0.23 | 0.17 | 0.95 | 0.57 | 0.75 | 3 |
| pot | 0.64 | 0.98 | 0.97 | 0.75 | 0.96 | 3 |
| crockery | 0.47 | 0.41 | 0.28 | 0 | 0.46 | 2 |
| cupboard | 0.39 | 0.11 | 0.96 | 0.6 | 0.67 | 2 |
| chair | 0.23 | 0.48 | 0.23 | 0.2 | 0.75 | 3 |
| table | 0.36 | 0.29 | 0.54 | 0.68 | 0.58 | 2 |
| tv | 0.73 | 0.43 | 0.34 | 0.24 | 0.46 | 2 |
| drawer | 0.5 | 0.08 | 0.97 | 0.72 | 0.58 | 2 |
| freezer | 0.7 | 0.02 | 0.97 | 0.77 | 0.87 | 3 |
| knife | 0.29 | 0.39 | 0.15 | 0.18 | 1 | 3 |
| coffee maker | 0.36 | 0.98 | 0.69 | 0.75 | 0.46 | 2 |
| spatula | 0.16 | 0.03 | 0.18 | 0.64 | 0.58 | 2 |
| oven | 0.67 | 0.13 | 0.15 | 0.76 | 0.58 | 2 |
| coffee mug | 0.27 | 0.02 | 0.73 | 0.72 | 0.58 | 2 |
| ladle | 0.42 | 0.02 | 0.97 | 0.76 | 0.46 | 2 |
| toaster | 0.14 | 0.06 | 0.12 | 0.11 | 0.46 | 2 |
| saucepan | 0.34 | 0.04 | 0.97 | 0.7 | 0.58 | 2 |
| egg | 0.54 | 0.17 | 0.91 | 0.74 | 0.46 | 2 |
| bread | 0.28 | 0.34 | 0.47 | 0.31 | 0.58 | 2 |
| butter | 0.06 | 0.42 | 0.27 | 0.2 | 0.58 | 2 |
| vegetable | 0.19 | 0.64 | 0.06 | 0.06 | 0.46 | 2 |
| lettuce | 0.16 | 0.19 | 0.07 | 0.06 | 0.67 | 2 |
| english muffin | 0.34 | 0.35 | 0.54 | 0.36 | 0.5 | 2 |
| hot sauce | 0.14 | 0.07 | 0.85 | 0.08 | 0.46 | 2 |
| cook | 0.81 | 0.11 | 0.76 | 0.78 | 0.58 | 2 |
| dish | 0.77 | 0.97 | 0.91 | 0.77 | 0.92 | 3 |
| skillet | 0.05 | 0.1 | 0.58 | 0.48 | 0.46 | 2 |
| bed | 0.23 | 0.03 | 0.21 | 0.24 | 0.08 | 0 |
| sofa | 0.16 | 0.41 | 0.07 | 0.18 | 0.25 | 1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| drink | 0.19 | 0.66 | 0.08 | 0.09 | 0.46 | 2 |
| fork | 0.28 | 0.11 | 0.93 | 0.11 | 0.58 | 2 |
| medicine | 0.18 | 0.61 | 0.08 | 0.06 | 0.46 | 2 |
| shower | 0.26 | 0.05 | 0.04 | 0.04 | 0.19 | 0 |
| bathtub | 0.26 | 0.05 | 0.04 | 0.04 | 0.19 | 0 |

Parts of Laundry Data

| item | quantity | plausible | typical | remarkable | salient | score |
|---|---|---|---|---|---|---|
| clothes | 1 | 0.15 | 0.46 | 0.14 | 0.13 | 0.82 |
| t_shirt | 1 | 0.53 | 0.71 | 0.11 | 0.05 | 0.58 |
| detergent | 1 | 0.89 | 0.9 | 0.69 | 0.95 | 0.6 |
| bleach | 1 | 0.48 | 0.48 | 0.32 | 0.39 | 0.58 |
| table | 1 | 0.43 | 0.27 | 0.95 | 0.45 | 0.67 |
| dryer | 1 | 0.49 | 0.34 | 0.89 | 0.74 | 0.56 |
| washer | 1 | 0.32 | 0.37 | 0.42 | 0.62 | 0.58 |
| wash_cloth | 1 | 0.68 | 0.57 | 0.13 | 0.11 | 0.46 |
| ironing_board | 1 | 0.87 | 0.92 | 0.57 | 0.92 | 0.62 |
| dirty_clothes | 1 | 0.45 | 0.36 | 0.78 | 0.6 | 0.56 |
| shelves | 1 | 0.23 | 0.26 | 0.87 | 0.19 | 0.46 |
| paper_towel | 1 | 0.04 | 0.31 | 0.69 | 0.52 | 0.46 |
| trash_can | 1 | 0.16 | 0.4 | 0.84 | 0.05 | 0.46 |
| towel_rack | 1 | 0.82 | 0.27 | 0.53 | 0.56 | 0.67 |
| water_heater | 1 | 0.01 | 0.17 | 0.02 | 0.01 | 0.66 |
| plumbing_pipes | 1 | 0.11 | 0.07 | 0.64 | 0.24 | 0.92 |
| drain | 1 | 0.73 | 0.14 | 0.01 | 0.59 | 0.58 |
| tile | 1 | 0.26 | 0.2 | 0.22 | 0.43 | 0.67 |
| laundry_bag | 1 | 0.25 | 0.45 | 0.21 | 0.25 | 0.81 |
| computer_desktop | 1 | 0 | 0.01 | 0 | 0 | 0.19 |
| stairs | 1 | 0.1 | 0.45 | 0.66 | 0.4 | 0.46 |
| sofa | 1 | 0.16 | 1 | 1 | 0.16 | 0.25 |
| shower | 1 | 0.2 | 0.4 | 0.17 | 0.16 | 0.09 |

Parts of Vacuum Data

| item | quantity | plausible | typical | remarkable | salient | score | vacuum |
|---|---|---|---|---|---|---|---|
| floor | 1 | 0.37 | 0.24 | 0.67 | 0.71 | 0.58 | 2 |
| linoleum | 1 | 0.17 | 0.57 | 0.01 | 0.1 | 0.46 | 2 |
| carpet | 1 | 0.25 | 0.33 | 0.96 | 0.62 | 0.46 | 2 |
| trash | 1 | 0.18 | 0.68 | 0.92 | 0.04 | 0.58 | 2 |
| bathtub | 1 | 0.2 | 1 | 0.85 | 0.08 | 0.4 | 2 |
| couch | 1 | 0.43 | 0.02 | 0.96 | 0.69 | 0.81 | 3 |
| sofa | 1 | 0.19 | 0.99 | 0.43 | 0.29 | 0.46 | 2 |
| coffee_table | 1 | 0.19 | 0.37 | 0.93 | 0.07 | 0.46 | 2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| picnic_table | 1 | 0 | 0.01 | 0 | 0 | 0.04 | 0 |
| closet | 1 | 0.32 | 0.31 | 0.96 | 0.28 | 0.46 | 2 |
| pantry | 1 | 0.22 | 0.04 | 0.93 | 0.08 | 0.67 | 2 |
| kitchen | 1 | 0.61 | 0.98 | 0.97 | 0.75 | 0.92 | 3 |
| bedroom | 1 | 0.34 | 0.2 | 0.98 | 0.69 | 0.58 | 2 |
| hallway | 1 | 0.16 | 0.17 | 0.2 | 0.2 | 0.87 | 3 |
| foyer | 1 | 0.22 | 0.02 | 0.978 | 0.64 | 0.58 | 2 |
| basement | 1 | 0.11 | 0.5 | 0.94 | 0.32 | 0.67 | 2 |
| stairwell | 1 | 0.25 | 0.12 | 0.94 | 0.25 | 0.67 | 2 |
| elevator | 1 | 0.11 | 0.11 | 0.95 | 0.52 | 0.46 | 2 |
| furniture | 1 | 0.46 | 0.98 | 0.6 | 0.45 | 0.46 | 2 |
| laundry_room | 1 | 0.73 | 0.1 | 0.97 | 0.77 | 0.75 | 2 |
| hotel_room | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| cupboard | 1 | 0.13 | 0.25 | 0.02 | 0.53 | 0.46 | 2 |
| refrigerator | 1 | 0.1 | 0.2 | 0.41 | 0.4 | 0.58 | 2 |
| front_door | 1 | 0.07 | 0.03 | 0.63 | 0.52 | 0.58 | 2 |
| gate | 1 | 0.13 | 62 | 0.16 | 0.17 | 0.46 | 2 |
| fence | 1 | 0.22 | 0.07 | 0.08 | 0.08 | 0.1 | 1 |
| locker_room | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| bathroom | 1 | 0.28 | 0.99 | 0.94 | 0.24 | 0.75 | 2 |