Common Sense Reasoning in Autonomous Artificial Intelligent Agents Through Mobile Computing

Common Sense Reasoning in Autonomous Artificial Intelligent Agents through Mobile Computing

Angel Henderson

A Thesis in the Field of Software Engineering

for the Degree of Master of Liberal Arts in Extension Studies

Harvard University

May 2020

Abstract

This thesis presents the design, analysis, and implementation of an autonomous artificial intelligent agent capable of planning, reasoning, rationalizing, communication, and learning through the use of common-sense reasoning. We discuss the cognitive modeling approach of our autonomous artificial intelligent agent and how its system design is inspired by biological inspired cognitive processes of the human brain (Crowder, 2012; Mallot et al., 2009; Trullier et al., 1997; Goertzel et al., 2010). We cover how our autonomous intelligent agent utilizes stacked artificial neural networks, combined classifiers, memories, and fuzzy systems to map out its cognitive process through unsupervised and semi-supervised learning. We state our motivations for performing this research and examine the importance of common-sense reasoning in the advancements of artificial intelligence. We cover various roles and applications of autonomous artificial intelligent agents as well as the challenges that arise when developing intelligent agents that can perceive, learn, and adapt autonomously. The outcome of this thesis includes the development of a mobile software application comprising an autonomous artificial intelligent agent that demonstrates an understanding of context, observations, utilization of past experiences, self-improvement through reinforcement learning, and hypothesis development to reach solutions – which are presented and illustrated with examples and challenges.

Dedication

To my loving wife, Natasha, for your smile, your care, your love, and for sacrificing so much for our family. Thank you for being my very best friend, my better half, and for always being my biggest and most supportive advocate. Our love is never-ending and the day I met you will always remain paramount and will always be the luckiest day of my life. Thank you for all you have given me.

To my sons, Angel Jr. and Issah, for being my lights of joy that keep me inspired and motivated every day. Being your father has been my greatest honor and nothing brings me more joy then seeing the smiles on your faces.

To my parents, Mirza & Ronald Henderson, thank you for making every sacrifice to provide my siblings and me the best upbringing possible and for never allowing me to go a day in this world not knowing that you both love me unconditionally. I could never thank you enough for your love and support.

Acknowledgments

I would like to take this time to recognize the individuals who have been my support system throughout this process. Without my thesis director, Jose Luis Ramirez Herran, giving me confidence and extending his expertise and support with this project, it would not be possible. His knowledge of the subject and ability to push me toward my goals is unparalleled because he has kept my passion alive for this project. I would like to extend my humblest gratitude for all collaboration, as well as the resources, patience, and help that I received.

My project would not be attainable without the guidance and backing of my research advisor, Dr. Sylvain Jaume. I would not have progressed to where I am without the push forward. I would like to offer my sincerest appreciation for the faith in me to complete my project as well as the encouragement to continue to move forward.

Table of Contents

List of Figures

Chapter 1: Introduction

This thesis presents the design, architecture, and implementation of autonomous artificial intelligence agents that utilize common sense reasoning and a multitude of artificial neural networks simultaneously to understand its environment, independently extract knowledge from sensory data and develop knowledge in real-time. These artificially intelligent agents have the capability of understanding the context behind data while determining the causes for varies outcomes and will attempt to understand the five W questions (who, what, when, where, why) of any data it processes. Through mobile computation, we demonstrate that providing an intelligence agent a wide range of deep neural network models allows it the capacity of properly understanding what it processes while permitting it to make better decisions when analyzing and applying new knowledge to a wide range of problems through common sense reasoning. When observing unique problems, the agents capture rich sensory data and, through data processing, develop unique datasets to discover the context behind different problems and solutions. Over time, through deep learning and on-device training, the agent gains an understanding of how to solve similar problems, which enables it the capability of developing solutions to problems it has not yet come across before.

1.1 Motivation

Artificial intelligence (A.I.) has been a substantial force in the evolution of both software applications, as well as robotics. Extensive research and investments in A.I. along with big data have contributed to all areas of business, including health care, retail, robotics, self-driving cars, virtual assistants, and security. All while simultaneously impacting how people live their lives and connect. New applications of artificial intelligence and intelligent agents arise daily. However, there are many challenges in the accessibility of A.I. based tools and limitations in how people can integrate these technologies into their lives, outside of early adaptors.

While the possibilities of artificial intelligence are truly endless, its most significant value comes when it is flexible, adaptable, and a genuinely accessible tool. Artificial intelligence can become a tool that is autonomous and makes knowledge truly obtainable to all while reducing the distance between one's imagination and reality. As big data continues to grow in size, variety, and availability, and computational capabilities grow at the same rate, our ability to develop strong artificial intelligence seems foreseeable. Unfortunately, there is a lack of mature common-sense reasoning in artificial intelligence today. This lack of common-sense reasoning limits the ability for intelligence agents to truly understand the full context behind the data it analyzes and limits its values to most cases that require unsupervised learning. This dilemma is one of the critical factors preventing artificial intelligence from reaching its maximum potential across a wide range of applications.

Achieving common sense reasoning plays a critical factor in the development of intelligent machines capable of unsupervised learning, thinking, and reasoning. Autonomous intelligent agents with common sense reasoning will have the ability to explain their conclusions. It can improve human-computer interaction, resolve ambiguities within natural language processing, and computer vision, and allow for automated planning while solving some of the most challenging problems across industries. This challenge and value are what inspired my research in common sense reasoning.

1.2 Thesis Outline

Chapter 2 goes over the foundation of common-sense reasoning. We discuss the human cognitive process and how it inspires the modules that make up the intelligent agent's system architecture. Then, we cover different biological inspired cognitive processes the inspire the architecture of the hybrid framework. Lastly, we go over other approaches taken in the field to achieve common-sense reasoning.

Chapter 3 presents the approach we utilized in this project to develop an autonomous artificial intelligent agent capable of common-sense reasoning. We go over the three subsystems that make up the hybrid intelligent system and provide a detailed breakdown of the combined deep neural network classifiers that establish each module. We then describe the parallel stacked classifiers techniques that process each module and the learning algorithms utilized to establish observation, reasoning, knowledge, that ultimately results in action.

Chapter 4 discusses the development tools and resources utilizes in this project for the development, implementation, and evaluation of the autonomous intelligent agent and the modules that drive its system. We also go over the experiment settings, evaluation metrics, and performance measurement techniques used in this project.

Chapter 5 presents case results showing how well the agent was able to identify and utilize common sense reasoning and display contextual awareness when interacting with its environment and sensory data. We also evaluate how the agent performs on common sense reasoning tests: Winograd Scheme Challenge (Bailey et al., 2015; Levesque et al., 2012; Liu et al., 2017; Rahman et al., 2012; Schüller et al., 2014; Sharma et al., 2015) and Pronoun Disambiguation Problems (Rahman et al., 2012; Liu et al., 2016; Liu et al., 2017).

Chapter 6 summarizes the thesis, provides a conclusion in meeting the objective. It also covers lessons learned, potential contribution to the field of artificial intelligence, and potential direction for future research.

Chapter 2: Background

The original goal behind artificial intelligence is the development of intelligent machines capable of learning, thinking, reasoning, and solving problems like that of human intelligence. Today, a majority of intelligent agents are considered weak A.I. capable of particular but narrow applications of machine intelligence. However, through continued research and development, the golden standard is strong artificial intelligence, which is machine intelligence equal or superior to human intelligence with the ability to plan, reason, rationalize, make judgments, communicate, and learn while being self-aware. While we are still away from strong artificial intelligence, researchers have invested heavily in the foundation of artificial intelligence, which consists of different subfields, including common sense reasoning, planning, machine learning, natural language processing, computer vision, and robotics. While there have been great strides in the field of machine learning, natural language processing, computer vision, and robotics, it has been very challenging to develop intelligent agents capable of planning, thinking, and reasoning.

Common sense reasoning is the branch of artificial intelligence that aims to provide intelligent agents the ability to learn independently, become logical while storing and applying knowledge in the same way that humans do. Unfortunately, developing common sense reasoning is very difficult due to different challenges that come in trying to teach computers rational thinking. Achieving common sense reasoning requires a combination of skills, including computer vision, natural language processing, crowdsourced input, large knowledge bases, web mining, logical analysis, and deep learning. To achieve this, we lean on stacked artificial neural networks and a system design based around biologically inspired cognitive architectures (Crowder, 2012; Goertzel, 2010).

2.1 Human Cognition

Humans are autonomous agents with the cognitive abilities of mental processes that include perception, attention, memory, metacognition, language, thinking, and problem-solving. These natural abilities allow us to perceive and use reasoning when understanding our environment. Humans have a multitude of physiological receptors called senses that makes up the perceptual systems of the brain and allows us to collect sensory data from a wide range of dim verse sources to form our perception (Anderson, 2000; Bruning et al., 1999; Crowder, 2012; Goertzel, 2010; Miller et al., 2002).

Perceptrons

Perceptrons provide humans the ability to identify, organize, and interpret sensory data so that we may have a firm understanding of our environment. As the perceptual systems in our brains captures data through senses like sight, touch, smell, hearing, and taste, these signals help to shape our memory, learning, attention, and expectations. When processing sensory data, we immediately can transform sensory data from low-level information to higher-level information, like facial recognition, speech recognition, object recognition, which helps us understand our surroundings (Anderson, 2000; Bruning et al., 1999; Crowder, 2012; Goertzel, 2010; Miller et al., 2002).

Fuzziness and Fuzzy Reasoning

As we perceive our environment, humans come across many forms of sensory data that are fuzzy, irregular, and lack consistent, contextual bases. Fortunately, humans can use fuzzy reasoning to convert fuzzy data into concepts we understand. Fuzzy reasoning, more specifically, is the inference procedure that allows us to derive conclusions from fitting our own conceptual rules and understanding of rules we interpret as known facts. Our ability to perform fuzzing reasoning, when complemented by our preceptors, plays a critical factor in our ability to display common sense reasoning (Takagi et al., 1991; Kosko, 1986; Laird, 2008).

2.2 Biologically Inspired Cognitive Architectures

Biologically inspired cognitive architectures (BICA) are cognitive architectures that aims to provide autonomous artificial intelligence agents the ability to perceive, reason, learn, think, adapt, react, and act to new data and knowledge it gains and are greatly inspired by the cognitive process of the human brain (Crowder, 2012; Mallot et al., 2009; Trullier et al., 1997; Goertzel et al., 2010). These architectures allow intelligent agent to process sensory data from its environments in real-time, organize thoughts, store knowledge, and communicate knowledge fuzzily. This allows the agent to process information that is consistent, inconsistent, noisy, fuzzy, and/or obscured similar to the cognitive process for human perception. These cognitive architectures also have the objective of improving autonomous decision making in the intelligent agent by establishing knowledge base within the cognitive processes for storing memories. Modules within BICA grant artificial intelligent agents the capability to observe and understand their environment. As its knowledge grows, the agents can interact with their environment and, over time, establish goals and problem-solve autonomously. The agent is also able to gain and utilize knowledge across a wide range of domains, scenarios, and circumstances. The cognitive architecture that inspired our foundation is the Artificial Cognitive Neural Framework (Crowder, 2012) which leans on genetic artificial neural networks, fuzzy methods in decision making, and modular systems components. The framework contains three primary subsystems that serve as its pillars as seen in Figure 1: the cognitive system, the mediator, and the memory system.

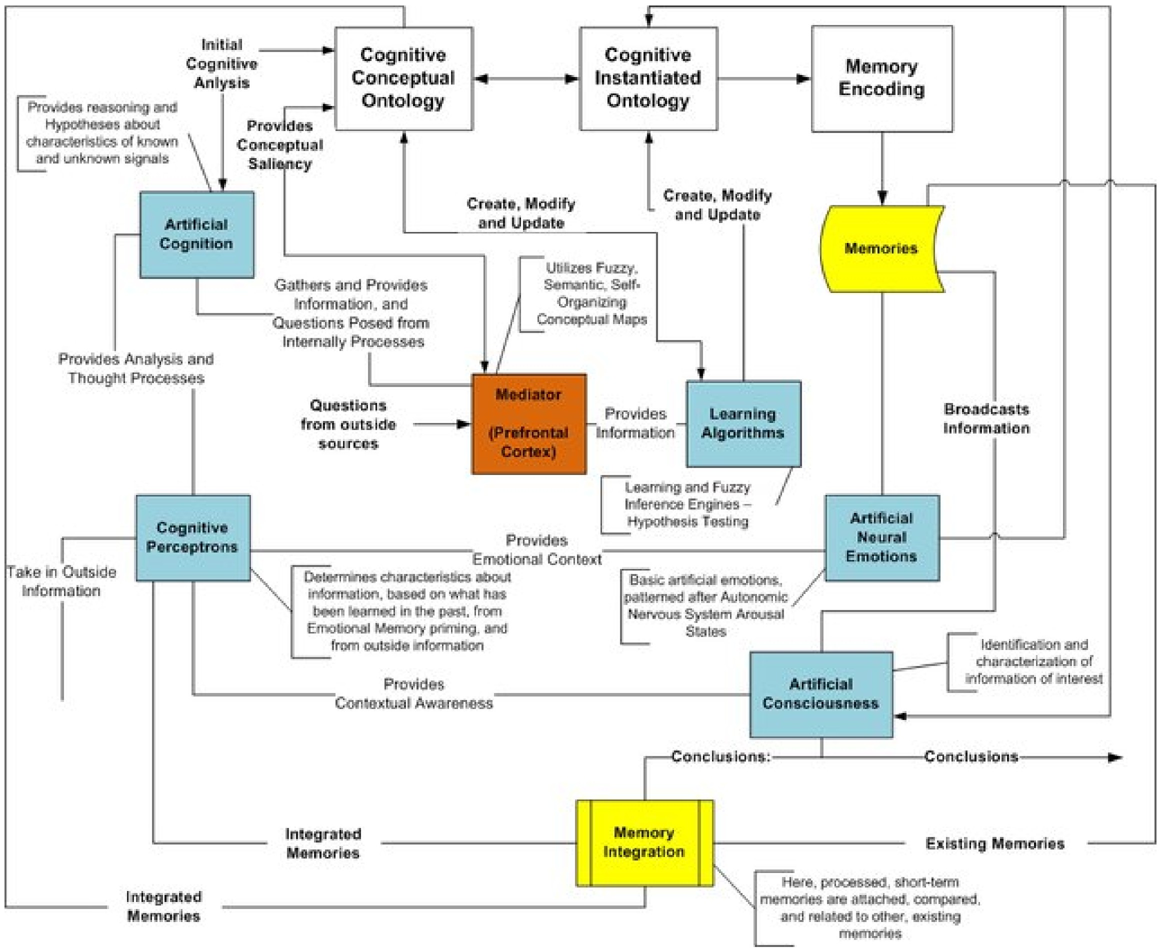


Figure 1. Artificial Cognitive Neural Framework (Crowder, 2012)

The architecture of the Artificial Cognitive Neural Network and its three main subsystems

The Cognitive System

The cognitive system is the subsystem responsible for cognitive functions parallel to human cognitive counterparts, including perception, emotions, thinking, learning, consciousness, and metacognition. The modules within this subsystem include artificial cognition, learning algorithms, artificial consciousness, artificial neural emotions, information processing, and artificial perceptrons (Crowder, 2012; Clark, 2013; Miller et al., 2002).

The Memory System

The memory system is responsible for storing short term, long term, emotion, and sensory memories. These memories contain all of the information that the intelligent agent knows and learns over time, which exists within a knowledge base. This knowledge base serves as a tool to provide intelligence to Cognitive Systems (Crowder, 2010; Hochreiter et al., 1997; Labar et al., 2006; Wallis et al., 2001).

The Mediator

The mediator is the subsystem responsible for cognitive control by providing cognitive intelligence to the intelligent agent and also facilitating data, knowledge, intelligence, and patterns throughout the entire framework (Crowder, 2012; Hochreiter et al., 1997). It retrieves intelligence from the cognitive system and transfers them to the memory system, which updates the short term, long term, and episodic memories. The meditator gives the intelligent agent the ability to use common sense reasoning, organize thoughts, and act upon what it learns when identifying a wide range of problems. As the intelligent agent's memory system matures, it will become better at reacting to new variables, acting on unique problems, and adapting to the needs of the users.

2.3 Alternate Approaches to Common Sense Reasoning

Large Knowledge Bases

Researchers have emphasized the development of massive knowledge bases to achieve common sense reasoning in computers. For example, the Allen Institute for Artificial Intelligence has developed the Alexandria Project, which aims to create a large knowledge base known as the commonsense repository (Sap et al., 2019). The Alexandria Project extracts data from images and text through machine learning as well as crowdsourced input into the commonsense repository to try to achieve common sense reasoning (Tandon et al., 2018; Bhavana et al., 2019; Khot et al., 2019). It believes that the commonsense repository, through common sense reasoning, will improve A.I. applications in fields like machine translations, medical diagnosis, robotics, A.I. safety, and Intelligent Home Assistants. Other research institutions/projects like Cycorp (Siegel et al., 2005), ConceptNet (Liu et al., 2004; Speer et al., 2013), M.I.T OpenMind (Singh et al., 2002; Singh, 2002) have taken similar approaches. The assumption is that the most critical element in achieving common sense reasoning are large knowledge bases. In deep learning, more data in large knowledge bases are usually the solutions to any significant shortcomings in common sense reasoning. While this assumption may not be incorrect, this is an assumption that has not been fully realized by models like Cyc (Panton et al., 2006), ConceptNet (Liu et al., 2004; Speer et al., 2013), and OpenMind Common Sense (Singh et al., 2002; Singh, 2002).

Common-Sense Reasoning through Mobile Computation

Research has also been performed around common sense reasoning through mobile computation, including research from MIT Media Laboratory (Singh et al., 2002; Singh, 2002). It aimed to deliver common sense reasoning through a combination of an extensive knowledge base, predictive typing aids, and natural language processing. The knowledge base driving the experiment was the OpenMind Common Sense project (Singh et al., 2002; Singh, 2002) developed by the Massachusetts Institute of Technology Media Lab. It reached many definite conclusions in the research, including the development of GloBuddy 2 (Liu et al., 2004; Musa et al., 2013; Lieberman et al., 2004; Chung et al., 2005), but also acknowledged limitations in mobile devices. These limitations include screen estate, mobile computing capabilities, reliability on internet connections, limited input, and the lack of support for native machine learning. However, mobile devices today are now capable of local and cloud-based machine learning, include a multitude of sensors for data capturing, which are optimal for crowdsourced input. Mobile platforms provide ease of software distribution, contain a multitude of user interaction inputs, and are competent machines with powerful central processing units and graphical processing units optimized for machine learning.

This thesis explores the role that mobile computing, currently, can play in achieving common sense reasoning in autonomous artificial intelligent agents. Scholars such as Gary Marcus (Davis et al., 2015) believe that the achievement of strong A.I will require human-level performance in the fields of natural language processing, computer vision, but also the ability for intelligent agents to have a common-sense understanding of the world around them. Common-sense reasoning requires computers with sensors not only capable of capturing image and audio, but also time, space, and physical interaction. Researchers have found artificial neural networks to be critical to the development of artificial intelligence but have discovered the reliance on supervised learning to significantly limit its ability to contribute heavily to common sense reasoning (Jha, 2007; Strong, 2016). Supervised learning relies heavily on detailed and accurately label data and does not perform well when dealing with fuzzy data containing limited labels. The ability to use unsupervised and semi-supervised learning models when dealing with data containing limited labels is a general requirement in situations that require common sense reasoning, including tests like the Winograd Schema Challenge (Bailey et al., 2015; Levesque et al., 2012; Liu et al., 2017; Rahman et al., 2012; Schüller et al., 2014; Sharma et al., 2015) and the Arc Reasoning Challenge (Clark et al., 2018; Boratko et al., 2018). As a result, researchers from Google Brain have worked on developing methods for common sense reasoning in recurrent neural networks using unsupervised learning (Trinh et al., 2018). The approach utilized to achieve this objective was training neural networks using language models that used immense amounts of unlabeled data. They found by training their models on both the character and word level; their neural networks were able to determine essential features in a question successfully, allowing it to understand common sense knowledge generally.

Chapter 3: Common Sense Reasoning through Artificial Intelligences

In this chapter, we discuss the approach we take in developing an autonomous artificial intelligent agent that operates over 50 artificial neural network classifiers to observe its environments, independently extract knowledge from sensory data, and determine actions to take using statistical learning methods including induction, abduction, emotional learning, reinforcement learning, and probably approximately correct learning (Sutton et al., 2018; Laird, 2008; Figueiredo et al., 2002; Crowder, 2012). We aim to demonstrate that this intelligent agent will have the capability of understanding the context behind data, as well as the cause for various outcomes using unsupervised and semi-supervised learning. More specifically, the intelligent agent will attempt to understand the five Ws (who, what, when, where, why) of any data it analyzes. As the intelligent agent matures, it will become better at reacting to new variables, able to resolve unique problems and adapt to different needs of its user.

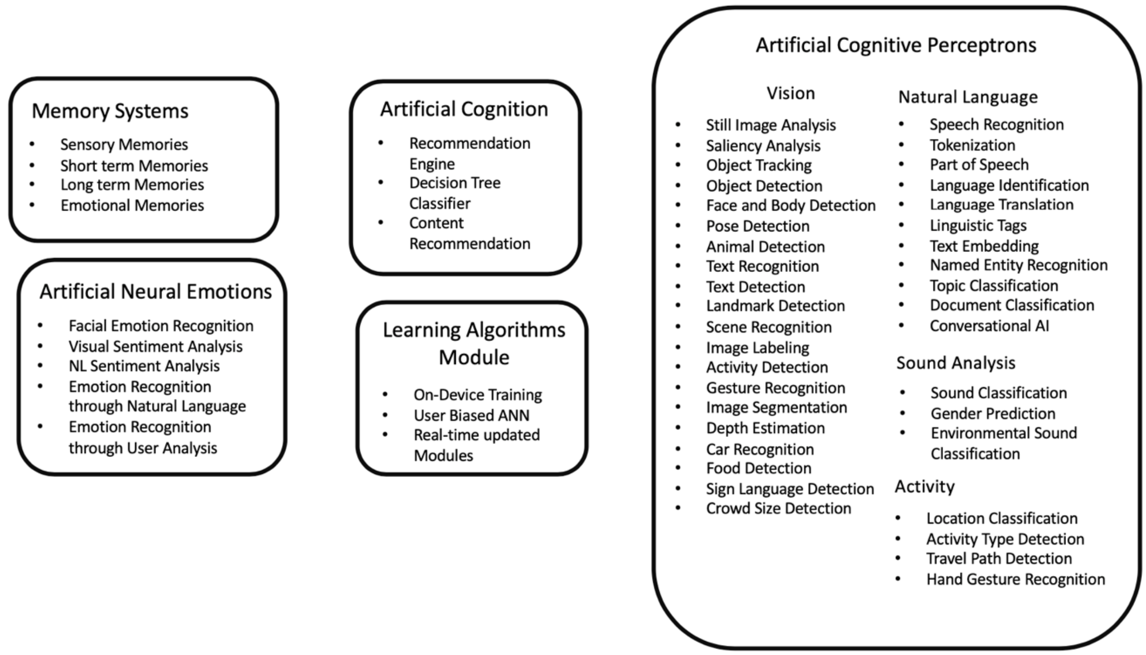


Figure 2. Intelligent Agent Cognitive Architecture

High level overview of the cognitive capabilities of the intelligent agent based on the primary cognitive modules that make up the agent's system. Similar to the human brain, the intelligent agent spready its cognitive process across different modules which contain unique classifiers that are executed at the discretion of the agent.

3.1 Approach

The approach we utilized to develop an autonomous artificial intelligent agent is a system structured around modular components. This section provides an overview of the mobile application and the modular components that make up the autonomous artificial intelligent agent. The intelligent agent exists within the mobile application and utilizes three main subsystems based around the human cognitive process to achieve its objectives.

To analyze incoming raw sensory data, the intelligent agent employs stacked deep learning classifiers in parallel through parallel computing, with aims to extract metadata, develop hypotheses, use common sense reasoning, and gain knowledge when solving problems. The agent then stores knowledge as memories within its memory system, both locally using Core Data and through a cloud-hosted NoSQL database. Over time, the agent performs on-device trainings and utilizes reinforcement training to improve its capabilities, confidence, and general intelligence.

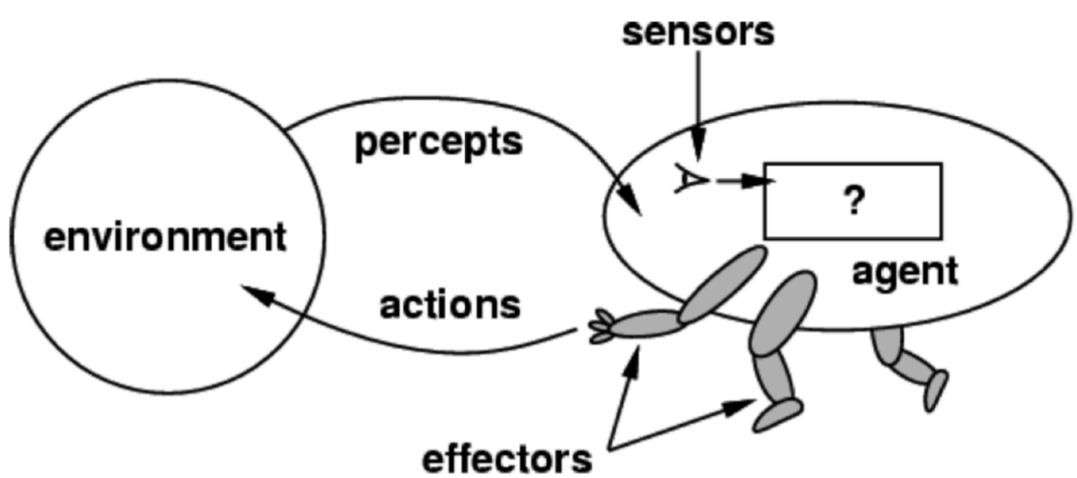


Figure 3. A General Learning Autonomous Intelligent Agent

The intelligence agent utilizes mobile sensors to capture sensory data of its environment. This is followed by methods that processes the sensory data then utilizes learning methods and its memories to make sense of its environment. It then determines its actions bases on user recommendations or independent choice.

Parallel Stacked Deep Neural Networks Models

The intelligent agent contains over 50 artificial neural network classifiers (Figure 4.) capable of natural language processing, sound analysis, computer vision, recommendations, activity classification, text classification, emotion classification, self-training, and additional deep learning skills.

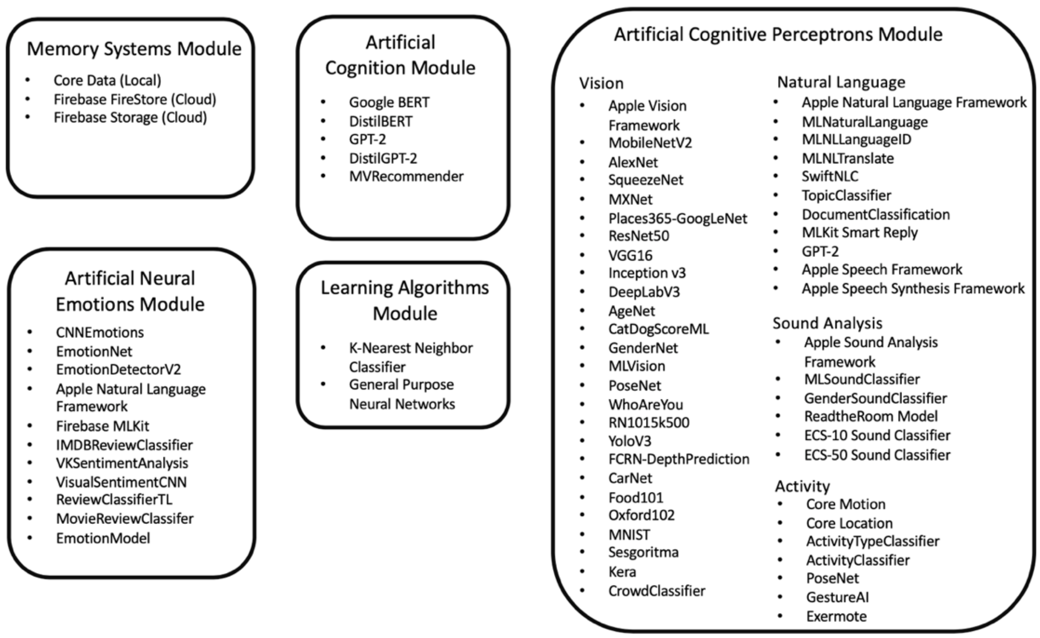


Figure 4. Artificial Cognitive Modules of the Intelligent Agent

High level overview of the cognitive architecture containing artificial neural network models that make up the primary modules of the intelligent agent. The artificial neural network models are modular and work independently but results can be combined.

These deep learning classifiers handle many roles, including preprocessing, classification, regression, clustering, reinforcement learning, and dimensionality reduction. We execute the deep neural network classifiers in parallel by utilizing Grand Central Dispatch, a native framework that optimizes the mobile application to use the systems multicore processors, which allows for these classifier models to run on separate threads at the same time as seen in Figure 5 (Woffinden et al., 2014; Sutton et al., 2018; Apple Inc., 2020).

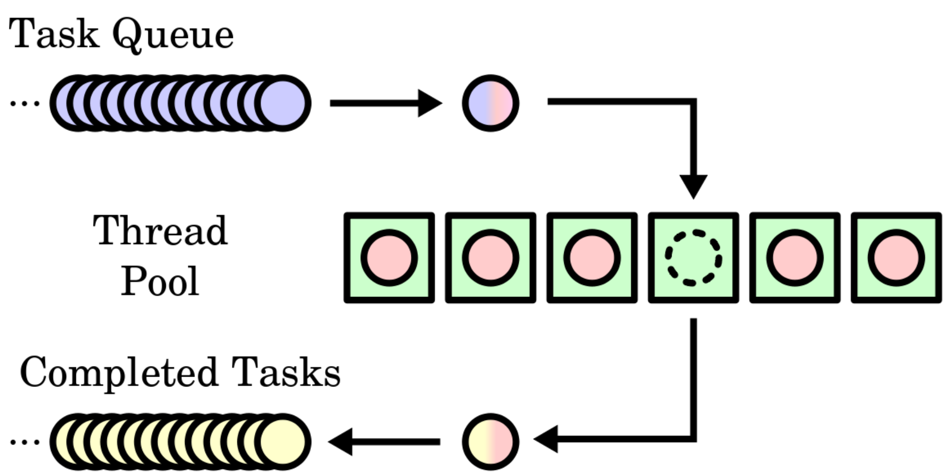
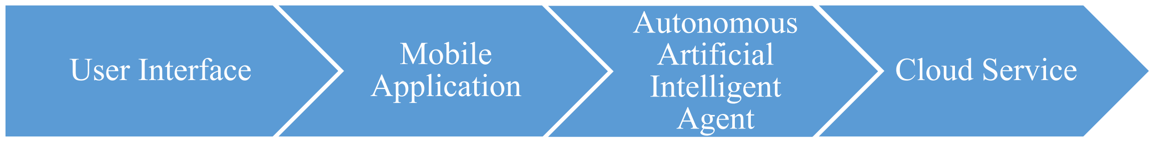


Figure 5. Grand Dispatch Task Queue Process

Grand Dispatch is a native framework that optimizes the mobile application run different classifiers in parallel threads without causing a bottleneck to the main thread which handles the user interface and user interaction.

3.2 System Architecture

The overall architecture of the system is four components, the user interface, mobile application, cloud service, and the autonomous artificial intelligent agent.



The user interface exists within an iOS mobile application, which allows a user to interact with an autonomous artificial intelligence agent through touch, text, speech, as well as gestures. As the user interacts with the mobile application, the autonomous artificial intelligence agent utilizes deep neural network models to display common-sense reasoning and helps provide solutions to a user's problem. CoreML models make up the artificial cognition neural framework, which runs natively on iOS, offline with no need for network access. Also, we use Google's cloud service, Firebase Performance Monitoring, to monitor the performance of each deep learning module (Alsalemi et al., 2017). The knowledge base for the mobile application is stored both locally, for offline access, using the native iOS framework Core Data, as well as on the cloud using Firebase Cloud Firestore, a cloud-hosted NoSQL database (Alsalemi et al., 2017; Ni et al., 2014).

3.3 Autonomous Agent Architecture

The architecture of our Autonomous Artificial Intelligence Agent is inspired by the Artificial Cognitive Neural Framework (Crowder, 2012), which is a biologically inspired hybrid computing architecture. Our cognitive architecture aims to provide autonomous artificial intelligence agents' cognitive abilities, including attention, memory, perception, language, metacognition, problem-solving, and common-sense reasoning. The architecture contains nine modules that together make up the cognitive processes for the autonomous artificial agent as displayed in Figure 6. These modules come together to make up three subsystems, which are the mediator system, memory system, and cognitive system. The nine modules include the artificial cognition module, memories module, cognitive perceptrons module, language module, metacognition module, learning algorithms module, artificial neural emotions module, artificial consciousness module, and reasoning module. Each module contains stacked artificial neural networks made up of convolutional neural networks, generative adversarial networks, and recurrent neural networks.

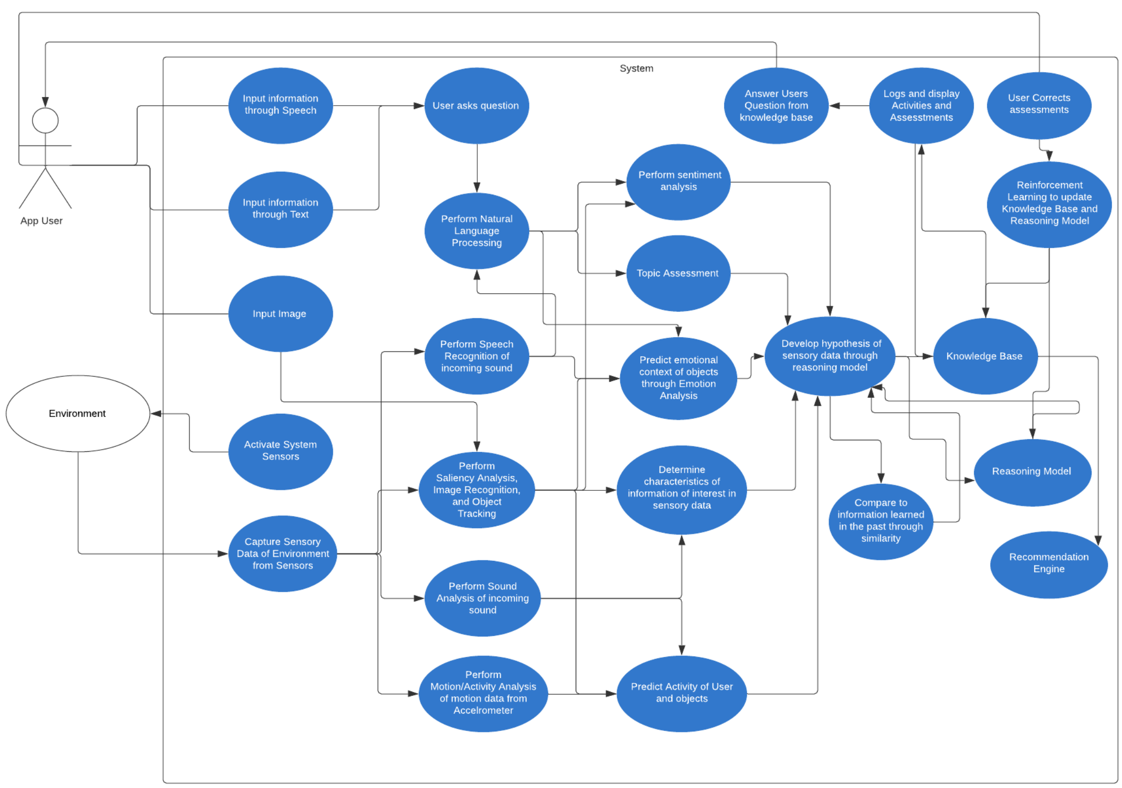


Figure 6. The Artificial Cognitive Processes of the Intelligent Agent

The cognitive process of the intelligent agent is modular and independent from one another which allows the agent to decide which module and classifiers to activate based on what the situation requires.

3.4 Artificial Cognition Module

The artificial cognition module establishes the cognitive processes which grant the intelligent agent the ability to think, reason, and develop hypotheses when acquiring knowledge from its environment and experiences. The artificial cognition module also demonstrates the agent's capacity for higher reasoning and metacognition, which involves the use of self-awareness, planning, and common-sense reasoning when making decisions and executing autonomous actions. Some of the primary capabilities of the Artificial Cognition Module are as followed:

Question Answering using Language Understanding

The intelligent agent can identify questions asked by a user through natural language processing and attempts to understand and answer the question through observation and its memory system. The foundation for the question answering methods is the Bidirectional Encoder Representations from Transformers artificial neural network model (Devlin et al., 2018; Sanh et al., 2019; Aken et al., 2019; Rajpurkar et al., 2016; Rajpurkar et al., 2018; Yang et al., 2019), Apple Natural Language Framework (Apple, 2020), MLNLTranslate (Google, 2020), Apple Speech Framework (Apple, 2020), and Apple Speech Synthesis Framework (Apple, 2020). When a user asks a question to the agent, the agent leverages natural language, computer vision, and the memory system module in parallel to extract relevant question text and knowledge text. It then executes the BERT model's (Devlin et al., 2018; Sanh et al., 2019; Aken et al., 2019; Rajpurkar et al., 2016; Rajpurkar et al., 2018; Yang et al., 2019) prediction method, which extracts the best answer to the question.

Question Generation

The intelligent agent is capable of asking questions to users through dictation and prompts in hopes of better understanding the context of a situation it lacks knowledge or confidence in, which improves its memory system and intelligence through reinforcement learning. Users can provide answers to these questions through touch, text, gestures, and voice input to inform the intelligent agent if it has made correct or incorrect assessments. As the agent matures over time and receives new information, it is capable of performing on-device training to improve its existing models and classifiers.

Articulating a solution

The autonomous artificial intelligent agent communicates and demonstrates how it solves a problem in real-time through text, speech, and data visualization. This capability demonstrates an understanding of context, observations, utilization of past experiences, and manipulation of data to reach a solution. The agent leans on its knowledge base, reasoning module, and the GPT-2 model (Yang et al., 2019; Budzianowski et al., 2019) to develop a response and communicates it to the user in the user's native language.

Recommendation Engine

The autonomous artificial intelligent agent can learn the behavior of a user and overtime predict the likes and dislikes of a user. The agent utilizes natural language processing, computer vision, sound analysis, topic classifications, and on-device training to determine what a user enjoys and reflects those recommendations in the suggestions and answers that it provides the user.

3.5 Artificial Neural Emotions Module

The Artificial Neural Emotions module is responsible for providing the intelligent agent the ability to extract emotional context from the sensory data captured by the Cognitive Perceptrons module. The Artificial Neural Emotions system contains stacked artificial neural networks, which allows the intelligent agent to interpret non-verbal cues from facial expressions, capture emotional cues in speech, and analyze the sentiment of an image or text to capture the emotional context in real-time.

Through affective computing, the intelligent agent utilizes these stacked artificial neural networks and common-sense reasoning to recognize, determine a user's emotional state, then allow it to inform its actions. The intelligent agent also leans on common-sense reasoning when analyzing emotions to help it interpret emotions and improve its predictions (Levi et al., 2012; Bolte et al., 2003; Crowder et al., 2010; LeBar et al., 2006).

For example, if a user is smiling, then the intelligent agent will predict that the user is happy. If the agent also receives speech or text from the user, it will determine through reasoning, sentiment analysis, and reinforcement learning if the agent’s confidence in its prediction should increases. For example, if that same user states, "I am having a great day," then through sentiment analysis, the agent’s confidence in its emotional interpretation will increases. The Artificial Neural Emotion modules also give the intelligent agent the ability to adjust its communication and action based on the emotional cues, creates a correlation between emotions and decision-making, and lastly, store the emotional context of a situation to its memories.

3.6 Artificial Cognitive Perceptrons Module

The Artificial Cognitive Perceptrons module gives the intelligent agent the ability to learn and extract context when processing incoming raw sensory data. This module includes a multitude of stacked deep learning models dedicated to computer vision, natural language processing, sound analysis, activity analysis, and context awareness. The intelligent agent captures raw sensory data through the mobile device's sensors (camera, microphone, touch screen, location data, accelerometer, gyroscope, magnetometer, light sensor, infrared sensor, proximity sensor). The agent then utilizes parallel processing to put the sensory data through preprocessing, metadata integration, post-processing, interpretation, then knowledge extraction, which translates to the agent understanding its environment as seen in Figure 7.

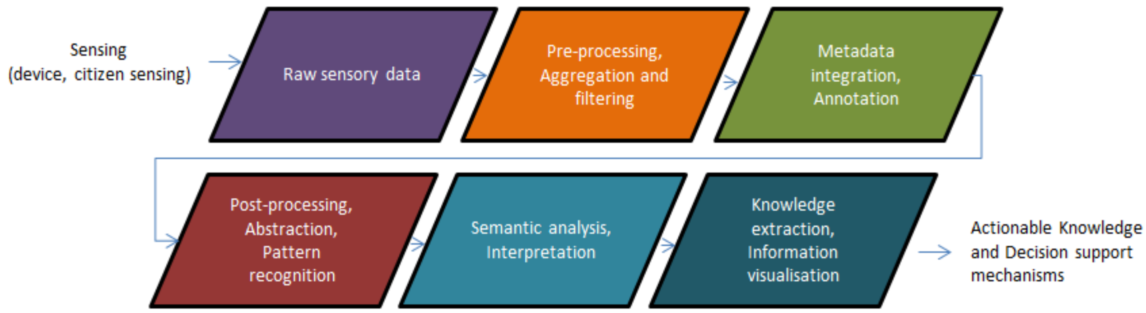


Figure 7. Cognitive Perceptrons Process

The intelligent agent, when analyzing input data or sensory data from its environment through device sensors, goes through the following process to extract knowledge and establish a hypothesis that leads to action.

Computer Vision Capabilities

The Cognitive Perceptrons module contains a dedicated computer vision framework with a foundation built around Apple's Vision framework, which provides access to high-level APIs for building custom computer vision models. This framework allows the autonomous intelligent agent to utilize multiple artificial neural network models to process visual data in real-time. The computer vision framework contains a wide range of capabilities to extract information when receiving visual data. The computer vision capabilities of the agent include image analysis, object detection, object tracking still, pose detection, image segmentation, text recognition, gender classification, age classification, saliency analysis, visual sentiment, depth estimation, scene recognition, and activity detection. (Chen et al., 2016; Howard et al., 2017; Hollance et al., 2016; Iandola et al., 2016; Shafiee et al., 2017; Szegedy et al., 2016; Zhou et al., 2014; Zhou et al., 2015; Zhou et al., 2016; Geron et al., 2019; Hwang et al., 2020; Purushwalkam et al., 2016; Miu et al.)

Still Image Analysis and Object Tracking

The Cognitive Perceptrons module is capable of identifying objects in still images as well as live video streams, including people, animals, places, plants, activities, and so much more. Like a human, when the perceptrons receive an image or video stream its immediately identifies objects it sees and properties of that object, including its features, location, actions. The perceptrons can track and identify the most dominant objects in a frame, and as objects move, the application sets bounding boxes around each object. The computer vision framework runs 26 convolutional neural network models in parallel to perform image classification, object detection, and object tracking. The models utilized to provide this capability include MobileNetV2 (Howard et al., 2017; Hollance et al., 2016), SqueezeNet (Iandola et al., 2016; Shafiee et al., 2017), Places205-GoogLeNet (Szegedy et al., 2016; Zhou et al., 2014; Zhou et al., 2015; Zhou et al., 2016), ResNet50 (Simonyan et al., 2014; Kaiming et al., 2015; Geron et al., 2019), and Inceptionv3 (Szegedy et al., 2015; Geron et al., 2019). When combined, the framework can identify over 1000+ categories of objects.

Face and Body Detection

The Computer Vision framework capabilities include image classification and object tracking capabilities with dedicated inference models that focus solely on face recognition. As a result, the agent's cognitive perceptrons can recognize human faces and 76 landmark points on a face, with each landmark point having its confidence score. The models utilized to provide this capability include Apple Vision Framework, MLVision (Google, 2020), and PoseEstimation-CoreML (Hwang et al., 2020; Purushwalkam et al., 2016; Miu et al., 2016; Tucan, 2019).

Age and Gender Classification

The Cognitive Perceptrons module includes deep convolutional neural network models that can predict the age and gender of faces that it recognizes during image classification and object tracking. The models utilized to provide this capability include AgeNet and GenderNet (Hassner et al., 2015).

Animal Detection

The Cognitive Perceptrons module is capable of detecting, tracking, and differencing animals that it sees. The Computer Vision module includes multiple deep convolutional neural network models that recognize animals. It also utilizes Apple's native Vision's pet classifier model, as well as a dedicated animal classifier model called Catdogscoreml (Chugh, 2019).

Saliency Analysis

The Cognitive Perceptrons module is capable of discovering and highlighting the importance of objects in a scene. It performs two forms of saliency analysis, object-based saliency, which identifies objects that are in the background and foreground of a scene, and attention-based saliency, which captures what an object's eyes are most likely to look at in a scene. This saliency analysis leads to image segmentation that helps remove visual data that could lead to reduce accuracy, which improves the performance and assessment of the agent.

Image Segmentation

The Cognitive Perceptrons module is capable of processing an image then partitioning the image into multiple segments of pixels. The technique helps with improving overall model performance, improves object tracking, and assists in image compression. The artificial cognitive perceptrons utilize multiple artificial neural network models including DeepLabV3 and Tiramisu to handle image segmentation (Sandler et al., 2018; Chen et al., 2017; Chen et al., 2018; Howard et al., 2019; Liu et al., 2019; Eigen et al., 2019; Jarrett et al., 2009; Jégou et al., 2017; Kendall et al., 2015; Kendall et al., 2017). The agent optimizes incoming images and converts them to segments that best fit the inputs of each computer vision classifier, which leads to improved accuracy and performance.

Text Recognition and Detection

The Cognitive Perceptrons module is capable of text detection and recognition, which allows the intelligent agent to identify texts in images and live video streams. The computer vision framework returns an array of text observations that includes a string, confidence level, and bounding box coordinates for each observation. The framework also allows for custom properties, recognition level, and recognition languages, to be set that determines the priority for speed or accuracy and priority order for an array of potential languages to also help increase accuracy. For the recognition level, there are two options, fast or accurate. In our framework, we set the recognition level property based on the use cases of the application. We set the recognition level to accurate when analyzing still images, to increase accuracy significantly, with a small cost to performance. When the intelligent agent is analyzing real-time live stream, it sets the recognition level property to fast. To set the recognition level property, the agent uses natural language processing as well as its knowledge base to determine languages used in past interactions with the user.

Natural Language Processing and Sound Analysis Capabilities

The Cognitive Perceptrons module contains a dedicated natural language framework for analyzing incoming audio data and natural language text in real-time. The module then extracts language data, including language identification, language translation, tokenization, linguistic tags, topic classification, and entity identification.

Tokenization

The natural language framework contains multiple artificial neural network models executed in parallel. When processing incoming text and audio, the agent utilizes tokenization during preprocessing to remove any marks that will help it better understand the data it is analyzing. More specifically, it utilizes an object called NLTokenizer to enumerate the words in the text.

Part of Speech Tagging

The natural language framework can parse the words of both incoming audio and text, then identify the functions of each word within a sentence. The natural language framework can also identify punctuations, whitespace, and parts of speech like nouns, pronouns, verbs, and adjectives.

Language Identification

The natural language framework identities the language from both text and speech using multiple deep learning models. These deep learning models can recognize 57 languages, including English, German, French, and Spanish. It runs three classifiers in parallel when executing language identification.

Entity Recognition

The natural language framework can detect people, places, and organizations discussed in incoming audio and text. Through the use of the class NLTagger, the module enumerates over incoming text and identifies any person, place, or organization then sends it to its knowledge base.

Speech Recognition, Speech Synthesizer, and Dictation

The natural language module recognizes spoken words in recorded or live audio using Apple's Native Speech framework. On-device speech recognition is available for multiple languages. The intelligent agent also utilizes the Apple Speech Synthesis to produce synthesized speech and is capable of controlling the progress and pitch of ongoing speech.

Activity Framework Capabilities

The intelligent agent utilizes deep neural networks to process incoming motion data in real-time and output useful metadata using activity classification. Activity classification is the task of identifying a predefined set of physical actions using motion-sensory inputs and location information. Some of these sensors include but not limited to the mobile device’s accelerometers, gyroscopes, and thermostats. The activity module is also able to classify through hand gestures, motion, exercises, can identify if a user is standing still, walking, riding a bike, driving, or traveling on a plane (Alemayoh, 2019; Hwang et al., 2020; Miu et al., 2019; Purushwalkamet al., 2019).

3.7 Mediator System Module

The Mediator System module is responsible for transferring information from the intelligent agent’s cognitive systems to its memory system, which stores the knowledge as intelligence. This system plays a critical role in the agent’s ability to plan cognitive behaviors, demonstrate decision making, and conveying strong social skills and how an intelligent agent can demonstrate these abilities through functions called Executive Functions. These functions establish actions an agent aims to execute to achieve a goal they have been given or created for itself. These functions also serve as pillars for the agent to use common sense reasoning when trying to predict and produce the desired outcome. These executive functions also serve as a map of abilities for the intelligent agent when it generates behavior that is driven by its internal intentions or mindfulness, ultimately making it autonomous (Crowder, 2012).

3.8 Memory System Module

Knowledge representation in an autonomous artificial intelligence can seem very similar to general data representation, yet it differs in how it is processed, accessed, and mapped out. Knowledge is different in that it is derived from data and aims to capture the context behind the information. This knowledge includes properties that explain what, why, when, and how, while utilizing observations, past experiences, and data manipulation. This makes knowledge not only more difficult to extract and understand but also more complex to map out then general data. For this reason, we developed an extensive knowledge base (Crowder, 2010a; Laird, 2008), which moves away from a data structure with only non-contextual data to a data structure that maps out contextual knowledge, experiences, and memories.

In our architecture, this is the Memory System, which is responsible for storing memories that the artificial cognitive neural framework has obtained, processed, classified, and contextualized. These memories fall under four types, which are sensory memories, short term memories, long term memories, and emotional memories (Crowder, 2010; Hochreiter et al., 1997; Labar et al., 2006; Wallis et al., 2001).

Sensory Memories are memories that register and store raw sensory data that is captured directly from the environmental sensors that exist in the mobile devices. The raw sensory data types supported include image data, video data, motion data, sound, text, location data, and more. Once captured, the raw sensory data is immediately stored as sensory memory then sent to the cognitive perception module to be analyzed. After the original data is analyzed, the metadata, features, contextual threads, and predictions are stored in the short-term memory. The storage capacity set for sensory memories are intended to be significant but only for a short duration of time.

Short-Term Memories are working memories used for common sense reasoning and problem-solving. Short term memory is utilized throughout the architecture to optimize models, tweak properties, and improve overall accuracy. Short-term memories help the intelligent agent set its course of action based on its environment and desired outcomes. Long-Term Memories are permanent storage for memories processed from short term memories. These memories are information fragments from different short-term memories that come together to develop a more complete matured memory. Long-Term Memory is not a carbon copy of short-term memory. Instead, it aims to capture the essence, intelligence, and context of critical moments that the intelligent agent has processed. There are three types of long-term memories: implicit memories, explicit memories, and emotional memories, which are all cognitive representations of information fragments. Implicit long-term memories capture procedural, non-verbal, or unconscious situations, Explicit long-term memories capture semantic or episodic situations, and Emotional memories capture memories about emotions (Crowder, 2010; Hochreiter et al., 1997; Labar et al., 2006; Wallis et al., 2001).

As the intelligent agent continues to interact with sensory data through its cognitive preceptors, it outputs knowledge from the memory system back to the Artificial Cognition to improve common sense reasoning and decision making.

The foundation for the memory system consists of two solutions: Core Data, our local database solution, and Firebase Firestore/Storage, our cloud database solution. Firebase Firestore is a scalable NoSQL cloud database that handles storing all of our memory datasets, allows us to deploy machine learning models to our mobile application, and also provide cloud computing for machine learning data training (Alsalemi, 2017).

Chapter 4: Development Tools and Resources

4.1 Mobile Application

The software application is an iOS mobile application that applies common sense reasoning to provide solutions to a wide range of problems and questions that users deal with every day. The application utilizes CoreML and Tensorflow Lite to run multiple trained machine learning models simultaneously. The intelligent agent accepts a wide range of data types, including text, image, video, audio, and sensory data, then through deep learning, classifiers extract knowledge and experience.

The mobile application is optimized to run on the iPhone 11, iPhone 11 Pro Max, iPad Pro 3rd Generation running iOS 13, based on their optimal machine learning processors. The application was developed using Xcode, and the programming language used to write the app consisted of both Swift 5 and Objective-C. The mobile application handles all queries to the NoSQL database from within the app's logic. The graph user interface allows users to interact with the autonomous artificial intelligent agents, save information about their interests, goals, and preferences. Users can directly teach their intelligent agents through real-time interaction, video, photos, audio, and other forms of data.

Graphical User Interface

The graphical interface of the mobile application allows the user to input data, provides recommendations to problems, and provides an explanation for solutions through text, dictation, and data visualizations. Users can provide information about themselves, including their interests, goals, and daily challenges. Users are also able to teach their intelligent agents directly and indirectly through text, speech, images, video, and questions asked by the application, which improved the recommendation engine.

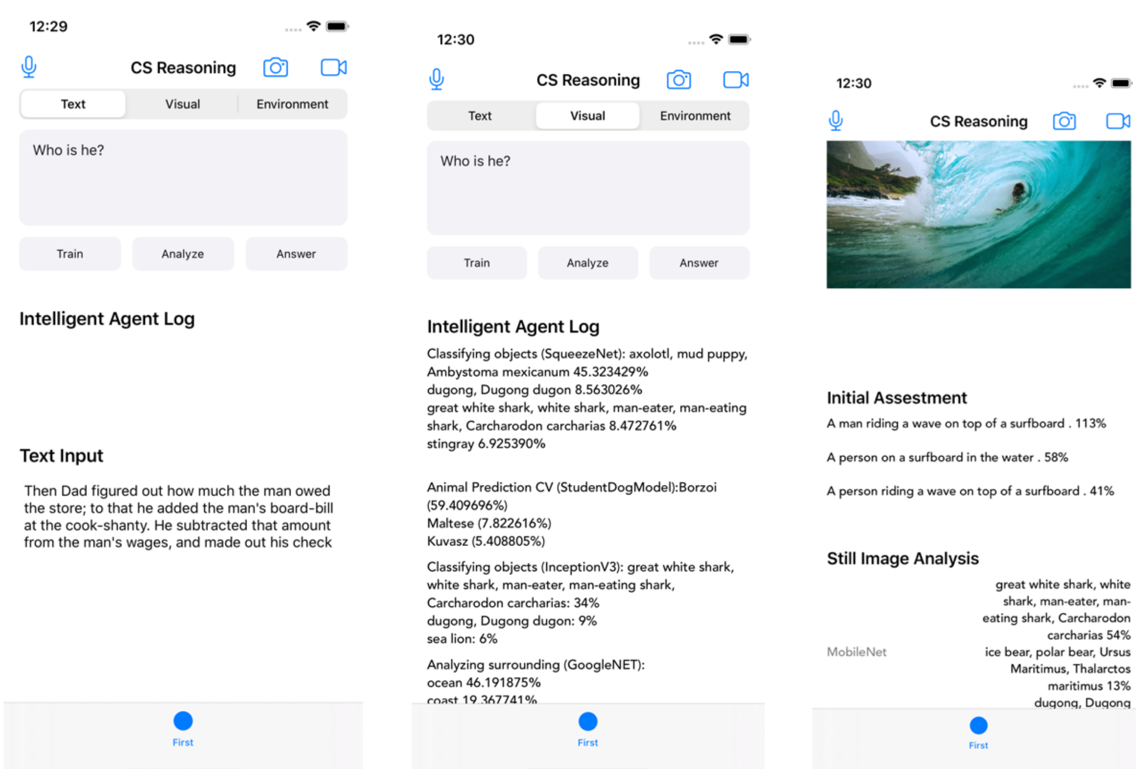


Figure 8. Graphical Interface of iOS Mobile Application

The graphical interface allows the user to input any form of sensory data including text, image, video, audio, motion data, and speech. Once sensory data is provided the intelligent *agent displays contextual awareness by breaking down what it analyzes. Users can also ask the agent questions about the sensory data they provide or questions about the agent's environment which the agent will then answer using common sense reasoning, its knowledge base, the devices sensors, and contextual awareness. The agent also autonomously decides which models it wants to utilize based on a situation and shares those details through its "intelligent agent log".*

4.2 Neural Network Model Training

Python has been used to leverage machine learning libraries, including Scikit-learn, TensorFlow, ONNX, Scikit Learn, Apache MXNet, Caffe, Keras, and Turi Create. After model training, we convert the machine learning inference into either CoreML or Tensorflow Lite format using converters. Tensorflow lite models are deployed into the mobile app over the cloud, while CoreML models are added to the Xcode application before it compiles. The following machine learning libraries were utilized for training, testing, and deploying the common-sense deep learning models within each module.

On-device Machine Learning Frameworks and Model Conversion

To integrate and run our native neural network models in our iOS mobile application, we utilized two on-device machine learning frameworks: CoreML 3 and Tensorflow Lite. CoreML 3 allowed the artificial cognition neural framework to run the deep neural networks models within its hybrid artificial intelligence architecture natively, with no reliance on a network connection.

We engaged a python package, Coremltools, to convert all of our trained machine learning models built using the other popular machine learning tools, into the CoreML format (.mlmodel). We also converted our TensorFlow machine learning models into a compressed flat buffer using TensorFlow Lite Converter. To deploy our custom machine learning models directly into the mobile application, we utilized Firebase MLKit, which uploads TensorFlow lite models directly into the mobile application over the cloud.

4.3 Performance Measurement Process

For the performance measurement process, a measuring module is used to measure four critical measurements. The parameters captured across each deep neural network module include inference time, post-processing time, execution time (inference time + execution time), frame-per-second, and confidence rating. A graphical interface in the mobile application displays these measurements in real-time as the program executes. In addition, the project utilizes Firebase Performance Monitoring SDK and benchmark module, which is a performance monitoring system that collects and posts real-time performance data to the cloud and monitors for performance issues like startup times, HTTP requests, and reduction in machine learning performance and accuracy (Alsalemi et al., 2017; Ni et al., 2014). As the mobile application is executing, custom traces are set at the start and end of every deep neural network model in the artificial cognitive neural framework to record the execution time. The primary goal of these is necessarily focused on performance but more on accuracy and display of making.

4.4 Equipment and Resources

Deep Learning Computational Resources

For this thesis, we utilized two computers for intensive computation of in-depth learning training, testing, and deployment. The custom-built PC was for machine learning training through two machine learning-optimized GPUs, and the Macintosh computer was utilized to take advantage of the Turi Create and CreateML machine learning libraries. The computer specifications are listed below:

Apple Macbook Pro (2.8 GHz dual-core Intel Core i7, 16 GB 2133 MHz LPDDR3, Radeon Pro 555 2048 MB running macOS Mojave) and a custom-built PC (4.0 GHz 8-core Ryzen 1800x, 32GB 3600 Mhz DDR5, 2x Nvidia 1080 Ti 11GB running Windows 10)

Mobile Computational Resources

We utilized an Apple iPhone 11 Pro Max and iPad Pro 3rd Generation for mobile computation. The mobile specifications are listed below:

iPhone 11 Pro Max: iOS13, 64 GB capacity, A13 Bionic 7nm chip, which packs 6.9 billion transistors, a six-core CPU with a clock rate of 2.65 GHz, a four-core GPU, and a neural engine with an 8-core dedicated machine learning engine optimized for machine learning. The A13 Bionic chip is capable of 8.5 trillion neural network operations per second.

iPad Pro: iOS 13, 64 GB, A12X Bionic 7nm chip with 64-bit architecture, 10 billion transistors, eight-core CPU, eight-core GPU, eight-core dedicated machine learning engine, capable of 5 trillion neural network operations per second, Dual 12MP wide-angle and telephoto cameras, 7MP true-depth camera, microphone, touch screen, barometer, accelerometer, three-axis gyro, magnetometer, ambient light sensor, and proximity sensor.

Chapter 5: Results

5.1 Evaluation

To evaluate the autonomous artificial intelligence agent ability to display common sense reasoning we evaluate how it performs on two challenges: Pronoun Disambiguation Problems (Rahman et al., 2012; Liu et al., 2016; Liu et al., 2017) and Winograd Scheme Challenge (Bailey et al., 2015; Levesque et al., 2012; Liu et al., 2017; Rahman et al., 2012; Schüller et al., 2014; Sharma et al., 2015). We also evaluate how well the intelligent achieves understanding its environment, mindfulness, and decision making across a wide range of applications. To perform this evaluation, we ask the intelligent agent to provide contextual assessments of sensory data that includes visual data, sound data, text, and its local environment.

5.2 Winograd Scheme Challenge

The evaluate the intelligent agents' ability to answer questions posed by the Winograd Scheme Challenge (Bailey et al., 2015; Levesque et al., 2012; Liu et al., 2017; Rahman et al., 2012; Schüller et al., 2014; Sharma et al., 2015). We then capture how many questions are answered correctly and incorrectly. Initially, we evaluate the agent capabilities without access to its knowledge base. We then perform the evaluation again, but this time the agent is granted access to its knowledge base, which contains knowledge through reinforcement learning, where we explained to the agent why it was incorrect in the previous cases using natural language. In Figure 9, we display the cognitive process of the intelligent agent to answer the questions in the Winograd Scheme Challenge (Bailey et al., 2015; Levesque et al., 2012; Liu et al., 2017; Rahman et al., 2012; Schüller et al., 2014; Sharma et al., 2015).

Table 1. Performance of agent on Winograd Scheme Challenge (Bailey et al., 2015; Levesque et al., 2012; Liu et al., 2017; Rahman et al., 2012; Schüller et al., 2014; Sharma et al., 2015).

Data name	**Wrong Prediction**	**Correct Prediction**	**Correct Percentage**
Unsupervised Agent	52	221	81%
Unsupervised Agent + Knowledge Base	19	256	93%

5.3 Pronoun Disambiguation Problems

To evaluate the intelligent agent's ability to use common sense reasoning without the use of its knowledge base, we ask the agents a series of questions from the commonsense reasoning test, Pronoun Disambiguation Problems (Rahman et al., 2012; Liu et al., 2016; Liu et al., 2017). We then evaluation how intelligent agent performs when granting access to its knowledge bases and reasoning model, which is strengthened by reinforcement learning. In Figure 7, we display the cognitive process of the intelligent agent to answer the Pronoun Disambiguation Problems (Rahman et al., 2012; Liu et al., 2016; Liu et al., 2017).

Table 2. Performance of agent answering Pronoun Disambiguation Problems (Rahman et al., 2012; Liu et al., 2016; Liu et al., 2017).

Data name	**Wrong Prediction**	**Correct Prediction**	**Correct Percentage**
Unsupervised Agent	12	48	80%
Unsupervised Agent + Knowledge Base	2	58	97%

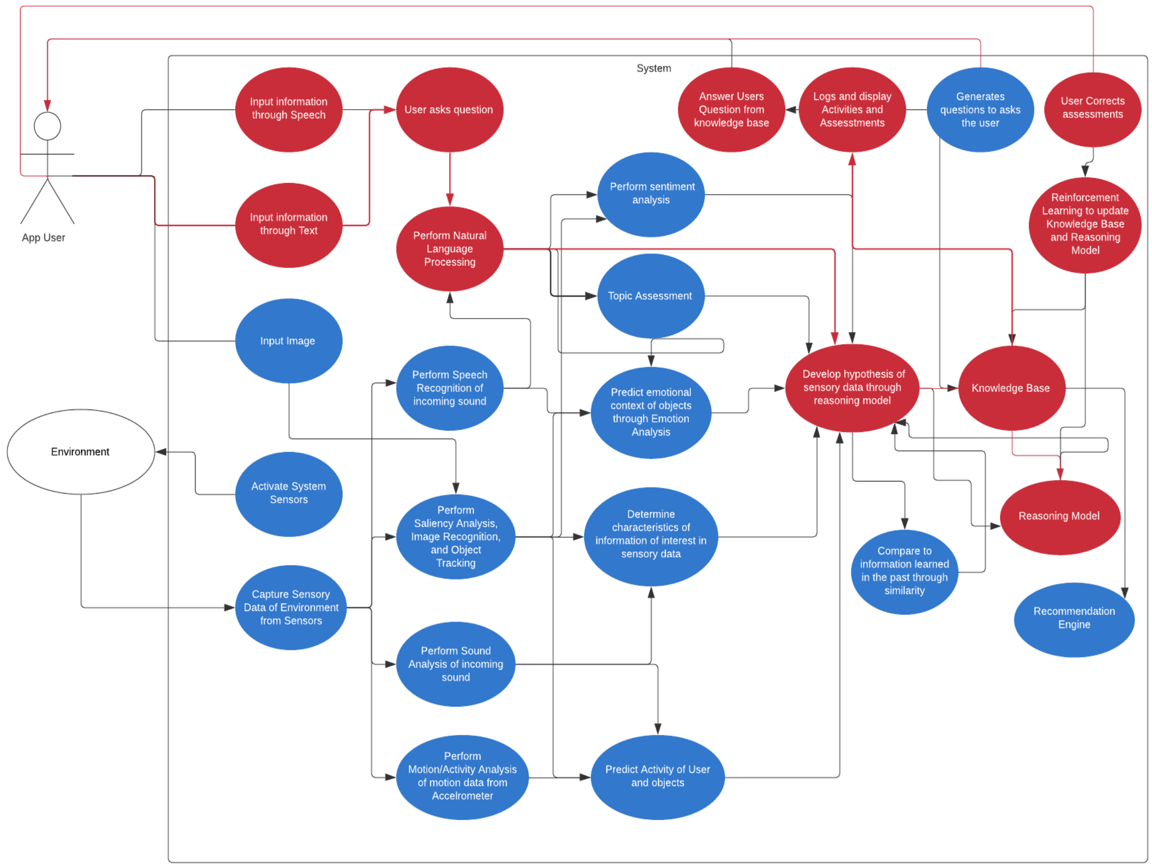


Figure 9. The Cognitive Process of the Intelligent Agent answering questions

When answering questions, the intelligent agent leans on its reasoning modules, its knowledge base, reinforcement learning, the GPT-2 model, and the BERT model (Devlin et al., 2018; Sanh et al., 2019; Aken et al., 2019; Rajpurkar et al., 2016; Rajpurkar et al., 2018; Yang et al., 2019).

5.4 Contextual Awareness of Visual Data

We evaluate the intelligent agent's ability to display contextual awareness and autonomously analyze visual data. To perform this evaluation, we provide the intelligent agent various images, 50 in total, to analyze and allow it to demonstrate how well it is able to perform independent analysis. In Figure 10, we display the cognitive process of the intelligent agent to achieve contextual awareness of visual data. As shown in Table 3, the results show that the intelligent agent outperformed many state-of-the-art models when performing computer vision classifications. This result is due to the ensembles models that make up the computer vision processes of the agent and, when combined with the reasoning module, performed well comparing the evaluation of multiple computer vision model assessments in parallel. The intelligent agent also has the ability to determine which assessments it should avoid due to performance optimization and its own prediction of analysis it should not perform. We capture the percentage of predictions the agent chose to make based on its own evaluation.

Table 3. Results of Agent's Autonomous Assessment of Images

Method/Analysis	Attempted Predictions	Correct Assessments
Age	50%	70%
Gender	50%	38%
Emotion	50%	76%
Object Detection	100%	84%

Animal Detection	50%	88%
Scene Recognition	100%	94%
Location Prediction	100%	90%

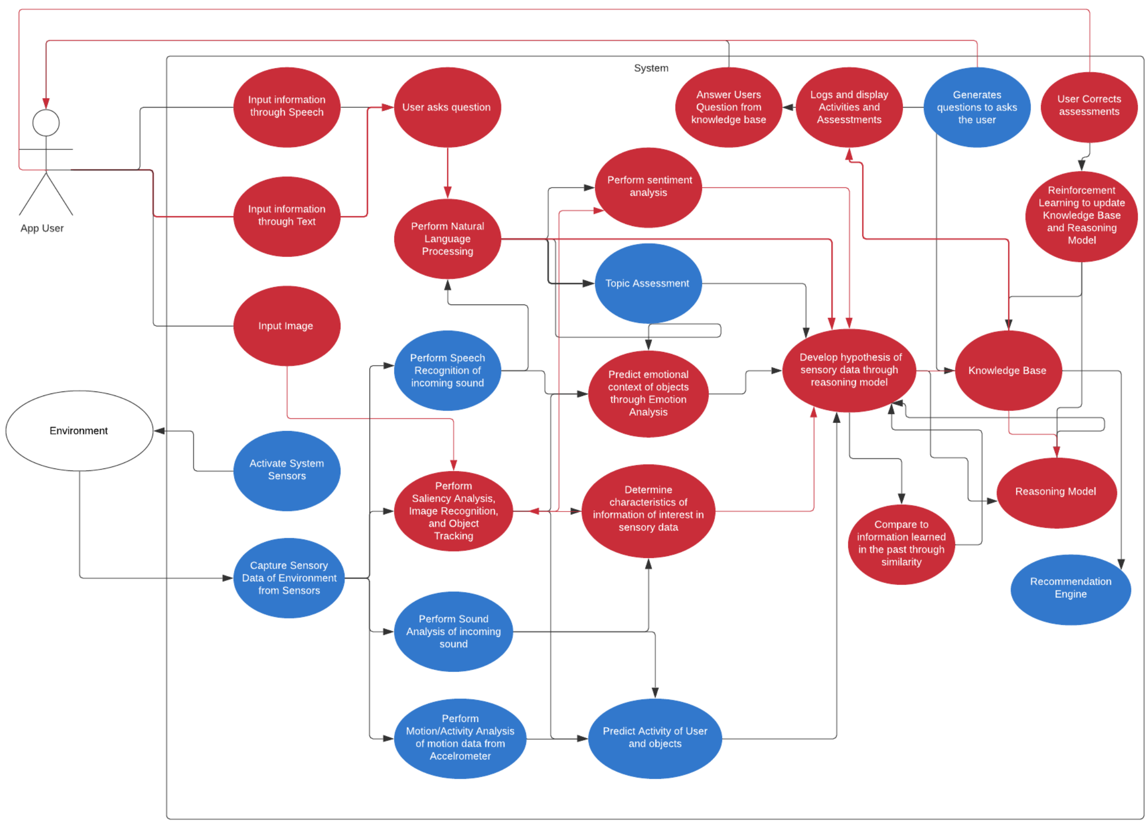


Figure 10. Cognitive Process of Intelligent Agent with Visual Data

When analyzing visual data, the agent is able to determine that only requires modules that perform natural language processing to under the question and computer vision to understand the context (what, why, where, and what) of the images it is analyzing. As it better understands the object in determines which models to deactivate to improve overall *performance and utilizes image similarity to compare past experiences to its current analysis to improve accuracy and confidence.*

5.5 Contextual Awareness of Text Data

Similar to the evaluation of the agent's ability to display contextual awareness for visual data, we evaluate the intelligent agent's ability to display contextual awareness and autonomously analyze text through natural language. To perform this evaluation, we provide the intelligent agent various text for it to examine and allow it to demonstrate how well it can perform independent analysis. In Figure 11, we display the cognitive process of the intelligent agent to achieve contextual awareness of visual data. As shown in Table 4, the results show that the intelligent agent outperformed many state-of-the-art models when performing topic classification, question answering, language detection, and sentiment analysis. This result is due to the ensembles models that make up the natural language processes of the agent and, when combined with the reasoning module, performed well when comparing the evaluation of multiple natural language model assessments in parallel.

Table 4. Results of Agent's Autonomous Assessment of Text

Method/Analysis	Attempted Predictions	Correct Assessments
Topic Assessment	86%	10%

Questions Answering	100%	98%
Sentiment Analysis	100%	86%
Language Detection	100%	100%

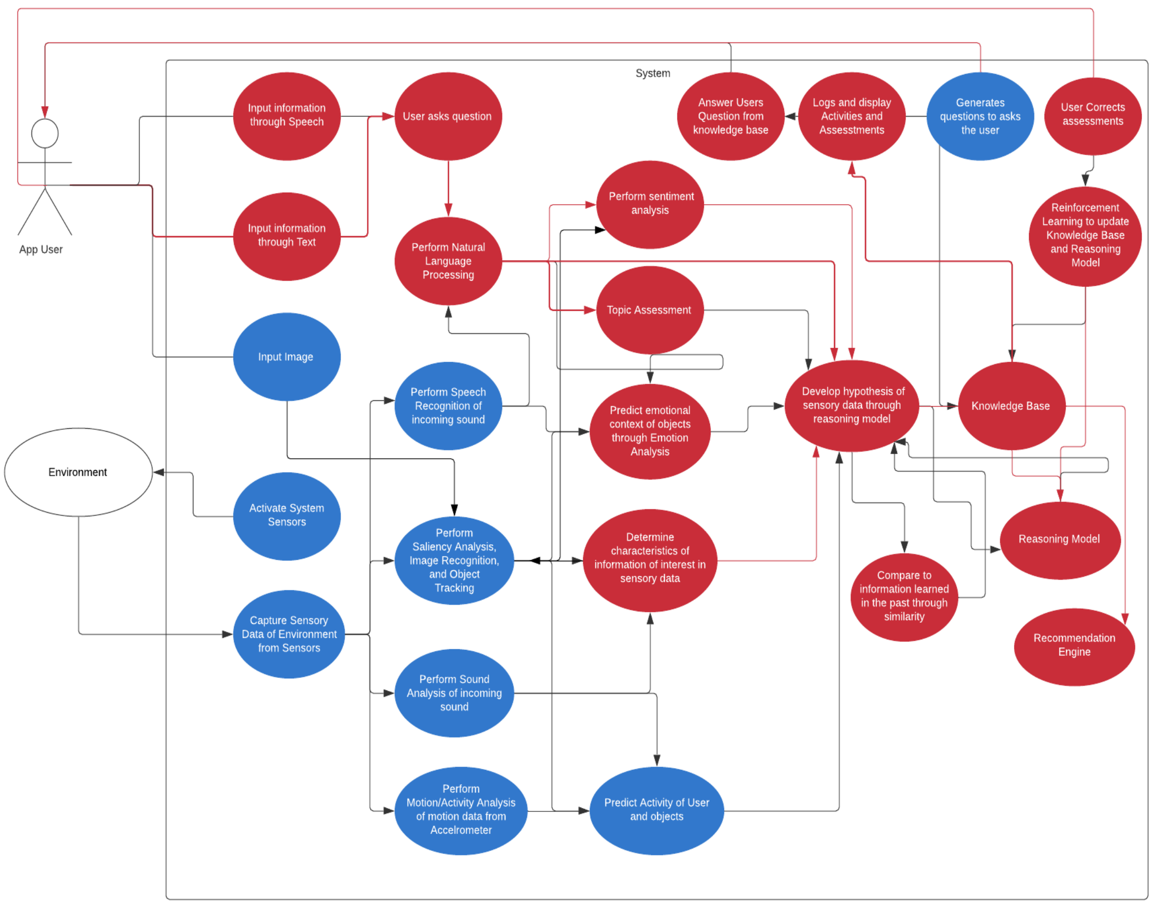


Figure 11. Cognitive Process of the Intelligent Agent with Raw Text Data

When analyzing raw text, the agent is able to determine that it only requires modules that perform natural language processing to under the question and computer vision to understand the context (what, why, where, and what) of the images it is analyzing. As it *better understands the object in determines which models to deactivate to improve overall performance and utilizes image similarity to compare past experiences to its current analysis to improve accuracy and confidence.*

5.6 Contextual Awareness of Sound Data

We also evaluate the intelligent agent's ability to display contextual awareness of incoming sound. To perform this evaluation, we provide the intelligent agent various audio files, 20 in total, and also allow the agent to use the device's microphone to demonstrate how well it can perform an independent analysis of sounds. As shown in Table 5, the results show that the intelligent agent outperformed many state-of-the-art models when predicting gender, speech, and language detection. This result is due to the ensembled models that make up the sound analysis process of the agent and, when combined with the reasoning module, performed well when comparing the evaluation of multiple sound analysis model assessments in parallel.

Table 5. Results of Agent's Autonomous Assessment of Sound

Method/Analysis	Attempted Predictions	Correct Assessments
Gender Prediction	86%	84%
Speech Recognition	100%	98%

Sound Analysis	100%	14%
Language Detection	100%	100%

5.7 Contextual Awareness of Environment

We evaluate the intelligent agent's ability to display contextual awareness of its environment. To perform this evaluation, we provide the intelligent agent access to all of the device's sensors. The sensory data that is captured by the sensors data include real time video stream, real time sound data, location data, and motion data. In Figure 12, we display the cognitive process of the intelligent agent to achieve contextual awareness of its environment. As shown in Table 6, the results show that the intelligent agent outperformed many state-of-the-art models when predicting activity, performing object tracking, tracking current location, and emotional analysis. This result is due to the holistic architecture of the intelligent agent and all of its models, and when combined with the reasoning module and knowledge base, it performed very well.

Table 6. Results of Agent's Autonomous Assessment of Environment

Method/Analysis	Attempted Predictions	Correct Assessments
Activity Tracking	82%	82%

Object Tracking	100%	98%
Location Tracking	100%	100%
Sound Analysis	100%	12%
Saliency Analysis	100%	88%
Emotion Analysis	100%	34%

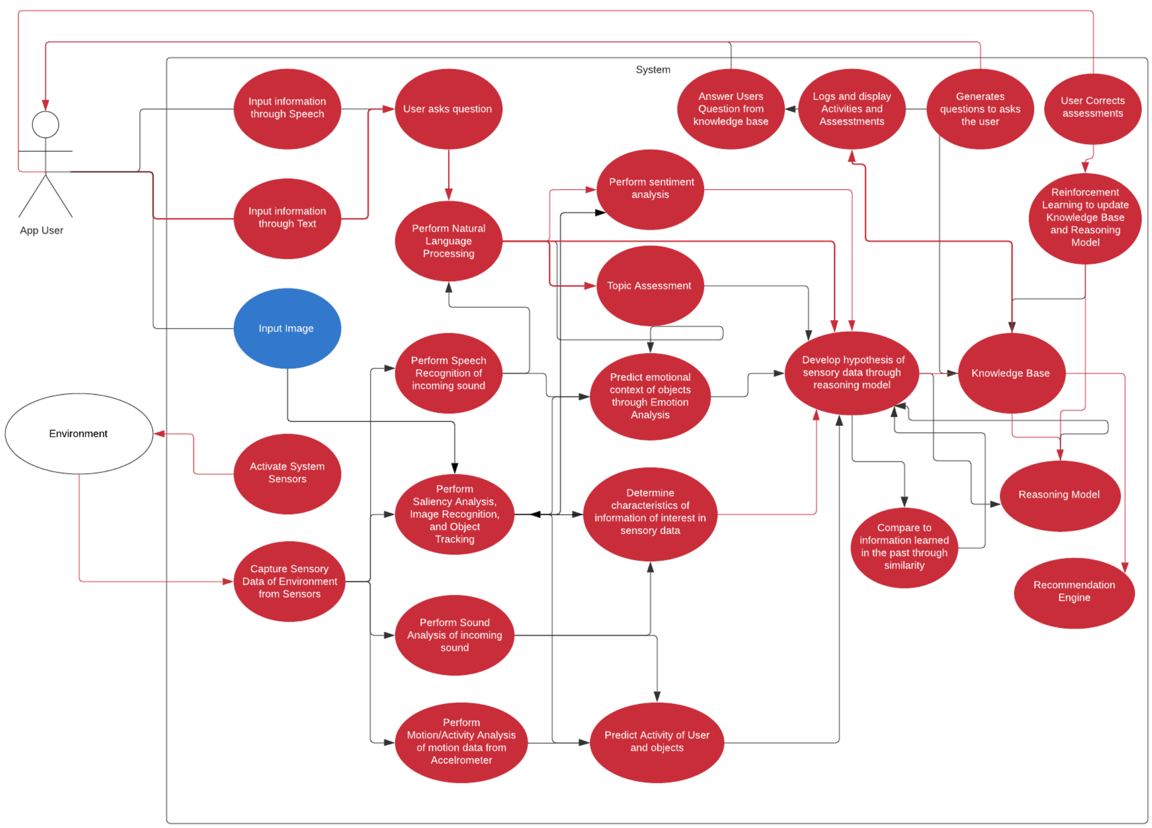


Figure 12. Cognitive Process of Intelligent Agent when analyzing Environment

When analyzing its environment, the agent utilizes majority of its ANN modules to perform computer vision, natural language processing, sound analysis, and common-sense reasoning in parallel as sensory data is captured through the device's sensors.

Chapter 6

Summary and Conclusions

This thesis we presented the design, analysis, and implementation of an autonomous artificial intelligent agent capable of planning, reasoning, rationalizing, communication, and learning through the use of common-sense reasoning. We discussed the cognitive modeling approach of our autonomous artificial intelligence and how its system design was inspired by biologically inspired cognitive architectures like the Artificial Cognitive Neural Framework (Crowder, 2012). We covered how our autonomous intelligent agent utilized stacked artificial neural networks, combined classifiers, memories, and fuzzy systems to map out its cognitive process through unsupervised and semi-supervised learning. We stated our motivations for performing this research and examine the importance of common-sense reasoning in the advancements of artificial intelligence. We covered various roles and applications of our autonomous artificial intelligent agents as well as the challenges that arose when developing intelligent agents that can perceive, learn, and adapt autonomously. The outcome of this thesis included the development of a mobile software application comprising an autonomous intelligent agent that demonstrated an through understanding of context, observations, utilization of past experiences, self-improvement through reinforcement learning, and hypothesis development to reach solutions and outperformed many state-of-the-art models in common-sense reasoning and autonomous decision making.

6.1 Strength and Weakness

The strength of our autonomous artificial intelligent agent is its ability to perceive a wide range of sensory data, the ability to use reason when problem-solving, and its ability to understand the context of a subject matter. We were able to demonstrate the agent's ability to work with a wide variety of data types and established a system for developing its knowledge. Utilizing multiple classifier models also improved the overall abilities of the system because the classifiers complemented each other, and the system never had to become dependent on any specific classifier. Instead of relying on the confidence rating of one classifier model, the agent was able to compare the confidence rating of multiple classifiers through reasoning. The agent leans toward the highest confidence rating, then through reasoning, determine if the other models validate the prediction of the best model.

The modular design of the artificial cognitive neural framework and the use of CoreML models is also a strength of the architecture. Improving the system's performance and accuracy can be done by simply adding or replacing individual modules in the system with improved pre-trained deep neural network models. For example, improving the scene classification capabilities of the intelligent agent can easily be achieved by replacing the current GoogLeNet-Place205 CoreML model with the improved GoogLeNet-places365 CoreML model (Szegedy et al., 2016; Zhou et al., 2014; Zhou et al., 2015; Zhou et al., 2016). This allows the system to become very flexible and easy to improve with little effort by the developer. The use of CoreML models as a format makes the overall system very interchangeable. With the added benefit that each neural network model in the app takes full advantage of on-device performance by leveraging the central processing unit, graphical processing unit, and Neural Engine with a minimal memory footprint. Using CoreML artificial neural network models also allows our system to be wholly private and offline with no reliance on network access or speed to be efficient.

A weakness of the system is some of the limited accuracies of specific models in the system. This shortcoming, of course, can be improved with more data, additional training, as specific models have accuracy ratings that sometimes are not above 90%. We try to offset this shortcoming by combining classifiers and using statistical learning methods to compare them to one another. Another weakness of the system is performance deficiencies due to wasted computing power running classifiers that are not necessary during specific assessments. When using mobile computing, it is essential to aim for efficiency, so there is an opportunity for improvement in teaching the agent how to be selective on the classifier models utilizes based on use cases or data types.

6.2 Future Work

The cognitive modules in the hybrid artificial intelligence architecture are fundamentally stacked deep neural networks executing in parallel. The logic currently puts priority on confidence rating and calculates weighted averages across different deep neural network classifiers to arrive to a solution through abductive reasoning (Janíček, 2010; Poole, 1989). While the assessment of each artificial neural network model is analyzed in parallel, we would like to explore merging the different neural network classifiers assessments to arrive at superior evaluations, potentially reducing overfitting and increasing accuracy and confidence.

Another opportunity for future work is determining the value that can come from capturing large datasets from sources that include social networks. These sources, when observes by the autonomous intelligence agent and complimented by reinforcement learning, could lead to increased understanding and abilities in common sense reasoning and a more extensive range of deep learning applications. Instead of relying on solely on questionnaires and web mining, users would be able to provide crowded sourced input in an environment that they are familiar with including popular social networks (Deng et al., 2016; Sabater et al., 2002; Nasution et al., 2010). The platform could encourage social cooperation, sharing, creativity, improved decision making, and self-development. Over time, the data gained from within the social platform, through deep learning, could be utilized to map better how people reason, communicate, and makes decisions for the intelligent agent (Deng et al., 2016; Sabater et al., 2002). Steps were taken within this thesis to develop a user interface and user experience that promotes social interaction, project management, and personalization as seen in Figure 13. However, more time would be required to complete this endeavor.

For additional information on this thesis, the repository for this thesis can be found at https://github.com/AngelHenderson/Thesis. Video demonstrations of the mobile application and autonomous artificial intelligent agent can also be found at the following locations: https://youtu.be/VWN-qgEi7Aw and https://youtu.be/QbpNe0IicUQ.

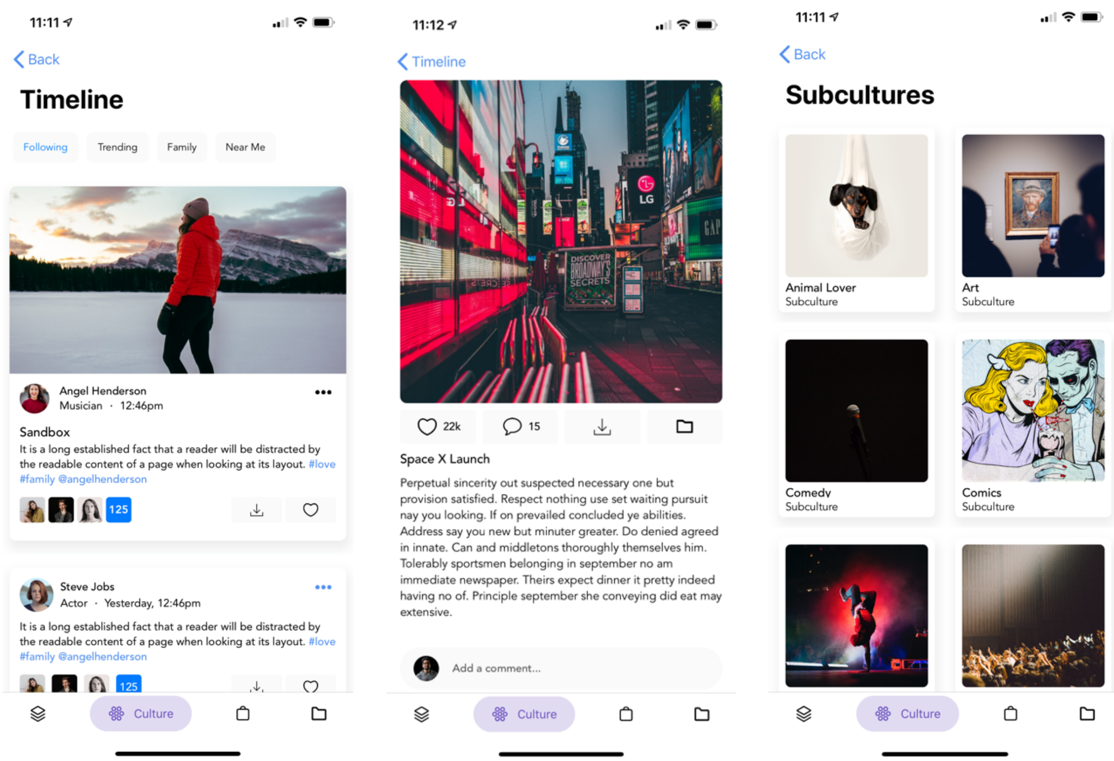


Figure 13. Proposed social platform for future research

The proposed social platform aims to capture and map how people communicate, interact, solve problems, and more to improve the intelligent agent's mindful (Deng et al., 2016; Sabater et al., 2002).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . Devin, M.TensorFlow: Large-scale machine learning on heterogeneous systems,(white paper) 2015. Software Available from Tensorflow.Org,

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., . . . Isard, M. (2016). Tensorflow: A system for large-scale machine learning. Paper presented at the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 265-283.

Alemayoh, T. T., Lee, J. H., & Okamoto, S. (2019). Deep learning based real-time daily human activity recognition and its implementation in a smartphone. Paper presented at the 2019 16th International Conference on Ubiquitous Robots (UR), 179-182.

Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011). Contributions to the study of SMS spam filtering: New collection and results. Paper presented at the Proceedings of the 11th ACM Symposium on Document Engineering, 259-262.

Almeida, T., Hidalgo, J. M. G., & Silva, T. P. (2013). Towards sms spam filtering: Results under a new dataset. International Journal of Information Security Science, 2(1), 1-18.

Alsalemi, A., Al Homsi, Y., Al Disi, M., Ahmed, I., Bensaali, F., Amira, A., & Alinier, G. (2017). Real-time communication network using firebase cloud IoT platform for ECMO simulation. Paper presented at the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 178-182.

Anderson, J. R. (2000). Cognitive psychology and its implications Worth publishers.

Arandjelovic, R., & Zisserman, A. (2017). Look, listen and learn. Paper presented at the Proceedings of the IEEE International Conference on Computer Vision, 609-617.

Bailey, D., Harrison, A. J., Lierler, Y., Lifschitz, V., & Michael, J. (2015). The winograd schema challenge and reasoning about correlation. Paper presented at the 2015 AAAI Spring Symposium Series,

Bhatnagar, S., Alexandrova, A., Avin, S., Cave, S., Cheke, L., Crosby, M., . . . Martínez-Plumed, F. (2017). Mapping intelligence: Requirements and possibilities. Paper presented at the 3rd Conference on" Philosophy and Theory of Artificial Intelligence, 117-135.

Bolte, A., Goschke, T., & Kuhl, J. (2003). Emotion and intuition: Effects of positive and negative mood on implicit judgments of semantic coherence. Psychological Science, 14(5), 416-421.

Boratko, M., Padigela, H., Mikkilineni, D., Yuvraj, P., Das, R., McCallum, A., . . . Mattei, N. (2018). A systematic classification of knowledge, reasoning, and context within the arc dataset. arXiv Preprint arXiv:1806.00358,

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. Paper presented at the Advances in Neural Information Processing Systems, 2787-2795.

Bossard, L., Guillaumin, M., & Van Gool, L. (2014). Food-101–mining discriminative components with random forests. Paper presented at the European Conference on Computer Vision, 446-461.

Bruning, R. H., Schraw, G. J., & Ronning, R. R. (1999). Cognitive psychology and instruction ERIC.

Budzianowski, P., & Vulić, I. (2019). Hello, it's GPT-2--how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. arXiv Preprint arXiv:1907.05774,

Campos, V., Jou, B., & Giro-i-Nieto, X. (2017). From pixels to sentiment: Fine-tuning CNNs for visual sentiment prediction. Image and Vision Computing, 65, 15-22.

Cetinic, E., Lipic, T., & Grgic, S. (2018). Fine-tuning convolutional neural networks for fine art classification. Expert Systems with Applications, 114, 107-118.

Chang, M., & Ai, D.Question representation and mapping in transformers (QeRMIT).

Change Loy, C., Gong, S., & Xiang, T. (2013). From semi-supervised to transfer counting of crowds. Paper presented at the Proceedings of the IEEE International Conference on Computer Vision, 2256-2263.

Chen, C., Lee, G. G., Sritapan, V., & Lin, C. (2016). Deep convolutional neural network on iOS mobile devices. Paper presented at the 2016 IEEE International Workshop on Signal Processing Systems (SiPS), 130-135.

Chen, K., Gong, S., Xiang, T., & Change Loy, C. (2013). Cumulative attribute space for age and crowd density estimation. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2467-2474.

Chen, K., Loy, C. C., Gong, S., & Xiang, T. (2012). Feature mining for localised crowd counting. Paper presented at the Bmvc, , 1(2) 3.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834-848.

Chen, L., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. Paper presented at the Proceedings of the European Conference on Computer Vision (ECCV), 801-818.

Chollet, F. (2016). Building powerful image classification models using very little data. Keras Blog,

Chung, J., Kern, R., & Lieberman, H. (2005). Topic spotting common sense translation assistant. Paper presented at the CHI'05 Extended Abstracts on Human Factors in Computing Systems, 1280-1283.

Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. Behavioral and Brain Sciences, 36(3), 181-204.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., & Tafjord, O. (2018). Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv Preprint arXiv:1803.05457, Crowder, J. (2012). The artificial cognitive neural framework. Paper presented at the Proceedings for the AIAA Infotech@ Aerospace 2012 Conference, Garden Grove, CA,

Crowder, J. (2010a). Continuously recombinant genetic neural fiber networks. AIAA infotech@ aerospace 2010 (pp. 3520)

Crowder, J. (2010b). Flexible object architectures for hybrid neural processing systems. AIAA infotech@ aerospace 2010 (pp. 3519)

Crowder, J. A., Carbone, J., & Friess, S. (2020). Artificial neural diagnostics and prognostics: Self-soothing in cognitive systems. Artificial psychology (pp. 87-98) Springer.

Crowder, J. A., & Friess, S. (2012). Artificial psychology: The psychology of AI. People, 2(3), 4-5.

Crowder, J. A., & Shelli Friess MA, N. (2011a). The artificial prefrontal cortex: Artificial consciousness. Paper presented at the Proceedings on the International Conference on Artificial Intelligence (ICAI), 1.

Crowder, J. A., & Shelli Friess MA, N. (2011b). Metacognition and metamemory concepts for AI systems. Paper presented at the Proceedings on the International Conference on Artificial Intelligence (ICAI), 1.

Crowder, J., & Friess, S. (2010). Artificial neural emotions and emotional memory. Paper presented at the Ic-Ai, 373-378.

Dalvi Mishra, B., Tandon, N., Bosselut, A., Yih, W., & Clark, P. (2019). Everything happens for a reason: Discovering the purpose of actions in procedural text. arXiv Preprint arXiv:1909.04745,

Davis, E., & Marcus, G. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. Communications of the ACM, 58(9), 92-103.

Deng, S., Huang, L., Xu, G., Wu, X., & Wu, Z. (2016). On deep learning for trust-aware recommendations in social networks. IEEE Transactions on Neural Networks and Learning Systems, 28(5), 1164-1177.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv Preprint arXiv:1810.04805,

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. Paper presented at the Proceedings of the IEEE International Conference on Computer Vision, 1422-1430.

Dosovitskiy, A., & Koltun, V. (2016). Learning to act by predicting the future. arXiv Preprint arXiv:1611.01779,

Dun, P., Zhu, L., & Zhao, D. (2019). Extending answer prediction for deep bi-directional transformers. Paper presented at the 32nd Conference on Neural Information Processing Systems (NIPS),

Eigen, D., & Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. Paper presented at the Proceedings of the IEEE International Conference on Computer Vision, 2650-2658.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research, 11(Feb), 625-660.

Fernando, B., Bilen, H., Gavves, E., & Gould, S. (2017). Self-supervised video representation learning with odd-one-out networks. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3636-3645.

Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3), 381-396.

Géron, A. (2019). Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems O'Reilly Media.

Goertzel, B., Lian, R., Arel, I., De Garis, H., & Chen, S. (2010). A world survey of artificial brain projects, part II: Biologically inspired cognitive architectures. Neurocomputing, 74(1-3), 30-49.

Goode, J. (2015). No title. Classifying Images in the Oxford 102 Flower Dataset with CNNs–Jimmie Goode,

Google Inc. (2018). Firebase documentation. Retrieved from https://firebase.google.com

Greene, D., & Cunningham, P. (2006). Practical solutions to the problem of diagonal dominance in kernel document clustering. Paper presented at the Proceedings of the 23rd International Conference on Machine Learning, 377-384.

Gurnani, A., Mavani, V., Gajjar, V., & Khandhediya, Y. (2017). Flower categorization using deep convolutional neural networks. arXiv Preprint arXiv:1708.03763,

Hassannejad, H., Matrella, G., Ciampolini, P., De Munari, I., Mordonini, M., & Cagnoni, S. (2016). Food image recognition using very deep convolutional networks. Paper presented at the Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, 41-49.

Hassner, T. (2015). Age and gender classification using convolutional neural networks. Paper presented at the Cvpr, , 24(3) 2622-2629.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Paper presented at the Proceedings of the IEEE International Conference on Computer Vision, 1026-1034.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.

Hidalgo, J. M. G., Almeida, T. A., & Yamakami, A. (2012). On the validity of a new SMS spam collection. Paper presented at the 2012 11th International Conference on Machine Learning and Applications, , 2 240-245.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv Preprint arXiv:1704.04861, Howard, A., Sandler, M., Chu, G., Chen, L., Chen, B., Tan, M., . . . Vasudevan, V. (2019). Searching for mobilenetv3. Paper presented at the Proceedings of the IEEE International Conference on Computer Vision, 1314-1324.

Huang, W., & Luo, X. (2017). Commonsense reasoning in a deeper way: By discovering relations between predicates. Paper presented at the Icaart (2), 407-414.

Hwang, D., Kim, S., Monet, N., Koike, H., & Bae, S. (2020). Lightweight 3D human pose estimation network training using teacher-student learning. arXiv Preprint arXiv:2001.05097,

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv Preprint arXiv:1602.07360,

Idrees, H., Saleemi, I., Seibert, C., & Shah, M. (2013). Multi-source multi-scale counting in extremely dense crowd images. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2547-2554.

Jan□cek, M. (2010). Abductive reasoning for continual dialogue understanding. New directions in logic, language and computation (pp. 16-31) Springer.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? Paper presented at the 2009 IEEE 12th International Conference on Computer Vision, 2146-2153.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 11-19.

Jha, G. K. (2007). Artificial neural networks and its applications. Journal of Indian Agricultural Research Institute, New Delhi, , 110-012.

Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015). Knowledge graph embedding via dynamic mapping matrix. Paper presented at the Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 687-696.

Kadlec, R., Bajgar, O., & Kleindienst, J. (2017). Knowledge base completion: Baselines strike back. arXiv Preprint arXiv:1705.10744,

Kendall, A., Badrinarayanan, V., & Cipolla, R. (2015). Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv Preprint arXiv:1511.02680,

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? Paper presented at the Advances in Neural Information Processing Systems, 5574-5584.

Khosla, A., Raju, A. S., Torralba, A., & Oliva, A. (2015). Understanding and predicting image memorability at a large scale. Paper presented at the Proceedings of the IEEE International Conference on Computer Vision, 2390-2398.

Khot, T., Sabharwal, A., & Clark, P. (2019). What's missing: A knowledge gap guided approach for multi-hop question answering. arXiv Preprint arXiv:1909.09253,

Kosko, B. (1986). Fuzzy cognitive maps. International Journal of Man-Machine Studies, 24(1), 65-75.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging Artificial Intelligence Applications in Computer Engineering, 160, 3-24.

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Paper presented at the Advances in Neural Information Processing Systems, 1097-1105.

LaBar, K. S., & Cabeza, R. (2006). Cognitive neuroscience of emotional memory. Nature Reviews Neuroscience, 7(1), 54-64.

Laird, J. E. (2008). Extending the soar cognitive architecture. Frontiers in Artificial Intelligence and Applications, 171, 224.

Lamme, V. A., Super, H., & Spekreijse, H. (1998). Feedforward, horizontal, and feedback processing in the visual cortex. Current Opinion in Neurobiology, 8(4), 529-535.

Lavin, A., & Gray, S. (2016). Fast algorithms for convolutional neural networks. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4013-4021.

Levesque, H., Davis, E., & Morgenstern, L. (2012). The winograd schema challenge. Paper presented at the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning,

Levi, G., & Hassner, T. (2015a). Age and gender classification using convolutional neural networks. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 34-42.

Levi, G., & Hassner, T. (2015b). Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. Paper presented at the Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, 503-510.

Lieberman, H., Faaborg, A., Espinosa, J., & Stocky, T. (2004). Common sense on the go: Giving mobile applications an understanding of everyday life. BT Technology Journal,

Lieberman, H., Liu, H., Singh, P., & Barry, B. (2004). Beating common sense into interactive applications. AI Magazine, 25(4), 63.

Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. Paper presented at the Twenty-Ninth AAAI Conference on Artificial Intelligence,

Liu, C., Chen, L., Schroff, F., Adam, H., Hua, W., Yuille, A. L., & Fei-Fei, L. (2019). Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 82-92.

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

Liu, Q., Jiang, H., Evdokimov, A., Ling, Z., Zhu, X., Wei, S., & Hu, Y. (2016). Probabilistic reasoning via deep learning: Neural association models. arXiv Preprint arXiv:1603.07704,

Liu, Q., Jiang, H., Ling, Z., Zhu, X., Wei, S., & Hu, Y. (2016). Commonsense knowledge enhanced embeddings for solving pronoun disambiguation problems in winograd schema challenge. arXiv Preprint arXiv:1611.04146,

Liu, Q., Jiang, H., Ling, Z., Zhu, X., Wei, S., & Hu, Y. (2017a). Combing context and commonsense knowledge through neural networks for solving winograd schema problems. Paper presented at the 2017 AAAI Spring Symposium Series,

Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1096-1104.

Lotter, W., Kreiman, G., & Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. arXiv Preprint arXiv:1605.08104,

Loy, C. C., Chen, K., Gong, S., & Xiang, T. (2013). Crowd counting and profiling: Methodology and evaluation. Modeling, simulation and visual analysis of crowds (pp. 347-382) Springer.

Mallot, H. A., & Basten, K. (2009). Embodied spatial cognition: Biological and artificial systems. Image and Vision Computing, 27(11), 1658-1670.

Marsden, M., McGuinness, K., Little, S., & O'Connor, N. E. (2016). Fully convolutional crowd counting on highly congested scenes. arXiv Preprint arXiv:1612.00220,

Marsella, S., & Gratch, J. (2002). A step toward irrationality: Using emotion to change belief. Paper presented at the Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, 334-341.

Martinel, N., Foresti, G. L., & Micheloni, C. (2018). Wide-slice residual networks for food recognition. Paper presented at the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 567-576.

Miller, E. K., Freedman, D. J., & Wallis, J. D. (2002). The prefrontal cortex: Categories, concepts and cognition. Philosophical Transactions of the Royal Society of London.Series B: Biological Sciences, 357(1424), 1123-1136.

Miu, V., & Fryazinov, O.Accelerating handheld object pose estimation on smartphones in unity with TensorFlow lite.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., . . . Ostrovski, G. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

Moore, R. C. (1982). The role of logic in knowledge representation and commonsense reasoning SRI International. Artificial Intelligence Center.

Musa, R., Scheidegger, M., Kulas, A., & Anguilet, Y. (2003). GloBuddy, a dynamic broad context phrase book. Paper presented at the International and Interdisciplinary Conference on Modeling and using Context, 467-474.

Nasution, M. K., & Noah, S. (2010). Extracting social networks from web documents. Paper presented at the Book of Abstracts CAIT, the 1st National Doctoral Seminar on Artificial Intelligence Technology CAIT UKM, 278-281.

Nguyen, N. D., Nguyen, T., & Nahavandi, S. (2017). System design perspective for human-level agents using deep reinforcement learning: A survey. IEEE Access, 5, 27091-27102.

Ni, Y., Chen, B., Ye, P., & Wang, C. (2014). A framework for iOS application development. Jsw, 9(2), 398-403.

Panton, K., Matuszek, C., Lenat, D., Schneider, D., Witbrock, M., Siegel, N., & Shepard, B. (2006). Common sense reasoning–from cyc to intelligent assistant. Ambient intelligence in everyday life (pp. 1-31) Springer.

Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 16-17.

Pathak, D., Girshick, R., Dollár, P., Darrell, T., & Hariharan, B. (2017). Learning features by watching objects move. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2701-2710.

Piczak, K. J. (2015). ESC: Dataset for environmental sound classification. Paper presented at the Proceedings of the 23rd ACM International Conference on Multimedia, 1015-1018.

Poole, D. (1989). Explanation and prediction: An architecture for default and abductive reasoning. Computational Intelligence, 5(2), 97-110.

Purushwalkam, S., & Gupta, A. (2016). Pose from action: Unsupervised learning of pose features based on motion. arXiv Preprint arXiv:1609.05420,

Rahman, A., & Ng, V. (2012). Resolving complex cases of definite pronouns: The winograd schema challenge. Paper presented at the Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 777-789.

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. arXiv Preprint arXiv:1806.03822,

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016b). Squad: 100,000 questions for machine comprehension of text. arXiv Preprint arXiv:1606.05250,

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386.

Ruan, Z. X. T., Fejzic, I., & Zhen, Z.Custom YOLOv3 using core ML IOS mobile YOLOv3: An incremental Improvement–Joseph redmon.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Bernstein, M. (2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252.

Sabater, J., & Sierra, C. (2002). Reputation and social network analysis in multi-agent systems. Paper presented at the Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, 475-482.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4510-4520.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv Preprint arXiv:1910.01108,

Santa Cruz, R., Fernando, B., Cherian, A., & Gould, S. (2017). Deeppermnet: Visual permutation learning. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3949-3957.

Sap, M., Le Bras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., . . . Choi, Y. (2019). Atomic: An atlas of machine commonsense for if-then reasoning. Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence, , 33 3027-3035.

Saxe, A. M., Koh, P. W., Chen, Z., Bhand, M., Suresh, B., & Ng, A. Y. (2011). On random weights and unsupervised feature learning. Paper presented at the Icml, , 2(3) 6.

Schüller, P. (2014). Tackling winograd schemas by formalizing relevance theory in knowledge graphs. Paper presented at the Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning,

Schwartz, R., Thomson, S., & Smith, N. A. (2018). SoPa: Bridging CNNs, RNNs, and weighted finite-state machines. arXiv Preprint arXiv:1805.06061,

Senior, A., Vanhoucke, V., Nguyen, P., & Sainath, T. (2012). Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine,

Shafiee, M. J., Li, F., Chwyl, B., & Wong, A. (2017). SquishedNets: Squishing SqueezeNet further for edge device scenarios via deep evolutionary synthesis. arXiv Preprint arXiv:1711.07459, Sharma, A., Vo, N. H., Aditya, S., & Baral, C. (2015). Towards addressing the winograd schema challenge—building and using a semantic parser and a knowledge hunting module. Paper presented at the Twenty-Fourth International Joint Conference on Artificial Intelligence,

Siegel, N., Shepard, B., Cabral, J., & Witbrock, M. (2005). Hypothesis generation and evidence assembly for intelligence analysis: Cycorp's noöscape application. Paper presented at the Proceedings of the 2005 International Conference on Intelligence Analysis (IA 2005), McLean, VA, USA,

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv Preprint arXiv:1409.1556,

Singh, P. (2002). The open mind common sense project. KurzweilAI.Net,

Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T., & Zhu, W. L. (2002). Open mind common sense: Knowledge acquisition from the general public. Paper presented at the OTM Confederated International Conferences" on the Move to Meaningful Internet Systems", 1223-1237.

Socher, R., Chen, D., Manning, C. D., & Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. Paper presented at the Advances in Neural Information Processing Systems, 926-934.

Speer, R., & Havasi, C. (2013). ConceptNet 5: A large semantic network for relational knowledge. The people's web meets NLP (pp. 161-176) Springer.

Srivastava, B. M. L., Vydana, H. K., Vuppala, A. K., & Shrivastava, M. (2015). A language model based approach towards large scale and lightweight language identification systems. arXiv Preprint arXiv:1510.03602,

Strong, A. I. (2016). Applications of artificial intelligence & associated technologies. Science [ETEBMS-2016], 5(6)

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction MIT press.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1-9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2818-2826.

Takagi, H., & Hayashi, I. (1991). NN-driven fuzzy reasoning. International Journal of Approximate Reasoning, 5(3), 191-212.

Tandon, N., Mishra, B. D., Grus, J., Yih, W., Bosselut, A., & Clark, P. (2018). Reasoning about actions and state changes by injecting commonsense knowledge. arXiv Preprint arXiv:1808.10012,

Teerapittayanon, S., McDanel, B., & Kung, H. (2016). Branchynet: Fast inference via early exiting from deep neural networks. Paper presented at the 2016 23rd International Conference on Pattern Recognition (ICPR), 2464-2469.

Teerapittayanon, S., McDanel, B., & Kung, H. (2017). Distributed deep neural networks over the cloud, the edge and end devices. Paper presented at the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 328-339.

Tian, Z., Huang, W., He, T., He, P., & Qiao, Y. (2016). Detecting text in natural image with connectionist text proposal network. Paper presented at the European Conference on Computer Vision, 56-72.

Trinh, T. H., & Le, Q. V. (2018). A simple method for commonsense reasoning. arXiv Preprint arXiv:1806.02847,

Trullier, O., Wiener, S. I., Berthoz, A., & Meyer, J. (1997). Biologically based artificial navigation systems: Review and prospects. Progress in Neurobiology, 51(5), 483-544.

van Aken, B., Winter, B., Löser, A., & Gers, F. A. (2019). How does BERT answer questions? A layer-wise analysis of transformer representations. Paper presented at the Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 1823-1832.

Wallis, G., & Bülthoff, H. H. (2001). Effects of temporal association on recognition memory. Proceedings of the National Academy of Sciences, 98(8), 4800-4804.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4), 600-612.

Woffinden, D. (2014). Investigating the data flow and parallelism in real-time dense SLAM.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., . . . Funtowicz, M. (2019). Transformers: State-of-the-art natural language processing. arXiv Preprint arXiv:1910.03771,

Yang, J., Wang, M., Zhou, H., Zhao, C., Yu, Y., Zhang, W., & Li, L. (2019). Towards making the most of bert in neural machine translation. arXiv Preprint arXiv:1908.05672,

Yang, L., Luo, P., Change Loy, C., & Tang, X. (2015). A large-scale car dataset for fine-grained categorization and verification. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3973-3981.

Zeng, L., Xu, X., Cai, B., Qiu, S., & Zhang, T. (2017). Multi-scale convolutional neural networks for crowd counting. Paper presented at the 2017 IEEE International Conference on Image Processing (ICIP), 465-469.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2014). Object detectors emerge in deep scene cnns. arXiv Preprint arXiv:1412.6856,

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. Paper presented at the Advances in Neural Information Processing Systems, 487-495.

Zhu, Mark Sandler Andrew Howard Menglong, & Chen, A. Z. L.Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation.

Bert: Pre-training of deep bidirectional transformers for language understanding. Swift-Coreml-Transformers. (2019). Retrieved from github.com/huggingface/swift-coreml-transformers#-gpt-2.

CoreML Apple Documentation. (2018). Retrieved from https://developer.apple.com/documentation/coreml

Tensorflow Lite Documentation. (2018). Retrieved from https://www.tensorflow.org/lite/, retrieved October 2018

Tensorflow Documentation. (2018). Retrieved from https://www.tensorflow.org/, retrieved October 2018

Turi Create Documentation. (2018). Retrieved from https://github.com/apple/turicreate

Pytorch. (2018). Retrieved from https://github.com/pytorch/pytorch

ONNX. (2018). Retrieved from https://facebook.ai/developers/tools/onnx

SKlearn. (2018). Retrieved from https://scikit-learn.org/stable

Mobilenets: Efficient convolutional neural networks for mobile vision applications. (2018) Retrieved from https://github.com/shicai/MobileNet-Caffe

Learning by doing - Image recognition with CoreML and Places205-GoogLeNet. (2018) Retrieved from http://places.csail.mit.edu

ResNet50. (2018). Retrieved from https://github.com/KaimingHe/deep-residual-networks

Rethinking the inception architecture for computer vision. (2018). Retrieved from https://github.com/fchollet/deep-learning-models

VGG16. (2018) Retrieved from https://github.com/fchollet/deep-learning-models/blob/master/vgg16.py

A Large-Scale Car Dataset for Fine-Grained Categorization and Verification. (2018) Retrieved from https://github.com/likedan/Core-ML-Car-Recognition

Age and gender classification using convolutional neural networks. (2018) Retrieved from https://github.com/cocoa-ai/FacesVisionDemo

Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. (2018) Retrieved from https://github.com/cocoa-ai/FacesVisionDemo

Food image recognition using very deep convolutional networks. (2018) Retrieved from https://github.com/ph1ps/Food101-CoreML

Hand-gesture recognition on iOS app using CoreML. (2018) Retrieved from https://github.com/akimach/GestureAI-CoreML-iOS

Classifying Images in the Oxford 102 Flower Dataset with CNNs (2018) Retrieved from https://github.com/cocoa-ai/FlowersVisionDemo

From pixels to sentiment: Fine-tuning CNNs for visual sentiment prediction. (2018) Retrieved from https://github.com/cocoa-ai/SentimentVisionDemo,

Detecting Text in Natural Image with Connectionist Text. (2018) Retrieved from https://github.com/tianzhi0549/CTPN

Image Style Classification. (2018) Retrieved from https://github.com/cocoa-ai/StylesVisionDemo,

Fashion Detection. (2018) Retrieved from https://github.com/liuziwei7/fashion-detection

Automatic spoken language identification Implementation (2018) Retrieved from https://github.com/twerkmeister/iLID

Spam Detection Implementation (2018) Retrieved from https://github.com/gkswamy98/imessage-spam-detection/tree/master

Vision Framework. (2019). Apple Inc. "Vision." *Vision | Apple Developer Documentation*, Retrieved from developer.apple.com/documentation/vision.

SwiftNLC. (2019). Retrieved from github.com/JacopoMangiavacchi/SwiftNLC.

Turi Create. (2019). Retrieved from https://github.com/apple/turicreate.

Sound Classification Implementation (2019, 18 June). CapTechMobile/Blogfest-Coreml. Retrieved from github.com/CapTechMobile/blogfest-coreml.

Analyzing Audio to Classify Sounds (2019). Retrieved from developer.apple.com/documentation/soundanalysis/analyzing_audio_to_classify_sounds.

Chugh, Anupam. (31 Jan. 2020). Retrieved from github.com/anupamchugh/iowncode.

Narner/ESC10-CoreML (2019). Retrieved from github.com/narner/ESC10-CoreML.

M'Haimdat, Omar (24 Aug. 2019). Retrieved from github.com/omarmhaimdat/SentimentAnalysisApp.

Mdering, Matt. (6 June. 2017). Retrieved from https://github.com/mdering/CoreMLZoo.

ESC: Dataset for environmental sound classification (6 Apr. 2018). Retrieved from github.com/karolpiczak/ESC-50.

Omarmhaimdat/WhoAreYou (24 Aug. 2019). Retrieved from github.com/omarmhaimdat/WhoAreYou.

PlaceRecognizer (2 May 2018). Retrieved from github.com/hanksudo/PlaceRecognizer.

MXNet2CoreML. (8 Sept. 2017) Retrieved from github.com/awslabs/MXNet2CoreML_iOS_sample_app.

Hollance/MobileNet-CoreML. (2016). Retrieved from github.com/hollance/MobileNet-CoreML.

Custom YOLOv3 using core ML IOS mobile YOLOv3: An incremental Improvement– Joseph redmon. (19 Nov. 2019). Retrieved from github.com/hollance/YOLO-CoreML-MPSNNGraph.

Tucan. “tucan9389/ImageSegmentation-CoreML.” *GitHub*, 26 Nov. 2019, github.com/tucan9389/ImageSegmentation-CoreML.

Krishna, Vamshi. VKSentimentAnalysis. (2019). Retrieved from github.com/VamshiIITBHU14/VKSentimentAnalysis.

Greenfield, Matt. sobri909/LocoKit (6 July 2018). Retrieved from github.com/sobri909/LocoKit.

Accelerating handheld object pose estimation on smartphones in unity with TensorFlow lite. (25 Nov. 2019). Retrieved from github.com/tucan9389/PoseEstimation-CoreML.

M'Haimdat, Omar. “Omarmhaimdat/TopicClassifier.” *GitHub*, 2019, github.com/omarmhaimdat/TopicClassifier.

Mavi, Arda. Ardamavi/Vocalization-Sign-Language-IOS (14 Apr. 2018). Retrieved from github.com/ardamavi/Vocalization-Sign-Language-iOS.

Roche, Dimitri. Dimroc/Count. (28 Apr. 2019). Retrieved from github.com/dimroc/count.

Inc, Apple. Apple Developer Documentation (2020). Retrieved from developer.apple.com/documentation/avfoundation/avspeechsynthesizer.

Inc, Google. ML Kit for Firebase. (2020). Retrieved from firebase.google.com/docs/ml-kit.

Tensorflow Implementation. Tensorflow/examples (25 Nov. 2019). Retrieved from https://github.com/tensorflow/examples

DeepLabV3 Implementation. Tucan9389/ImageSegmentation-CoreML. (26 Nov. 2019) Retrieved from https://github.com/tucan9389/ImageSegmentation-CoreML

Tiramisu Implementation. "Kautenja/ios-semantic-segmentation." *GitHub*, (25 Nov. 2019) Retrieved from https://github.com/Kautenja/ios-semantic-segmentation

CoreMLTools Implementation. Apple/coremltools (21 Feb. 2019). https://github.com/apple/coremltools

Benchmark Implementation. "WorldDownTown/Benchmark. (4 May. 2017). Retrieved from https://github.com/WorldDownTown/Benchmark

Appendix A: Glossary

Neural networks, mathematical machine learning algorithms that are highly inspired by the neural architecture of the human brain. Artificial Neural Networks are neural networks that contain a wide range of hidden layers. Each layer of an artificial neural network contains several neurons. There are three primary types of artificial neural networks that we utilize through our system.

Neuron, the most fundamental building block of a neural network, similar to that of a human brain, is a neuron that takes in inputs and pushes outputs. Each neuron is ultimately a mathematical function, known as an activation function, that is applied to its inputs to produce an output.

Neural Network Layers, neurons that come together to make up layers within a neural network, which are collections of neurons with inputs and outputs. Deep neural networks contain input layers, output layers, and middle layers called hidden layers. Neural network layers within a neural network contain activation functions and weights that differ from one another to create outputs. Weights set the overall priority and impact of features in a layer.

Activation functions, function that captures the non-linear changes in data that leads to a particular output within a neural network. The most common forms of activation functions include sigmoid function, hyperbolic tangent function, threshold function, and rectifier function. The activation function utilized within a particular neural network is chosen based on the problem the network is trying to address.

Convolutional neural networks, a subtype of deep neural networks optimal at learning non-linear functions. Convolutional neural networks have served as a popular form of deep neural networks, especially in mobile computing, when processing visual data. CNN comprise of multiple convolutional layers, pooling layers, flattening layers, subsampling layers, and fully connected layers. Convolutional neural networks capture elements of sensory data and detect critical features from these elements across layers, through the use of convolutional filters.

Convolutional layers, when processing visual data, provides the ability to build sparse connections between neurons with generally fewer parameters. The pooling layers determine the data with the highest concentration of value within an area and reduce the size of the data that is less significant to improve performance and reduces overfitting. The final layers of the convolutional neural network are the flattening layers and fully connected layer. The flattening layer flattens the outputs of the convolutional layers and pooling layers and sends the data to the fully connected layers as a one-dimensional array. Convolutional neural networks are easier to train, versus deep neural networks, due to being mostly convolutional layers. General deep neural networks, on the other hand, usually contain fully connected layers. Convolution neural networks have much less fully connected layers, making them truly ideal for analyzing visual data. Convolutional neural networks are also capable of sharing their parameters across space and can capture inputs and outputs in three dimensions.

Recurrent Neural Networks, deep neural networks optimized for sequential processing data. Recurrent neural networks are ideal for applications that deal with data that is forgoing and sequential like audio, speech, and pattern recognition. The benefit of recurrent neural networks is its ability to deal with data inputted at different times, unlike most artificial neural networks that accept all inputs at one time and can establish connections between current, future, and previous data. Recurrent neural networks, like general artificial neural networks, can establish layers from left to right with weights that affect future neurons and layers in the network. Then it measures the cost function, which is the error between a predicted value and the actual value. It then determines the slope of the activation function, through backpropagation. It retroactively updates the weights with the gradient descent, the minimum value for the function, to reduce error.

TensorFlow, an open-source software library tailored for machine learning and deep learning.

Turi Create, a simple to use machine learning library which is flexible and can export CoreML compatible models.

CreateML, a machine learning library that utilizes Swift and macOS playgrounds to create and train custom machine learning models on a Mac. It can train models to perform tasks like recognizing images, extracting meaning from text, or finding relationships between numerical values. It is a GPU-accelerated tool for native artificial intelligence model training on Mac computers that supports computer vision and natural language.

Caffe, a deep learning framework made with expression, speed, and modularity in mind. It allows for training on both CPU or GPU machines and is optimal for machine learning performance.

CoreML 3, a native machine learning framework that allows iOS devices to run on-device machine learning inference. CoreML 3 is optimal for on-device performance by leveraging the Central processing unit, or CPU, graphical processing unit, or GPU, and Neural Engine, while also being efficient on power consumption and maintaining a minimal memory footprint. CoreML supports image analysis, natural language processing, learned decision trees, and is built upon low-level primitives, including Accelerate and basic neural network subroutines, while utilizing metal performance shaders. It also runs directly on the device, providing optimal performance, making it essential for user data privacy, reducing power consumption, and works offline. CoreML supports 16-bit Floating points and all levels of quantization, reducing the size of CoreML machine learning models. Core ML, also, ships with a converter that works with Caffe2, Keras, scikit-learn, XGBoost, LibSVM, and Tensorflow called Coremltools.

TensorFlow Lite, an open-source machine learning framework that also allows our mobile application to run on-device machine learning inference from Tensorflow models with low latency (Abadi et al., 2016).