5-13-2022

Incorporating spatial relationship information in signal-to-text processing

Jeremy Elon Davis

Mississippi State University, davis.jeremy.e@gmail.com

Incorporating spatial relationship information in signal-to-text processing

By

Jeremy Elon Davis

Approved by:

Cindy L. Bethel (Major Professor)
Derek T. Anderson (Co-Major Professor)
Amy E. Bednar
J. Edward Swan, II
John E. Ball
T.J. Jankun-Kelly (Graduate Coordinator)
Jason M. Keith (Dean, Bagley College of Engineering)

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Science
in the Department of Computer Science and Engineering

Mississippi State, Mississippi

May 2022

Name: Jeremy Elon Davis

Date of Degree: May 13, 2022

Institution: Mississippi State University

Major Field: Computer Science

Major Professor: Cindy L. Bethel

Title of Study: Incorporating spatial relationship information in signal-to-text processing

Pages of Study: 215

Candidate for Degree of Doctor of Philosophy

This dissertation outlines the development of a signal-to-text system that incorporates spatial relationship information to generate scene descriptions. Existing signal-to-text systems generate accurate descriptions in regards to information contained in an image. However, to date, no signal-to-text system incorporates spatial relationship information. A survey of related work in the fields of object detection, signal-to-text, and spatial relationships in images is presented first. Three methodologies followed by evaluations were conducted in order to create the signal-to-text system: 1) generation of object localization results from a set of input images, 2) derivation of Level One Summaries from an input image, and 3) inference of Level Two Summaries from the derived Level One Summaries. Validation processes are described for the second and third evaluations, as the first evaluation has been previously validated in the related original works. The goal of this research is to show that a signal-to-text system that incorporates spatial information results in more informative descriptions of the content contained in an image. An additional goal of this research is to demonstrate the signal-to-text system can be easily applied to additional data sets, other than the sets used to train the system, and achieve similar results to the training sets. To achieve this goal, a validation study was conducted and is presented to the reader.

Key words: Object Detection, Signal-to-Text, Scene Understanding, Spatial Relationships in Images, Natural Language Processing

DEDICATION

To my wife, Brittny. You are the love of my life and I could not imagine going on life's journey with anyone other than you. Thank you for your continued support and encouragement during this stressful time of my life. I hope I am always the best husband and father that I can be as you deserve all the happiness in the world. Without you, my life would be empty.

To my children, Amelia and Hayden. I hope that as you were watching me go through this process you learned that anything is possible if you dedicate yourself. I hope I am always a person that you can look up to and hope that you both know you are loved unconditionally.

To my grandmother, Dorothy, my mother, Myra, and my aunt, Mary. Thank you for all you have done and continue to do to make me the man I am today. Mama Dorothy and Mom, even though you aren't with me on this earth anymore, you are still with me every day and I hope this makes you proud of me.

To Uncle Terry and Aunt Cheryl, thank you both for your unwavering love and support throughout my life. Thank you both for being there for me when I was going through tough times and being a pillar of support and providing a shoulder to lean on. It continues to mean the world to me.

To Dr. John Foley, you have been very influential as both a mentor and friend to me. I cannot express how much I appreciate you taking a chance on me when I was an undergraduate student. I did not know at the time I would be making a friend/mentor for life but I am very grateful.

To Dr. Chris Goodin, thank you so much for giving me the opportunity to spread my wings and fly while I worked under your guidance. I can't express how grateful I am to call you my friend and I hope for nothing but the best for you and your family.

To Dr. Dampier, your unwavering support of me from my first day of attending MSU through the end of my Master's degree is appreciated more than words can express. You were a great influence on my academic career and life. I would not have signed up for the Ph.D. program if not for your encouragement and belief in me.

To Rikk Anderson, you are one of my most trusted friends. I am proud to follow in your footsteps and finish my Ph.D. I did not know that I was signing up for a lifelong friendship when I signed up for Dr. Yuan's Intro to AI course as an undergrad, but I cannot express how thankful I am that I did.

To Dae and Tan, thank the both of you for your support and for always being there for me in the clutch and for listening to me stress about everything. I hope both of your lives are filled with adventures and enjoyment as you travel abroad.

To Dr. Bethel, thank you for agreeing to be my committee chair and taking a chance on a non-traditional student. Your support and mentorship throughout this process is greatly appreciated.

To Dr. Anderson, thank you for making me realize that I can in fact "do math." I learned so much from your classes and I am proud that you agreed to help me and mentor me in the field.

To Dr. Swan, thank you for stepping in and agreeing to be part of my committee. Your undergraduate and graduate classes were not lost on me and I appreciate all of the valuable lessons I've learned from you.

To Dr. Bednar, thank you for being my “cheerleader.” It was always a pleasure to work alongside you at ERDC and I am proud to continue to still be able to do so.

To Dr. Ball, thank you so much for coming in clutch for me. I do not know where I would be if I had not been able to lean on you. You have my deepest gratitude.

To Col. Brandon Haynie, thank you for all of your support and encouragement. It is a great honor to call you my friend and a pleasure to work alongside you.

To Chayne Thrash, one day I will be able to return the favor and be your sounding board when you embark on your Ph. D. adventures. Thank you for always being willing to listen and giving advice.

Finally, I would like to dedicate this to my lord and savior, Jesus Christ. Without him, nothing I do would matter.

ACKNOWLEDGEMENTS

I would like to thank and acknowledge Dr. Jim Keller and Dr. Pascal Matsakis for providing me with the Histogram of Forces code used for experiments in this dissertation.

I would also like to thank Dr. Adrian Rosebock for creating an invaluable resource with his website, PyImageSearch [52].

Finally, I would like to thank anyone and everyone that has helped in any way during this journey.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

INITIALISMS AND ABBREVIATIONS

AP Average Precision

AWS Amazon Web Services

BLEU Bilingual Evaluation Understudy Score

BRNN Bidirectional Recurrent Neural Network

CNN Convolutional Neural Networks

COCO Common Objects in Context

DNN Deep Neural Network

DPM Deformable Parts Model

FS Fuzzy Set

FIS Fuzzy Inference System

GIOU Generalized Intersection Over Union

H-LDA Hierarchical Latent Dirichlet Allocation

HOF Histogram Of Forces

HOG Histograms of Oriented Gradients

IAA Interval Agreement Approach

IOU Intersection Over Union

JSON Javascript Object Notation

LDA Latent Dirichlet Allocation

LSTM Long Short-Term Memory

LSVM Latent Support Vector Machine

mAP Mean Average Precision

MIT Massachutesetts Institue of Technology

NLP Natural Language Processing

NMF Non-Negative Matrix Factorization

NMS Non-Maxima Suppression

R-CNN Regions with CNN Features

ReLU Rectified Linear Units

R-FCN Region-based Fully Convolutional Networks

RNN Recurrent Neural Networks

ROI Regions Of Interest

RPN Region Proposal Networks

S2T Signal-to-Text

SIFT Scale Invariant Feature Transform

SSD Single Shot MultiBox Detector

SVD Singular Value Decomposition

SVM Support Vector Machine

TF-IDF Term Frequency-Inverse Document Frequency

VOC Visual Object Classes

YOLO You Only Look Once

CHAPTER 1

INTRODUCTION

This research provides a novel signal-to-text (S2T) framework that couples existing technologies to create a system that generates more informative scene descriptions. This chapter first presents the motivation for developing such a system, which is primarily due to the lack of existing systems' abilities to infer spatial reasoning information. Following, a number of research questions and corresponding hypotheses are presented to the reader. These research questions and hypotheses will be the primary focus of the remainder of this dissertation including evaluations and validations performed toward determining the level of support for the formulated hypotheses. Two novel concepts will be introduced throughout this research, Level One and Level Two Summaries, and these summaries explored in depth in Chapters 4 and 5 respectively.

1.1 Motivation

Current signal-to-text systems [65] [66] [70] are limited by the lack of incorporation of spatial reasoning information. These current systems largely center around a combination of object detection frameworks and natural language processing (NLP), where the NLP labels are simply constructed based around the output of the object detection frameworks. While these systems do generate valid labels for scene descriptions, the lack of spatial reasoning, a valuable piece of information, between objects in an image limit inference accuracy.

Image processing techniques are being adopted across a broad range of applications. While image processing is not new research, the advent of deep learning brought about a resurgence of research interest in image processing. In particular, convolutional neural networks (CNN) caused an expansion in the research and development of object classification and object detection frameworks.

Object classification frameworks generate a label and corresponding accuracy score for objects believed to exist in an image. Object detection frameworks also generate a label and classification score for objects in an image, but also compute the outline of each object in an image via bounding boxes or masks. Numerous open source frameworks for object classification and detection have been developed for use. The research outlined in the conducted evaluations made use of the TensorFlow [1] and You Only Look Once (YOLO) [46] frameworks for object detection.

Recent research has centered around signal-to-text (S2T) systems. S2T, also referred to as scene understanding or scene description systems, attempt to generate a concise, natural description of the interactions between objects in images. S2T systems designed around the deep learning methodology, such as Show-and-Tell [70], use labels generated by an object detection framework combined with the natural language processing (NLP) capabilities of recurrent neural networks (RNN) to construct high-level descriptions of object interactions. S2T systems such as the hierarchical framework developed by Anderson, et al. in [3] made use of fuzzy inference for constructing the high-level descriptions of object interactions in images.

One area of S2T systems that can be improved is the ability to take spatial context into account when generating scene descriptions. At the time of this writing, no work has been performed in the area of using object bounds to compute spatial relationships or ensure logical orderings of objects in scene captions that contain spatial context. Thus, a primary goal of this research was to build upon current S2T methodologies by incorporating a spatial relationship component into the S2T framework. Very little work has been performed toward using spatial relationships in the context of S2T systems and an overview of the existing attempts and their drawbacks is given in Chapter 2.3. As a result of incorporating spatial information into S2T systems, this research determined an effective strategy for constructing NLP descriptions of scenes that are driven by the inferred spatial relationships.

1.2 Research Questions and Hypotheses

This research first derived concise natural language descriptions of spatial relationships between objects in an image. These are considered "Level One Summaries" and provide a high level description about the relationships between each tuple of objects in an image. This research demonstrates that Level One Summaries can be derived in a manner that accurately informs the S2T system constructed.

This research then inferred a natural language scene description based on the derived Level One Summaries. These "Level Two Summaries" provide a high-level description about the overall relationships between the set of object tuples in an image. In other words, the Level One Summaries informed the next phase of the pipeline, which constructed the Level Two Summaries. The expectation of this research was to show that Level Two Summaries that incorporated spatial information provided a more informative natural language scene description when compared to existing frameworks that do not take spatial information into account.

Each of the Level One and Level Two Summaries depend upon a constrained set of grammar terms that were defined during the evaluations. A constrained grammar allowed the S2T system to accurately map images to Level One Summaries and then Level One Summaries to Level Two Summaries.

This research presents methods and evaluations designed to answer the following research questions (RQ) and validate the hypotheses (H).

RQ1: How can concise and meaningful natural language descriptions (Level One Summaries) of spatial relationships between all object tuples in an image be derived?

RQ2: How can the Level One Summaries provide meaningful information to the creation of informative, high-level, natural language descriptions (Level Two Summaries) of the object interactions in a scene?

RQ3: When applying the S2T framework to a data set other than the data set used to construct the system, will the results be as reliable as the initial data set results?

RQ4: How can the S2T framework developed in this research be used as a mechanism for applying scene interaction labels that incorporate spatial reasoning information to a previously unlabeled data set?

H1: Histogram of Forces (HOF) [40] and Generalized Intersection Over Union [51], both rigorously validated methods with strong mathematical proofs, will provide concise natural language descriptions of spatial relationship information between object tuples in a scene.

H2: By utilizing a fuzzy inferencing approach in the generation of scene labels, the uncertainty in spatial relationship reasoning can be effectively modeled and incorporated into scene descriptions generated by the S2T system.

H3: Since the S2T system was modeled as a fuzzy inferencing system at its core, the rule-base will be data set agnostic and as such the developed S2T system will be as reliable on an additional data set as it was on the training data set.

H4: The developed S2T system requires no a priori *information regarding scene labels, thus applying the framework to a previously unlabeled data set will not limit the effectiveness of the system in regards to generating accurate scene descriptions.*

1.3 Summary

This chapter introduced the concept of object detection and S2T systems in terms of their uses in providing natural language scene descriptions. Also introduced was the concept of Level One and Level Two Summaries for S2T systems. These Level One and Level Two Summaries are expected to provide a more informative natural language labeling for images as opposed to state-of-the-art systems that do not incorporate spatial information. In particular, Level One Summaries provide an additional piece of information between object tuples, in the form of spatial reasoning, that could prove to be invaluable for computer vision data sets. Level Two Summaries will provide refined scene descriptions that incorporate spatial relationship reasoning, allowing the scene labels to much more accurately describe the interactions occurring in the scene. The S2T system provided by this research can be used as a standalone image annotation system, or be used to label a previously non-annotated data set. This property of the system allows it to be used as a tool for training supervised learning algorithms that require lots of training data via automating the data labeling process.

The remainder of this dissertation is composed of the individual parts of the S2T system, where each component will be analyzed with a discussion of results. To begin, background work related to this research is discussed in Chapter 2. Chapter 3 provides details of the methodologies used to extract object localization results. Chapter 4 provides implementation details of the algorithms used to construct the Level One Summaries (spatial relationship reasoning for object tuples) from the object localization results. Chapter 5 combines object localization results with Level One Summaries in order to produce Level Two Summaries, which are high level NLP descriptions of interactions occurring in the scene. Chapter 6 applies the S2T system to a data set not previously encountered along with an analysis of the results obtained. To conclude, Chapter 7 provides a discussion of the overall results of the system along with a presentation of potential future work.

CHAPTER 2

RELATED WORK

The presented research used existing methods in the areas of object detection and spatial reasoning, where spatial reasoning was an integral part of the proposed Signal-To-Text (S2T) framework. The following sections discuss existing object detection, spatial relationship, and S2T systems as they relate to this research. The proposed research improves upon existing S2T systems that do incorporate spatial reasoning by the use of well-defined algorithms such as Histogram of Forces [40] and Generalized Intersection Over Union [51]. By incorporating these methods for spatial reasoning, the proposed system provides additional fine-grained detail about spatial relationships between objects, compared to the existing methods, which only indicate one of three cardinal directions [20].

2.1 Object Detection

A vital component of S2T systems that warrants an initial discussion is object detection. Object detection builds upon the task of object classification, which provides a set of labels for objects that exist in an image. Object detection provides, in addition to the set of labels, localization information for each object. State-of-the-art object detection methods provide localization information by using either bounding boxes or pixel-wise object masks.

There has been extensive prior work in the area of object detection that has evolved over the years. Early models for object detection involved using a combination of methods to first estimate features and then perform classification on the estimated features. Recently, the focus has heavily shifted toward the use of deep neural networks, specifically convolutional nerual networks (CNN), for object detection. Recent approaches use CNNs to estimate features to form objects, classify which object the estimated features represent, and form bounding regions around each of the classified objects.

The following subsections first discuss some of the earliest works in object recognition. Then a general discussion of non-deep learning object recognition frameworks is presented. Finally, a brief overview of some of the most recent and widely accepted deep learning-based object detection frameworks are discussed in further detail as these networks are the current state-of-the-art.

2.1.1 Frameworks Based on Human-Engineered Models

One of the earliest works for object detection is described in Robust Real-Time Object Detection [67] and is commonly referred to as the Viola-Jones detector. The original Viola-Jones detector was developed to perform real-time face detection in images. The Viola-Jones detector operated by first computing Haar-like features that allow fast feature evaluation, which the authors term an integral image. The integral image was then subjected to a boosting process, namely AdaBoost, in which several weak classifiers were allowed to work on the subset of features from the integral image in order to construct strong classifiers out of their weaker counterparts. Once these strong classifiers had been obtained, a cascade classifier architecture was developed using the strong classifiers. This cascade classifier approach allowed the Viola-Jones detector to first denote the promising regions of the image, and then classify with further scrutiny any region that looked promising. By doing this, the speed of the Viola-Jones detector was dramatically increased.

By applying their novel approach, Viola and Jones were able to achieve real-time results with accuracies similar to existing facial detection methods. The Viola-Jones detector was one of the first major works in the field of object detection. It is a very specialized object detection framework that was optimized to detect human faces. Retraining the framework would require additional training of the AdaBoost component to account for other object instances, as well as hand-crafting a set of features for each additional object instance.

A novel framework for human detection was proposed in Histograms of Oriented Gradients for Human Detection [11]. The framework proposed was robust in that it could detect humans in various poses and at various scales. This framework also introduced a novel feature descriptor in the form of a histogram of oriented gradients (HOG). These features were used in conjunction with a classification algorithm to detect objects in an image.

The authors note that an object can often be localized by examining the distribution of the local intensity gradients and edge directions without precise knowledge of where the object is in the image. HOG features are computed by first dividing the image into n grid cells where each cell computes the gradient and direction of the gradient for the average value of the pixels in the cell. Once these cells have been computed, m block regions are computed. These block regions are an accumulation of the values in the smaller cells. Each of these blocks can then be used to normalize the smaller cell regions, and the normalized value of the cells that compose the block regions are what the authors define as a HOG descriptor block. The combination of the m descriptor blocks into a single vector is what the authors define to be a HOG feature.

The HOG features are then used in conjunction with a classifier to detect objects, humans in this case, in an image. The authors used a linear support vector machine (SVM) to speed up execution of their algorithm in [11]; however they do note that a Gaussian kernel SVM showed an accuracy increase of 3% at the cost of a much longer running time. Dalal and Triggs [11] note that, at the time of writing, the proposed framework reduced the false positive rates of the state-of-the-art human detectors by more than an order of magnitude. The proposed framework achieved near-perfect accuracy on the Massachusetts Institute of Technology (MIT) pedestrian database. As such, the authors also created a freely available pedestrian database aimed to be more challenging than its predecessor. Finally, while the HOG features are efficient on pedestrian images where the pedestrian is fully visible, any occlusions are prone to introduce false negative classification errors and it is assumed that a parts-based model with higher spatial invariance would lower these false negative rates.

The HOG feature-based approach to classification gave excellent results when working with images where objects were fully visible. The deformable parts model (DPM) framework proposed in Object Detection with Discriminatively Trained Parts-Based Models [15] builds upon the work on Dalal and Triggs in [11] by adding a parts-based model into the pipeline to avoid high false negative rates introduced by occlusion. The authors note that historically, complex models such as the one proposed, have performed poorly when applied to non-trivial data sets. This is likely due to the difficulty in training these complex models and the authors' proposed framework aims to bridge that gap.

The first contribution of the DPM framework was centered around the previously discussed HOG features. The DPM framework formed its part-based model by constructing a star-shaped graph hierarchy where the root node was a HOG filter as described in [11] and each subsequent node of the hierarchy was a combination of parts filters and deformation models. The root filter approximately reasoned about an entire object in an image whereas each subsequent parts filter performed higher resolution reasoning about smaller parts of the coarse object. Thus, the position of the root filter defined the detection window and the parts filters defined the smaller bounding boxes that constructed the individual parts of the entire object. To accurately detect objects, an overall score for each root filter was computed. High scoring root filters were indicative of an object, and the location of the parts that resulted in the high scoring root filter defined the full object. The authors noted that by defining an overall score for each root filter, they were able to detect multiple instances of the same object in an image and also that the placement of the root filters were analogous to using a sliding window approach such as those used in the Dalal-Triggs filter defined in [11]. In these terms, the score of the root filter can be viewed as the score of a sliding window placement. The DPM framework also employed principal component analysis on the HOG feature vectors, reducing the dimensionality of each with no noticeable loss in performance and an increase in execution speed. The DPM framework was designed to be feature invariant, but the authors chose to use HOG descriptors as features for their experimentation.

This particular framework used partially labeled imagery. The labels were bounding boxes around a region of interest, with no associated class label. The second contribution of the DPM framework was the ability to train using this partially-labeled data. The authors developed a technique they referred to as latent SVM (LSVM). The latent variables in the LSVM model were the specifications of the proposed object configurations. LSVM operated on the concatenation of the root filter, part filters, and deformation models (the star hierarchical model), β, and the feature vectors that represented each proposed object $\iota(x, z)$. By maximizing the inner product of $\beta * \iota(x, z)$, a bounding box for x was generated if its score was above an arbitrary minimum threshold for detection.

The data set used for experimentation was the PASCAL Visual Object Classes (VOC) 2006 data set and consisted of images labeled with bounding boxes [14]. The authors were able to successfully train the DPM framework using a three-phase approach. First, the LSVM model was trained on the ground truth data using gradient descent. Then the root filters of the DPM were initialized to n components, where n was the number of objects in the data set. The root filters for each component were trained using a standard SVM approach that resulted in positive examples. Negative examples for the SVM training were provided by randomly selecting regions from images in the data set that did not contain bounding boxes. Finally, the part filters were initialized by greedily assigning six components per root filter that maximized the energy covered in the filter specified by the root region. During testing, the greedy assignment of the parts filters resulted in overlap. Using non-maximum suppression, the overlapping regions were eliminated. The authors note that this method of training allowed them to train an accurate, complex model on non-trivial data.

In general, the DPM framework performed well on a wide variety of object classes and achieved accuracies similar to state-of-the-art approaches for most categories. The DPM model was not subjected to the errors of occlusion, as was the case for state-of-the-art techniques at the time of writing. The majority of false positives were due to intra-class confusion, such as the difference between a car and bus or horse and cow. The DPM framework was rigid in terms of bounding box prediction. The authors note that their model reported high false positives in the cat category because the training data contained only images of cat faces and the testing data contained faces and bodies. This is a common pitfall of most object detection systems.

2.1.2 Machine-Learned Framework Methods

Lecun et al. pointed out that one reason for the increased use of deep learning techniques for image classification and object detection was that traditional machine learning approaches were unable to process input images in their raw form. CNNs can process images as raw input and also craft features using the network architecture as opposed to using hand-crafted features or features formed using traditional feature extraction methods. Each convolutional layer abstracts the image to form features, for example, the first convolutional layer may detect edges, the second lines, the third shapes, and so on. Each of these layers generated a subset of features, and as such, feature extraction was performed naturally within the CNN. This inherit ability of CNNs along with the accuracy CNNs achieve on image data sets led to a wide adoption of deep learning techniques by the computer vision community [32].

Numerous CNN variants have been developed to date, as can be evidenced by a quick online literature search. However, the general structure of a convolutional neural network remains the same. As mentioned above, the input layer of a CNN was the image itself. One or more convolutional layers were then applied to the input image, resulting in a feature map. These convolutional layers performed a filtering operation on the input data in the form of a discrete convolutional filter, which was where the convolutional in convolutional neural network originates. After the first convolution operations, the outputs were passed through an activation function, the most common being either rectified linear units (ReLU), softmax, or hyperbolic tangent. These activation functions generated the feature map for an individual convolutional layer. The feature maps were then passed through a pooling layer, which simply merged similar features into one feature by computing the maximums of local patches of features in a feature map(s). One or more combinations of convolutional layers plus max pooling layers were stacked together in the CNN architecture. The final two layers of general CNN architecture were a fully connected layer and an output layer. To date, CNNs were generally trained using traditional backpropagation to adjust weights and minimize loss [32]. The networks discussed in the following subsection all adhere to this general architecture for CNNs, but with some tweaking in order to perform a specific task.

Initial work using convolutional neural networks focused heavily on the idea of region proposals and forming object descriptions based on the proposed valid regions. In Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation [18], the authors described the regions with CNN (R-CNN) features algorithm that "combined CNNs with bottom-up region proposals in order to localize and segment objects." The authors state that the progress on object detection frameworks for most of the 2000's was based around feature extraction and constructing an ensemble framework that used the features to extract object bounds in images. While these ensemble methods gave acceptable results, most of the work from 2010-2012, for example [63], [6], and [64] resulted in small gains in mean average precision (mAP) percentages because the frameworks developed were essentially variants of other ensemble frameworks based around feature extraction and recognition.

Existing works in image recognition using CNNs show that CNN architectures result in higher mAP accuracy compared to existing frameworks based on feature extraction and classification. However, these CNN architectures were unable to successfully segment objects within the images they were classifying [31]. The primary goal of the R-CNN framework was to bridge the gap that existed between image classification and object detection using CNNs. The R-CNN framework was the first framework proposed that showed a CNN could achieve higher mAP accuracy on the PASCAL VOC data set when compared to the generally-accepted feature-based frameworks [14]. The authors compared the proposed framework to an existing framework, OverFeat [57], which combined a sliding window approach with a CNN architecture in order to perform object detection.

The primary disconnect between image classification and object detection was that object detection required the localization of any objects that existed within the image. To solve this task, the proposed framework generated approximately 2000 class-independent region proposals for each input image. A feature vector of fixed-length was extracted from each proposal using a CNN, and these feature vectors were classified using a category-specific linear support vector machine. The CNN architecture used in the R-CNN framework was the same CNN architecture proposed by Krizhevsky and Hinton in [31]. A selective search was used to extract the region proposals and forward pass them through the CNN architecture. The features generated by this forward pass were then passed to a set of SVM classifiers to score each extracted feature vector.

The R-CNN framework was tested on the PASCAL VOC 2010-12 data set and compared with the original deformable parts model (DPM) and variants of the DPM [14]. R-CNN achieved a mAP of 50.2%, which was a noticeable improvement over the DPM mAP of 33.4%. The R-CNN framework was applied to the ILSVRC2013 data set and compared to the OverFeat framework. Where OverFeat achieved a mAP of 24.3%, R-CNN was able to achieve a mAP of 31.4%. The authors note that while the mAP was improved, the OverFeat model was approximately nine times faster than the R-CNN model.

Fast R-CNN [17] was an improvement to the R-CNN framework proposed in [18]. Fast R-CNN builds on the work of the R-CNN framework by improving training and testing speeds of the framework and also increasing detection accuracy. As discussed above, the R-CNN framework was a three-stage pipeline. The proposed framework in this paper streamlines the model into a single pipeline that learns to categorize the object proposals and then refines their spatial locations. The authors note that training the R-CNN was slow due to performing a forward pass through the CNN for each region proposal without sharing any computations.

To speed up the training time, region proposals were fed into the CNN architecture alongside the training image where they were processed by several convolutional and max pooling layers. These layers produced a convolutional feature map. After constructing the feature map, each region proposal was processed by a region of interest (ROI) pooling layer. This ROI layer extracted a feature vector of fixed-length. Fast R-CNN processed each feature vector using a sequence of fully connected layers that fed into two softmax output layers. One output layer produced probability estimates over *n* object categories while the second output layer output a 4-tuple for each of the *n* object classes. Each 4-tuple corresponded to the bounding box positions for one of the *n* classes.

Constructing the framework as a single pass pipeline allowed Fast R-CNN to detect objects by forward passing an input image and pre-computed region proposals through the CNN. The authors do not explicitly denote which method was used to pre-compute the region proposals, but their experiments indicated that the proposals could be computed using DPM or selective search. The ROI features were compressed using a truncated singular value decomposition (SVD) approach in order to speed up the execution time of performing object detection.

The authors compared the performance of Fast R-CNN with the original R-CNN framework. On the VOC 2012 set, Fast R-CNN achieved a mAP of 68.4% where the improved R-CNN framework detailed in [19] achieved a mAP of 62.4%. Fast R-CNN reduced the training time of R-CNN by a factor of nine, decreasing the time to train the network from 84 hours to 9.5 hours. Fast R-CNN processed images 146 times faster than R-CNN when not using truncated SVD ROI compression and processed images 213 times faster when using truncated SVD ROI compression.

An incremental improvement to Fast R-CNN, Faster R-CNN, was proposed in [50] and was shown to operate in real-time. Though Fast R-CNN could process images in under a second, the computation time required to perform the region proposal was not taken into account. Performing region proposals was the bottleneck of performing real-time object detection at the time of this writing. To eliminate this bottleneck, the authors introduced region proposal networks (RPN) that allowed computation of region proposals in approximately 10 ms per image.

The Faster R-CNN model was composed of two modules. The first module was a RPN that computed the region proposals for the input images. The second module was the Fast R-CNN detector. These modules comprised a single network for object detection. To share computation time with Fast R-CNN, the RPN module was constructed such that it shared the convolutional layers that computed feature maps for region proposals with the Fast R-CNN module. Regions were generated by sliding a small window over the feature maps generated by the convolutional layers. These windows generated a lower dimensional feature vector from the feature maps generated in the convolutional layers. The window feature maps were then processed by two fully connected layers, a box-regression layer and a box-classification layer. To prevent RPN and Fast R-CNN from modifying the convolutional layers independently, the authors proposed three techniques that allowed sharing of convolutional layers between the two network architectures. All of the experiments listed in this paper made use of the alternating training strategy. Alternating training first, trained the RPN and used the proposal to train the Fast R-CNN, and the resulting trained Fast R-CNN was then used to initialize the RPN architecture. The authors suggested that using these methods to propose regions eliminates the bottleneck that existed in previous region proposal frameworks.

In R-FCN: Object Detection via Region-based Fully Convolutional Networks [10], the authors proposed a framework that used a fully convolutional network coupled with a RoI pooling layer for region segmentation. The framework was denoted as region-based fully convolutional networks (R-FCN). Similar to R-CNN, the R-FCN framework incorporated the two-stage pipeline of a) region proposal and b) region classification. The authors use a region-based approach for their implementation because such approaches demonstrated a higher overall accuracy when compared to non-region based proposal frameworks for several benchmark data sets at the time of writing.

The R-FCN framework used the same RPNs that were used in the Faster R-CNN architecture to propose candidate regions [50]. The region proposals were then shared with the fully convolutional network (FCN). The FCN was used to classify object categories and background. The final layer of the FCN was a RoI pooling layer similar to the RoI pooling layer introduced by Fast R-CNN [17]. This RoI layer computed an output score for each candidate region. By combining the RoI layer with the convolutional layers, the R-FCN framework learned specialized scores for each proposed region that were position sensitive.

The authors used the PASCAL VOC data set for experimentation [14]. Compared to Faster R-CNN, which achieved a mAP of 68.9%, the R-FCN framework achieved a mAP of 76.6%.

However, this required the use of position sensitivity scoring in the RoI layer. Without the position sensitivity score, the authors note R-FCN failed on the VOC data set.

YOLO [46] is a framework aimed to re-purpose object detection via a CNN into a single-pass pipeline. YOLO used a single CNN to simultaneously predict bounding boxes for objects and classification probabilities for each bounding box found in a single pass through the network architecture. The primary reason for using a single-pass pipeline for object detection was to optimize execution speed to provide a framework that could perform object detection in real-time. YOLO was the first framework proposed that provided a real-time solution for object detection using a CNN.

YOLO first composed the input image into a $N \times N$ grid of cells, where each cell was responsible for predicting bounding boxes and confidence scores for each bounding box. The confidence score for each bounding box was an indication of how confident the YOLO framework was that a particular bounding box concisely encapsulated an object. Thus, if an object fell into a particular cell, that cell was responsible for the object prediction. Each bounding box was a 5-tuple that consisted of the width and height of the bounding box, the (x, y) coordinate of the bounding box in the image, and the confidence score of the bounding box. To classify objects, the YOLO framework predicted a class conditional probability for each object m in the set of known objects. These conditional probabilities were predicted for each cell in the SxS grid in the image. The combination of the class probabilities for each cell and the bounding box confidence scores provided a class-specific confidence score for each bounding box. The authors note that these scores encoded the probability that a class appeared in a bounding box as well as how well a bounding box fit an object.

At the time of writing, the only real-time object detection system that had been implemented was a variant of the deformable parts model developed by Sadeghi et al. [55] that ran at 30 fps. Similar work using CNN architectures for real-time object detection achieved maximum execution speeds of 18 fps with a mAP% of 26.1. YOLO achieved an execution speed of 45 fps with a 63.4 mAP%. The authors tested a smaller YOLO architecture, referred to as Fast YOLO, which achieved an execution speed of 155 fps with a mAP% of 52.7. Other region-based proposal networks, such as R-CNN, achieved higher mAP%s but were unable to achieve the FPS speeds of the YOLO architectures. YOLO redefined state-of-the-art in regards to real-time object detection.

Although the YOLO framework does perform object detection faster than previous CNN object detection frameworks, it does not achieve the same accuracy. Specifically, the authors note that the framework struggles to localize smaller objects that usually are grouped together such as a flock of birds. YOLO also does not generalize to objects that appear in unknown configurations since the model learns to predict bounding boxes from input training data. Lastly, errors in small bounding boxes versus errors in large bounding boxes are penalized the same in the YOLO framework. The authors note that their localization errors occurred due to the lack of differentiation on the bounding box error penalization. The previous results were all obtained from experimentation on the PASCAL VOC 2007 data set [14]. The authors applied a YOLO model that was trained on this data set to a generalized problem of person detection in artwork. YOLO exhibited the capability to generalize well to a new domain based on experimentation. YOLO was also the first framework to exhibit capability to perform live detection on incoming frames from a live webcam.

The first improvement to the original YOLO detector was proposed in YOLO9000: Better, Faster, Stronger [47], dubbed YOLO v2 by the authors. The authors noted that current object detection data sets were relatively small compared to object classification data sets, with the primary drawback being labeling of objects in images, which was usually performed via user specification.

A goal of YOLO v2 was to expand object detection to the scale of object classification. The authors proposed a new method that harnessed the vast amount of classification data and expanded it to the object detection domain. This method jointly trained an object detector on both detection and classification data sets. In this manner, YOLO v2 leveraged the labeled object detection data sets to precisely localize objects and used the classification data sets to increase the robustness and vocabulary of YOLO v2. By incorporating this training strategy, the authors note that YOLO v2 can detect over 9000 different object categories.

A second goal of YOLO v2 was to improve upon the recall and localization errors of YOLO v1. Rather than scaling up the original YOLO architecture, the authors proposed to simplify the network architecture and make the representation easier to learn by using a combination of techniques including: batch normalization, high resolution classifiers (448×448 convolutional layers as opposed to 256×256 in YOLO v1), using anchored bounding boxes to predict bounding boxes, predicting the anchored bounding boxes using K-Means clustering, using fine-grained features to eliminate small object localization errors, and using multi-resolution data for training. With the addition of these features, YOLO v2 achieved a mAP of 77.8% compared to 63.4% mAP for the original YOLO model on the Pascal VOC 2007 [14] data set.

A third goal of YOLO v2 was to improve the detection speed of YOLO without sacrificing the model accuracy. To speed up processing speeds, the authors proposed a new CNN architecture, Darknet-19, that was used as the base of the YOLO v2 model. This new architecture reduced the number of operations required from 30.69 billion for the VGG-16 CNN to 8.52 billion operations for YOLO v2. By using this CNN architecture the processing speed was increased to 59 fps compared to the 45 fps of the original YOLO model.

In YOLOv3: An Incremental Improvement [48], the authors proposed the YOLO v3 framework as an incremental improvement to the YOLO v2 framework [47]. The first improvement resulted from replacing the softmax classification with logistic regression. The authors noted that softmax was not necessary for good performance, and using logistic regression helped when moving on to more complex data sets. A second improvement came from allowing YOLO v3 to perform predictions across three different scales of bounding boxes. Another incremental improvement involved replacing the base architecture proposed in YOLO v2 (Darknet-19) with a newly developed CNN architecture, Darknet-53, that combined the structure of Darknet-19 and the recently developed residual networks. This new CNN architecture demonstrated the highest measured GPU operations per second and thus improved the execution speed of YOLO v2.

These incremental improvements were tested on the COCO data set. YOLO v3 exhibited a mAP of 41.9% compared to a mAP of 35.5% for YOLO v2. The single-shot multi-box detector (SSD) still achieved a higher mAP of 49.8% but the authors note that YOLO v3 runs at three times the speed of SSD on the COCO data set, although none of the timings were listed in this work.

The most recent update to the YOLO framework, YOLO V4: Optimal Speed and Accuracy of Object Detection [4], sought to improve both the accuracy and real-time execution speed of the existing YOLO frameworks. YOLO V4 leveraged the use of GPU architectures for training and running networks to improve real-time speed. To improve accuracy, the self-adversarial training methodology was employed. Also used to improve accuracy were the “bag of freebies” and “bag of specials” methods selected by the authors which were composed of various tuning and regularization techniques for the convolutional network. YOLO V4 [4] was able to achieve a mAP score of 43% on the COCO [35] data set while operating at 70 FPS.

SSD, proposed in [37], was similar to the YOLO framework in that it used a single deep neural network to perform bounding box localization and bounding box classification. The SSD framework was also designed with real-time object detection in mind. The authors noted that their framework was simple to train and to integrate into embedded systems, which made it ideal for a real-time solution for object detection. At the time of writing, increased speed in similar object detection frameworks came as a trade-off for detection accuracy and the SSD framework aimed to provide more accurate detections in real-time. The SSD framework was the first deep network framework that did not resample features from proposed bounding boxes and was as accurate as approaches that do, such as methods based on the R-CNN [18] architecture.

The SSD framework improved existing single-shot framework detection rates by making a modification to the types of filters used to classify object categories. By using small convolutional filters to predict object categories in proposed bounding boxes and using separate filters for various aspect ratio detections of the same object categories. These filters were applied at progressive stages of the network in order to perform detection at multiple scales within the image. The authors note that this modification allowed the SSD framework to achieve high accuracy in low-resolution imagery which, in turn, further decreased execution speeds.

The core of the SSD model was constructed based on the VGG16 [59] architecture, minus the classification layers. The framework produced a fixed-size set of bounding boxes and scored the presence of object categories that existed in the bounding boxes in a manner similar to YOLO

[46]. However, all bounding boxes were allowed to remain intact and a non-maxima suppression step removed any bounding boxes that were subsets of larger boxes. To perform object detection, several progressively decreasing convolutional feature layers were added to the end of the VGG16 base network that allowed the SSD framework to perform class predictions at multiple scales. Similar to other single shot approaches, the SSD framework tiled the input image into grid cells, but the SSD framework proposed default bounding boxes of varying resolutions at each grid scale to account for different object aspect ratios.

The SSD framework was able to provide speeds of 59 fps with a mAP of 74.3% on the VOC2007 data set. This was a substantial improvement to the execution speed of Faster R-CNN [18], which ran at 7 fps, as well as a substantial improvement to the 63.4% mAP of YOLO [46], which ran at 45 fps. While Fast YOLO ran at 155 fps, the authors noted that the increased execution speed comes at the price of a 22% decrease in accuracy when compared to the SSD framework. The authors also noted that the improvement in speed was a result of eliminating bounding box proposals and pixel/feature resampling. Compared to R-CNN, the authors noted that SSD performed better at localization because it "directly learns how to regress the object's shape and classify object categories instead of using two decoupled steps." SSD did show a higher error rate when localizing smaller object categories. This was likely due to SSD being very sensitive to bounding box size. In other words, since the SSD framework used various aspect ratios to detect objects, the coarsest filters may have little to no useful information for smaller objects. The authors noted that SSD achieved very high accuracy on any larger object category and was robust to aspect ratios in larger objects due to the use of the decreasing ratio convolutional filters.

The authors noted that using a variety of bounding box shapes during SSD initialization greatly increased bounding box generalization during testing. One of the major contributions of the SSD framework were these different bounding boxes at different scales on the network output layers. By combining these various resolution bounding boxes, the SSD framework achieved comparable accuracy to Faster R-CNN on lower resolution input imagery. The authors also noted that they believe this was the reason that the SSD framework was considerably more accurate than the YOLO framework. Finally, the authors noted that due to the simplicity of the SSD framework, it could be used as a viable building block for a system that required a real-time object detection component.

The most recent development in machine-learned framework methods was centered around the computation of object masks in images. A mask is a pixel wise binary visualization of an object in an image, a value of 1 indicates that a pixel belongs to an object, whereas a value of 0 indicates an object does not. By using pixel masks, it is possible to get a tighter set of bounds that encompass an object due to a mask representing a set of pixels that belong to an object whereas a bounding box is composed of only the four corners of the box enclosing an object in an image.

In Mask R-CNN [21], the authors described a framework that predicted an object mask in parallel to predicting object bounding boxes. Mask R-CNN builds upon the concept of Faster R-CNN [50]. Faster R-CNN provided as output a bounding box and class label for each object in the image. The proposed framework added in a third branch that output a mask for each object in the image. The mask branch provided much finer-grained localization detail. To accomplish this, the authors made use of the RPNs and ROI proposals used by Faster R-CNN and then used the ROI proposals to construct the pixel-wise masks for each object detected. The labels computed for each binary mask made use of the ROI classification step of Faster R-CNN to assign a label to the object mask while assigning labels to each ROI proposal.

The authors tested the proposed Mask R-CNN [21] on the COCO [35] data set and compared their results to state-of-the-art methods in instance segmentation [9], [34], [58]. Mask R-CNN outperformed all previous methods when using the average precision(AP) scoring metric. The authors also noted that the Mask R-CNN framework outperformed the Faster R-CNN framework [50] on the COCO data set in terms of the AP metric. Lastly, the authors noted that the Mask R-CNN framework can be easily extended to other tasks such as human pose estimation. The authors noted that Mask R-CNN generated comparable results to those obtained by state-of-the-art pose estimation methods.

2.2 Signal-to-Text

S2T is the process of generating a semantic description of an image and the objects contained within. Scene understanding frameworks generally consist of an object detection and localization component combined with a NLP component that generates semantic descriptions from the output of the object detection components.

The earliest frameworks that performed scene description took a hierarchical approach. In these frameworks, many individual components were stacked on each other to build a scene description system. Recent advances in the field of deep learning led to the usage of CNN for the object detection and localization component and long short-term memory recurrent neural networks (LSTM RNN) for the NLP component of scene understanding frameworks.

2.2.1 Frameworks Based on Hierarchical Models

In Toward Total Scene Understanding: Classification, Annotation, and Segmentation in an Automatic Framework [33], the authors proposed a hierarchical generative framework for scene understanding. At the time of writing, no algorithm had been developed that performed the classification, object annotation, and object segmentation of images in a single framework. This process of high level scene classification, object annotation, and object segmentation was denoted by the authors as the total scene understanding framework and combining these tasks into a single framework was a primary goal of this paper. The authors noted that earlier object detection approaches simply offered a single label or list of labels to an image without localization of where the objects were in the scene, disallowing effective semantic descriptions of the scene. The proposed framework computed the localization of the annotated objects in the scene, allowing more meaningful semantic scene descriptions to be generated.

The framework first generated a visual component and annotation component for each image. For the generative visual component, a multinomial distribution over the probability of objects in the scene was used. The image was sampled for each object in the distribution and known features of each class object was compared to features from the sampled patch. The features used by the framework were scale invariant feature transform (SIFT) features, first proposed by Lowe [38]. The annotation component was generated at the same time as the image component. For the generative annotation component, a region index was sampled from a uniform distribution over the number of classes that was responsible for assigning each image region a corresponding tag.

After construction of the generative models for the image and annotation components, the framework used these models to perform the tasks of classification, segmentation, and annotation.

The goal of segmentation in this framework was to extract exact pixel locations of each object in the image. The generative image model component was used to both propose object locations and also to infer these object locations. Once object localization had been performed, the generative annotation component was used to assign labels to each object in the image by treating the names of the segmented object as the object annotation. The overall goal of the classification step in regards to the framework proposed was to estimate the most likely label for an input image given the labels and object locations inferred by the visual and annotation generative models. The classification step generated a scene label, taken from one of eight learned classes in this work, that described the entire image. In other words, the scene label was composed as a hierarchical structure where the label concisely described the inferred tags and the inferred tags were composed of their corresponding localized object(s).

As mentioned previously, this was the first proposed work that incorporated high level classification, segmentation, and annotation into one framework. Of particular interest was the framework's ability to segment objects without any *a priori* knowledge about object localization or labels during training. The framework successfully computed object locations and labels, which were then assigned to one of eight (the number of scene classes proposed by the authors) scene labels. The framework proposed achieved state of the art accuracy for object segmentation and annotation when compared to existing methods at the time of this writing. However, the framework could not infer semantic descriptions of objects in a scene dynamically. In other words, the framework was dependent upon a user-defined set of high-level scene classes in which the annotated objects were binned. In recent works, the task of assigning a semantic scene description was performed by applying NLP methods to labels generated during object detection allowing the high-level classification to be performed at the segmentation level, a more dynamic method of generating scene descriptions. These methods will be discussed in further detail below.

In Linguistic Summarization of Video for Fall Detection Using Voxel Person and Fuzzy Logic [3], the authors proposed a hierarchical framework that was designed for fall detection, which operated on a continuous video feed. The proposed framework was designed to be used in a facility such as a senior care center, where fall detection was a significant concern. The authors designed their framework with privacy in mind so humans were distinguished by silhouettes and the video feeds were not retained. Using the silhouettes, a fuzzy inference approach was used to classify the current state of the person. Linguistic summaries were then constructed based on the temporal states of the silhouettes classified by the fuzzy inference system.

To construct the human silhouettes, the authors used multiple cameras that each viewed the environment at different angles. Once each camera's silhouette had been computed, the intersection of each silhouette was used to construct a three-dimensional silhouette of a human, which the authors labeled as a voxel person. Using the voxel person representation remedies problems such as occlusion that arose in standard single camera setups. The voxels computed for each image were then classified by a fuzzy inference system using a set of rules such that their membership degree existed in a set of pre-determined states. For the purposes of experimentation, the proposed framework used three states of membership to classify activity: Upright, On-the-ground, and In-between. The authors noted that it was important to point out that being on the ground (a state) does not imply a fall (an activity). The result of performing reasoning about the state of a voxel person at time t was thus three membership values for the set of pre-determined states that were used to calculate temporal linguistic summaries from video.

The goal of performing the temporal linguistic summaries was to take time periods of resident activity and provide a succinct semantic description of the activity performed. As an example the authors used “the resident was preparing lunch in the kitchen for a moderate amount of time” and “the resident has fallen in the living room and is down for a long time.” As the authors point out, reporting linguistic summaries for each video frame would result in too much information for the user to process, thus performing the summaries in a temporal manner reduced the amount of information presented to the end user, limiting information overload. The linguistic summaries were constructed in human readable language, such as the examples above, using the form:

$$X_c \text{ is } S_i \text{ in } P_k \text{ for } T_j \tag{2.1}$$

where X denoted the voxel person, S denoted the state the voxel person was in, P denoted the part of the “world” the voxel person was in, and T denoted the amount of time the voxel person had spent in state S in P area. The authors also noted that X was obtained from crisp sets whereas S, P, and T all resulted from fuzzy set membership values. Unlike previous approaches based on hidden Markov models, the framework proposed in this research used fuzzy set theory to construct annotations. Using fuzzy set theory allowed the system to be adaptable to special cases and thus could be modified to fit individual needs, in this case individual patients in a facility.

The authors performed experiments on data that was generated at the Computational Intelligence Laboratory at the University of Missouri. Experiments were not performed using elderly citizens as the risk of obtaining fall data would be too high, thus students were used as subjects to simulate falls. Working with nurses to establish a threshold for true positive fall detection alerts, the proposed framework was able to successfully classify 100% (14 out of 14) of fall activities as true positives and only generate false positives for 6% (2 out of 32) of non-fall activities. The authors noted that the false positive alerts were generated by activities that required the voxel person to be on the ground such as doing lying leg lifts.

The proposed hierarchical framework based on fuzzy set memberships proved to be very robust and easily adaptable to other use cases besides the one discussed in this paper. The proposed framework successfully generated linguistic scene summaries that provided valuable information to the end user in a concise but descriptive manner, unlike previous frameworks that just provided one-word labels generated in a hierarchical manner. The primary drawback of this system was that in order to adapt the framework for a new usage, the rules for the fuzzy inference system must be hand-crafted in order to meet the needs of the new system, which could be a time consuming task. However, recent methods based upon deep neural networks required the training and testing of the neural networks in order to fit new and specific use cases, which was a very time intensive task itself.

2.2.2 Frameworks Based on Machine-Learned Models

In Show and Tell: A Neural Image Caption Generator [65], the authors described a framework for signal-to-text that used a single pipeline based on deep neural nets that first assigned a list of labels to an image and then generated a sentence structure natural language description of an image. The authors denote this framework as Show and Tell.

To generate the natural language descriptions, the framework used a RNN by first allowing an encoder RNN to read the source sentence and then allowing a decoder RNN to generate a verbose target sentence. The authors replaced the encoder RNN with a CNN that performed image classification, which generated a list of labels that were passed into the decoder RNN to produce the final description of the scene. The decoder RNN used was a LSTM RNN that had shown state of the art performance for the tasks of translation and sequence generation due to its ability to handle exploding/vanishing gradients [23]. The authors indicated that a batch normalization CNN model was used for generating the image labels.

To evaluate the model, the authors tested Show-And-Tell on several data sets and evaluated the resultant image captions using two metrics. The first metric was based on human ratings of the input images. To accomplish this, the authors set up an Amazon Mechanical Turk experiment where each image was labeled by two workers to provide a ground truth description. The second metric was based on the bilingual evaluation understudy (BLEU) score, which ranked the precision of word n-grams between the generated captions and ground truth captions. The authors indicated that the labels generated by Show-And-Tell aligned well with the BLEU scores, but the model fared less well versus the labels constructed by human raters. The authors noted that a major area of improvement was making the captions align more closely with the human rater labels.

This model was one of the first works in the field of signal-to-text using deep neural networks and successfully generated semantic sentence descriptions of images in a data set. However, this work did not take spatial context between objects into consideration as the framework used a CNN strictly for image classification and not object detection.

Improvements and alterations to the original Show-And-Tell framework was proposed in Show and Tell: Lessons Learned [66] for the Microsoft common objects in context (COCO) image captioning challenge [35]. One such alteration was testing with the GoogleLeNet CNN model.

This model resulted in sub-par performance when compared to the batch normalization model previously used.

The first improvement made to the model was to allow the LSTM RNN to fully train before fine tuning the CNN used to generate the image captions. Previous training strategies trained both models simultaneously and the authors noted that training in this fashion allowed the noisy gradients coming from the LSTM model to creep into the CNN model and corrupt it. To counteract this, the authors trained the CNN for 500K iterations and then froze the parameters before jointly training the LSTM alongside the CNN.

A second improvement involved allowing the LSTM model to perform scheduled sampling to more loosely predict the next word in the sequence. In other words, as opposed to forcing the model to strictly predict the next word in the sequence, which resulted in overfitting and poor evaluation performance, the framework was allowed to use the LSTM model predicted word (even if it was wrong) during training, which resulted in higher accuracy on the evaluation set. The final improvements involved constructing an ensemble of these models and aggregating their captions and reducing the search space of the beam search algorithm used to construct the proposed image captions, which resulted in more novel image captions.

The improvements to the original model resulted in higher accuracy on the MS COCO data set as well as image captions more in line with the hand-crafted image captions. However, the improvements to the model still did not take into account the location of objects in the scene and thus could not provide spatial context for any image captions.

A framework was proposed that was based on a combination of a CNN model and LSTM RNN model that aimed to move toward providing a dense description of a scene in Deep Visual-Semantic

Alignments for Generating Image Descriptions [28]. The proposed framework attempted to infer the alignment between segments of a sentence and the region of the image they described, and to use a multimodal RNN that accepted an input image and generated a dense textual caption of the input image.

To detect the image regions, the framework used the R-CNN framework discussed in [18] and kept the top 19 detected locations in the image. To compute the sentence representations of the detected locations, the authors proposed a bidirectional RNN (BRNN) that learned the word representations of the visual components at each of the detected locations. To align the text segments to image segments, the framework treated each alignment as hidden variables in a Markov random field where the interactions between neighboring words resulted in alignment to the same region. The scene descriptions were then generated using the multimodal LSTM RNN. This RNN operated in a similar fashion to the model discussed in [65] with the addition of a conditioning process that conditioned the generative process of the RNN on the content of the input images, which resulted in more accurate contextual scene captions.

The proposed framework outperformed state-of-the-art models at the time of writing on both the Flickr30K and MS COCO [35] data sets in terms of precision and recall. The authors noted that the model was subject to limitations. The first limitation was that the framework could only perform at a fixed resolution. Secondly, the RNN only received image information from additive bias, which were less verbose than using multiplicative interactions between the CNN and the RNN models. Finally, the authors noted that this was not a single stage pipeline and converting it into one remains an open problem. The example figures listed in the paper do show that the model does generate concise, descriptive examples of testing images from the data set. While this model does use a CNN model that is well-suited for object detection, the only locality information used by the authors was to generate text snippets of each location. Thus, the model did not make full use of the spatial relationship information available when generating the final image caption.

The framework proposed in Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention [70] used an attention-based model to describe the contents of images. The authors showed via visualization how the model learned to fix its gaze on salient objects while generating image captions. The authors highlighted the ability to saliently gaze as it mimicked a human's ability to compress a vast amount of information into a descriptive language.

The framework approached caption generation by incorporating both a hard attention mechanism trainable by maximizing an appropriate lower bound and a soft attention mechanism trainable by standard backpropagation. Visualizing these attention mechanisms provided insight into what and where the attention was focused. Similar to [65], the framework used a CNN model, Google LeNet, to generate image labels and a LSTM RNN model to construct the image captions. To represent the hard attention component, the framework extracted the visual features from the CNN model when generating each word of the output image caption sequence. To generate the soft attention component, the approximate marginal likelihood of a vector representing all potential salient image features was optimized using backpropagation.

Experimentation showed that the proposed framework was able to achieve state-of-the-art performance in terms of BLEU and Metric for Evaluation of Translation with Explicit Ordering (METEOR) scores on the MS COCO [35], Flickr8K, and Flickr30K data sets. The authors also noted that the visual attention models corresponded very well to human intuition but there was no metric to judge this. While this model did use individual image features to determine which salient objects to focus on, the framework did not take into account positioning of the objects and thus could not provide spatial context regarding the image captions generated. However, the visual examples provided in the supplementary section of this paper do show that the addition of visual attention did allow for more verbose image captions that eliminated some of the noise that previous models could not.

2.3 Spatial Relationships in Images

There have been few works in the realm of image spatial relationships. This section presents two different frameworks, one based on hierarchical models and one based on machine-learned models. This research will implement the framework discussed in Section 2.3.1 combined with object detection methods in an attempt to improve upon the machine-learned models discussed in Section 2.3.2.

2.3.1 Frameworks Based on Hierarchical Models

The framework described in A New Way to Represent the Relative Position Between Aerial Object [40] introduced the concept of histogram of forces (HOF), or F-Histograms. The approach treated two-dimensional objects as longitudinal sections (one-dimensional objects) in order to leverage the power of integral calculus to compute the relationship between each object in an image. The spatial relationships between two objects were constructed in a building block manner: point-couple spatial relationships informed line segment couple spatial relationships; line segment couple spatial relationships informed longitudinal section couple spatial relationships; and finally, F-Histograms were computed on longitudinal sections that determined the degree of confidence in computed spatial relationships.

The framework took two images, Image A and Image B as input. Image A represented the argument image, whereas image B represented the referrant image. Images A and B were either a binary mask representing a crisp object in space, or a grey-scale mask with values in the range $[0-255]$ that represented a fuzzy object in space. As such, the F-Histogram framework was able to successfully compute either crisp or fuzzy spatial relationships between two objects. The primary goal of the framework was to answer the proposition "A is in direction α of B" where α was a finite set of spatial directional relationships.

As mentioned above, the spatial relationships were constructed in a building block manner. For the handling of points, each point formed the argument image and was compared with each point in the referrant image. Let M and N be a point of A and B respectively, then the equation

$$\varphi(X_M - X_N) \tag{2.2}$$

provided the weight that supported the proposition "A is in direction α of B." φ was computed for each point in A and B for each direction α in the finite set of directions. The result was a set of weights supporting each point couple's directional relationship for all finite directions.

To compute the relationships between the oriented line segments contained in the argument and referrant images, the couple (I, J) was used to support the proposition "A is in direction α of B." Both I and J could be described by points M and N described above. The authors noted that it followed naturally that the set of weights, f, describing the directional relationships between each oriented line segment (I, J) could be computed by summing the set of weights, φ, that described each point that composed the tuple (I, J).

$$f(x, y, z) = \int_{y+z}^{x+y+z} (\int_{0}^{z} \varphi(u - v)dv)du \tag{2.3}$$

Equation 2.3, in the general form, described the integral calculus solution to summing over each point in the combined line segments.

The authors noted that there exists one set, and only one, of mutually disjoint segments that can be constructed by the above mentioned line segments. These mutually disjoint segments represented the longitudinal segments for the argument and referrant images. The authors considered the couple $(A_{\theta v}, B_{\theta v})$, where θ was a direction from the finite set of directions and v a real-valued number, an argument put forward to support the proposition "A is in direction α of B." Again, it followed naturally to estimate the set of weights, F, supporting those arguments by summing the previously computed set of weights f that corresponded to the couple $(A_{\theta v}, B_{\theta v})$. The equation

$$F(\theta, A_{\theta v}, B_{\theta v}) = \sum_{i\in 1\ldots n, j\in 1\ldots m} f(d_{I_i}, D^{\theta}_{I_i J_j}, d_{J_j}) \tag{2.4}$$

represented the set of weights supporting the proposition for each longitudinal section. In Eq. 2.4, d_{I_i} represented the line segment from I, d_{J_j} represented the line segment from J, and $D^{\theta}_{I_i J_j}$ represented the distance between the line segments I and J.

Lastly, the authors described the computation of the F-Histograms, $F^{\theta}(A, B)$ over the argument and referrant objects. Building upon the weights computed for the longitudinal segments, F, the F-Histograms represent the total sum of the weights that support the proposition "A is in direction α of B." The equation

$$F^{\theta}_{2}(A, B) \tag{2.5}$$

represented the scalar result of elementary forces of gravity. In other words Eq. 2.5 represented the forces exerted by the longitudinal sections of the argument image, A, on the referrant image, B, each tending to move B in direction α. The authors noted that this property was why the function $F_2^\theta(A, B)$ was called the histogram of forces associated with (A, B) via F.

The authors noted that only disjoint objects were examined in this literature and that fuzzy relationships were computed using the level cuts of the fuzzy objects, which were crisp representations of the fuzzy objects. Experimentation was performed on test raster images that expressed the relationships: to the right of, to the left of, above, and below. Comparisons were performed with a previously implemented method, histograms of angles [42]. The authors conducted experiments using three different techniques, M(histograms of angles), M0 (general force F-Histograms) and M2 (gravitational force F-Histograms). Results indicated that the methods M and M0 were fundamentally equivalent. However, M preferred the horizontal and vertical directions, while M0 did not. M2 directly took metric information into account and produced specific histograms. For simple configurations, the results achieved by M and M2 were comparable and for some configurations they were not comparable at all. The authors noted that the "best" method depended on the application being considered. Finally, the authors noted that when M2 provided completely different results, the opinions expressed by M2 were still able to be completely explained rationally and these opinions were not able to be expressed by M because histograms of angles did not take metric information into account.

2.3.2 Frameworks Based on Machine-Learned Models

To date, very little work has been performed regarding spatial relationships of objects in images in the context of scene understanding. Some initial work used a CNN image classification model coupled with the use of a heat map to designate the important areas of the image.

In Identifying Spatial Relations in Images using Convolutional Networks [20], a framework was described that used the VGGNet CNN image classification model coupled with a proposed heat map to extract spatial context for objects in the scene in an attempt to provide better captions. The framework used image features extracted from the VGGNet model as inputs to a multi-layer perceptron, which generated the corresponding heat map for each image. The heat map in this framework modeled image features in the range of not important (blue) to most important (red).

Once the heat map was constructed, it was used to determine the spatial relationships between each object in the image by examining the positions of the red regions of the heat map since each object in the image had a high likelihood of being represented in the red (most important) range. The framework developed generated descriptions such as “a man is beside a dog” or “a ball is behind a lamp.” For the simple case where the objects were juxtaposed, the heat map picked the first object encountered and constructed the description of the form “object x is beside object y.” For the more complicated case where the objects overlap, the framework used the heat map to determine which object was in the foreground and which object was in the background to generate descriptions of the form “object y is behind object x.”

To verify their model, the authors used a synthetic data set and tested the accuracy of the CNN model. The model achieved 69% accuracy on the synthetic data set. No baseline existed for the spatial relationship accuracy, so the authors reported their findings in terms of learning the expected spatial relationships given the synthetic data set. The authors noted that the model performed with 100% accuracy when classifying the beside and behind relationships independently. However, the authors also noted that when incorporating images with the “beside”, “behind”, and “above” relationships present, the accuracy drops to 8.4% for beside and 20.43% for behind.

This research presented an initial approach to incorporating spatial relationships between objects in a scene understanding system. However, the proposed framework only generated relationships for objects beside or behind each other. To construct a more robust and descriptive system, accounting for object bounds and using them to compute spatial relationships would provide more fine-grained detail than the heat maps presented in this model could. Also, the framework presented in this paper does not account for the logical ordering of objects when generating scene labels. This was likely due to the proposed model using only the object labels and spatial orderings to construct the final semantic description.

2.4 Summary

This chapter explored previous work in the areas of object detection, S2T systems, and computing spatial relationships in images. Much of the existing work will be leveraged in order to construct the S2T system in Chapters 4 and 5. The existing S2T systems alone do not provide any concept of object-to-object localization. A primary goal of the system was to leverage the work in the area of object spatial relationships in order to provide more informative scene descriptions by using object-to-object localization information.

CHAPTER 3

OBJECT LOCALIZATION

This chapter presents the implementation and results for several existing frameworks that were used to extract various pieces of information from images. Information obtained from these existing frameworks includes: image object segmentation via mask pixel segmentation, object bounding boxes, and image category labels.

To begin, this chapter presents a diagram of the S2T pipeline that was implemented in this paper along with examples of original unprocessed images that were passed through the pipeline shown in Figure 3.1. A description of the implementation methods used for object segmentation and metadata generation along with results are presented next. Results were analyzed to determine which results met the criteria to be used as input to the Spatial Relationship stage of the pipeline shown in Figure 3.1. Also, an analysis and explanation of results that were “good” detections, but not suitable to be used as input to the Spatial Relationship evaluation is presented.

The aforementioned frameworks were used as-is, out of the box. As such, validation was not performed on the methodologies presented in this chapter. However, a discussion of previous validation results from the original works are presented. The goal of this activity was to provide meaningful input to the evaluations conducted in Chapter 4. Therefore, the results analysis in

Section 3.6 serves as a means of validating the useful results that proceed to the next stage of the pipeline.

3.1 S2T Pipeline

To begin, let us start by exploring each stage of the S2T system as represented by the pipeline shown in Figure 3.1. The system starts by accepting an image in its unprocessed form, such as the example images shown in Figures 3.3(a) - 3.8(a), as input. The image is then processed via the image segmentation and image labeling methods, which are the primary focus of this chapter. The "Object Localization" stage shown in Figure 3.1 outputs a set of masks or bounding boxes, that are used as input to the "HOF/GIOU Computation" phase discussed in Chapter 4. The Inception [61] and Resnet [22] models provide image metadata that is used as input to the algorithms discussed in Chapter 5 in conjunction with the spatial relationship results generated in Chapter 4.

The "HOF/GIOU Computation" task consists of two stages, Generalized Intersection Over Union (GIOU) to determine object overlap and HOF to determine locality between object tuples in an image. The result of the HOF/GIOU computation are the Level One Summaries for the raw input image. These Level One Summaries are then used as input to the Fuzzy Inference System (FIS) Computation stage of the pipeline. It is important to note that Level One Summaries can be used as the final system output as is shown in Figure 3.1. Level One Summaries, viewed as a system output, consist of the spatial relationship information for each localized object tuple in an image. Consider the image in Figure 3.3(a). The Level One Summary for this image would be of the form: "person_1 overlaps and is above surfboard_1." Thus, the Level One Summaries themselves do provide meaningful information when used as system output.

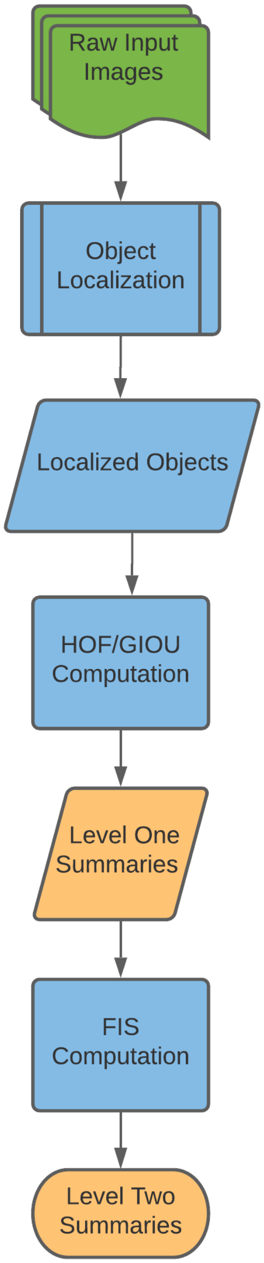


Figure 3.1

S2T System Pipeline

Each Level One Summary serves as input to the FIS Computation stage of Figure 3.1. This stage generates a set of labels that describe both the general interactions between all object tuples in an image, which we denote as “general category interactions” and the interactions between a person or persons and objects in an image, which we denote as “person domain specific interactions.” Given that an image has no person(s) detected, the general category interactions are returned as the output of the S2T system. When an image does have a person(s) detected, both the general category interactions for all object tuples, including people, and the interactions between people and each object in the image is returned as the S2T system output. The general category and person domain interactions both utilize their own sets of fuzzy rules for the FIS Computation stage of Figure 3.1. Chapter 5 explains the construction of the fuzzy rules and the FIS in greater detail. Consider again the image in Figure 3.3(a). The general category interactions generated by the FIS Computation stage of Figure 3.1 for this image would be of the form: “person_1 interacting with surfboard_1.” Additionally, the person domain interactions generated by the FIS Computation stage would be of the form: “person_1 riding surfboard_1” since the person both overlaps and is in the “above” direction of the surfboard. These sets of labels are the final output of the S2T system constructed in this research for the raw input image shown in Figure 3.3(a). As mentioned in Chapter 1, a primary focus of this research is to determine if spatial relationship information can be leveraged to construct more informed scene descriptions than previous works have accomplished. Consider the images in Figure 3.3(a) and 3.4(a). Current S2T systems may generate the same label of “person riding surfboard” but clearly in Figure 3.4(a), the person is holding and not riding the surfboard. Chapters 4 and 5 expand upon the use of spatial relationship information in constructing the output Level Two Summaries as shown in Figure 3.1 and show that incorporation of such information does allow the system to differentiate between a person holding and a person riding a surfboard, among various other interactions.

3.2 Object Localization via Pixel-wise Segmentation

This section discusses the evaluation design and implementation of the Mask-RCNN [21] algorithm. Mask-RCNN generates pixel level object localizations and provides a tighter bound than bounding box segmentations. As mentioned above, mask localizations are the preferred method used in this pipeline.

3.2.1 Mask-RCNN Implementation

This section details the Mask-RCNN [21] implementation and evaluation design including the software packages and libraries used, descriptions of how the mask image segmentation was performed, and details how the results of the evaluation were stored for retrieval. Afterwards, a small sample of the Mask-RCCN results are presented.

3.2.1.1 Software and Libraries Used

The Mask-RCNN algorithm was implemented using the OpenCV [5] open source framework using the Python [45] programming language. An Anaconda [2] virtual environment was used to allow the software designed for this evaluation to be easily shared between machines. The Tensorflow [1] library provided weights for the Mask-RCNN model that were pre-trained on the COCO [35] data set. As such, no model training for the Mask-RCNN model was required and the model was used as-is out of the box. Results were stored using a combination of Pandas [62] data frame, CSV files, and Numpy [43] arrays.

3.2.1.2 Evaluation Design Details

Algorithm 3.1 Object Localization

```
Input: Raw input images from the COCO data set
Output: Object localization results for the input image
for all $img \in$ COCO Images do
    Set $img_{mask} = MaskRCNN(img)$
    Set $img_{yolo} = YOLOv3(img)$
    Set $img_{ssd} = SSD(img)$
end for
```

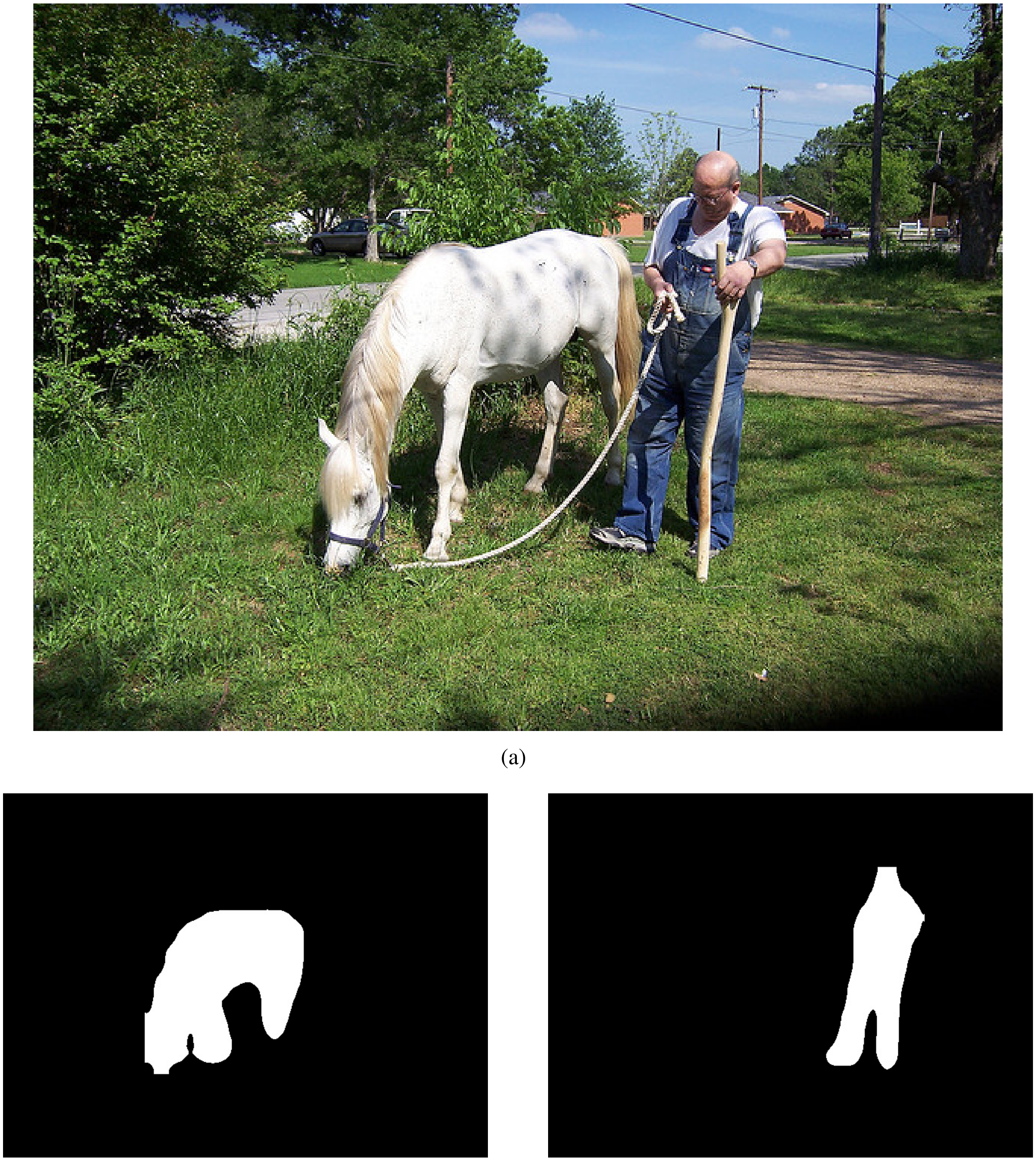
Algorithm 3.1 shows the general process used to obtain object localization results. The process is performed for each of the images in the COCO [35] data set and the output are the three sets of object localization results for each of the input images in the data set. Mask RCNN [21], YOLOv3[48], and SSD [37] were all applied to each input image to generate object localization results. The algorithm returns each of these sets of results as individual output sets. The mask segmentations were the preferred method for this research, as they provide tighter object bounds. However, as Algorithm 3.1 shows, if no mask results were available, the system does generate bounding box results using both the YOLO and SSD algorithms. Doing this allowed the system to have “fallback” results if an image contained no mask segementations. It is important to note that it is possible that none of the object localization algorithms generate valid results for an image. In these instances, the object localization results are set to the empty set, and thus are not used in Level One or Level Two Summary, Chapters 4 and 5 respectively, computation.

Each image from the COCO [35] data set was loaded from disk using the OpenCV [5] library and processed sequentially using the algorithm described in this section. The Mask R-CNN [21] model, set up as described in Section 3.2.1.1, was used to process each of the images.

The Mask R-CNN [21] model generated the following results for each object in an image: a set of coordinates for the bounding box surrounding the object, a class label for the object, a confidence score indicating how confident the model was that the label was accurate, and a set of coordinates that corresponded to the pixels defining the object mask. It was possible for the Mask R-CNN model to detect the same object multiple times. In order to counteract this, non-maxima suppression (NMS) [24] with a threshold value of 0.5 was performed on the bounding box output to ensure that only boxes with the tightest bounds around each object were preserved. The threshold value was determined by choosing 100 sample images from the COCO [35] data set, varying the threshold value, and then inspecting the bounding box results to determine the threshold that provided the most accurate bounding boxes for the 100 sample images. The threshold was varied between 0.1 and 0.9 during evaluation. For mask segmentation, a threshold value above 0.5 generated too many undesirable extra localization outputs as artifacts, where a threshold value below 0.5 removed too many good localization outputs from the data set to be considered reliable. Optimizing the NMS threshold for Mask R-CNN is beyond the scope of this work, and is the subject of other research in the field of object detection. The S2T system constructed in this research assumes *a priori* optimization of any object detection/segmentation algorithm used to generate the object localization data set to input to the evaluations conducted in Chapters 4 and 5.

The pixel-wise masks were then generated on an object-by-object basis using the chosen bounding box for each object. Each mask was represented as a two-dimensional array where a pixel coordinate that lies within the object had a value of 255 and pixels that do not had a value of 0. The masks had dimensions that corresponded to the width and height of the bounding box for the object in question. The end result of mask generation was *n* numerical arrays, where *n* was the number of objects in an image and each segmentation mask had dimensions $w \times h$ where w and h were the width and height of the bounding box that corresponded to the object being extracted. As an example, in Figure 3.7(b) there were four objects: person 1, person 2, person 3, and motorcycle 1. Thus, there were four bounding boxes extracted, one for each object in the image. The pixel-wise masks were then extracted using the bounding boxes for each object.

All of the evaluation results discussed in this section were stored for retrieval to avoid recomputing the results. Bounding boxes, labels, and confidence scores for each image were written to Pandas [62] data frames during the Mask R-CNN [21] evaluation. Pixel-wise segmentations were stored to disk as Numpy [43] "npy" compressed binary files. Along with the numerical arrays, pixel-wise segmentation masks were stored to disk as a grayscale image. The grayscale images were required as input to the HOF [40] algorithm discussed in Chapter 4. These images were constructed by creating an image the same size as the original input image and setting all pixel values that do not belong to the mask to 0 (black) and all pixel values that belong to the mask to 255 (white). An example of segmentation masks is shown in Figure 3.2 (b) and (c) for the original image in (a). Similar to the Numpy [43] arrays, each image had n grayscale mask images associated with them, where n was the number of objects detected in an image and these masks were stored in a directory that corresponded to the image path. All values from the Pandas [62] data frames were then stored in CSV files for retrieval. Along with the bounding boxes, labels, and confidence scores, the image relative path and directory to the mask greyscale images and mask numerical arrays for each image were also stored in the CSV file.



(a)

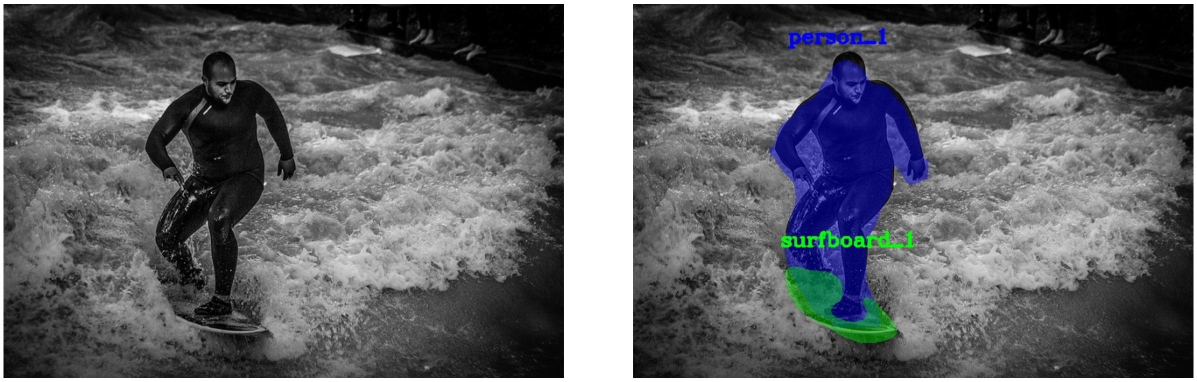
(b)

(c)

Figure 3.2

Pixel-Wise Segmentation Masks

(a) Original image (b) Horse pixel mask (c) Person pixel mask

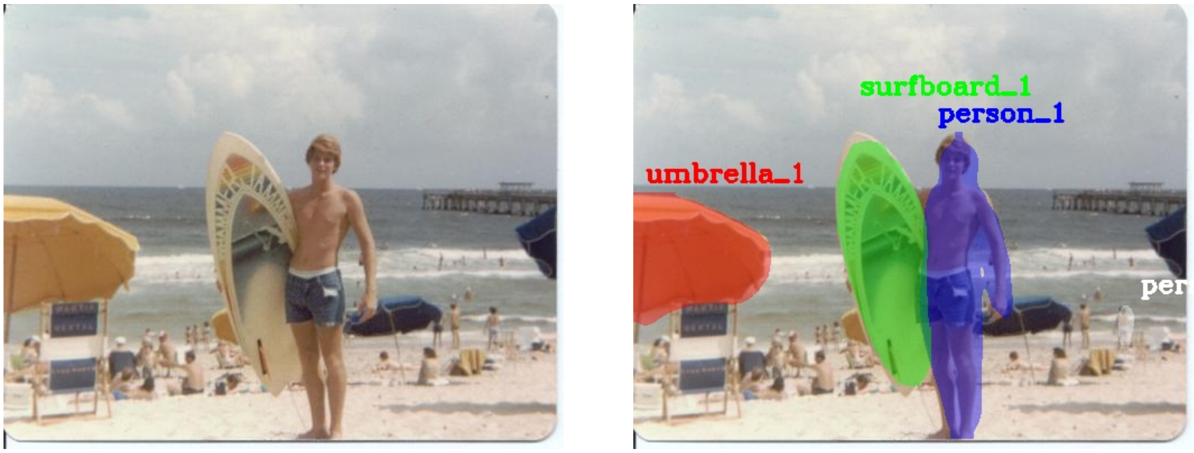


(a) (b)

Figure 3.3

Person riding surfboard

(a) Original image (b) Mask localization output

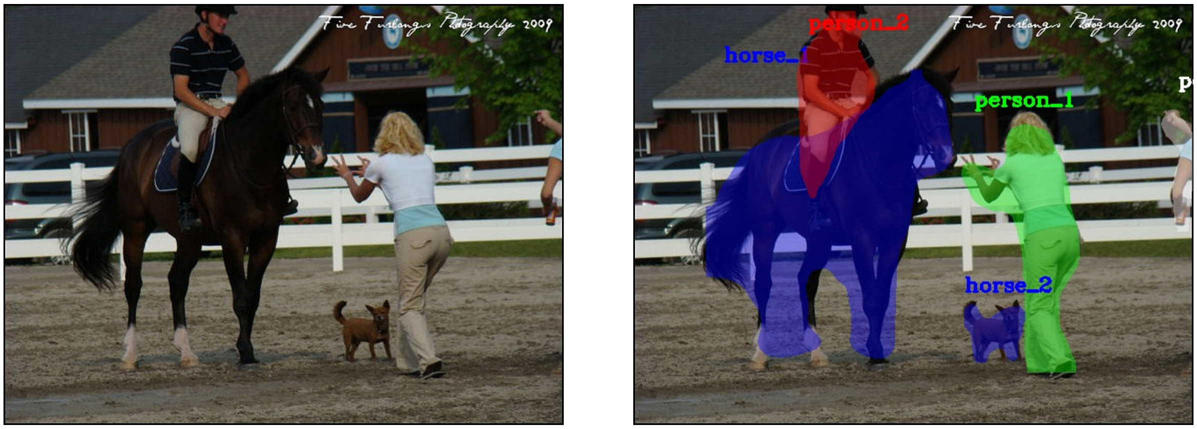


(a) (b)

Figure 3.4

Person holding surfboard

(a) Original image (b) Mask localization output



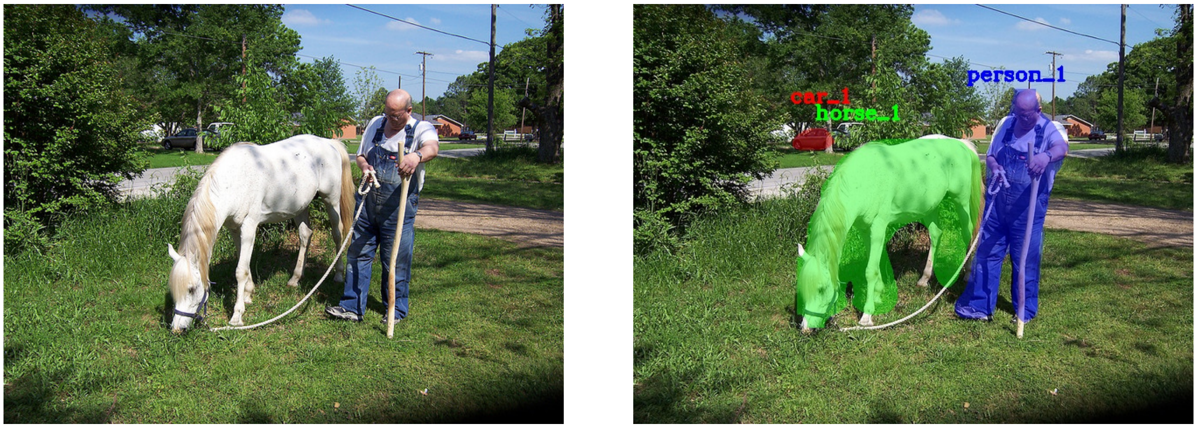
(a)

(b)

Figure 3.5

Person riding horse

(a) Original image (b) Mask localization output



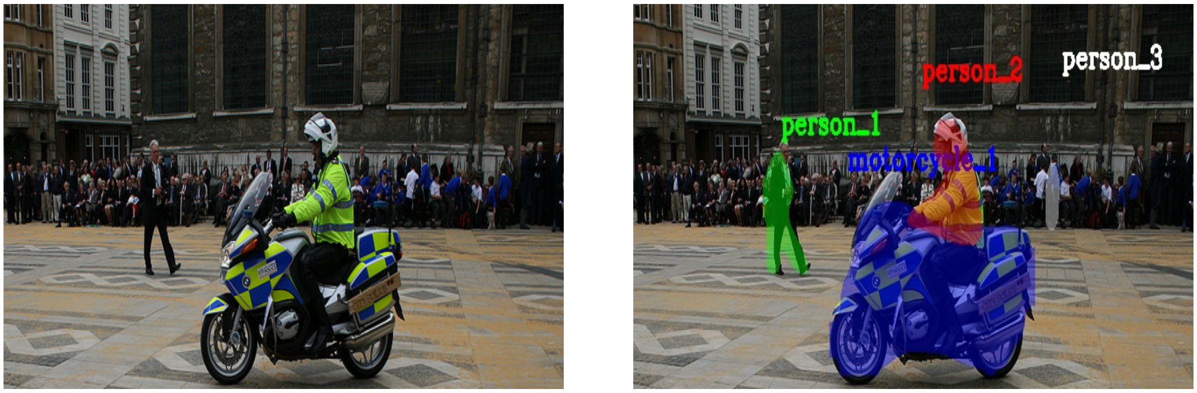
(a)

(b)

Figure 3.6

Person beside horse

(a) Original image (b) Mask localization output



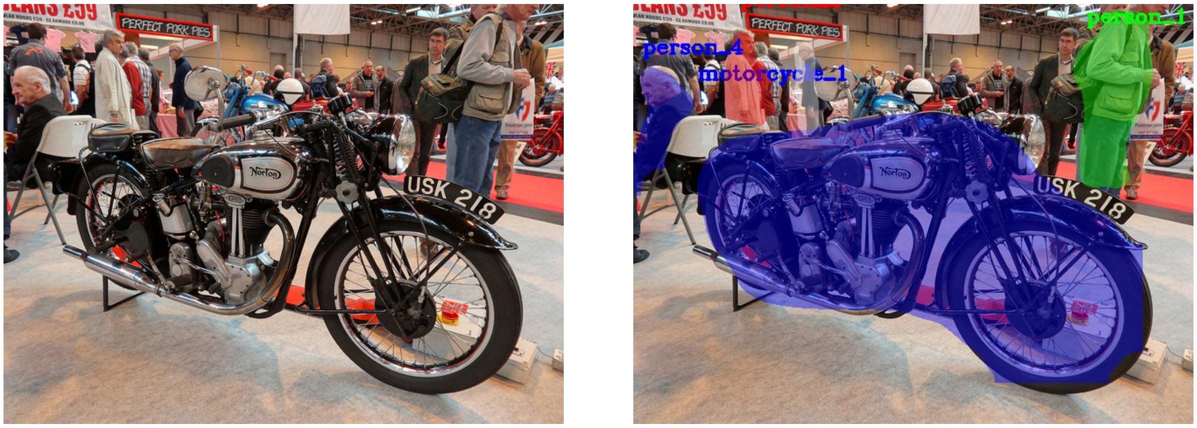
(a)

(b)

Figure 3.7

Person riding motorcycle

(a) Original image (b) Mask localization output



(a)

(b)

Figure 3.8

People beside motorcycle

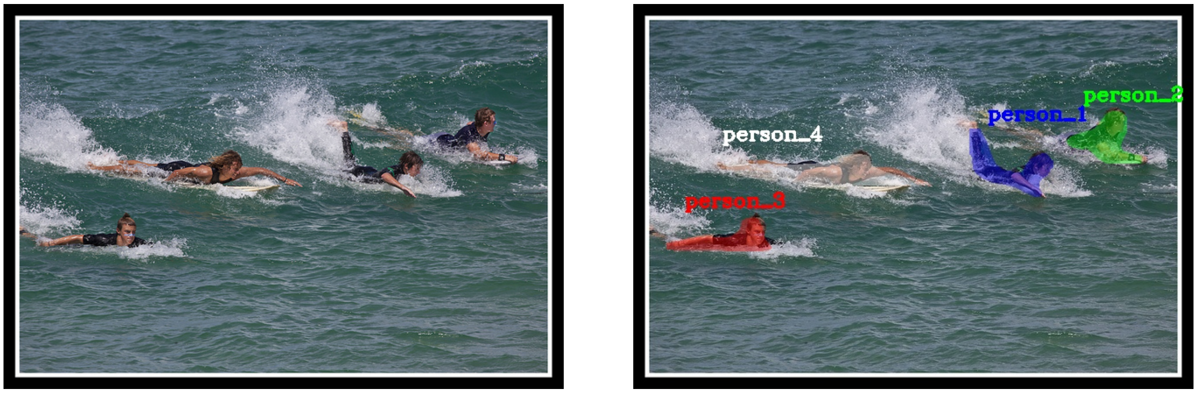
(a) Original image (b) Mask localization output

3.2.1.3 Evaluation Results

Figures 3.3(b) - 3.8(b) visualize the results generated by the Mask R-CNN [21] for the original images in Figures 3.3(a) - 3.8(a). In each figure, (a) shows the original unprocessed image, where (b) shows the localization output obtained by applying pixel-wise mask segmentation. Visualization of these results was performed using the OpenCV [5] library via loading the original image and the pixel-segmentation mask arrays discussed in Section 3.2.1.2. After loading the masks from disk for the original image, each pixel coordinate that corresponded to an object was overlaid with a transparent color, chosen randomly, to signify the pixels that correspond to the detected object. The labels for each object, also discussed in Section 3.2.1.2, were placed above the detected object and assigned the same color as the mask pixels to signify the label and mask pixel correspondence.

Labels for each object were assigned a sequential numbering scheme, which was necessary for the spatial relationship evaluation discussed in Chapter 4. For example, consider Figure 3.5(b), Without a way to distinguish between each object of the same class, it would be impossible to determine which person was riding the horse. Additionally, Figure 3.5(b) also shows that object segmentation results are not always perfect since the dog in the image also was detected as a horse. Given this, the question is now “which person is riding which horse?” The numbering scheme for like objects in images allowed us to pass more useful information to the evaluations in Chapters 4 and 5 so that these questions can accurately be answered and incorrect labels for individual objects in images become less of a concern.

Section 3.6 discusses the results that were deemed usable as input to the spatial relationship evaluation discussed in Chapter 4 and Section 3.6 details the inspection process used in order to determine the usable results. There were instances where the Mask R-CNN [21] model generated good object localization results, but these results were not usable for computing the scene descriptions described in Chapter 5, such as the images in Figure 3.9(b) where there is only one class of object detected, a person. The detection results were accurate, however, there were no objects detected with which a person could interact. Even when using the metadata generated via the Inception [61] model, there would be no way to generate a scene description for this image. However, these localization results can still be used as input to the spatial relationship evaluations of Chapter 4. The pipeline process simply stops after computing spatial relationship information in these cases.



(a) (b)

Figure 3.9

Good Mask R-CNN Localization; Unusable for Level Two Summaries

(a) Original image (b) Mask localization output

3.3 Object Localization via Bounding Boxes

This section discusses the evaluation design and implementation of the bounding box localization algorithms used in the pipeline. Algorithm 3.1 shows that both YOLOv3 [48] and SSD

[37] were applied to each of the images in the COCO [35] data set. Bounding box localization results were used when there were no available mask results, and this section discusses the results obtained by applying both the YOLO and SSD algorithms to each of the input images. Both methods generate the same type of results, so the remainder of the discussion will just simply refer to "bounding boxes" as opposed to the differentiation between the methods. As the name suggests, bounding box localization generated bounding boxes for each object detected in an image.

3.3.1 Bounding Box Localization Implementation

This section details the implementation of the two bounding box localization methods used for the S2T pipeline. The packages and libraries used are listed, followed by an explanation of the evaluation design. Lastly, a small sample of bounding box localization results are visualized and discussed.

3.3.1.1 Software and Libraries Used

Bounding box localization used two different models, YOLOv3 [48] and SSD [37]. The OpenCV [5] library was used for both of the models and all code was implemented using the Python [45] programming language. An Anaconda [2] virtual environment was used in this evaluation as well to allow the software to be easily shared between machines. The Tensorflow [1] library provided pre-trained model weights for the SSD [37] model that were trained on the COCO [35] data set. YOLOv3 pre-trained model weights for the COCO data set are available for download alongside the paper from [48]. As with the Mask R-CNN [21] implementation discussed in Section 3.2.1.1, the YOLOv3 and SSD models were used as-is out of the box, so no model training was performed.

As mentioned previously, Mask R-CNN [21] segmentation is the preferred method for localization, but bounding box localizations were also computed to ensure there were as many usable localization results as possible for all images in the data set. The methods discussed in this section and in Section 3.2 were applied to each image in the data set, but not all results were used as input to the evaluation outlined in Chapter 4. Section 3.6 describes the methods used for choosing which localization results were used for each image in the data set.

3.3.1.2 Evaluation Design Details

The process for computing bounding box localization was similar to the process for Mask R-CNN [21] localization described in Section 3.2.1.2. Each image from the COCO [35] data set was loaded and processed sequentially through the YOLOv3 [48] and SSD [37] models using the OpenCV [5] library via the setup discussed in Section 3.3.1.1. Both the SSD [37] and YOLOv3 [48] models generate the same set of information as output from a forward pass through the model. That information included the following for each object detected in an image: a set of coordinates for the bounding box surrounding the object, a class label for the object, and a confidence score that indicates how confident the model was that the class label was accurate. Similar to Mask R-CNN [21], it was possible for both YOLOv3 and SSD to generate multiple bounding boxes for the same object in an image. NMS [24] with a threshold of 0.4 for SSD and 0.7 for YOLOv3 was used to eliminate duplicate detections. These threshold values were chosen by using a random sample of 100 images from the COCO [35] data set, using varying threshold values for each model, and inspecting the bounding box detections to determine the optimal threshold value for the sample images. Similar to the NMS [24] threshold selection for mask segmentation, the threshold was varied between values of 0.1 and 0.9. Values lower than 0.4 and 0.7 for SSD and YOLOv3 respectively, removed too many correct bounding box outputs. Values higher than 0.4 and 0.7 for SSD and YOLOv3 respectively, generated too many artifact bounding box predictions, making the bounding box outputs unreliable. As mentioned previously, it is beyond the scope of this work to determine the optimal NMS [24] threshold for SSD and YOLOv3 in the general case and it is assumed that the bounding box localization method used in the constructed S2T system has been previously optimized to generate accurate object localization data to use in the evaluations conducted in Chapters 4 and 5.

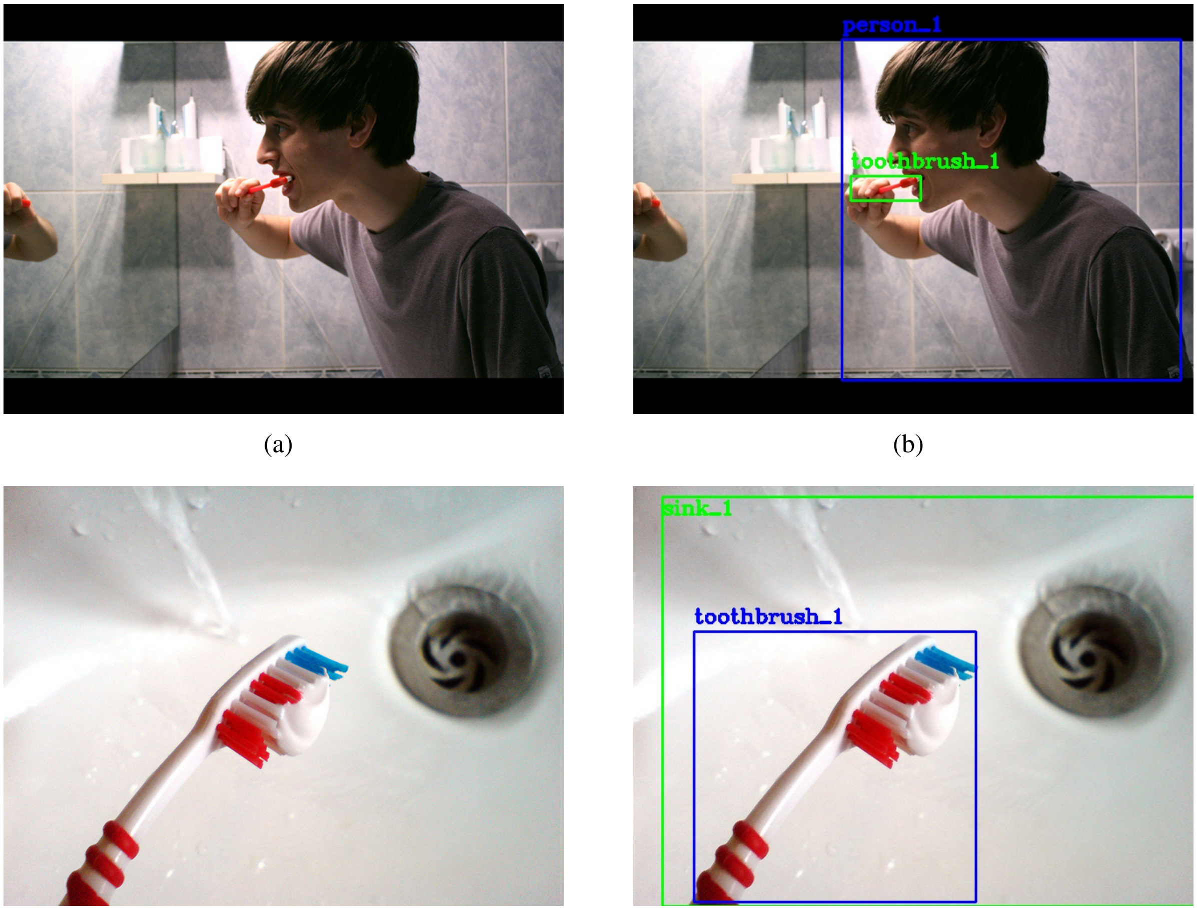
Unlike the Mask R-CNN [21] model, the bounding boxes were the localization results for the YOLOv3 [48] and SSD [37] models, so no further processing was required after they were generated. The end result of bounding box localization was *n* sets of bounding box coordinates where *n* was the number of objects detected in an image. Consider the bounding box localization results in Figures 3.10(b) and 3.10(d) for the raw images shown in Figures 3.10(a) and 3.10(c) respectively. There are two object localization results for Figure 3.10(a): “person 1” and “toothbrush 1”. Similarly, there are two localization results for Figure 3.10(c): “toothbrush 1” and “sink 1”. Thus, in each of these original images, two bounding boxes were extracted.

Again, all of the evaluation results discussed previously in this section were stored for retrieval to avoid repeated computation of the results. Bounding boxes, object labels, and confidence scores were written to Pandas [62] data frames during computation. As mentioned above, both SSD [37] and YOLOv3 [48] model results were computed, thus there were two different sets of results for the COCO [35] data set written out to Pandas data frames, one for the YOLOv3 results and one for the SSD results. All values stored in Pandas data frames were then stored in CSV files for retrieval with an indicator of the type of detector used so results for an individual model could be retrieved.

3.3.1.3 Evaluation Results

Figures 3.10(b) and 3.10(d) visualize the bounding box results generated for the original images in Figures 3.10(a) and 3.10(c) respectively. Visualization of these results was performed using the OpenCV [5] library via loading the original image and using the library to draw the bounding boxes and labels computed in Section 3.3.1.2 on the image. Bounding box colors were chosen randomly and the class labels that correspond to a bounding box use the same color as the bounding box and were placed above the bounding box in the image. Similar to mask localization, labels for each object used a sequential numbering scheme to allow for usability in the spatial relationship evaluations detailed in Chapter 4.

Section 3.6 discusses bounding box localization results that were deemed usable as input to the evaluations implemented in Chapter 4 and Chapter 5 as well as the methods used to determine which bounding box localization results were usable. As with mask localization, there were instances where bounding box localization generated good object localization results which were not usable for the evaluation in Chapter 5. Consider the bounding box results shown in Figure 3.11(b) for the original image Figure 3.11(a). In this instance, there was only one class of object detected, apples. There are no interactions that apples can logically perform, so this would not be a suitable result to use for the S2T system. Figure 3.11(d), on the other hand, shows two different classes of objects for the original image shown in Figure 3.11(c): an oven and a microwave. There are no reasonable interactions that these two objects can perform, so this is another example of good



(a)

(b)

(c)

(d)

Figure 3.10

Bounding Box Localization Results

detection results that were not suitable to use as input to the evaluations in Chapter 5 that generate scene descriptions. Although these localization results cannot be used as input to the evaluations of Chapter 5, Level One Summaries, detailed in Chapter 4, can still be computed. The S2T pipeline will simply return the Level One Summaries as a final output as opposed to computing the scene descriptions.

3.4 Metadata Generation via Inception

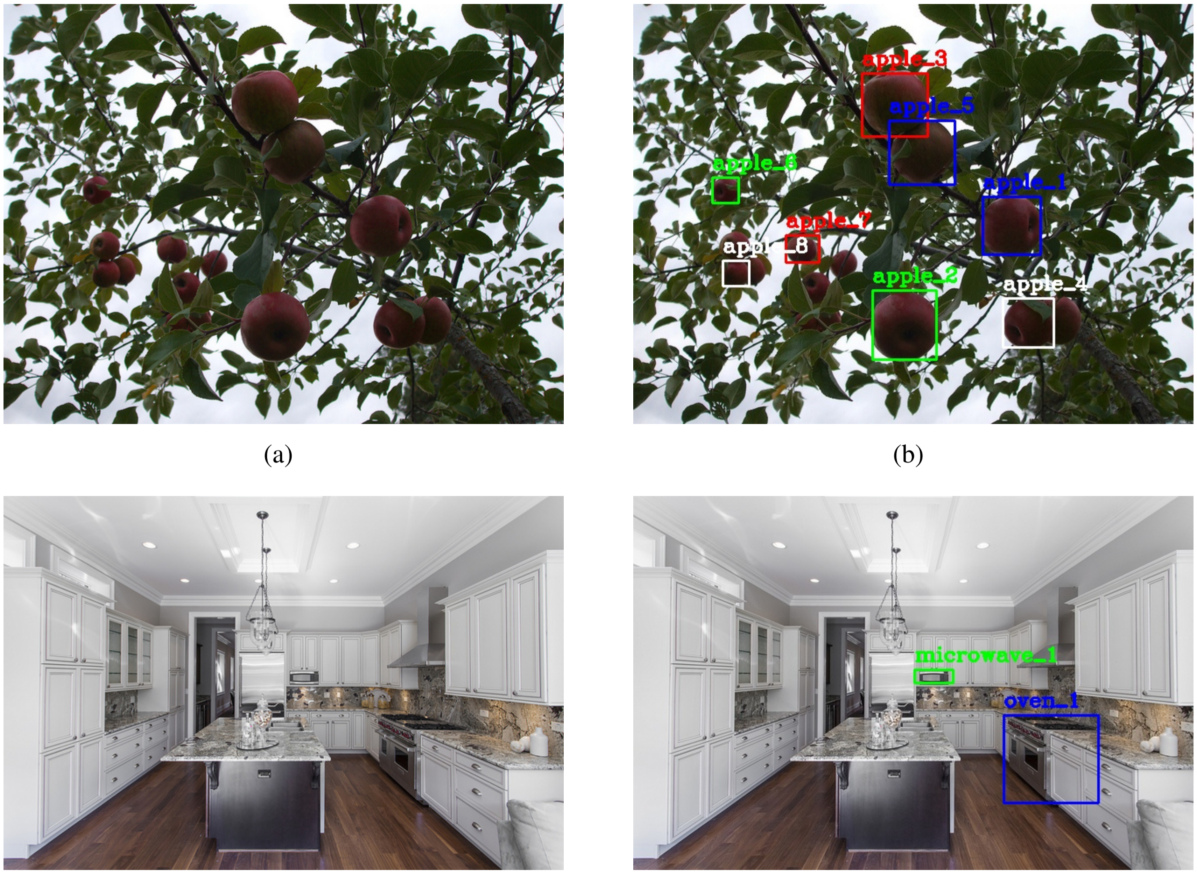
This section discusses the evaluation design and implementation of the metadata generation algorithm used in the pipeline. Metadata will be used as input to the evaluations in Chapter 5 to provide additional information about the contents of an image. Unlike the previously discussed localization algorithms, metadata generation does not provide a localization of objects in an image.

3.4.1 Metadata Generation Implementation

This section outlines the implementation of the metadata generation algorithm. First, the libraries and software packages used are presented, followed by an explanation of the evaluation design. This section concludes by providing an example of when metadata generation will be used as additional input to the Level Two Summaries constructed in Chapter 5.

3.4.1.1 Software and Libraries Used

Metadata was generated using the Inception [61] model loaded using the OpenCV [5] library. Pre-trained model weights were provided by the Caffe [26] library for the Inception model. No training on the model was performed as it was used out of the box as-is. All of the evaluation



(a)

(b)

(c)

(d)

Figure 3.11

Good Bounding Box Localizations; Not Usable for Level Two Summaries was performed using the Python [45] programming language and all packages were loaded into an Anaconda [2] virtual environment to allow the code to be easily shared.

3.4.1.2 Evaluation Design Details

Each image from the COCO [35] data set was loaded and passed through the Inception [61] model using the OpenCV [5] library per the setup described in Section 3.4.1.1. Once an image was processed through the Inception model, a set of labels and confidences were returned as output. These labels corresponded to objects the Inception model believed exist in the image and the confidence scores corresponded to how confident the Inception model was that the labels were accurate. If more than five labels were detected, only the top five labels for each image were preserved as evaluation showed that result quality quickly deteriorated after the first five labels.

As each image was processed, the top five label results were stored in Pandas [62] data frames for each image. After all images were processed, the information stored in the data frames was written to a CSV file for retrieval in order to avoid repeated computation.

3.4.1.3 Evaluation Results

As mentioned in Section 3.4, the metadata results were used in order to provide additional information to the inference of Level Two Summaries discussed in Chapter 5. This metadata information became useful when object localization labels were vague. A brief example of such a scenario follows:

Consider the image in Figure 3.12. The localization methods discussed in Sections 3.2 and 3.3 detected the soccer ball as a “sports ball.” For the Level One Summaries, discussed in Chapter 4, the vague label did not make much difference since we only cared about where objects were in relation to one another. However, for the Level Two Summaries computed in Chapter 5, the vague label did not provide the necessary information that inferred the people in the scene were playing soccer. Inception [61] labels allow this information to be extracted from images and thus allowed for more well informed input to the Level Two Summary computations. The Inception labels for Figure 3.12 were: "soccer ball" and "rugby ball", a much more accurate description compared to "sports ball" with regards to Level Two Summary input information.

3.5 Existing Framework Validation

As mentioned previously, all of the models used in this evaluation were used as-is and have been previously validated. This section gives a brief discussion of how each model was validated, as well as the performance of the model on the validation data.

Mask R-CNN [21] was validated on the COCO [35] data set and achieved a mAP score of 58.0% during validation. The underlying architecture used during validation was the Resnet-FPN [36] [22] architecture. Mask R-CNN achieved the highest mAP score of any of the models used in this chapter on the COCO data set. All validation details for the Mask R-CNN architecture can be found in the original work in [21].

The YOLOv3 [48] model was validated on two data sets. For the Pascal VOC [14] data set, the model achieved a mAP score of 77.8%. For the COCO [35] data set, the model achieved a mAP score of 41.9%. All validation details for the YOLOv3 model can be found in the original work in [48].

The SSD [37] model was also validated on the Pascal VOC [14] and COCO [35] data sets. SSD achieved a mAP score of 74.3% on the Pascal VOC data set, and a mAP score of 43.5% on the COCO data set. All validation details for the SSD model can be found in the original work in [37].

The Inception [61] model was validated during the Imagenet Large Scale Visual Recognition Competition [54] on the data set for the challenge. The model achieved an error rate of 6.67% and was the top performing model for the challenge. All validation details for the Inception model can be found in the original work in [61].

3.6 Discussion

This chapter detailed the methods used to generate input information to the Level One and Level Two Summaries discussed in Chapters 4 and 5 respectively. As has been previously mentioned, pixel-wise mask segmentation was the preferred method used for object localization, with bounding box localization being used as backup. The evaluations detailed in this chapter in effect implement the “Object Localization” stage of the S2T system pipeline in Figure 3.1 and the output of the “Object Localization” stage were the “Localized Objects” in regards to the figure. It is worth noting that this section discusses only the results for the localization methods of Sections 3.2 and 3.3. As Inception [61] results are only used as metadata, we will refer to Section 3.4.1.3 for the results analysis.

The evaluations described in this chapter first created an “Object Localization” data set out of the images of the COCO [35] data set. This object localization data set was then further refined to remove any data points that were not useful for the evaluations discussed in Chapter 4. An additional refinement was then performed that pruned any data points that were deemed unusable for Level Two Summary computation, described in Chapter 5, from the object localization data set. The remainder of this section describes the final object localization data set in detail.

3.6.1 Object Localization Data Set Description

The COCO [35] data set contained a total of 1558 images. These were the “Raw Input Images” of the pipeline in Figure 3.1. As shown in Table 3.1, 571 images contained good detections suitable for input to the Level One Summaries, 506 images contained detections suitable for input to the Level One and Level Two Summaries evaluations, and 987 images did not contain any useful information for either Level One or Level Two Summaries.

Table 3.1

Localization Data Set Totals

	Total
Input Images	1558
Level One Usable	571
Level Two Usable	506
Unusable	987

Usable information for Level One and Level Two Summaries are briefly discussed in both Section 3.2.1.3 and 3.3.1.3, but it is worth revisiting the topic in this section. Usable information for the Level One Summaries evaluation of Chapter 4 can be viewed as an image with multiple objects, regardless of if they are the same class or not, whose localization outputs were good. In other words, Level One Summaries can be constructed on an image as long as there are multiple good localization outputs for an image. Figures 3.3(b), 3.4(b), 3.5(b), 3.6(b), 3.7(b), 3.8(b), 3.9(b), 3.10(b), 3.10(d), 3.11(b) and 3.11(d) all show localization outputs that are usable as input for Level One Summaries because each image contained multiple good localization outputs.

Usable detections for Level Two Summaries, described in Chapter 5 are a subset of the usable detections for Level One Summaries. In addition to the multiple good object localization outputs requirement, the Level Two Summary inputs require that more than one class of object has been detected. Furthermore, the classes of objects detected must be able to reasonably interact with each other. In other words, in order to infer Level Two scene descriptions for an image, the image must have two or more classes of objects within that can logically interact. As an example, the localization outputs in Figures 3.3(b), 3.4(b), 3.5(b), 3.6(b), 3.7(b), 3.8(b), 3.10(b) and 3.10(d) all contain more than one class of object and there are logical interactions between at least two of the object classes in an image. Consider Figure 3.10(b), the logical interaction between “person 1” and ”toothbrush 1“ is “person 1 brushing teeth”. As an additional example, consider Figure 3.3(b) where the logical interaction between “person 1” and “surfboard 1” is “person 1 riding surfboard 1”. These are examples of Level Two Summary outputs of the S2T system. It is left to Chapters 4 and 5 to provide in depth details on how these scene descriptions were inferred. Table 3.2 gives detailed counts for pixel-wise and bounding box localization data points in the object localization data set and Table 3.3 gives total counts of the rejected images/localization outputs.

Table 3.2

Usable Localization Totals

	Total
Mask Level One Usable	520
Mask Level Two Usable	460
Bounding Box Level One Usable	51
Bounding Box Level Two Usable	46

Table 3.3

Unusable Localization Totals

	Total
No Localization Results	58
One Localization Result	511
Poor Localization Results	418

It is important to also discuss the data points that were not usable for either of the evaluations in Chapters 4 or 5. The first case were images that contained no localization outputs at all. In total, there were 58 images from the COCO [35] data set where the Mask R-CNN [21], YOLOv3 [48], and SSD [37] models detected no localization output. A second case where data points did not meet the criteria for either Level One or Level Two Summaries was when there was only one object localization in an image. In total, there were 511 images where only one localization was found no matter the localization model used. A final case where localization data points did not meet the criteria for Level One or Level Two Summaries were images where the localization data points were poor. Poor localizations included: localizations where the masks/bounding boxes were in the wrong location for an object, object class labels were inaccurate and could not be used, or masks/bounding boxes were too large or too small for proper localization. In total, there were 418 images from the COCO data set that contained poor localization results.

3.6.2 Manual Inspection of Localization Results

The first two cases for rejecting localization data points mentioned in Section 3.6.1 were accomplished with simple queries of the data using Pandas [62]. The final case required a manual inspection process. The manual inspection process was also used to sift the good localization data points down to those that could be used only for Level One Summaries and those that could be used in both Level One and Level Two Summaries. The manual inspection process is the focus of this section.

Manual inspection of the data set was performed as a two stage process. A Python [45] script was used that visualized the images in the same manner discussed in Sections 3.2.1.3 and 3.3.1.3 in both stages. During the first pass, images were first visualized and then two options were presented in the Python console: “accept result” and “reject result”. This pass separated the 412 poor localization data points discussed in Section 3.6.1. This process was first performed on pixel-wise mask localizations and all images with good mask localization outputs were tracked by file name. After the pass on the mask localization outputs, the same process was performed on bounding box localization outputs for any image where masks had less than one object detection. Both SSD [37] and YOLOv3 [48] localization outputs were inspected in this pass. This first pass of manual inspection allowed extraction of the 577 localization outputs deemed usable as input to Level One Summaries.

After the “good” data points were filtered out by the first pass, a second pass was performed to determine which localization outputs were usable for Level Two Summaries. As with the first pass of manual inspection, the localization outputs were visualized and two options presented in the Python [45] console: “usable for Level Two Summaries” and “not usable for Level Two Summaries”. In this secondary manual inspection pass we only looked for data points that met the criteria discussed in Section 3.6.1; outputs with more than one localization with more than one object class, and at least two classes of objects that could logically interact with each other. This secondary manual inspection allowed identification of the 509 localization outputs deemed usable as input to Level Two Summaries.

This chapter presented all of the localization and metadata generation methods used in the S2T pipeline. In Chapter 4, the object localization data set was used to inform inference of the Level One Summaries, which corresponds to object tuple spatial relationships. In Chapter 5, the object localization data set and the image metadata, were combined with the Level One Summaries and used as input to the inference of Level Two Summaries, which corresponded to a natural language scene description.

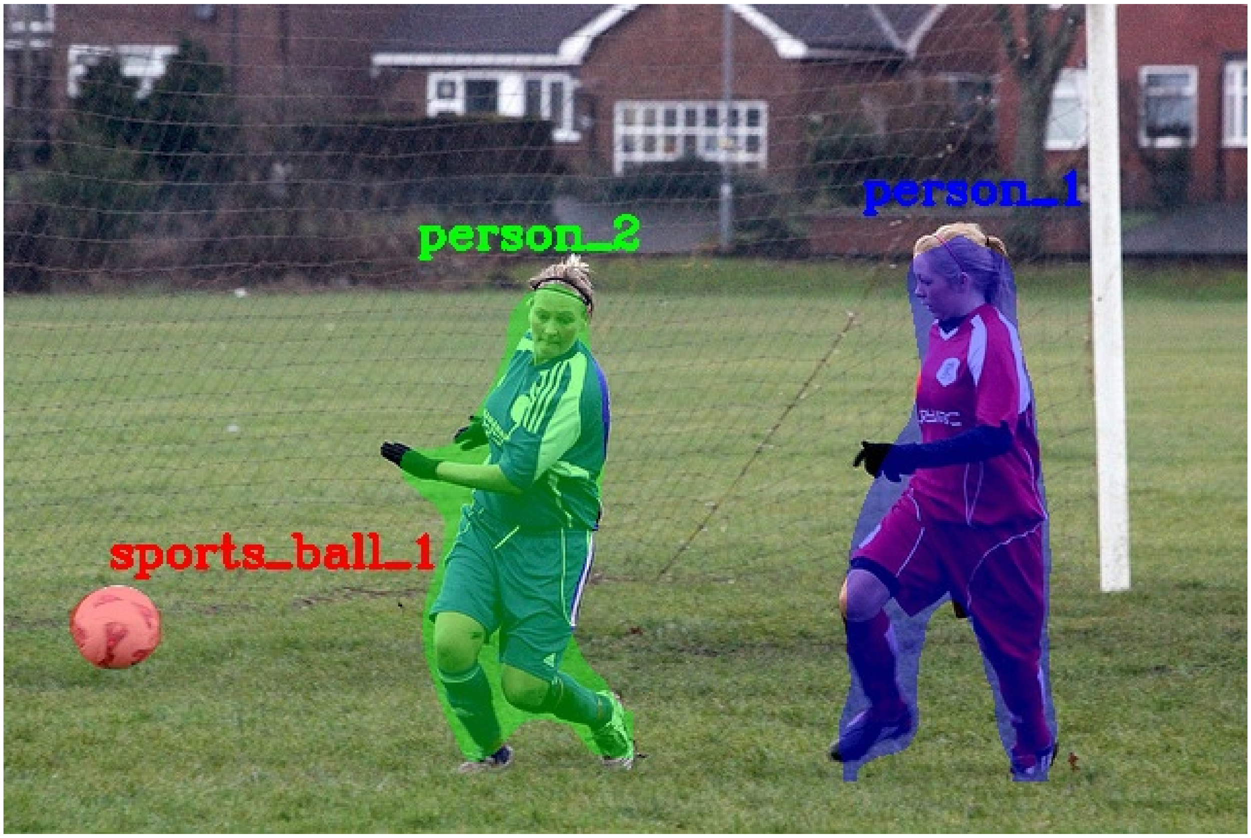


Figure 3.12

Metadata Example Image

CHAPTER 4

EVALUATION 1: LEVEL ONE SUMMARIES

Level One Summary generation is the next phase of constructing the S2T pipeline and system. Level One Summaries provide the spatial relationships between each tuple of objects in an image. This chapter begins by revisiting the S2T pipeline diagram shown in Figure 4.1 to discuss the current stage of the process. Afterwards, the implementation of the two methods used to generate Level One Summaries, HOF [40] and GIOU [51], are discussed in detail. Following implementation details, an analysis of the results obtained from combining the HOF and GIOU algorithms are presented. Level one Summaries are, in essence, the results obtained by combining the output of the HOF and GIOU algorithms. This chapter concludes with a discussion of the Level One Summaries in general as well as how Level One Summaries pertain to input to the Level Two Summary evaluations of Chapter 5.

HOF [40] and GIOU [51] both have been previously validated in their respective original works. As such, no validation was performed on the algorithms as they were used without modification. Section 4.8 serves as validation of the results as they relate to the input to Level Two Summary evaluation of Chapter 5 and the validation of the HOF and GIOU algorithms is reserved to the original works in [40] and [51] respectively.

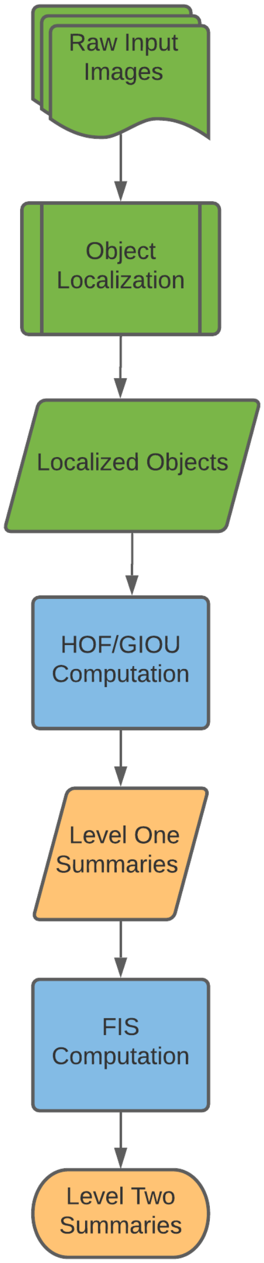


Figure 4.1

Level One Summaries in the S2T Pipeline

4.1 S2T Pipeline

Let us begin this chapter by revisiting the S2T pipeline diagram in Figure 4.1 and examine the current as well as completed stages of the system. Evaluations in Chapter 3 performed the "Object Localization" operations and the results are the "Localized Objects" as it relates to the S2T pipeline diagram. "Localized Objects" serve as input to the "HOF/GIOU Computation" stage and the result is the "Level One Summaries" in reference to the diagram. This chapter presents the implementation of the "HOF/GIOU Computation" stage and the results analysis examines the "Level One Summaries" output by the HOF and GIOU computations. Each of the completed stages of the pipeline diagram in Figure 4.1 is shown in green and the stages yet to be completed are shown in blue. Finally, nodes shown in orange, correspond to outputs of the S2T system. With this in mind, the diagram shows the Level One Summaries, computed in this chapter, and Level Two Summaries, computed in Chapter 5, as shown in Figure 4.1 can both be used as output of the S2T system. As mentioned previously, the Level One Summaries are used as the final output when there are no object localization results that can logically interact with each other. Sections 3.2.1.3 and 3.3.1.3 both discuss these cases in further detail.

4.2 Level One Summary Approach

This section briefly introduces the algorithms and methods used for constructing Level One Summaries as well as the software packages and libraries used to implement the algorithms. Thorough implementation details of each method are reserved to the individual Sections 4.3, 4.4 and 4.5 below. The combination of these methods is used to construct Level One Summaries and this process is discussed in detail in Section 4.6. Results for the individual methods, as well as the

Level One Summary output, is reserved to Section 4.7 as the combination of results is the most logical way to present them.

4.2.1 Software Packages and Libraries

All evaluations detailed in this chapter made use of an Anaconda [2] virtual environment to allow the S2T system to be easily configured on multiple machines. Object tuple computation and GIOU were implemented in Python [45] and made use of the Numpy [43] library for set operations. HOF was implemented using Matlab [39] code obtained from the authors of the original work in [40] that ran in a Python script. GIOU and HOF results were stored using Pandas [62] data frames, which were then written to CSV files for retrieval. Visualization of results were performed using the OpenCV [5] library for displaying images and text labels were assigned to each of the images. These text labels will be displayed in tables for the purposes of this evaluation.

4.2.2 Level One Summary Methods

Construction of Level One Summaries relied on output from three individual methods. Each method accepted the set of localization results computed in Sections 3.2 and 3.3 as input. The first task performed was the computation of the object-two tuples for each localization result in the set of results. This process is detailed in Section 4.3. These object tuples were used to map the localization results for use as input to the other two tasks for constructing Level One Summaries. First, object overlap and object proximity was computed using the GIOU [51] algorithm with implementation details given in Section 4.4. Following, object tuple spatial relationships were computed using the HOF [40] algorithm, detailed in Section 4.5. Finally, a combination of the results from GIOU and HOF were used to construct Level One Summaries for each localization result in the result set. This process is described in detail in Section 4.6. Level one Summary results are presented in Section 4.7. Tuple, GIOU, and HOF results are embedded in the Level One Summaries so visualization of the individual results are shown alongside the final Level One Summaries.

4.3 Object Tuple Computation

The first task performed was the generation of the set of object tuples for each image localization result. Localization results, discussed in Sections 3.2.1.3 and 3.3.1.3, were loaded from CSV files into Pandas [62] data frames and then a set of tuple object permutations was computed for each image using the object labels contained in the localization results. This set of tuple object labels for each image was then mapped to their corresponding bounding box or mask localization result. These mappings served as the input to the GIOU [51] and HOF [40] algorithms discussed in Sections 4.4 and 4.5. Algorithm 4.1 shows psuedocode for the process for computing the tuple of objects in an image.

Algorithm 4.1 Compute Object Tuples for Image

```
let $R$ be the set of localization results $\geq 2$
for all $r \in R$ do
    let $L$ be the set of labels for $r$
    set $twotuples_L = permutation(L, 2)$
    set $twotuples'_L = twotuples_L - inverse\ relationships$
end for
```

Algorithm 4.1 shows that the tuple computation process is only performed on image object localization outputs with two or more localization results. This process was performed on the labels generated during object localization, discussed in Chapter 3, and the result of this process was a set of tuple label permutations for each individual image's localization results. Permutations with $n = 2$ were generated from the set of labels for an image and then inverse relationships were removed such that only one pair of tuples was preserved for a pair of labels. By doing this, the number of computations performed for both GIOU [51] and HOF [40] was halved. As an example, consider Figure 3.12, where there are three labels: "person_1", "person_2", and "sports_ball_1". A logical conclusion can be drawn that if "person_1" has the relationship of "to the right of" "sports_ball_1", then "sports_ball_1" must be "to the left of" "person_1". Thus, only one set of results for spatial relationships need to be computed for a tuple of objects and the bidirectional permutation does not need to be preserved. It follows that the same conclusion can be drawn for a tuple of objects that overlap as well. The labels were ordered according to an object hierarchy, discussed in further detail in Chapter 5, such that people took precedence over animate objects and animate objects took precedence over inanimate objects. This was done to ensure that the person-object Level One Summaries were computed and not the other way around. For the case of no person detections, the system prefers to look at interactions in the form of animate-inanimate Level One Summaries. The final result of Algorithm 4.1 is the set of tuple object pairs with no inverse relationships and these result pairs were used to index the object localization results for computing GIOU and HOF labels, which are discussed in the following sections.

4.4 Generalized Intersection Over Union

Generalized intersection over union (GIOU) [51] was originally implemented for computing loss between predicted bounding boxes and ground truth bounding boxes. The original work provides details on other uses of GIOU including computation of proximity and object overlap.

This work used the proximity and overlap properties of the GIOU algorithm to determine a) if object tuples overlap and b) determine the proximity between object tuples. The remainder of this section discusses the implementation of the GIOU algorithm used in this work, a discussion of the validation of the GIOU algorithm from the original work in [51], and a brief discussion of the results. The complete discussion of the results is reserved to Section 4.7 where both the GIOU and HOF [40] results are presented.

4.4.1 GIOU for Computing Object Overlap and Proximity

The original GIOU implementation in [51] was implemented with bounding box localization in mind. As such, the GIOU algorithm was used as described in the original work for the bounding box localization results derived in Section 3.3, and GIOU was computed using the underlying bounding boxes for the mask localization results derived in Section 3.2.

GIOU was implemented identical to the original work in [51]. Algorithm 4.2 shows the pseudocode of the implementation for the purposes of the evaluations in this chapter. The input to the algorithm were the bounding boxes corresponding to the argument and referrant images, which are discussed in Section 4.5. For the purposes of the GIOU algorithm, the argument and referrant images may be viewed as two distinct bounding boxes corresponding to two distinct objects. The algorithm proceeds by then computing a minimal enclosing bounding box, *C*, for the *Arg* and *Ref* bounding boxes. Afterwards, the intersection over union (IOU) was computed by dividing the intersection of the pixels in the *Arg* and *Ref* bounding boxes by the union of the pixels in the *Arg* and *Ref* bounding boxes. Lastly, GIOU was computed by subtracting a normalizing factor from the IOU. The normalizing factor, $\frac{|C-(Arg \cup Ref|}{|C|}$, represents the ratio of the area occupied by

C excluding *Arg* and *Ref* divided by the area occupied by *C*. This normalizing factor allowed a measure that focused on the empty area between *Arg* and *Ref*. In this manner, GIOU scores reflected the distance between two arbitrary bounding boxes, whereas IOU scores did not. Both IOU and GIOU were scale invariant, which was necessary given that most bounding boxes were of dissimilar sizes.

Algorithm 4.2 Generalized Intersection Over Union

```
input: Two bounding boxes: $Arg, Ref \in Img$
output: IOU and GIOU score for the $Arg$ and $Ref$ objects
Find smallest enclosing bounding box $C$ for $Arg$ and $Ref$
$IOU = \frac{|Arg \cap Ref|}{|Arg \cup Ref|}$
$GIOU = IOU - \frac{|C - (Arg \cup Ref|}{|C|}$
```

The input for the GIOU [51] computation were the bounding box localization results computed in the evaluations of Chapter 3, separated into tuples using the method described in Section 4.3. As mentioned previously, GIOU used the underlying bounding boxes for mask segmentation localization results discussed in Section 3.2, and the bounding boxes were used as is for bounding box localization results discussed in Section 3.3. The output of the GIOU algorithm was two labels: a) *Arg* overlaps/does not overlap *Ref* and b) *Arg* is in proximity *p* of *Ref*. Any *Arg* and *Ref* pair with $IOU > 0$ was assigned the “overlaps” label, whereas any pair with $IOU = 0$ was assigned the “does not overlap” label. The overlap computation was the trivial case in that any IOU score with a value greater than 0 indicates that at least some portion of the bounding boxes overlap. IOU was chosen for the overlap case because it reflected any potential overlap between objects better than GIOU since there was no normalizing factor for IOU. The proximity case relied upon the range $-1 \leq GIOU \leq 1$, where decreasing GIOU values indicated that *Arg* and *Ref* were further away from each other and increasing GIOU values indicated that *Arg* and *Ref* were closer to each other. For the case of $GIOU = 1$, the objects completely surrounded each other and for the case $GIOU = -1$, the objects were as far away from each other as possible. Results obtained from the GIOU computations were stored in a CSV file for easy retrieval.

In order to construct the FIS system detailed in Chapter 5, the overlap and proximity values were modeled using fuzzy membership functions. Figure 4.2 shows the trapezoidal membership functions used to represent “Overlap”. Overlap was modeled as a trapezoidal membership function in order to allow the system to capture the wide ranges where overlap and no overlap were certain. As any value greater than 0 indicated overlap and any value less than zero indicated no overlap, these trapezoidal membership functions more accurately model those ranges as opposed to triangular membership functions. Similarly, proximity was also modeled using trapezoidal membership functions. The use of trapezoidal membership functions allowed modeling the cases where more than one value corresponds to a specific proximity, as shown in Figure 4.3.

Construction of membership functions is described in great detail by Ross in [53]. Figure 4.2 shows that any value $IOU \leq 0$ corresponds to “No Overlap” where any value $IOU > 0$ corresponds to “Overlap.” In essence this is just a Boolean-valued function, but for the purposes of constructing Level Two Summaries, overlap was represented as a fuzzy variable. For Level One Summaries, the overlap label was a result of performing defuzzification of the IOU input value and returning the maximum membership of the “Overlap” membership functions. Equation 4.1 represents this computation in mathematical form. The overlap membership functions and

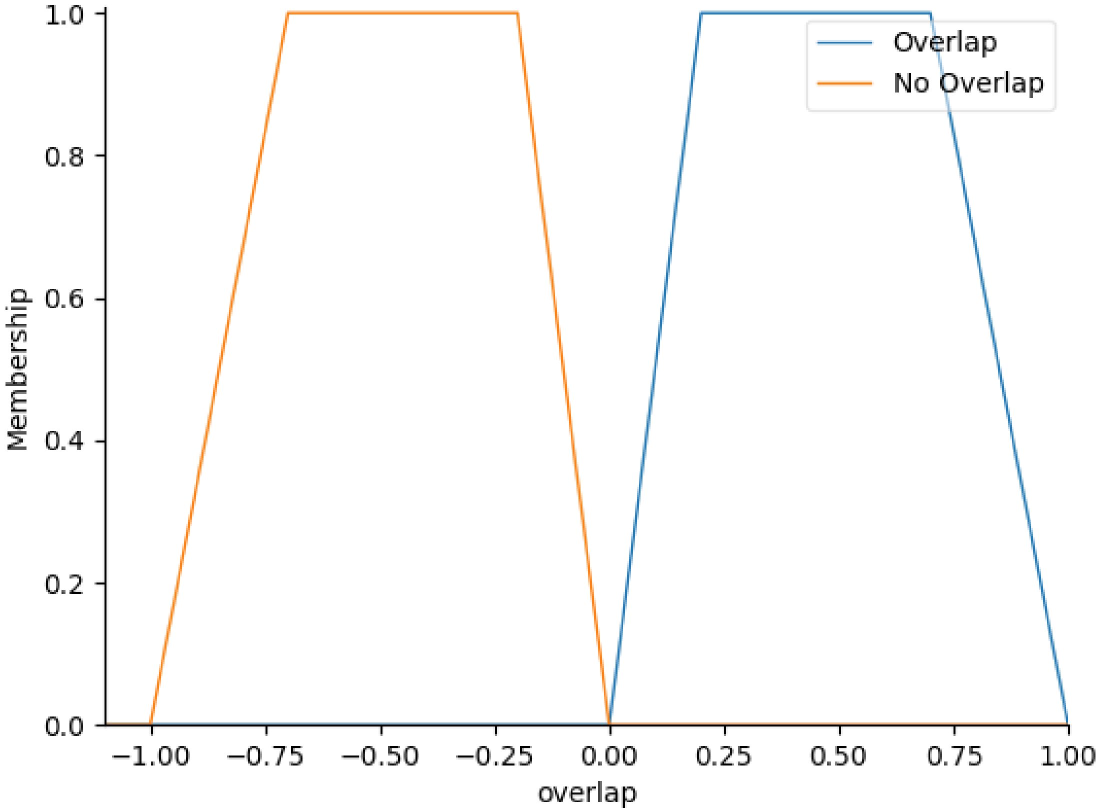


Figure 4.2

Overlap Membership Functions

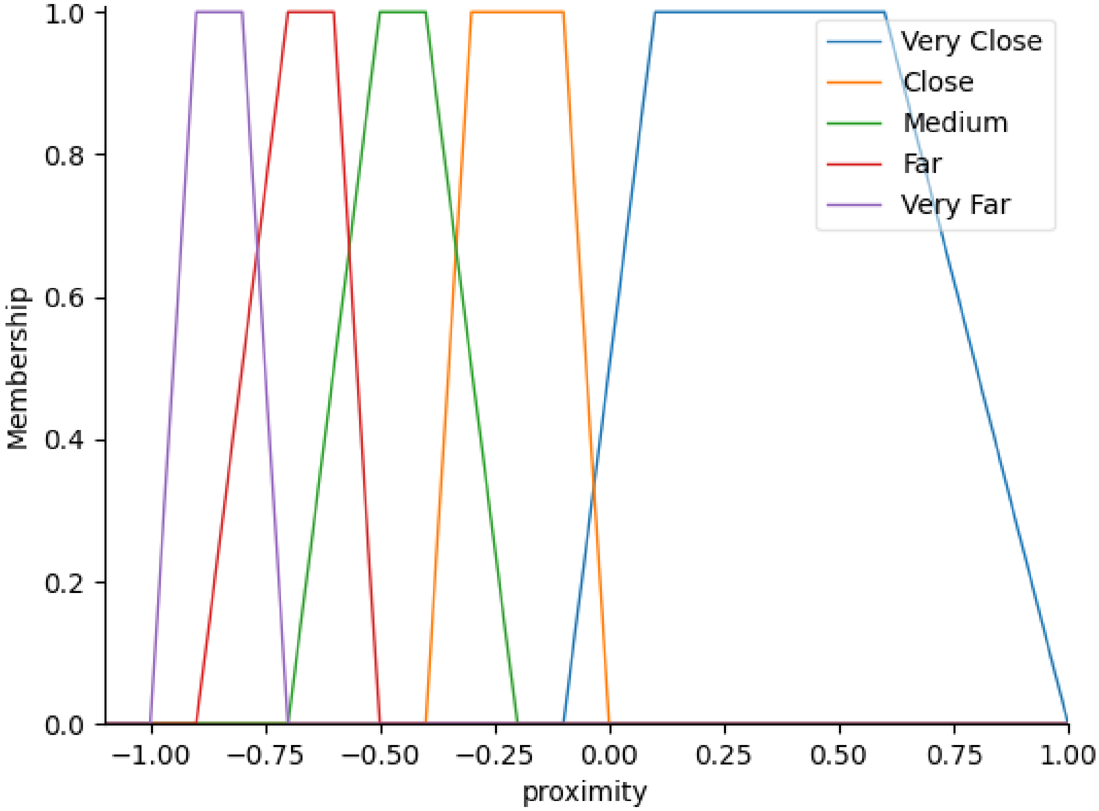


Figure 4.3

Proximity Membership Functions

defuzzification operations were implemented using the SKFuzzy [69] library using Python [45] scripts.

$$overlap = max(overlap(IOU), no\_overlap(IOU)) \tag{4.1}$$

Figure 4.3 shows the trapezoidal membership functions used to represent "Proximity." As shown in the figure, the proximity membership functions were more aligned with the classical definition of a fuzzy variable in that the input variable could exist in multiple outputs. Figure 4.3 shows that the range $-1.0 \geq GIOU > -0.75$ corresponded to "Very Far", $-0.85 \geq GIOU > -0.5$ corresponded to "Far", $-0.7 \geq GIOU > -0.25$ corresponded to "Medium", $-0.35 \geq GIOU > 0.0$ corresponded to "Close", and $-0.1 \geq GIOU > 1.0$ corresponded to "Very Close". While proximity was not used specifically for Level One Summaries, its result could be computed by performing the defuzzification of the $GIOU$ input value and returning the maximum of the memberships of the "Proximity" membership functions. Equation 4.2 represents this computation in mathematical form. The proximity membership functions and defuzzification operations were implemented using the SKFuzzy [69] library using Python [45] scripts.

$$\begin{split} proximity = max(very\_close(GIOU), close(GIOU), \\ medium(GIOU), far(GIOU), very\_far(GIOU)) \end{split} \tag{4.2}$$

The overlap crisp output that results from defuzzification of the IOU value, computed via Algorithm 4.2, combined with the output of the HOF [40] algorithm, discussed in Section 4.5 were used to construct the Level One Summary outputs. An in depth discussion of how the final Level One Summaries were constructed is presented in Section 4.7. The proximity output from the GIOU [51] computations was used strictly as latent input as a fuzzy variable, in addition to the combined overlap/HOF results represented as fuzzy variables, to the Level Two Summary computations presented in Chapter 5.

4.4.2 GIOU Validation

No additional validation was performed on the GIOU [51] algorithm itself. The authors of the original work used a mAP score metric for the task of computing bounding box loss, i.e. the relative position of the predicted bounding box compared to the ground truth value. The authors reported an average mAP improvement of 9.78% for YOLOv3 bounding box predictions and an average mAP improvement of 2.60% for Mask-RCNN [21] bounding box predictions. The evaluations constructed in this work used an unmodified version of GIOU, and as such we relied on the original validation as proof the algorithm works properly. A manual inspection of the crisp variables obtained via defuzzification for overlap, using Equation 4.1, and proximity, using Equation 4.2, was performed to ensure the fuzzy membership functions were performing as expected. This manual inspection showed that both the overlap and proximity computations were accurate for the object localization results obtained from Chapter 3. Specifically, overlap and proximity values were computed for each tuple in the 571 images usable for Level One Summaries from Chapter 3. Of these 571 images, all of the overlap and proximity results were accurate. This was an expected behavior as the GIOU [51] computation was deterministic and the only verification necessary was the proper construction of the fuzzy membership functions.

4.4.3 GIOU Discussion

GIOU [51] was performed on the set of object localization results that contained more than one object localization deemed to be usable for either Level One or Level Two Summaries. Table

3.2 shows totals for the usable object localization results. GIOU was performed on both mask and bounding box usable localization results. As mentioned previously, the GIOU values for each image's object localization results were stored in CSV files for easy retrieval. Specifically, the IOU and $GIOU$ values were retained and stored as floating point values to be used as input information for the Level Two Summaries computed via FIS, discussed in detail in Chapter 5.

The following section details the evaluations used to compute the spatial relationships between the set of tuples of objects in an image. The combination of these results is presented in Section 4.6 to construct the Level One Summaries. Visualizations of the GIOU [51] results is reserved to Section 4.7 where the visualizations are presented in tandem with the results from Section 4.5.

4.5 Histogram of Forces for Spatial Relationships Between Objects

The Histogram of Forces (HOF) algorithm implemented in the original work by Matsakis in [40] was used to compute spatial relationship information for the object localization results generated in Chapter 3. Each image's localization results were separated into object tuple pairs using the method detailed in Section 4.3 and the HOF algorithm was used to compute the spatial relationships for each tuple pair. First a description of the implementation of the HOF algorithm and its output as it relates to the S2T system constructed is presented. A discussion of the original validation is presented following the implementation details. A discussion of the results of the HOF algorithm is reserved for Section 4.7 where the combined results of GIOU and HOF will be described.

4.5.1 HOF Implementation

Algorithm 4.3 Spatial Relationship Computation

```
input: Object localization results for an image
output: Tuple spatial relationships for localization results
Set T = tuple pairs generated using Algorithm 4.1
for $(t_a, t_r) \in T$ do
    Compute binary images for $t_a$ and $t_r$
    $F0_t, F2_t, FH_t = HOF(t_a, t_r)$
    if $F0_t = F2_t = FH_t$ then
        $SR_t = FH_t$
    else if $F0_t = F2_t$ then
        $SR_t = F2_t$
    else if $F0_t = FH_t\textbf{or}F2_t = FH_t$ then
        $SR_t = FH_t$
    end if
end for
```

The HOF [40] algorithm used for the evaluations made use of the original Matlab [39] code provided by the authors of the original work. A Python [45] script was implemented to call the Matlab functions and pass inputs to the HOF algorithm and to store the outputs of the HOF algorithm for retrieval.

The original HOF [40] algorithm accepted as input an argument and referrant object as binary images, where the spatial relationship computed was of the form: "Argument is in direction α of Referrant." The assignment of the argument and referrant objects was hierarchical and based on object categories, discussed in detail in Chapter 5, where person detections took precedence and were used to provide a relative space in which the HOF algorithm operated.

Algorithm 4.3 details the process used in this work to compute the spatial relationships for object two tuples in an image and this algorithm was run on each localization result in the set of usable localization results described in Chapter 3 and shown in Table 3.2. The algorithm accepted as input a single image's object localization results, in the form of masks or bounding boxes, described in Sections 3.2 and 3.3 respectively. The HOF algorithm was shape-invariant so the mask and bounding box localization results were used as-is. Object tuple pairs were computed for the localization results of an image using Algorithm 4.1 and the argument and referrant objects were assigned as the first and second entries in the tuple respectively. For the argument and referrant objects, an individual binary image was constructed using the OpenCV [5] and Numpy [43] libraries, where a value of 1 indicated a pixel belongs to an object and a value of 0 indicated a pixel did not belong to the object. These argument and referrant binary images were then passed as input to the HOF Matlab [39] implementation. As mentioned previously, the inverse tuple relationships were not preserved. As such, the HOF algorithm was only performed once per tuple pair and spatial relationship symmetry was exploited for the referrant-argument relationship once the argument-referrant relationship was computed. As an example, if the argument object was "to the left of" the referrant object, it followed that the referrant object was "to the right of" the argument object.

HOF [40] outputs a set of three histograms: constant force (F0), gravitational force (F2), and hybrid force (FH) histograms, where hybrid force histograms were a combination of the F0 and F2 force histograms. Each histogram contained entries in up to 360 bins, where the bins corresponded to degrees on the unit circle. The largest value bin corresponded to the direction that the argument object was in reference to the referrant object. Further descriptions of the F0, F2, and FH histograms is given in the original work by Matsakis in [40].

Algorithm 4.3 shows that the F0, F2, and FH histograms were output for each tuple in the set of tuples for the object localization results. Then, a consensus value was chosen to represent the spatial relationship, SR_t in Algorithm 4.3, for a tuple object pair. The output values of the

HOF [40] algorithm for each type of force histogram were integer values that corresponded to the maximum angle detected. After computing SR_t $\forall t \in T$, the F0, F2, and FH histogram values were stored in a CSV file for easy retrieval.

In order to use the spatial relationship information as input to the Level Two Summary computation, described in Chapter 5, the spatial relationships were represented as triangular membership functions [53] as shown in Figure 4.4. Triangular membership functions were used to model the spatial relationship ranges because each cardinal direction has only one value that indicates that exact direction. For example, consider the HOF output of 90 and consider it represented on the unit circle in degrees. This value corresponds strictly to the “above” direction, whereas the value 91 lies somewhat in both the “above” and “left” directions on the unit circle, and the value 89 lies in somewhat both the “above” and “right” directions. Using triangular membership functions allowed the modeling of these directions such that only four membership functions were required and also allowed modeling the uncertainty when HOF outputs did not strictly exist in one of the four modeled directions.

Table 4.1 shows the ranges each direction of the membership functions fall into in a more readable format. The degree ranges were separated into nine directions, each of which corresponded to values on the unit circle. The consensus angle, SR_t, was defuzzified in order to extract the crisp output spatial relationship direction, shown in Table 4.1, for each object tuple. These crisp outputs were combined with the overlap results, discussed in Section 4.4, in order to construct Level One Summaries. The fuzzy value of SR_t was used as input to the Level Two Summary computations detailed in Chapter 5. The spatial relationship membership functions and defuzzification operations were implemented using the SKFuzzy [69] library in Python [45] scripts.

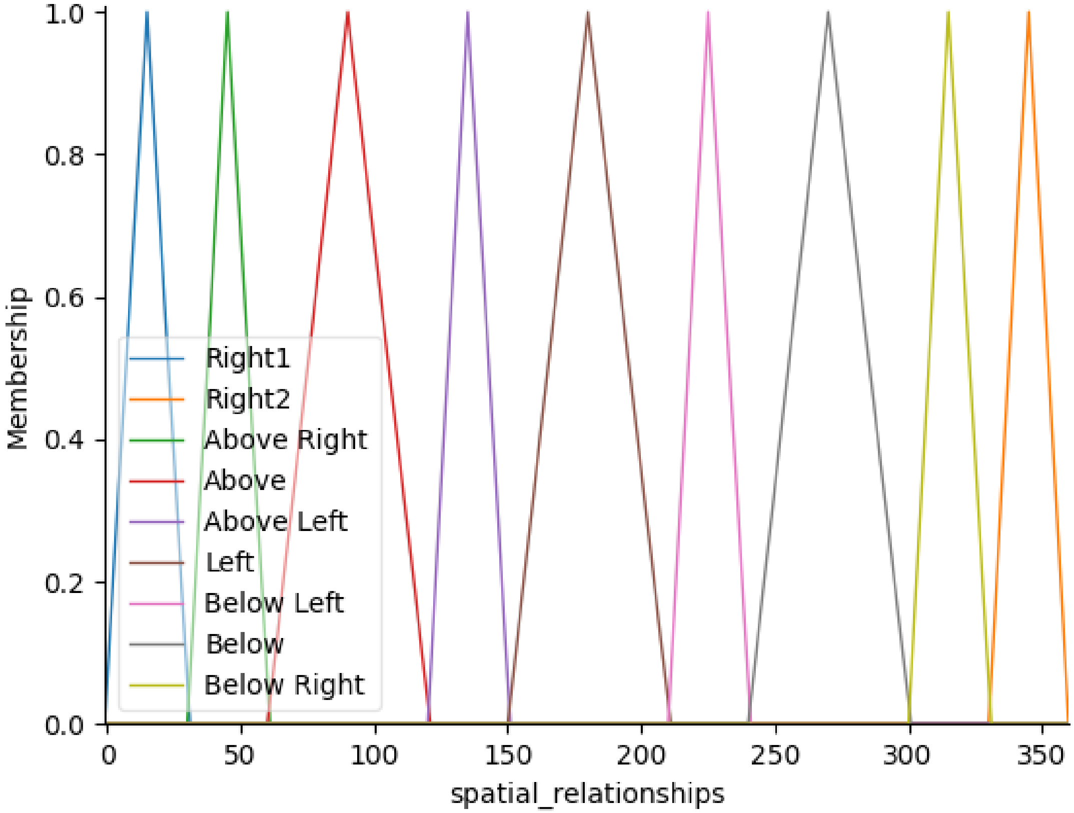


Figure 4.4

Spatial Relationship Membership Functions

Table 4.1

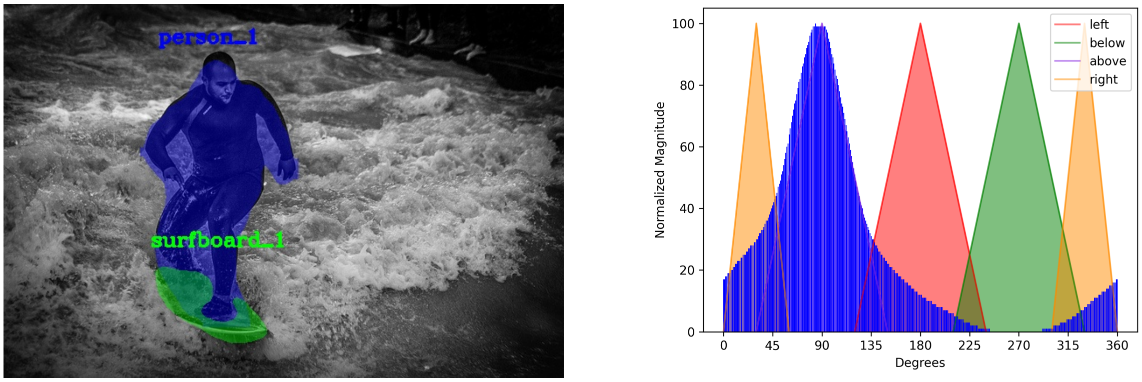
Spatial Relationship Directions

Range	Direction
$0° < SR_t \leq 30°$	to the right of
$30° < SR_t \leq 60°$	above and to the right of
$60° < SR_t \leq 120°$	above
$120° < SR_t \leq 150°$	above and to the left of
$150° < SR_t \leq 210°$	to the left of
$210° < SR_t \leq 240°$	below and to the left of
$240° < SR_t \leq 300°$	below
$300° < SR_t \leq 330°$	below and to the right of
$330° < SR_t \leq 360°$	to the right of

4.5.2 HOF Validation

No additional validation was performed on the original HOF work of [40]. The authors compared HOF results from the work to a similar method, Histogram of Angles [41]. The aim of the validation was to show that HOF generalized well to all shapes when compared to the Histogram of Angles method. Fundamentally, the F0 histograms were equivalent to the output of Histogram of Angles, and as such the results were nearly identical during comparison. The authors demonstrated in [40] that the F2 and FH histograms provided richer descriptions of spatial relationships because of several factors. First, the F2 and FH histograms specifically provide context-specific contextual knowledge based on metrics presented by Gapp in [16], and furthermore, the F2 and FH histograms can always be rationalized whereas the Histogram of Angles output cannot. Non-disjoint objects were handled using the HOF algorithm, but there was no way to handle these objects in the Histogram of Angles method. Lastly, the HOF algorithm was set up to handle fuzzy objects, whereas the Histogram of Angles method could only handle crisp disjoint objects. While the authors noted that the Histogram of Angles method was a viable method, the HOF method outperforms the Histogram of Angles method due to its additional properties discussed. Further validation details of the HOF algorithm is reserved to the original work in [40].

However, a manual validation was performed on the defuzzification process to ensure that the membership functions shown in Figure 4.4 generated the correct crisp spatial relationship for the object tuples. Similar to the process discussed in Section 4.8.2, each of the 571 image localization results deemed usable as input for Level One Summaries were visualized in a Python [45] script using the OpenCV [5]. For each image, the crisp output for each object tuple spatial relationship was presented as a text output for inspection to ensure that the spatial relationship between the tuple-object pair was accurate.



(a)

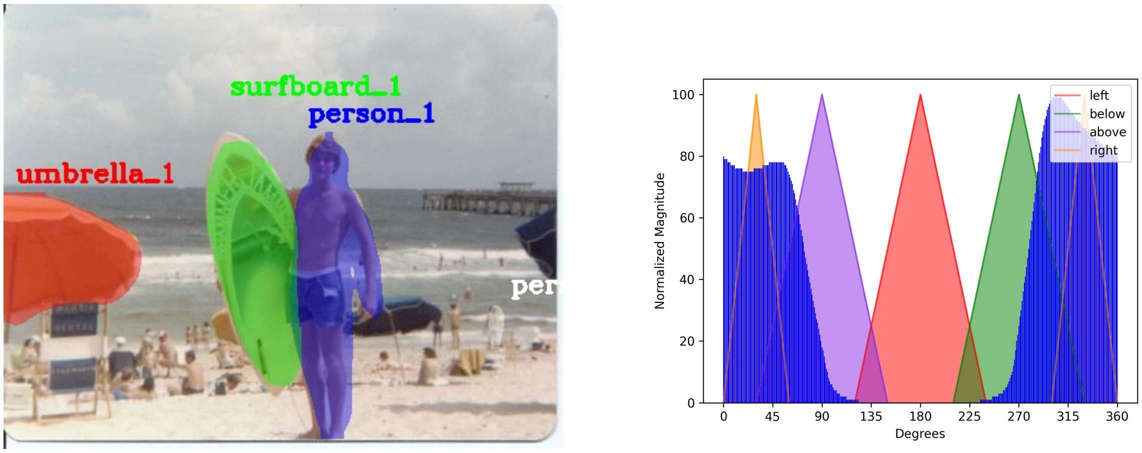
(b)

Figure 4.5

Person riding surfboard HOF output

(a) Mask localization output (b) Spatial relationship membership function

Figures 4.5 - 4.11 show examples of spatial relationship membership function outputs obtained by applying the HOF [40] algorithm to localized object results. For brevity, only one tuple



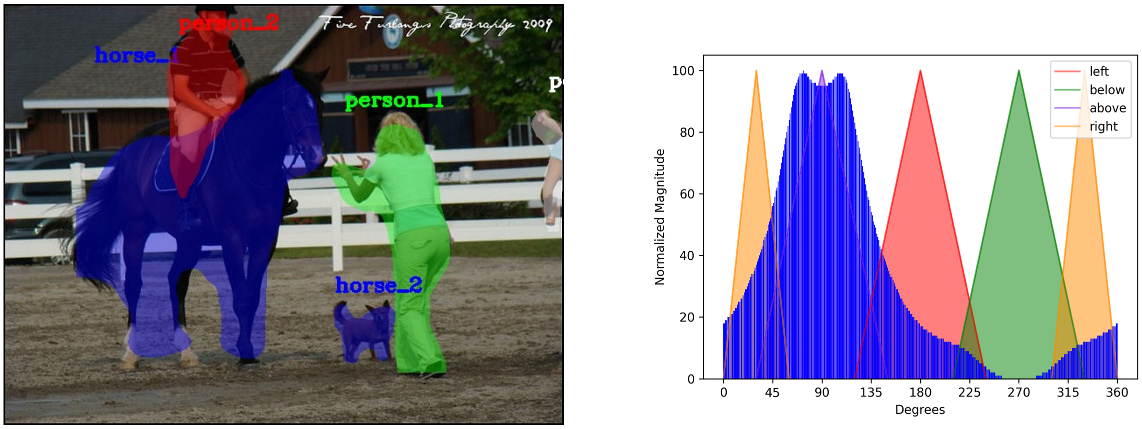
(a)

(b)

Figure 4.6

Person holding surfboard HOF output

(a) Mask localization output (b) Spatial relationship membership function



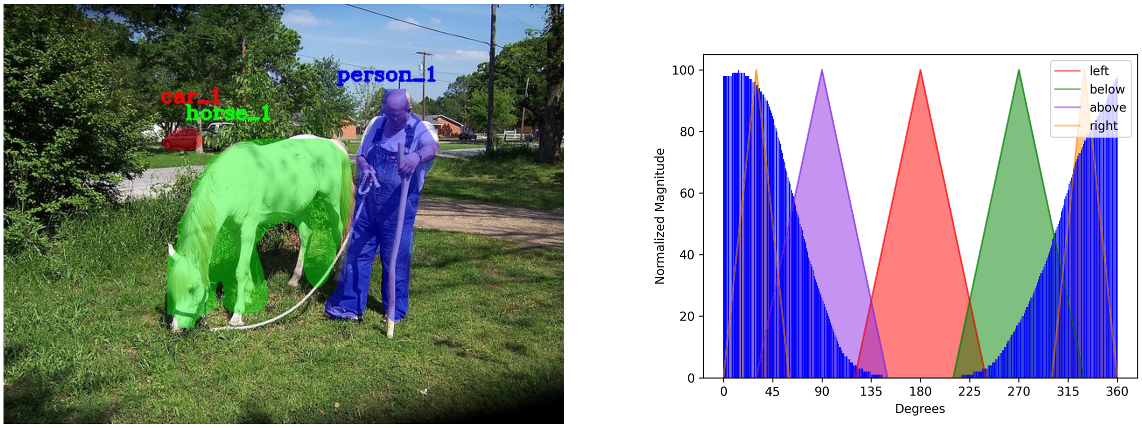
(a)

(b)

Figure 4.7

Person riding horse HOF output

(a) Mask localization output (b) Spatial relationship membership function



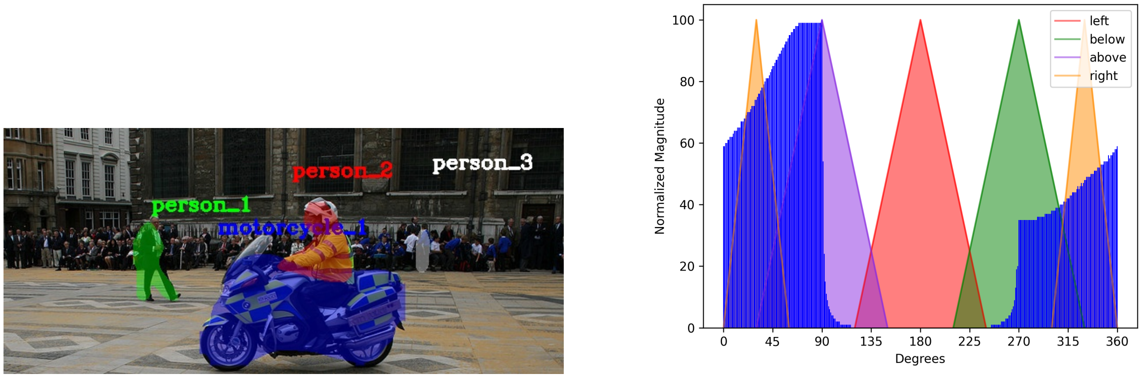
(a)

(b)

Figure 4.8

Person beside horse HOF output

(a) Mask localization output (b) Spatial relationship membership function



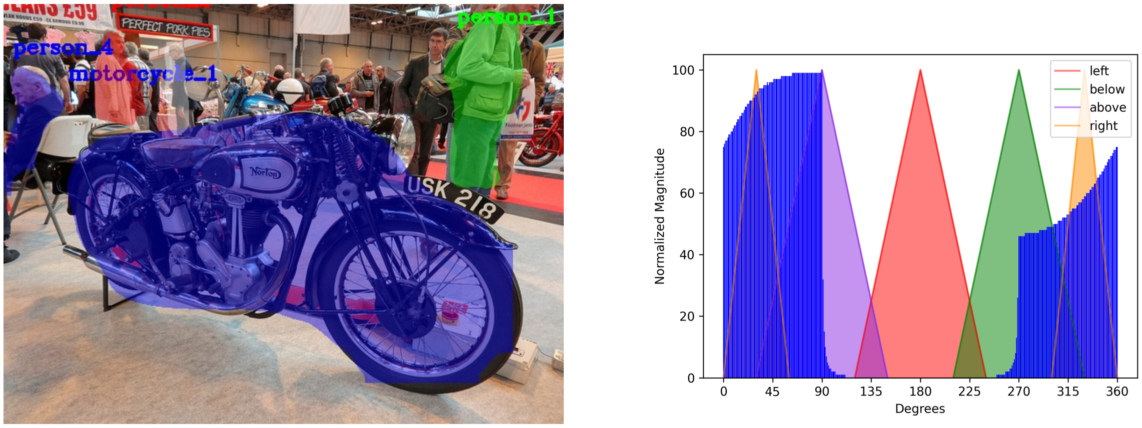
(a)

(b)

Figure 4.9

Person riding motorcycle HOF output

(a) Mask localization output (b) Spatial relationship membership function

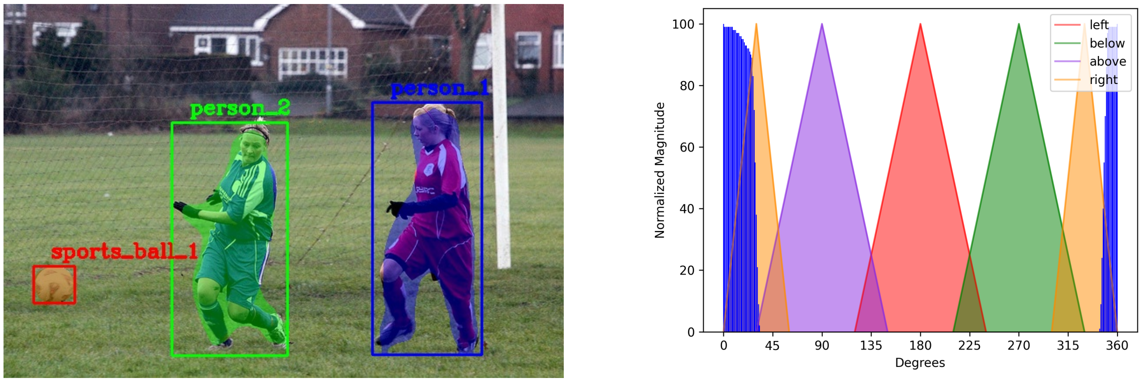


(a) (b)

Figure 4.10

Person beside motorcycle HOF output

(a) Mask localization output (b) Spatial relationship membership function



(a) (b)

Figure 4.11

Person playing soccer HOF output

(a) Mask localization output (b) Spatial relationship membership function

membership function output was displayed for the mask localization results. The membership function outputs for each of the images were chosen to be the most interesting tuple in the image except in instances such as Figure 4.10 where there were no “interesting” interactions. The membership result images, (b) in each of the images, shows the degrees of the unit circle on the x-axis, and the magnitude for each degree found by HOF is shown on the y-axis. The membership functions for the four cardinal directions (left, below, above, and right) are shown on each image, and the magnitude is superimposed on the membership functions such that the defuzzified output can be inferred from the image.

Interpreting the membership function outputs allowed us to infer in what direction the HOF algorithm thought the tuples were with respect to each other. Figure 4.5(b) shows that the person was above the surfboard. Figure 4.6(b) shows that person_1 was to the right of the surfboard in the image. Figure 4.7(b) shows that person_2 was above horse_1 in the image. Figure 4.8(b) shows the person was to the right of the horse in the image. Figure 4.9(b) is interesting in that it shows person_2 was both above and right of the motorcycle, thus the person can be said to be “above and to the right” of the motorcycle. Figure 4.10(b) shows a similar “above and to the right” result for the person_1 and motorcycle_1 tuple. Lastly, Figure 4.11(b) shows that person_1 was to the right of the soccer ball. Section 4.8.2 goes into further detail about the manual validation process for both overlap and spatial relationships, the combination of which were the final Level One Summaries, and includes detailed statistics for the entire data set.

4.5.3 HOF Discussion

Similar to the GIOU [51] discussion in Section 4.4.3, HOF [40] was performed on the set of object localization results deemed usable for either Level One or Level Two Summaries. Table 3.2 shows the total count of the images with valid object localization results. As mentioned in Section 4.5.1, the HOF algorithm was shape-invariant, so the HOF algorithm worked with both mask and bounding box localization results as-is. The integer values for the F0, F2, and FH histograms were all stored in CSV files using the image file name as a key to the corresponding object localization results. The consensus value for SR_t, shown in Algorithm 4.3, was computed during the construction of Level One Summaries discussed in Section 4.6.

The next section discusses the construction of Level One Summaries, which was the primary objective for the evaluations conducted in this chapter. This and the proceeding sections detail the building blocks for constructing these summaries. As the spatial relationships computed with HOF [40] and object overlap/proximity values computed using GIOU [51] were combined to construct Level One Summaries, the primary discussion of the spatial relationship and overlap/proximity results is reserved to Section 4.7, where a detailed description of the combined outputs is discussed.

4.6 Construction of Level One Summaries

Level One Summaries were constructed by combining the defuzzified GIOU [51] output and defuzzified HOF [40] output, discussed in Sections 4.4 and 4.5 respectively, into a concise description of proximity/spatial relationships for each object-tuple pair in an image. In the case that an object tuple overlapped, the Level One Summary contained the "overlap" label. For example, an object tuple that overlapped would have a Level One Summary of the form: "Argument overlaps and α of Referrant", where α corresponded to one of the nine directions described in Table 4.1. In the case where the object tuples did not overlap, the "overlap" label was omitted from the Level One Summary. An example of this instance is: "Argument α Referrant." The spatial relationship, α, was always included in the Level One Summary.

Level one Summaries could be viewed as the function L_1 that maps each tuple of objects, (a, r), where a and r are the argument and referrant objects respectively, from a set of object localization results for an image (I) to a Level One Summary L_1S for all object tuples (T) in I:

$$\begin{gathered} L_1(a, r) \mapsto L_1S \\ \forall (a, r) \in T \in I \end{gathered} \tag{4.3}$$

The mapping function in Eq. 4.3 can be viewed as the combination of the HOF [40] and GIOU [51] outputs for each tuple (a, r) in T:

$$L_1(I) = \sum_{n=1}^{p} GIOU(a_n, r_n) + HOF(a_n, r_n) \tag{4.4}$$

where p is the number of objects in an image. By exploiting the symmetry, Eq. 4.4 becomes:

$$L_1(I) = \sum_{n=1}^{p/2} GIOU(a_n, r_n) + HOF(a_n, r_n) \tag{4.5}$$

Eq. 4.5 was applied to the usable object localization results, discussed in Sections 3.2.1.3 and 3.3.1.3, and shown in Table 3.2. The results were the set of Level One Summaries generated for the usable object localization results.

The following section provides a detailed description of the Level One Summary results in the form of tables. Section 4.8.2 provides a discussion on the validation process used for the Level One Summaries computed.

4.7 Evaluation Results

Table 4.2

Level One Summaries for Figure 3.3

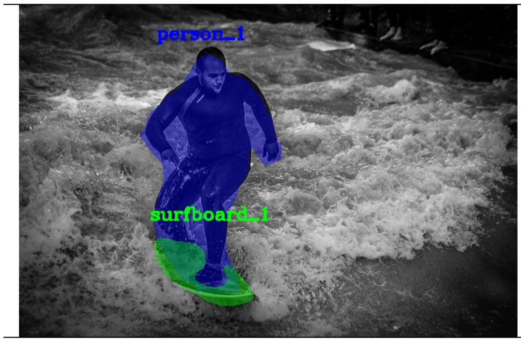
Localization Result	Level One Summaries
	• person_1 overlaps and is above surfboard_1

Table 4.3

Level One Summaries for Figure 3.4

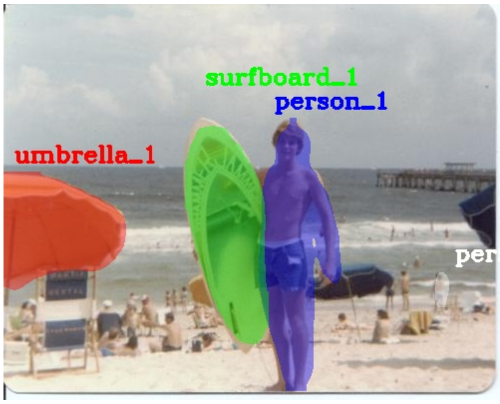
Localization Result	Level One Summaries
	• person_1 overlaps and is below and to the right of surfboard_1 • person_1 is to the right of umbrella_1 • person_1 is to the left of person_2 • person_2 is to the right of surfboard_1 • person_2 is to the right of umbrella_1 • surfboard_1 is to the right of umbrella_1

Table 4.4

Level One Summaries for Figure 3.5

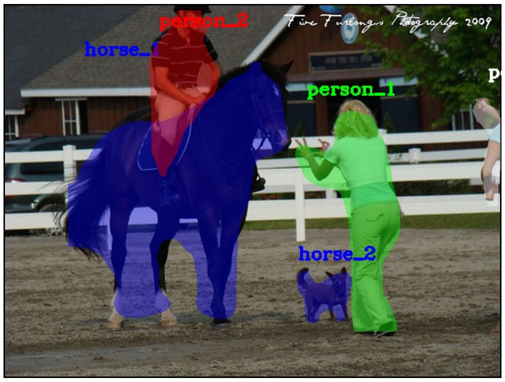
Localization Result	Level One Summaries
 person_2 horse_1 person_1 horse_2 P Fine Furlongs Photography 2009	• person_1 is below and to the right of person_2 • person_1 overlaps and is to the right of horse_1 • person_1 is to the left of person_3 • person_2 is above and to the left of horse_2 • person_2 is to the left of person_3 • person_2 overlaps and is above horse_1 • person_3 is to the right of horse_1 • person_3 is above and to the right of horse_2 • horse_1 is to the left of horse_2

Table 4.5

Level One Summaries for Figure 3.6

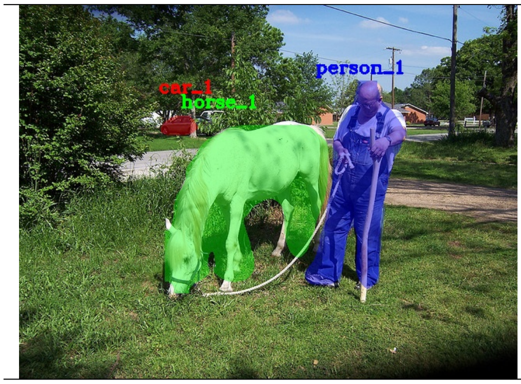
Localization Result	Level One Summaries
	• person_1 overlaps and is to the right of horse_1 • person_1 is to the right of car_1 • horse_1 overlaps and is below and to the right of car_1

Table 4.6

Level One Summaries for Figure 3.7

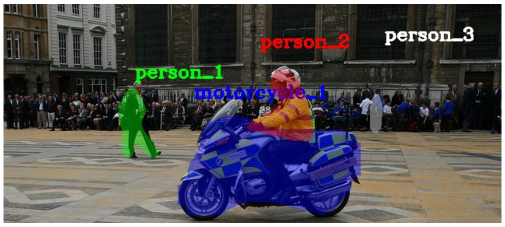
Localization Result	Level One Summaries
	• person_1 is to the left of person_2 • person_1 is to the left of motorcycle_1 • person_1 is to the left of person_3 • person_2 is to the left of person_3 • person_2 overlaps and is above and to the right of motorcycle_1 • person_3 is to the right of motorcycle_1

Table 4.7

Level One Summaries for Figure 3.8

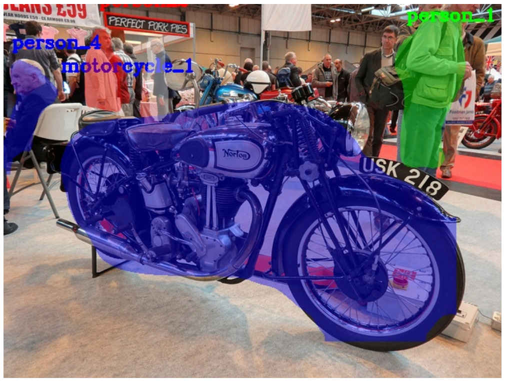
Localization Result	Level One Summaries
 person_1 person_4 motorcycle_1 PERFECT PORK PIES USK 218	• person_1 is to the right of person_2 • person_1 is to the right of person_3 • person_1 is to the right of person_4 • person_1 overlaps and is to the right of motorcycle_1 • person_2 is to the left of person_3 • person_2 is to the left of person_4 • person_2 overlaps and is above motorcycle_1 • person_3 is to the right of person_4 • person_3 overlaps and is above and to the left of motorcycle_1 • person_4 overlaps and is to the left of motorcycle_1

Table 4.8

Level One Summaries for Figure 3.9

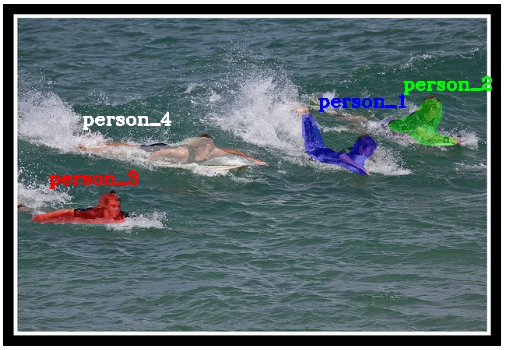
Localization Result	Level One Summaries
	• person_1 is to the left of person_2 • person_1 is to the right of person_3 • person_1 is to the right of person_4 • person_2 is to the right of person_3 • person_2 is to the right of person_4 • person_3 is to the left of person_4

Table 4.9

Level One Summaries for Figure 3.10(b)

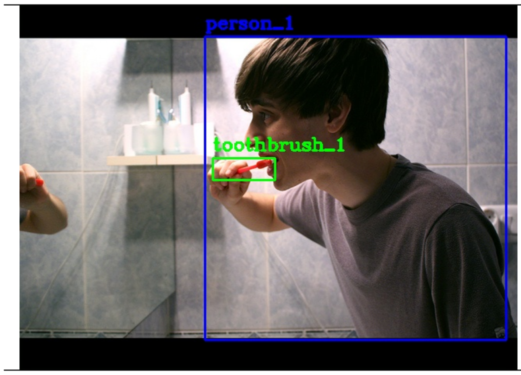
Localization Result	Level One Summaries
	• person_1 overlaps and is below and to the right of toothbrush_1

Table 4.10

Level One Summaries for Figure 3.10(d)

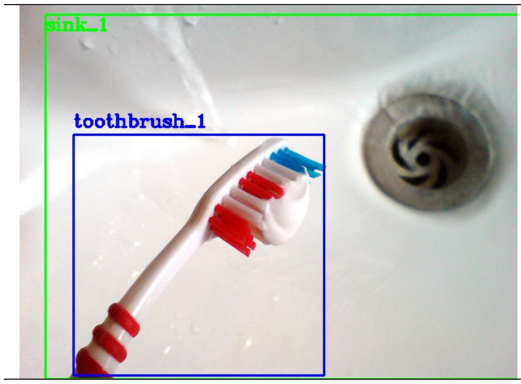
Localization Result	Level One Summaries
	• toothbrush_1 overlaps and is to the left of sink_1

Table 4.11

Level One Summaries for Figure 3.11(d)

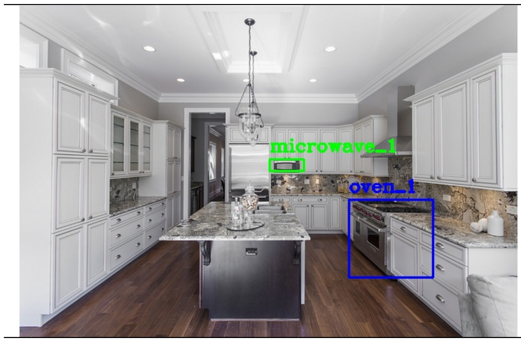
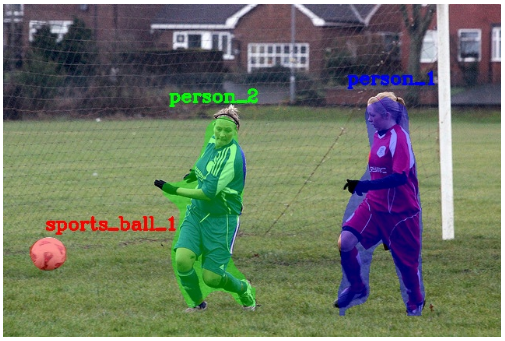
Localization Result	Level One Summaries
	• oven_1 is below and to the right of microwave_1

Table 4.12

Level One Summaries for Figure 3.12

Localization Result	Level One Summaries
	• person_1 is to the right of person_2 • person_1 is to the right of sports_ball_1 • person_2 is to the right of sports_ball_1

This section visualizes the Level One Summary results generated using the evaluations detailed in the previous sections of this chapter. Each of the visualizations show the Level One Summaries for the localization result images computed using the techniques described in Chapter 3. Each result in Tables 4.2 - 4.12 displays the localization result on the left and the Level One Summaries for the localization result on the right. Tables 4.4 - 4.8 and Table 4.12 show the Level One Summaries for the mask localization results computed in Section 3.2. Symmetrical results were omitted from the tables for brevity. As discussed in Section 4.3, the symmetrical inverse relationship can be inferred given one label for a tuple of objects. As an example, consider the Level One Summary shown in Table 4.2, where the Level One Summary shows that “person_1 overlaps and is above surfboard_1”. It follows naturally that given this relationship, it must be true that “surfboard_1 overlaps and is below person_1”. Table 4.2 is also an example of object localization and Level One Summary results that can be used as input to the Level Two Summary computations, discussed in depth in Chapter 5, since the objects could logically interact and their Level One Summaries support an interaction. Specifically, the Level One Summary for Table 4.2 supports the scene description “person_1 riding surfboard_1”.

Table 4.8 is an example of object localization and Level Two Summaries that can not be used as input to the Level Two Summary computations, as there are no logical interactions between the objects. However, as shown in the table, the Level One Summary outputs could be used as a final output of the S2T system as a general scene description.

Tables 4.9 - 4.11 show the Level One Summaries for the bounding box object localization results computed in Section 3.3. Table 4.9 once again shows the symmetrical property of Level One Summaries, but in addition, also shows the reason masks were considered to be more reliable than bounding box localizations when computing spatial relationships. The output of the HOF [40] algorithm was correct in that the majority of the bounding boxes for the person was to the right of and below the bounding box for the toothbrush. However, a tighter bound, computed via mask localization, would likely have ensured that the HOF results would indicate the toothbrush was to the left of the person and the person was to the right of the toothbrush. The overlap property computed using GIOU [51] provided the necessary information to the S2T system to indicate that the person was most likely holding the toothbrush. Using the combination of the GIOU and HOF properties was what allowed the S2T system to use the bounding box localization results reliably as input to the Level Two Summary evaluations of Chapter 5.

Table 4.11 shows an example of object localization results that were not usable as input to the Level Two Summaries of Chapter 5. As with Table 4.8, the Level One Summaries visualized in Table 4.11 could still be used as a final output of the S2T system. The following section discusses the data set used for Level One Summary evaluations and the manual inspection of the Level One Summaries that served as the validation for the evaluations of this chapter.

4.8 Discussion

This chapter detailed the methods used to generate the Level One Summaries for the object localization results generated in the evaluations of Chapter 3. Level one Summaries were used as the primary input to the “FIS Computation” stage that constructed the “Level Two Summaries”, discussed in Chapter 5, as was shown in the S2T pipeline diagram in Figure 4.1. Level one Summaries were computed by combining the outputs of the GIOU [51] and the HOF [40] algorithms. This combined output detailed if an object tuple a) overlapped/did not overlap and b) in what direction the object tuples are in relation to each other. Figure 4.1 also shows that Level One Summaries could be the final output of the constructed S2T system in this work. Section 4.7 discussed examples of when Level One Summaries were the final output of the S2T system. As a brief recap, Level One Summaries were the final output of the S2T system when object localization results contain no results that could logically interact to perform an action in the scene. The evaluations detailed in this chapter in effect implement the “HOF/GIOU Computation” stage of the S2T pipeline in Figure 4.1 and the output of the “HOF/GIOU Computation” stage were the “Level One Summaries” with respect to the pipeline diagram. Once the Level One Summaries were generated, the summaries were then a) passed as input to the “FIS Computation” stage of the pipeline or b) returned as the final output of the S2T system. The remainder of this section describes the data set used for evaluation and the manual validation of the Level One Summaries.

4.8.1 Level One Summary Data Set Description

Table 3.2 shows the localization results that were usable for Level One and Level Two Summaries. As can be seen in the table, there were 571 localization results that were usable for Level One Summaries and 506 localization results that were usable for Level Two Summaries. The 506 Level Two usable images were a subset of the 571 Level One usable images, and as such, the Level One Summaries were computed for the 571 images with Level One usable localization results. As discussed in Section 3.6.1, Level One usable localization results were those results that contained more than two object localizations, regardless if they were the same class of object, and whose masks/bounding boxes, along with labels, were in appropriate locations for all objects in an image.

Table 4.13

Level One Summary Totals

	Total
Input Images	571
Level One Summaries (Masks)	7482
Level One Summaries (Bounding Boxes)	184
Level One Summaries (Masks) Usable for Level Two	6274
Level One Summaries (Bounding Boxes) Usable for Level Two	120
Level One Summaries (Masks) Unusable for Level Two	1208
Level One Summaries (Bounding Boxes) Unusable for Level Two	64

Table 4.13 shows that for the 571 images with good localization results, 7482 Level One Summaries were computed for the mask localization results and 184 Level One Summaries were computed for the bounding box localization results. Table 4.13 also shows that of the 7482 Level One Summaries computed on the mask localization results, 6274 of them were usable for Level

Two Summaries and 1208 of them were only usable for Level One Summaries, according to the metrics discussed in Section 3.6.2. Similarly, Table 4.13 also shows that of the 184 Level One Summaries computed for bounding box localization results, 120 of them were usable for Level Two Summaries and 64 of them were only usable for Level One Summaries, again based on the metrics discussed in Section 3.6.2.

The 1274 Level One Summaries that could not be used as input to the Level Two Summary construction of Chapter 5 were the pending final output of the S2T system constructed in this work for the object localization results deemed unusable for Level Two Summaries. The 6394 Level One Summaries computed on object localization results deemed usable for Level Two were the pending input to the Level Two Summary evaluations of Chapter 5. Section 4.7 discusses each of the cases of when a Level One Summary was usable for Level Two Summary input and when a Level One Summary was the final output of the S2T system in detail. The following section discusses the manual inspection process used to ensure the Level One summaries generated were valid. This process served to prune the initial Level One Summary totals shown in Table 4.13.

4.8.2 Manual Inspection of Level One Summaries

As has been previously mentioned, both the GIOU [51] and HOF [40] algorithms were extensively validated in their respective original works. As such, no validation was performed on the individual methods used to generate the Level One Summaries. Rather, the individual validation of GIOU and HOF was relied upon when computing spatial relationships and overlap and it was assumed that the HOF and GIOU computations were consistent when constructing the Level One Summaries. Manual inspection of the Level One Summaries was performed afterwards to ensure that the output of the evaluations of this chapter were reliable and accurate. The remainder of this section details the manual inspection process used to verify the accuracy of the Level One Summaries computed.

A Python [45] script was written that loaded the Level One Summaries and object localization results into Pandas [62] data frames for an image from the corresponding CSV files. Afterwards, the mask or bounding box localization results were visualized using methods identical to those discussed in Sections 3.2.1.3 and 3.3.1.3 respectively. The Level One Summaries for the localization results of an image were then output to the Python console for inspection.

Table 4.14

Correct Level One Summaries

	Total	Correct Results
Level One Summaries (Masks)	7482	7482
Level One Summaries (Bounding Boxes)	184	184

The Level One Summaries for the 571 images with good localization results were inspected to verify the accuracy. Similar to the manual inspection of object localization results discussed in Section 3.6.2, the Level One Summaries were subjected to a pass/fail criteria test. This pass/fail test was constructed as a Python [45] script where the object localization visualizations and Level One Summaries for an image were displayed with the option to either accept or reject the Level One Summaries. It is important to note that only "overlap/does not overlap" was inspected for the Level One Summaries, and not the latent proximity information computed using the GIOU [51] algorithm. Table 4.14 shows that of the 7482 Level One Summaries for masks, all results were accurate. Table 4.14 also shows that of the 184 Level One Summaries for bounding boxes, all results were accurate. These results were expected because the Level One Summaries were constructed using two methods that have been previously validated extensively in their respective original works. This manual validation process also served to verify that the fuzzy membership functions and defuzzification to crisp outputs used for overlap and spatial relationships were implemented correctly. Both GIOU [51] and HOF [40] are deterministic processes, so they are not subject to randomness. By introducing the fuzzification and defuzzification process, a degree of randomness was introduced to both processes. Thus, the manual inspection validation process also served as a way to validate that the randomness introduced did not degrade the quality of the underlying algorithms. Table 4.14, shows that the randomness introduced did not degrade the performance of either the GIOU [51] nor the HOF [40] algorithms.

4.8.3 Conclusion

The results presented in Section 4.7, specifically the Level One Summary information conveyed in Tables 4.2 - 4.12 provide positive evidence to support that our first hypothesis correctly supports our first research question from Chapter 1:

RQ1: How can concise and meaningful natural language descriptions (Level One Summaries) of spatial relationships between all object tuples in an image be derived?

H1: Histogram of Forces (HOF) [40] and Generalized Intersection Over Union [51], both rigorously validated methods with strong mathematical proofs, will provide concise natural language descriptions of spatial relationship information between object tuples in a scene.

This chapter provided implementation details on how Level One Summaries were generated. A detailed explanation of how the individual methods, HOF [40] and GIOU [51], were used in the S2T system was given and the manner in which the HOF and GIOU results was combined resulted in the final Level One Summaries for each image with valid object localization results. The Level One Summaries that were deemed usable for Level Two Summaries will be used as input to the evaluations conducted in Chapter 5. The Level One Summaries that were not deemed usable as input to Level Two Summary computation were the final output of the S2T system for the object localization results of the image in question. Chapter 5 details the construction of a natural language scene description (Level Two Summary) by using object localization output, a result of the evaluations in Chapter 3, combined with Level One Summaries, the results of the evaluations conducted in this chapter.

CHAPTER 5

EVALUATION 2: LEVEL TWO SUMMARIES

Level Two Summary generation was the final phase of constructing the S2T pipeline. Level two Summaries provide natural language scene descriptions for an image. Level one Summaries were used as input to the Level Two Summary computation, and as such, Level Two Summaries have ingrained spatial relationship information. This chapter begins by revisiting the S2T pipeline diagram shown in Figure 5.1. Following, a discussion of how the objects were categorized into an object hierarchy is given. Level Two Summaries were constructed for the general domain, where any object interactions were considered, and for the person domain, where only person interactions with other objects were considered. A discussion of the implementations of a fuzzy inference system (FIS) for each domain is given following the object hierarchy discussion. Afterwards, details of the validation procedures for both the general domain FIS and person domain FIS are presented, followed by a discussion of the results of Level Two Summaries computed for the COCO [35] data set. Due to the nature of two-dimensional data, the Level Two Summaries computed by this evaluation were subject to inherent limitations, and these limitations are discussed in depth following the discussion of the general and person-domain Level Two Summaries. Finally, a discussion of the overall results obtained by the evaluations performed in this chapter are given.

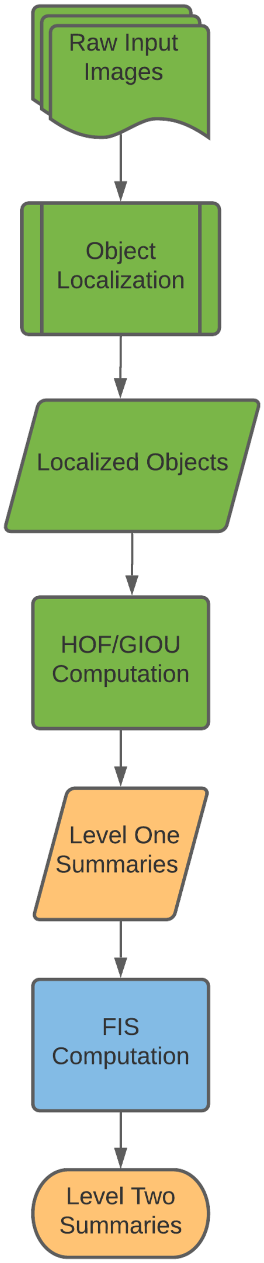


Figure 5.1

Level Two Summaries in the S2T Pipeline

5.1 S2T Pipeline

To begin, let us revisit the updated S2T pipeline diagram shown in Figure 5.1 and explore the current and completed stages of the system. Evaluations in Chapter 3 successfully implemented the “Object Localization” stages with the “Localized Objects” as output. “Localized Objects” served as input to the “HOF/GIOU Computation” stage of the pipeline and the evaluations conducted in Chapter 4 successfully implemented the “HOF/GIOU Computation”. Output of the evaluations detailed in Chapter 4 were the “Level One Summaries” in the S2T pipeline. Figure 5.1 shows that “Level One Summaries” can be the final output of the S2T system constructed (denoted by orange). Section 4.7 provides an in-depth analysis for when the Level One Summaries are the final output of the S2T system. Figure 5.1 also shows that “Level One Summaries” are used as input to the “FIS Computation” stage of the pipeline. Implementation, results, and analysis of the “FIS Computation” stage is the primary focus of this chapter. Figure 5.1 also shows that “Level Two Summaries” are the final output of the S2T system, denoted in orange, when the Level One Summaries were sufficient to be used as input to the “FIS Computation” stage. As such, an in-depth analysis and discussion of the Level Two Summaries generated by the evaluations of this chapter are given.

5.2 Object Hierarchy

Chapter 4 briefly introduced the concept of the object hierarchy used for the S2T pipeline. The remainder of this section discusses the implementation of the object hierarchy in detail. It is important to note that the ranking of objects was chosen based around the generation of person-domain Level Two Summaries, discussed in Section 5.4. Thus, the object hierarchy is an entirely subjective, but necessary construct, as the object hierarchy was used as the premise to construct the fuzzy rule base for the person-domain Level Two Summaries detailed in Section 5.4. A discussion on the creation of the object hierarchy for the general domain Level Two Summaries follows next.

Before segmenting the set of objects into general and domain specific categories, an object hierarchy was developed that served to rank order the tuples of objects in an image in terms of most important to least important interactions. Three hierarchy categories were decided upon based on an examination of the COCO [35] data set: "person," "animate," and "inanimate" objects. The "person" object was assigned its own category, and was given precedence over all other objects in the data set. The evaluations of Section 5.4 focus primarily on person interactions with other objects, and as such, the hierarchy was developed to allow person detections to exist outside the realm of other categories. By doing so, the object hierarchy was able to be constructed based around the idea that the "person" category formed the basis of the hierarchy where all other categories served as supporting categories with which the person would interact. The "animate" hierarchy consists of any objects that are living. While it was true that the only animate objects in the COCO [35] data set were people and various animals, separating people from animals allowed the segmentation to give higher importance to person interactions over animal interactions. The "inanimate" category was the most enumerated category and contained all other objects that a person or an animate object could interact with.

The COCO [35] data set was used alongside the object hierarchy as a basis for constructing the hierarchy. The data set contained 90 different objects that correspond to objects the mask [21] and bounding box [37] [48] object detection algorithms could positively identify. Each of these objects were assigned a general category and a sub-category. The general categories, as the name suggests, served as a general description of the type of object. For example, the “vehicle” general category is assigned the objects: bicycle, car, motorcycle, airplane, bus, train, boat, and truck. The sub-categories were constructed based on how a person would interact with the object. The constructed person domain FIS hinged upon how a person would interact with the objects and as such, the FIS rule bases revolved around the object hierarchy general and sub-categories. With this in mind, each sub-category can be viewed as a way to segment the interactions a person would perform with an object. For example, the vehicle category was broken into two sub-categories: “personal” and “passenger.” The “personal” sub-category contained the following vehicles, which a person would drive: bicycle, car, motorcycle, truck, and boat. The “passenger” sub-category contains the following vehicles, which a person would ride in: airplane, bus, and train. Table 5.1 shows the hierarchy broken down into general categories and sub-categories.

Table 5.1

Object Hierarchy

General Category	Sub-Categories
animal	pet, ridden, large animal
vehicle	personal, passenger
urban	sidewalk, road
sports	frisbee, skiing, snowboarding, flying kite
sports	baseball, skateboarding, surfing, tennis, soccer
clothing	worn, carried
food/tableware	eaten, drank, used
furniture/decor	sat on, used
electronics	watched, talked on, used
household	used, held

The general-domain Level Two Summaries, discussed in Section 5.3, were not dependant upon the use of the object hierarchy. These summaries were not concerned with the actions being performed, rather these summaries just indicated whether two objects were interacting. The general-domain Level Two Summaries provided a general summary of interactions between object tuples in an image. The person-domain Level Two Summaries, discussed in Section 5.4, made use of the sub-categories of the object hierarchy. These Level Two Summaries provided more refined detail centered around people and how they interacted with various objects of differing categories. As such, the person-domain Level Two Summaries provided a summary of the actions being performed in an image.

It is important to note that any number of domains could be constructed given the set of objects from the COCO [35] data set. The person domain category was chosen for this dissertation as an example of how to construct a domain specific set of rules that used a FIS to make inferences about scene interactions. Additionally, the person domain was chosen for these evaluations because it was the most interesting domain given the general data set used. This research could be applied to any number of domains with the caveat that a domain expert would be required to construct rules for very specific domains. Conversely, the general-domain rules could be applied as-is to any new domain.

The remainder of this chapter is devoted to the construction, results, and validation of the Level Two Summaries for both the general domain and the person domain. The following section provides an in-depth discussion of the general-domain Level Two Summaries.

5.3 General-Domain Level Two Summaries

This section describes the design and implementation of the FIS used to generate the general-domain Level Two Summaries. To start, a description of the fuzzy rule base for the FIS was given, followed by a description of the implementation of the FIS itself. Afterwards, a discussion and analysis of the results of applying the general-domain FIS to the Level One Summaries, described in Chapter 4, is presented. Finally, the validation and computed metrics for the general domain FIS is detailed.

5.3.1 General-Domain Fuzzy Rule Base

The first task in developing the general-domain FIS was constructing the fuzzy rule base for the system. Developing the rule base for a FIS was a manual task and required domain knowledge for constructing appropriate rules that reflected the desired outputs during inference. The general-domain rule base was very succinct, as the general-domain Level Two Summaries simply described the object tuples in an image that were interacting in some form. As such, the object hierarchy categories, discussed in Section 5.2, were not used in the development of the general-domain fuzzy rule base. Instead, the object hierarchies discussed in Section 5.2, were used to structure the tuples of objects in each image.

The general-domain Level Two Summaries were meant to provide a high-level description of interactions in the scene. Therefore, the rule base was kept very simple and there was no ingrained domain-specific information in the fuzzy rule base. Instead, the general-domain fuzzy rule base was constructed based around the premise that if two objects were overlapping, or in very close proximity, then they were likely interacting with each other. This property allowed the general-domain rule base to be applied to additional data sets with very little to no work at all. Table 5.2 shows the fuzzy rule base used for the general-domain FIS computations.

Table 5.2

General-Domain FIS Rule Base

IF *Overlap* AND *Very Close* OR *Close* THEN *Interacting*
IF *No Overlap* AND *Very Close* THEN *Interacting*
IF *No Overlap* AND NOT *Very Close* THEN *Not Interacting*
IF *Overlap* AND NOT *Very Close* OR *Close* THEN *Not Interacting*

Table 5.2 shows the entire rule base used to construct the general-domain FIS for these evaluations. As seen in the table, the general-domain FIS relied solely on the proximity and overlap membership functions, which corresponded to the output of the GIOU [51] algorithm computed in Section 4.4.1 for the set of input images. A detailed description of the overlap and proximity membership functions was given in Section 4.4.1 for the reader. The following section provides the implementation details for the general-domain FIS used in these evaluations, which used the general-domain fuzzy rule base outlined in this section.

5.3.2 General-Domain FIS Implementation

The SKFuzzy [69] library was used to implement the FIS for this evaluation. All code for the implementation was written using the Python [45] programming language. After generating the general-domain Level Two Summaries, the results were stored in CSV files for easy retrieval, using the image name as a key for mapping between input and output values. Input values to the general-domain Level Two Summary computation were the Level One Summary numerical values for proximity and overlap, computed in Section 4.4.1. The general domain FIS made use of the proximity and overlap membership functions, shown in Figures 4.3 and 4.2 respectively, to fuzzify the overlap and proximity input values. As discussed in Section 5.3.1, the fuzzy rules were kept high-level for this implementation and doing so allows extension to any domain with very minimal work. The remainder of this section describes the steps used to generate the general-domain Level Two Summaries in detail.

$$\mu_y(y) = min(\mu_{y^1}(y), \mu_{y^2}(y), \ldots \mu_{y^n}(y)), \text{ for } y \in Y \tag{5.1}$$

$$y_t = max[min[\mu_{y^1}(y), \mu_{y^2}(y), \ldots \mu_{y^n}(y)], \text{ for } y \in Y \tag{5.2}$$

Algorithm 5.1 General-Domain Level Two Summary Computation

```
input: Level One Summaries for an input image
output: General-Domain Level Two Summaries
    Set T = Hierarchically ordered Level One Summary results
    for $t \in T$ do
        Set $p_t$ to the proximity score of the Level One Summary
        Set $o_t$ to the overlap score of the Level One Summary
        Set $p_t$ and $o_t$ as the input of the General-Domain FIS
        Set $y_t$ to the aggregate of the conjunctive rules using Equation 5.2
        Set $y_t^*$ to the centroid defuzzified value of $y_t$
        Assign $y_t^*$ as the general-domain Level Two Summary of $t$
    end for
```

The FIS implementation was based around a conjunctive system of rules. That is, the union of the antecedents was taken to compute the consequent. The FIS was modeled as a Mamdani [53] fuzzy control system using the SKFuzzy [69] library, which aggregated the set of rules shown in Table 5.2 and the resulting consequent output of the system were the fuzzy membership values for

“Interacting” and “Not Interacting.” Once the fuzzy consequent values were obtained, centroid defuzzification [60] was used to assign the crisp output label to the object tuple.

Equation 5.1, also given by Ross in [53], shows the method used to compute the output of the individual rules. Since the rules of the system were conjunctive rules, the minimum operator was used to aggregate the input membership functions. Each μ_{y^n} corresponds to an individual membership function, such as the proximity and overlap membership functions shown in Figures 4.3 and 4.2 respectively. Each membership function takes as input a fuzzy value for y. The final output, μ_y denotes the fuzzy consequent value for an individual rule, such as the rules shown in Table 5.2. Stated another way, the consequent value for each rule was obtained by applying the union (AND) operator to each fuzzy input. Taking Table 5.2 as an example, there will be four μ_y values computed, one for each of the rules in the table.

Equation 5.2, shows the method used to aggregate the outputs of the individual rules of the FIS. As shown in Equation 5.2, the method is just an extension of Equation 5.1, where the *max* operation was performed on the conjunctive rule outputs computed using Equation 5.1 to find the maximum membership of the output of the aggregated conjunctive rules. The value μ_y can be viewed as the output of the individual rules, whereas the value y_t can be viewed as the fuzzy output of the aggregation of the consequents of all rules in the rule base. Performing the *max* operation allows the FIS to assign the most likely consequent fuzzy value given the set of inputs. Viewed another way, Equation 5.2 allowed the FIS to assign the most likely output fuzzy membership value to a set of inputs. Centroid defuzzification [60] was used to generate the final crisp output for the fuzzy value y_t.

Algorithm 5.1 details the process for computing the general-domain Level Two Summaries on the set of input images. The algorithm was performed on the results for a single image's Level One Summaries, computed in Chapter 4 and each input image had *n* Level One Summaries where *n* corresponded to the number of object tuples in an image. The output of the algorithm was *n* general-category domain Level Two Summaries, or in other words, a Level Two Summary for each Level One Summary. While each Level One Summary had a corresponding Level Two Summary in the general-category domain, it is important to note that only Level Two Summaries for objects that were interacting were presented in the results, which served to reduce the noise of the output labels for an image. Any object tuple assigned the "Not Interacting" label were omitted from the results presented to the user.

The Level One Summaries were ordered hierarchically, using the method described in Section 5.2. Using this ordering, Algorithm 5.1 processed Level One Summaries in the following order:

- person - person interactions
- person - animate interactions
- person - inanimate interactions
- animate - animate interactions
- animate - inanimate interactions
- inanimate - inanimate interactions

For each t in the hierarchically-ordered Level One Summaries T, the algorithm first assigned p_t as the proximity score of the Level One Summary and o_t as the overlap score of the Level One Summary. The values p_t and o_t were set as input to the fuzzy control system, implemented using SKFuzzy [69], and the fuzzy control system computed the aggregate of the conjunctive rules using Equation 5.2 for the general domain FIS. Note that the spatial relationship values for the

Level One Summaries were not used in the general-domain FIS computation. This is because the general-domain FIS was meant to be a very high-level description of the interactions in the scene, and the proximity and overlap values were sufficient to determine such interaction labels. The value y_t was then defuzzified into a crisp scalar using the centroid defuzzification method [60] and the resulting value y_t^* was assigned as the general-domain Level Two Summary for the input Level One Summary t. This process continued for all of the input Level One Summaries for an image. Additionally, the algorithm was applied to the set of input images for which there were adequate Level One Summaries, described in Section 4.8. The following section discusses in depth the results obtained from applying Algorithm 5.1 to the Level One Summaries deemed viable for Level Two Summary computation.

5.3.3 General-Domain FIS Results

Table 5.3

General-Domain Level Two Summaries for Figure 3.3

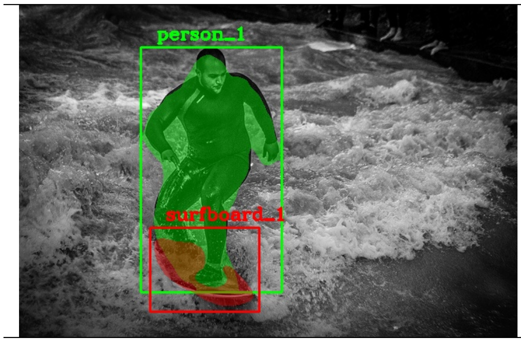
Localization Result	General Level Two Summaries
	• person_1 interacting with surfboard_1

Table 5.4

General-Domain Level Two Summaries for Figure 3.4

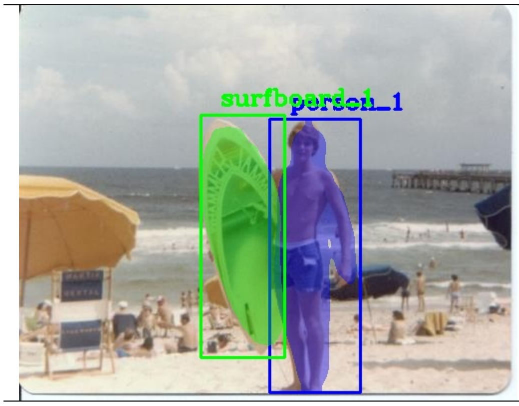
Localization Result	General Level Two Summaries
	• person_1 interacting with surfboard_1

Table 5.5

General-Domain Level Two Summaries for Figure 3.5

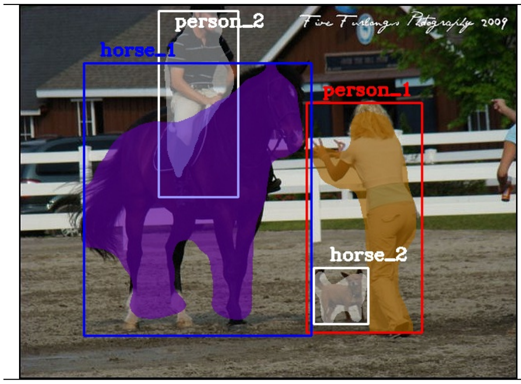
Localization Result	General Level Two Summaries
 person_2 Five Furlongs Photography 2009 horse_1 person_1 horse_2	• person_1 interacting with horse_1 • person_1 interacting with horse_2 • person_2 interacting with horse_1

Table 5.6

General-Domain Level Two Summaries for Figure 3.6

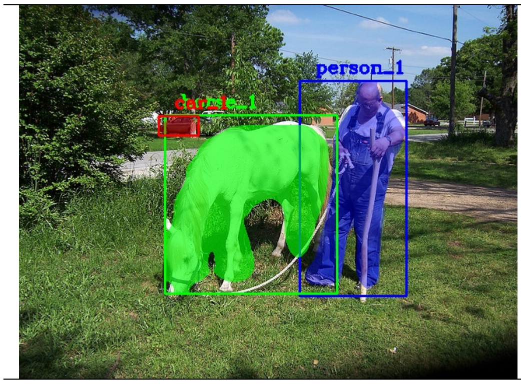
Localization Result	General Level Two Summaries
	• person_1 interacting with horse_1 • horse_1 interacting with car_1

Table 5.7

General-Domain Level Two Summaries for Figure 3.7

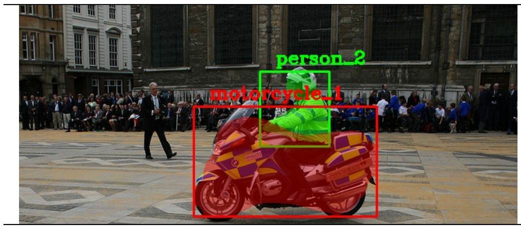
Localization Result	General Level Two Summaries
	• person_2 interacting with motorcycle_1

Table 5.8

General-Domain Level Two Summaries for Figure 3.8

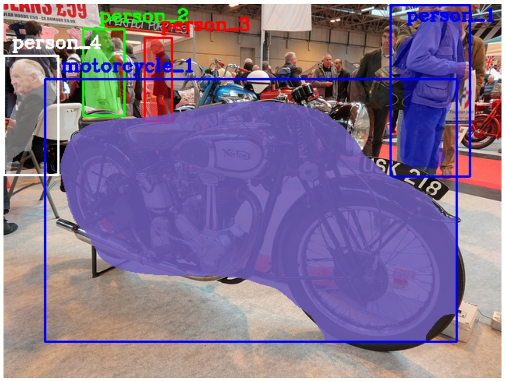
Localization Result	General Level Two Summaries
 person_2 person_3 person_1 person_4 motorcycle_1	• person_1 interacting with motorcycle_1 • person_2 interacting with motorcycle_1 • person_3 interacting with motorcycle_1 • person_4 interacting with motorcycle_1

Table 5.9

General-Domain Level Two Summaries for Figure 3.10(b)

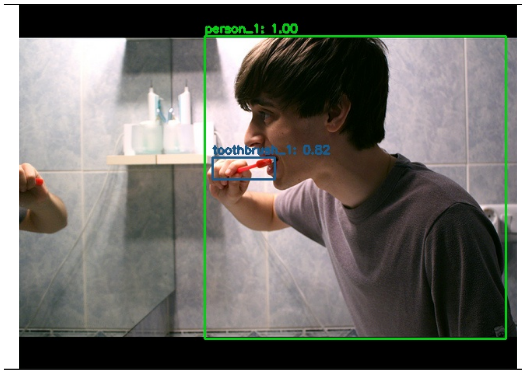
Localization Result	General Level Two Summaries
	• person_1 interacting with toothbrush_1

Table 5.10

General-Domain Level Two Summaries for Figure 3.10(d)

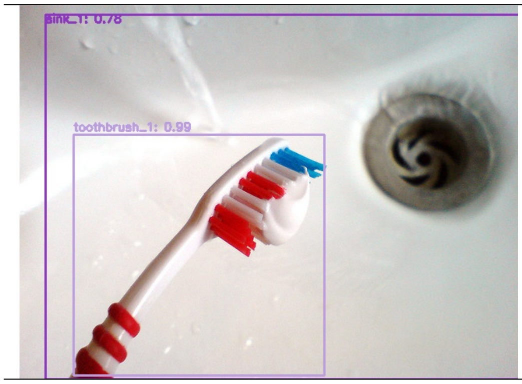
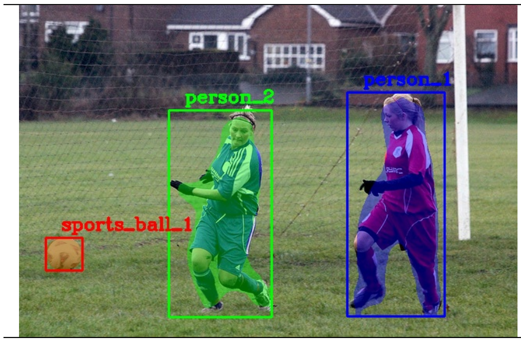
Localization Result	General Level Two Summaries
	- toothbrush_1 interacting with sink_1

Table 5.11

General-Domain Level Two Summaries for Figure 3.12

Localization Result	General Level Two Summaries
	- No applicable interactions apply

Tables 5.3 - 5.11 show the results generated by the general-domain FIS. Each table displays the localization results on the left with the corresponding general-domain Level Two Summaries enumerated on the right. For mask localization results, the bounding box localization results were also visualized, such as in Table 5.3. The bounding box localization results were visualized because the GIOU [51] algorithm required bounding boxes as input, and as such the proximity

and overlap values were computed based on box localization results. As mentioned previously, the general-domain summaries were meant to provide very high-level details about interactions in the scene. As such, the same label could be applied to different images even though the objects were performing different interactions. As an example, Table 5.3 and Table 5.4 both contained the single label of "person_1 interacting with surfboard_1." These labels were considered correct labels for the general-domain summaries. Section 5.4 discusses the process of generating labels that provide more refined descriptions of interactions in the scene.

Tables 5.3 - 5.11 also show that general-domain summaries were computed for every object two tuple interaction in the scene. For example, Table 5.6 shows the interaction label: "horse_1 interacting with car_1." Person-domain Level Two Summaries, discussed in Section 5.4, only apply labels based on person - object interactions. These labels were a refined set of labels that described in great detail what actions a person was performing in an image. However, for images with no person detection results, the general-domain Level Two Summaries could still be used to apply a general set of interactions occurring in the scene, with the caveat that the information will be very high-level and not refined.

Table 5.11 shows an interesting case where there were no general-domain Level Two Summaries computed, but as can be seen in Section 5.4.3, a person-domain label was generated for the image successfully. No general-domain interactions were computed due to the lack of any object tuple being in close proximity to each other. This highlights the case where general-domain person - object interactions were not required in order to generate a person-domain Level Two Summary. Specifically, for the soccer example, it was a likely scenario that no person would be detected in close proximity to the soccer ball, as was the case in Table 5.11. Additionally, as will be discussed in Section 5.4.3, Table 5.11 also shows a localization result where Inception [61] metadata was necessary to determine the actions being performed. Namely, the "sports_ball_1" detection was not descriptive enough to determine the correct rule to apply when constructing person-domain Level Two Summaries.

Table 5.5 shows that both person_1 and person_2 are interacting with horse_1, and that person_1 is interacting with horse_2. While horse_2 was a bad object detection result, for the purposes of the general-domain summaries, the label was considered correct. Section 5.4.3 shows that the invalid detection was nullified in the person-domain labels, as person_1 was not detected as riding horse_2. Table 5.6 shows that person_1 is interacting with horse_1 and horse_1 is interacting with car_1. Table 5.7 shows that person_2 is interacting with motorcycle_1. Table 5.8 shows the following interactions: person_1 interacting with motorcycle_1, person_2 interacting with motorcycle_1, person_3 interacting with motorcycle_1, and person_4 interacting with motorcycle_1. Table 5.9 shows that person_1 is interacting with toothbrush_1. Lastly, Table 5.10 shows that toothbrush_1 is interacting with sink_1. The statistics for the general-domain Level Two Summaries are reserved for Section 5.6. The following section discusses the method of validation used for the general-domain Level Two Summaries.

5.3.4 General-Domain FIS Validation

The validation procedure performed for the general-domain Level Two Summary validation followed a procedure similar to the one used to validate the Level One Summaries, computed in Chapter 4. Because there was no ground truth baseline for general-domain Level Two Summaries for the COCO [35] data set, and because the general-domain level summaries were, to an extent, subjective, a manual validation strategy was decided upon to verify that the labels produced by the general-domain FIS were an adequate reflection of the interactions being performed in the scene.

A Python [45] script was written that loaded the general-domain Level Two Summaries and object localization results into Pandas [62] data frames. For each object tuple in an image, the corresponding localization results were displayed using the OpenCV [5] library, identical to the methods discussed in Sections 3.2.1.3 and 3.3.1.3. Whereas the visualizations of the general-category Level Two Summaries only contained the objects that were determined to be interacting, the manual inspection process displayed all of the general-category Level Two Summaries to the user in the Python console for inspection. By doing this, the manual inspection process could be used to determine if an object tuple that was determined to be “Not Interacting” was a false negative, or in other words should have been labeled as “Interacting”. This process was performed for each of the 506 images that were deemed usable as input to Level Two Summary computation. The metrics used to determine if an image’s localization results were usable as input to Level Two Summary computation was discussed in detail in Section 3.6.1.

Section 5.6 provides further analysis for the general-domain Level Two Summaries, but for the purposes of validation, consider Table 5.25. Table 5.25 shows that the general-domain Level Two FIS performed well for the purpose of computing general object interactions for a S2T system. The remainder of the discussion of the general-domain Level Two Summary results along with in depth analysis of computed statistics is reserved for Section 5.6.

As a final note, Table 5.25 shows that the inaccurate general-domain Level Two Summaries were all due to the problem of missing depth information. This occurred when a system failed to correctly label an object interaction because it was not possible to determine the depth between two objects in strictly a two-dimensional space. Consider the general-domain interactions in Table 5.6. While the "person_1 interacting with horse_1" label was accurate, the "horse_1 interacting with car_1" label was not. This shows an example where depth information could alleviate invalid labels, but such information was not available for the object detection algorithms used for this research at the time of writing.

5.4 Person-Domain Level Two Summaries

This section presents the design and implementation details of the FIS used to generate the person-domain Level Two Summaries. To begin, the construction of the fuzzy rule base for the person-domain FIS is discussed. Following the discussion of the rule base, the implementation of the person-domain FIS is detailed. Afterwards, an in-depth analysis of the generated person-domain Level Two Summaries is given, followed lastly by a discussion of the validation procedures and statistics obtained from the generated Level Two Summaries for the input object localization results and Level One Summaries, computed in Chapters 3 and 4 respectively.

5.4.1 Person-Domain Fuzzy Rule Base

The first task for constructing the person-domain Level Two Summaries was generating a rule base to use for developing the person-domain FIS. Similar to the process discussed in Section 5.3.1, constructing a rule base was a manual task, which required domain-specific knowledge to construct rules that appropriately reflected the desired outputs during inference. Unlike the general-domain rule base, the person-domain rule base was a very large and detailed rule base. This was a requirement because computing the person domain interactions changed based on the type of object a person was interacting with. The object hierarchy categories, discussed in Section

5.2 were used extensively when developing the fuzzy rule base for the person-domain and when determining which rules should be applied to the person - object tuple being examined. The object hierarchies, also discussed in Section 5.2, were used to structure the person - object tuples such that “person - animate” interactions were computed first, followed by “person - inanimate” interactions. It is important to note that no rules were developed for “person - person” interactions. During evaluation, it was determined that pose and depth estimation would be essential for computing such interactions and this information was not available from the algorithms used to generate the Level One Summaries, discussed in Chapter 4. It is also important to note that given enough manpower and time, any number of domains can be developed using the provided object hierarchy, but this research focused only on the person-domain as a proof of concept to construct a S2T system that provided refined labels, which incorporated spatial relationship information.

The person-domain Level Two Summaries were meant to provide refined descriptions of person - object interactions in the scene. In order to accomplish this task, a rule base was constructed for each of the object categories shown in Table 5.1. Each rule base contained encoded information around how a person would interact with objects from the category in question. Additionally, unlike the general-domain FIS implementation, a FIS was required for each category, and the object category determined which FIS should be called for the person - object tuple. These FIS implementations are discussed in detail in Section 5.4.2. It is important to note that the sub-categories of a category could potentially require the construction of different rules to determine the appropriate interaction being performed. As such, the sub-categories of the object hierarchy were also used to determine the appropriate set of rules to apply to a person - object tuple. Table 5.12 shows the person - animal rule base, which exhibited the properties of both a rule base limited to a specific object category, but also rules that fired based on the sub-category of the object the person was interacting with.

Table 5.12

Person-Animal Domain FIS Rule Base

IF *Pet* AND *Overlap* AND *Very Close* OR *Close* AND NOT *Below* THEN *Petting*
IF *Pet* AND *No Overlap* AND *Very Close* AND NOT *Below* THEN *Petting*
IF *Ridden* AND *Overlap* AND *Very Close* OR *Close* AND *Above* THEN *Riding*
IF *Ridden* AND *No Overlap* AND *Very Close* AND *Above* THEN *Riding*
IF *Large Animal* AND *Overlap* AND *Very Close* THEN *Interacting*

Consider, as an example, Table 5.12, which shows the rule base for the person - animal domain. This table shows the rules for the person - animal domain interactions. Only the rules for positive interactions are shown in order to keep the table succinct. The entire list of rules for the person - animal domain, as well as all other domains, is shown in Appendix A. Table 5.12 shows that the person - animal domain used the overlap, proximity, and spatial relationship fuzzy input values, which were represented by the fuzzy membership functions in Figures 4.2, 4.3, and 4.4 respectively. The underlying algorithms, GIOU [51] and HOF [40], used for computing these fuzzy values were discussed in Sections 4.4.1 and 4.5.1 respectively. Table 5.12 shows an instance of a rule base that was applied based on the general category of the object hierarchy, namely the animal category. Additionally, Table 5.12 shows how an individual category used the sub-category information to determine the appropriate rules to apply. As can be seen in the table, the animal category contained rules for three sub-categories: pets, ridden animals, and large animals. Each sub-category contained a specific interaction that a person and an animal could perform: petting, riding, and interacting. For the *Petting* interaction, the rule base shows that a person needed to be relatively close and not below the animal in order to be considered to be petting the animal. The *Riding* interaction shows that a person needed to be relatively close and above the animal in order to be considered as riding the animal. For large animal interactions, the rule was similar to the rules for general-domain interactions, where the person needed to be overlapping and very close to fire the *Interacting* label. The COCO [35] data set, only contained images for large animals (bears, giraffes, and zebras) where a person was either in close or far proximity to the animal. As such, the rule was modeled to follow similarly to the general-category rules where a person either was or was not interacting with the animal.

As mentioned previously, the person-domain Level Two Summaries required an extensive and fairly large set of rules. These rules were necessary to infer appropriate Level Two Summaries centered around a person and object interaction. For brevity, the entire rule base is not presented in this chapter, rather the person-domain rule base for Level Two Summaries is reserved to Appendix A. The following section discusses the implementation of the FIS for the person-domain Level Two Summaries in detail.

5.4.2 Person-Domain FIS Implementation

This section describes the overall implementation details for the person-domain Level Two Summary FIS. As mentioned in the previous section, an individual FIS was constructed for each object category in the object hierarchy, but the process for implementing the FIS remained the same throughout with the exception of rules used. The SKFuzzy [69] library was used to implement each FIS and all code was written using the Python [45] programming language. The person-domain Level Two Summaries were stored in CSV files for easy retrieval and the image name was used as

Algorithm 5.2 Person-Domain Level Two Summary Computation

```
input: Level One Summaries for an input image
output: Person-Domain Level Two Summaries
    Set T = Hierarchically-ordered Level One Summary results that contain a person
    for $t \in T$ do
        Set $p_t$ to the proximity score of the Level One Summary
        Set $o_t$ to the overlap score of the Level One Summary
        Set $sr_t$ to the spatial relationship score of the Level One Summary
        Set $c_t$ to the object hierarchy category of the object the person is interacting with
        Set $FIS_{c_t}$ to the FIS corresponding to the object category $c_t$
        Set $p_t$, $o_t$ and $sr_t$ as the input of $FIS_{c_t}$
        Set $y_t$ to the aggregate of the conjunctive rules using Equation 5.2
        Set $y_t^*$ to the centroid defuzzified value of $y_t$
        Assign $y_t^*$ as the person-domain Level Two Summary of $t$
    end for
```

a key to map between the input image and output person-domain Level Two Summaries. Inputs to the person-domain Level Two Summaries consisted of the Level One Summary numerical values for overlap, proximity, and spatial relationships, which were computed in Sections 4.4.1 and 4.5.1. The person-domain FIS used the overlap, proximity and spatial relationship membership functions, shown in Figures 4.2, 4.3, and 4.4 respectively, to fuzzify the input values. Section 5.4.1 shows an example rule base for the person - animal interaction FIS and described the complexity of the entire rule base. The entire rule bases used to construct the person domain Level Two Summary FIS are reserved to Appendix A. The remainder of this section expands on the implementation details for a person domain Level Two Summary FIS.

Similar to the process discussed in Section 5.3.2, each FIS constructed for the person-domain Level Two Summaries were based around a conjunctive system of rules where the union of the antecedents were taken to compute the consequent. These FIS were modeled as Mamdani [53] fuzzy control systems using the SKFuzzy [69] library. The rules were aggregated using the

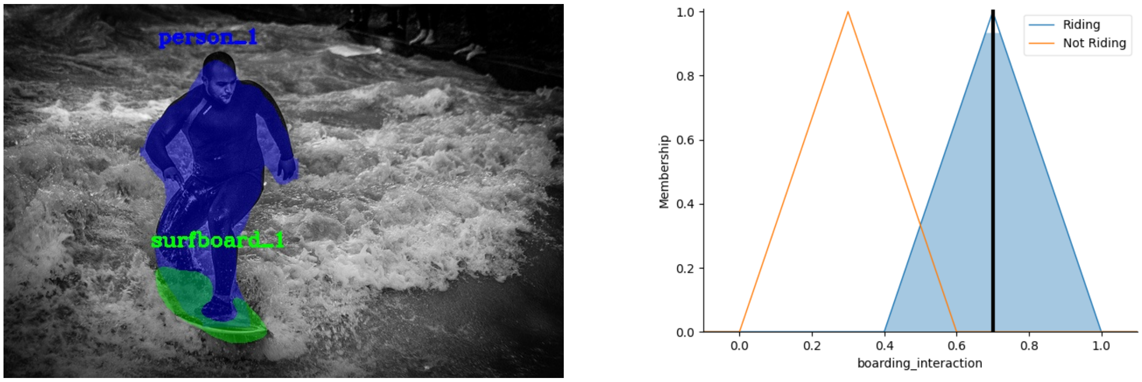
Mamdani fuzzy control systems and centroid defuzzification [60] was performed on the output consequent values to assign the crisp output to the corresponding person - object tuple. The reader is directed to Appendix A for a full listing of the fuzzy rules, which were used to construct the individual control systems.

Because the person-domain Level Two FIS used the same underlying process, namely the Mamdani [53] fuzzy control system, Equations 5.1 and 5.2 were used in each FIS to aggregate the rule output of each system. Once rules were aggregated, centroid defuzzification [60] was applied to the consequent output to determine the final label to assign as the person-domain Level Two Summary, identical to the process discussed in Section 5.3.2. Computation of the person-domain Level Two Summaries does follow a similar process to computation of general-domain Level Two Summaries (Algorithm 5.1) but with a few additions to incorporate necessary information to make a more informed decision about the person - object interactions occurring in the scene. This process and the additions are shown in Algorithm 5.2.

Algorithm 5.2 details the process used to compute the person-domain Level Two Summaries from the set of Level One Summaries implemented in Chapter 4. Like the general-domain Level Two Summary computations, the person-domain Level Two Summary computations were performed on a single image's Level One Summaries. Thus, Algorithm 5.2 was applied to each image's Level One Summaries that contained at least one person - object interaction. In other words, any images with Level One Summaries that did not describe a person interacting with an object were omitted from the person-domain Level Two Summary computation. This restricted set of Level One Summaries is denoted by T in Algorithm 5.2. The output of Algorithm 5.2 was the set of person-domain Level Two Summaries for an input image. It is important to note that for the purposes of visualizing the results of this research, only the positive person interaction labels from the person-domain FIS were presented as the output of the FIS to reduce the noise in the results section. For example, consider the person - horse interactions, where one of “riding” or “not riding” would be assigned as a label. While the system itself would output either of these labels, the results in Section 5.4.3 omit the “not riding” labels.

Algorithm 5.2 begins by ordering the set T hierarchically from greatest to least: “person animate object interactions” and “person inanimate object interactions” where each tuple $t \in T$ was processed using the process that follows. The values p_t, o_t, and sr_t were assigned as the proximity score, overlap score, and spatial relationship score of the Level One Summary corresponding to t. The value c_t was set to the object hierarchy category of the object in the Level One Summary that the person was interacting with. Based on c_t, the values sp_t, o_t, and sr_t were assigned as input to FIS_{c_t}, where FIS_{c_t} corresponded to the FIS for the object category c_t. Algorithm 5.2 then followed the same process used in Algorithm 5.1, where the value y_t was assigned to the aggregate of the conjunctive rules of FIS_{c_t} using Equation 5.2. The value y_t was then defuzzified using centroid defuzzification [60] to generate the crisp output label y_t^* and this value was assigned as the person-domain Level Two Summary of the tuple t. This process was performed for all images with Level One Summaries that contained a person - object interaction. The initial pruning of the Level One Summaries to determine the viability for use in Level Two Summary computations was discussed in Section 4.8, but an additional pruning process was performed to limit the person-domain Level Two Summary computation to those Level One Summaries with a person - object interaction.

Figures 5.2 - 5.8 show examples of the output obtained by processing the input images we have been working with throughout this dissertation through the person-domain FIS. All images were generated using the SKFuzzy [69] library in Python [45]. Similar to Figures 4.5 - 4.11, FIS output was generated for only one tuple, deemed to be the most interesting interaction for each image, for brevity. The membership function outputs, (b) in each image, show the degree of membership of the positive/negative interaction being performed. On the x-axis, the interaction type is shown, where positive interactions ranged from 0 to 1 and negative interactions ranged from 0 to -1. The black solid vertical line indicates what the FIS determined was the label to assign, and this determination was made via centroid defuzzification. This assigned label was essentially the output of the Level Two Summary FIS for the person-domain. Figure 5.2(b) shows that the person was detected as riding the surfboard in the image. Figure 5.3(b) shows that the system detected person_1 was not riding the surfboard in the image. Figure 5.4(b) infers that person_2 was riding horse_1 in the image. Figure 5.5(b) indicates that the person was not riding the horse. Figure 5.6(b) presents a result where there was some degree of membership for both the positive and negative interactions, but the end result from defuzzification generated a label indicating the person was riding the motorcycle. Figure 5.7(b) shows that person_1 (and all other people, excluded for brevity) was detected as not riding the motorcycle in the image. Finally, Figure 5.8(b) was another instance where both the positive and negative interactions had some degree of membership, but the end result was that the system detected that person_1 was playing soccer with sports_ball_1. The following section details the results obtained from applying Algorithm 5.2 to the limited set of Level One Summaries in detail.

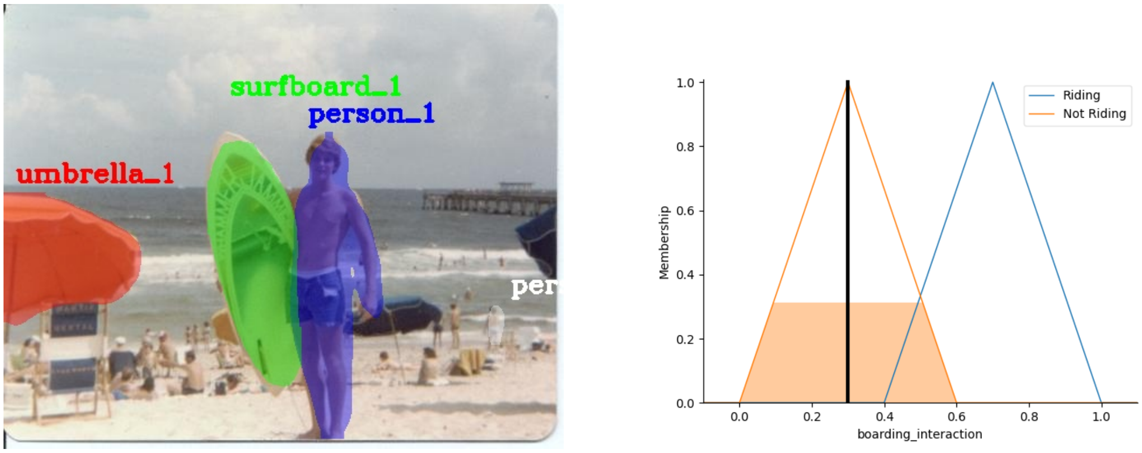


(a) (b)

Figure 5.2

Person riding surfboard FIS output

(a) Mask localization output (b) FIS output

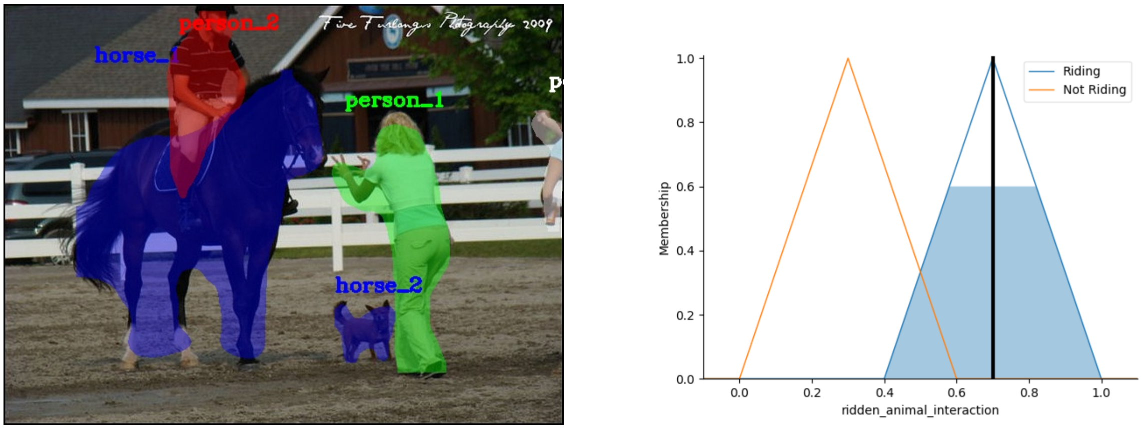


(a) (b)

Figure 5.3

Person holding surfboard FIS output

(a) Mask localization output (b) FIS output

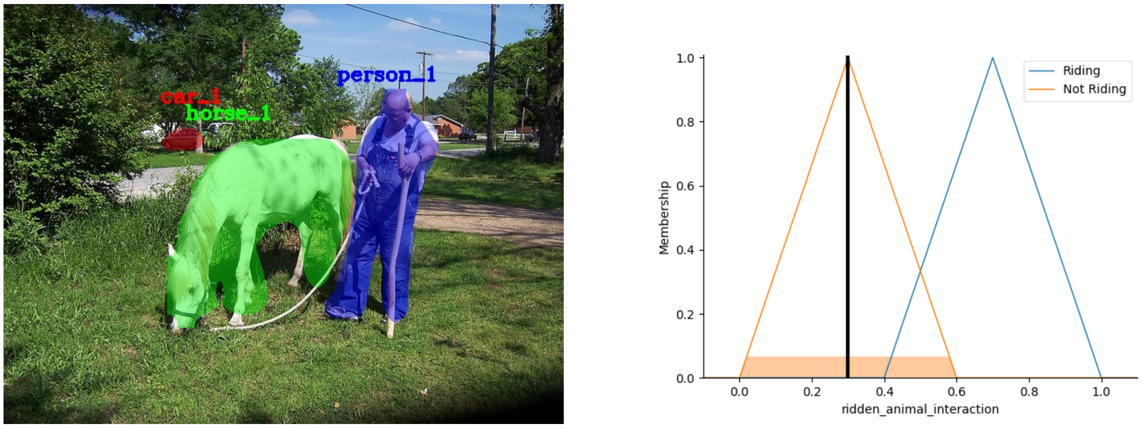


(a) (b)

Figure 5.4

Person riding horse FIS output

(a) Mask localization output (b) FIS output

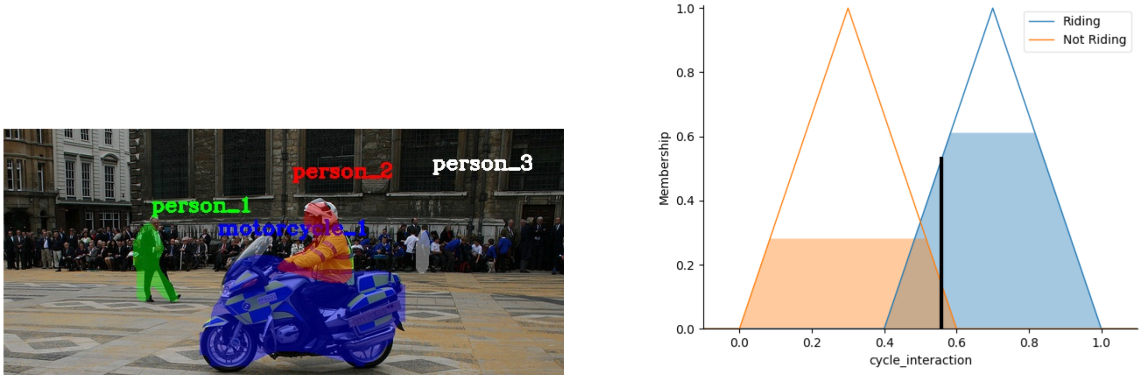


(a) (b)

Figure 5.5

Person beside horse FIS output

(a) Mask localization output (b) FIS output



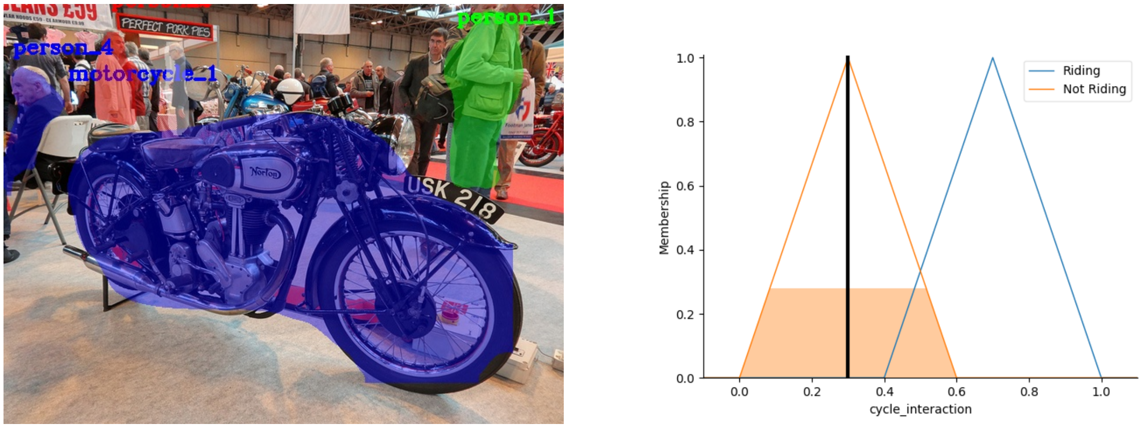
(a)

(b)

Figure 5.6

Person riding motorcycle FIS output

(a) Mask localization output (b) FIS output



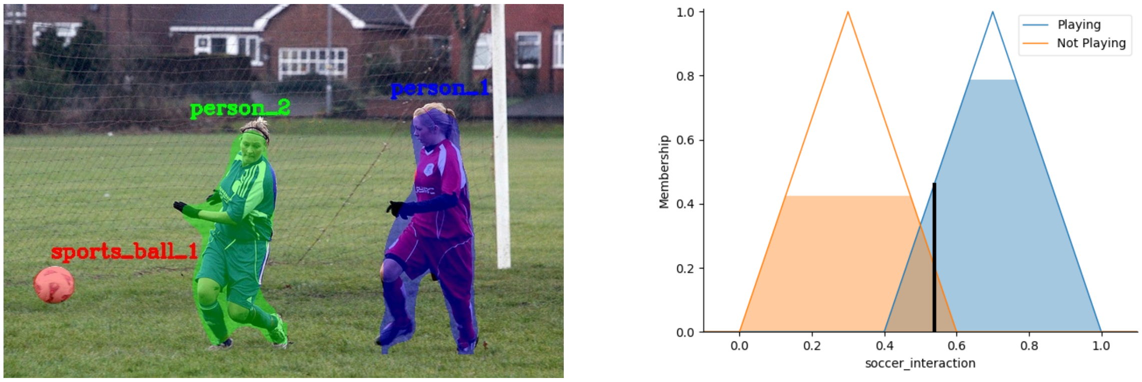
(a)

(b)

Figure 5.7

Person beside motorcycle FIS output

(a) Mask localization output (b) FIS output



(a) (b)

Figure 5.8

Person playing soccer FIS output

(a) Mask localization output (b) FIS output

5.4.3 Person-Domain FIS Results

Table 5.13

Person-Domain Level Two Summaries for Figure 3.3

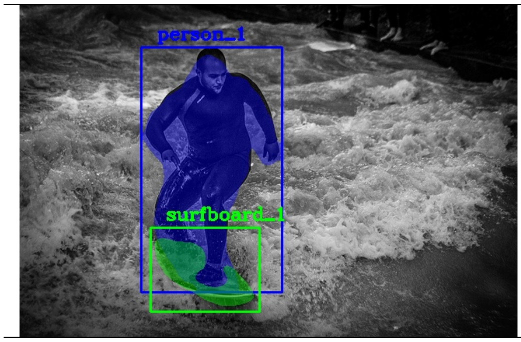
Localization Result	Person Level Two Summaries
	• person_1 riding surfboard_1

Table 5.14

Person-Domain Level Two Summaries for Figure 3.4

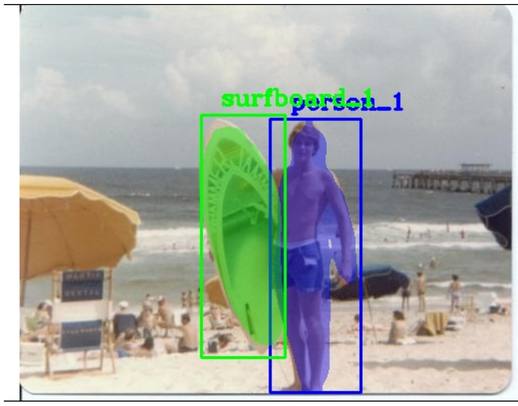
Localization Result	Person Level Two Summaries
	• person_1 overlaps and is below and to the right of surfboard_1

Table 5.15

Person-Domain Level Two Summaries for Figure 3.5

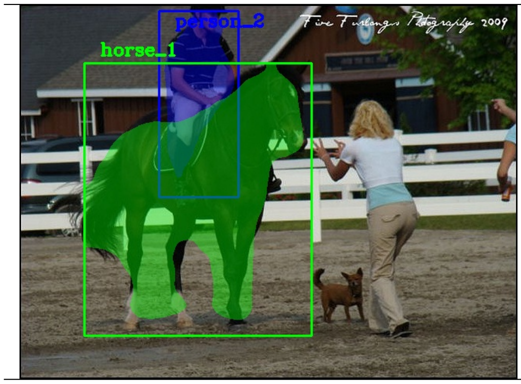
Localization Result	Person Level Two Summaries
	• person_2 riding horse_1

Table 5.16

Person-Domain Level Two Summaries for Figure 3.6

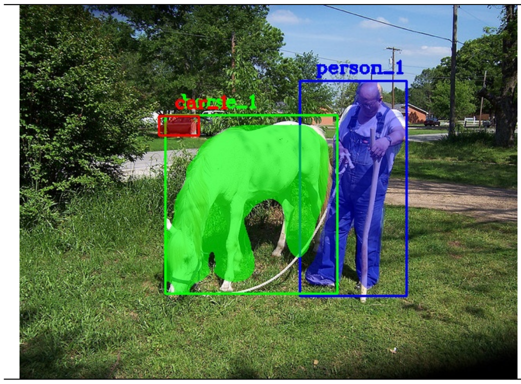
Localization Result	Person Level Two Summaries
	• person_1 overlaps and is to the right of horse_1

Table 5.17

Person-Domain Level Two Summaries for Figure 3.7

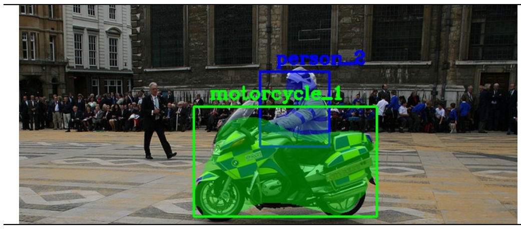
Localization Result	Person Level Two Summaries
	• person_2 riding motorcycle_1

Table 5.18

Person-Domain Level Two Summaries for Figure 3.8

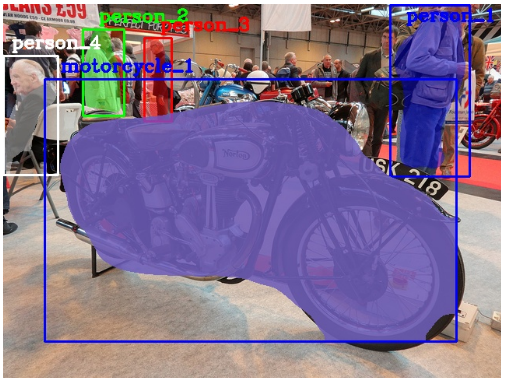
Localization Result	Person Level Two Summaries
 person_2 person_3 person_4 motorcycle_1 person_1	• person_1 overlaps and is to the right of motorcycle_1 • person_2 overlaps and is above motorcycle_1 • person_3 overlaps and is above and to the left of motorcycle_1 • person_4 overlaps and is to the left of motorcycle_1

Table 5.19

Person-Domain Level Two Summaries for Figure 3.10(b)

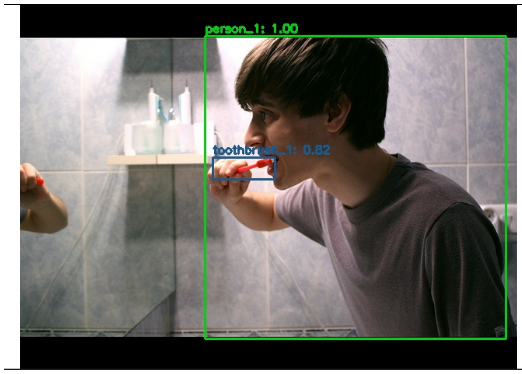
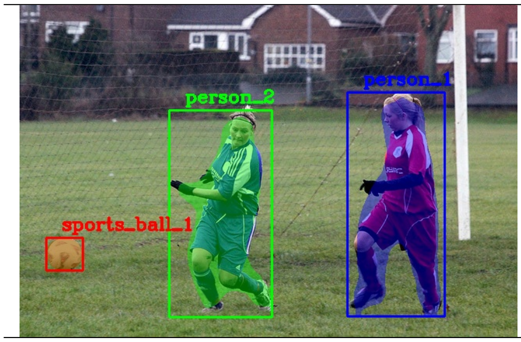
Localization Result	Person Level Two Summaries
	• person_1 brushing teeth with toothbrush_1

Table 5.20

Person-Domain Level Two Summaries for Figure 3.12

Localization Result	Person Level Two Summaries
	• person_1 playing soccer with sports_ball_1 • person_2 playing soccer with sports_ball_1

Tables 5.13 - 5.20 show the results generated by the person-domain FIS. Each table displays the localization results on the left with the corresponding person-domain Level Two Summaries on the right. Additionally, the localization results were limited to the person and the set of objects they were interacting with for these summaries. If an image contained a mask localization, the bounding box localization for that object was also displayed as the bounding box information was used as input to the GIOU [51] algorithm and thus used for constructing the fuzzy input values for proximity and overlap. The person-domain Level Two Summary labels were meant to provide very specific details about how a person(s) was interacting with objects in the scene. As mentioned previously, the negative labels that indicated a person was not performing an interaction were omitted from the results shown in Tables 5.13 - 5.20.

Also previously discussed was the ability to use Level One Summaries as the final output of the system. In order to provide a more interesting description of the scene, it was decided that in instances where the person was not performing a specific action, the Level One Summary was used to give a bit more descriptive information. As an example, consider Table 5.14, where the person was holding the surfboard. Rather than assigning the output label “person_1 not riding surfboard_1,” the Level One Summary, “person_1 overlaps and is below and to the right of surfboard_1,” was used as an output label. As can be seen in Appendix A, the rule base for the person-domain is large and would easily explode exponentially if all interactions were accounted for. While it is indeed possible to construct such systems for any number of interactions the end user is interested in, for the purposes of this research, the rule base was limited to keep the evaluations tractable in time. It is also important to note that while Table 5.10 contained a general-domain Level Two Summary for the toothbrush and sink interaction, there was no person-domain Level Two Summary computed for this image due to the lack of a person detection.

Table 5.13 shows the person-domain Level Two Summary for the original raw input image in Figure 3.3(a). The Level Two Summary of “person_1 riding surfboard_1” was an adequate reflection of what action the person was performing in the scene. Table 5.14, which was previously discussed as an example, shows the Level One Summary, “person_1 overlaps and is below and to the right of surfboard_1,” used as output of the S2T system for the original raw input image in Figure 3.4(a). Table 5.15 shows an interesting property of the person-domain Level Two results for the original raw input image in Figure 3.5(a). While the interaction of “person_2 riding horse_1” was computed correctly, the omission of “person_1 riding horse_2” was also important. Considering the general-domain Level Two Summaries shown in Table 5.5, one would expect to see the aforementioned person-domain Level Two Summary. This highlights the ability of incorporating proximity and spatial relationship information to generate accurate S2T scene descriptions. This lack of result also highlights that even if an object detection algorithm is incorrect, the system can account for incorrectness with appropriate fuzzy rule base design.

Table 5.16 was another instance where the Level One Summary was used as the final output of the S2T system. This shows the label “person_1 overlaps and is to the right of horse_1” and was a valid summation of the person’s interaction with the horse in the scene for the original input image in Figure 3.6(a). Table 5.17 shows the person-domain Level Two Summary “person_2 riding motorcycle_1,” for the raw input image shown in Figure 3.7(a), which was a correct and accurate description of not only what actions were being performed in the scene, but also of who was performing the action. Table 5.18 shows the robustness of the system when dealing with tricky images. This can also be attributed to the constructed fuzzy rule base for person - vehicle interactions, where the rules were robust enough to encode that even though some of the people are above and overlapping the motorcycle, the degree of overlap and proximity to the motorcycle, as well as the cardinal direction of the person in relation to the motorcycle were not sufficient enough to warrant the “riding” interaction. Table 5.18 shows the Level One Summaries as output for each person in relation to the motorcycle for the original input image in Figure 3.8(a). Table 5.19 shows the person-domain Level Two Summary: “person_1 brushing teeth with toothbrush_1” which was an accurate reflection of the action being performed by the person in the scene for the original input image in Figure 3.10(a). Table 5.19 was also an instance where no mask detections were available but the S2T system was still able to construct valid labels when using the bounding box localization results for a spatial relationship computation.

Table 5.20 was a very interesting result as it showed a number of unique properties of the S2T system. As discussed in Section 5.3.3, it was not necessary for two objects to be interacting in the general-domain in order to have an accurate person-domain Level Two Summary. Furthermore, Table 5.20 shows the ability to incorporate metadata, specifically the Inception [61] metadata computed in Section 3.4, when constructing the person-domain Level Two Summaries. This meta data was ingrained in the fuzzy rule base for sports such that any time a “sports_ball” object detection result was input to the person-domain FIS, the Inception metadata was passed to the person - sports category FIS to make a determination of the most likely sport being played in the scene. Namely, soccer, baseball, and tennis used the Inception meta data as they were all instances of a category where a “sports_ball” detection could occur in the COCO [35] data set. The person-domain Level Two Summary for Table 5.20 shows that both “person_1 and person_2” were “playing soccer with sports_ball_1” as a result for the raw input image in Figure 3.12.

The following section describes the validation procedure used for person-domain Level Two Summaries. Additionally, the metrics used and the results of inference using the person-domain Level Two FIS are presented.

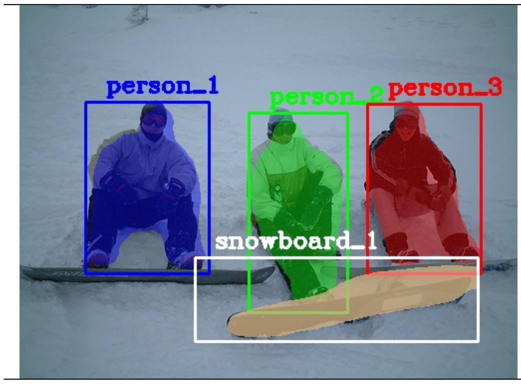
5.4.4 Person-Domain FIS Validation

A process similar to the validation process of Section 5.3.4 for general-domain Level Two Summaries was used to validate the person-domain Level Two Summaries. As with the general-domain summaries, there was no ground truth baseline data for the person-domain Level Two Summaries with regards to the COCO [35] data set. Also, the FIS used to generate the person-domain Level Two Summaries was based upon a rule base that was subjective to the authors of this research, who were acting as the domain experts required to construct a S2T rule base for the person-domain. With these factors in mind, a manual validation strategy was used to verify that the labels generated by the person-domain Level Two FIS adequately described the person - object interactions occurring in the scene.

A Python [45] script was written that first loaded the person-domain Level Two Summaries and object localization results into Pandas [62] data frames. For each person - object tuple in an image, the OpenCV [5] library was used to display the object localization results, identical to the methods in Sections 3.2.1.3 and 3.3.1.3. Although the results presented were limited to only those which showed a person performing an interaction with an object, all of the person-domain Level Two Summaries were presented on the Python console for inspection during validation. Each person-domain Level Two Summary was validated individually for each image. Once visualized, the user was presented with a prompt to "accept" or "reject" the person-domain Level Two Summary. This process was performed for each of the person-domain Level Two Summaries computed for the 506 input images deemed as usable as input to Level Two Summaries, which was discussed in Section 4.8.1. As mentioned previously, a pruning process was also used to limit the Level One Summaries used for Level Two inference to those that contained at least one person - object interaction. Thus, even though the person-domain Level Two summary FIS were applied to each of the 506 input images, there were images that did not have any person - object interactions and as such, no person-domain Level Two Summaries.

Table 5.21

Incorrect Level Two Summary

Localization Result	Person Level Two Summaries
	• person_1 riding snowboard_1 • person_2 riding snowboard_1 • person_3 riding snowboard_1

Section 5.6 provides further insight into the accuracy of the person-domain Level Two Summaries along with detailed statistics. Table 5.26 is discussed in detail in Section 5.6, but for the purposes of validation, the table shows that the system adequately performed the task of computing Level Two Summaries in the person-domain. The only person-domain Level Two Summary that was determined to be invalid during the validation process is shown in Table 5.21. As can be seen in the localization result visualization, this image poses a rather tricky case for a S2T system to process in terms of determining which person was riding which snowboard.

General-domain Level Two Summaries were subject to the problem of invalid labels due to missing depth. Person-domain Level Two Summaries were also subject to this problem, in addition to the problem of occlusion and the bad localization result problem. These three known limitations are the focus of the following section.

5.5 Known Limitations

Table 5.22

Incorrect Level Two Summary - Missing Depth

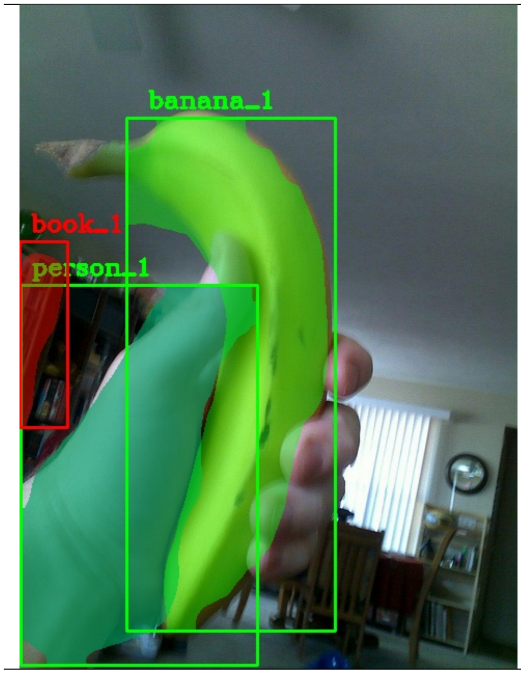
Localization Result	Person Level Two Summaries
	• person_1 eating banana_1 • person_1 reading book_1

Table 5.23

Incorrect Level Two Summary - Occlusion

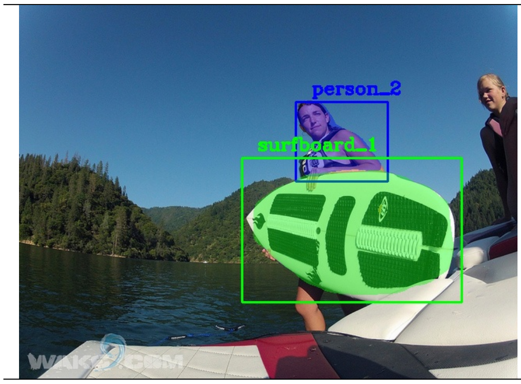
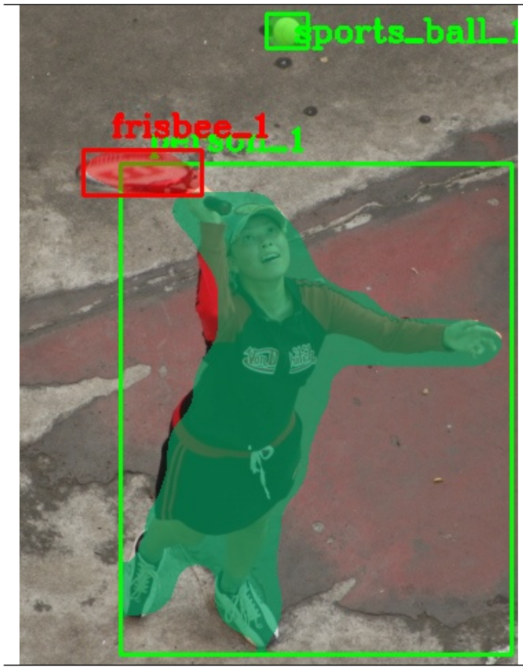
Localization Result	Person Level Two Summaries
	• person_2 riding surfboard_1

Table 5.24

Incorrect Level Two Summary - Bad Localization

Localization Result	Person Domain Level Two Summaries
	• person_1 playing tennis with sports_ball_1 • person_1 throwing frisbee_1

This section details the known limitations of the constructed S2T system. These limitations were briefly mentioned in Sections 5.3.4 and 5.4.4 and consisted of missing depth, occlusion, and bad localization results.

The missing depth problem is highlighted in Table 5.22. As is shown in the localization results, the "person_1 eating banana_1" label was accurate, but the "person_1 reading book_1" label was not due to the book being some distance away from the person. It is easy for a human eye to infer depth information, and doing so lets us come to the logical conclusion that the book was not at the same depth as the person in the image. However, it was not possible for the underlying bounding box [37] [48] and mask [21] localization algorithms used in the S2T system to make such an inference in strictly a two-dimensional space. Furthermore, the localization results showed that the person and book bounding box localization results overlapped and were in close proximity in the two-dimensional space, coupled with the mask localization results that showed the person was to the right of the book in the two-dimensional space. These fuzzy inputs would have triggered to fire for a person reading a book (shown in Appendix A) that required the person to be overlapping and in close proximity and not below the book in order to be considered reading. The missing depth problems account for the majority of inaccurate labels for both the general- and person-domain when constructing Level Two Summaries, as shown in Tables 5.25 and 5.26 respectively.

The easiest way to remedy the depth problem would be to make use of algorithms that compute object localization results in three dimensions, such as some of the methods discussed by Wang in [68]. However, these techniques all rely on point cloud information, which was not readily available for standard object detection data sets like COCO [35]. Another possible remedy for the depth problem would be to use methods that attempt to infer three-dimensional objects given two-dimensional image object localization results, such as the system proposed by Huang in [25]. At the time of performing the object localization evaluation, none of the proposed methods were readily available, and these methods still require supervised training of a model that will perform the three-dimensional projection, a process for which training data is not readily available.

An additional way to remedy the depth problem would be to use algorithms that incorporate three-dimensional information into spatial relationship computations. Reed et al. [49] proposed an implementation of HOF [40] that computes force histograms in three-dimensional space. Similarly, Kaur et al. [29] propose an implementation of HOF that computes force histograms in three-dimensional space using fast Fourier transforms. The incorporation of these algorithms could serve to alleviate issues that arise from the depth problem in future S2T system designs. At the time of writing, these implementations weren't readily available for open source use. Additionally, to use the algorithms for computing overlap and proximity, an alternative to the GIOU [51] algorithm would be required to be developed that works with the three-dimensional force histogram computations. The original implementation of HOF [40] was chosen for this research because it allowed easy separation of the proximity/overlap computations from the spatial relationship computations. This was a design choice for the constructed S2T system to show a proof of concept, and it is important to note there is no right or wrong answer when choosing the underlying methods used.

The occlusion problem is highlighted in Table 5.23. As shown in the localization results, the mask and bounding box location reflect that the person was indeed above and overlapping the surfboard, hence the "person_2 riding surfboard_1" label. Again, it is easy for a human to infer that the entirety of the person is not above the surfboard, but this is not an easy task for a machine. This image in particular shows that the surfboard is occluding the majority of the lower torso of the person, and as such, the object detection algorithm only detected the upper torso of the person, leading to the inaccurate label. General-domain Level Two Summaries were not subjected to the problem of occlusion, as an argument object either was or was not interacting with a referrant object. Thus, the general-category Level Two Summary of "person_2 interacting with surfboard_1" was still valid. The problem of occlusion, then, only arises when computing person-domain Level Two Summaries.

One potential solution to this problem would be the improvement of object detection algorithms that could accurately compute objects that were segmented due to occlusion. At the time of writing, no such object detection algorithms exist and the improvement is beyond the scope of this research. Table 5.26 shows that the occlusion problem accounts for four invalid person-domain Level Two Summaries.

The bad localization result problem is highlighted in Table 5.24. The object localization results show that the tennis racket was incorrectly predicted as “frisbee_1”. This invalid detection in effect trickles down to the person-domain Level Two Summary of “person_1 throwing frisbee_1.” We as humans know this object localization was incorrect, but the machine had difficulty distinguishing between a tennis racket and frisbee disk in this particular instance. Also shown in the localization results and person-domain Level Two Summaries were that the person was playing tennis with “sports_ball_1” which was a valid detection.

In this particular instance, the argument can be made that since the sports ball was inferred to be a tennis ball, using the Inception [61] meta data, the system could be designed to infer that object localization results were likely incorrect. While that is true for instances where a person was interacting with more than one object, inferring incorrect object localization results was not possible when there existed only one object in the scene the person was interacting with. The simplest way to alleviate the bad localization problem is to improve the underlying object detection algorithms used. While a near-perfect object detection algorithm would surely alleviate the bad detection problem, improvement of object detection algorithms is beyond the scope of this research.

General-domain Level Two Summaries were not subject to the bad localization result problem for the COCO [35] data set used in construction of the S2T system. As with the occlusion problem, bad localization results were not considered for the general-domain because an argument object either was or was not interacting with a referrant object. Table 5.26 shows that 16 person-domain

Level Two Summaries were deemed inaccurate due to the bad localization result problem. The following section discusses the results and computed statistics of the general-domain Level Two Summaries and person-domain Level Two Summaries in extensive detail.

5.6 Discussion

This section first presents an in-depth analysis of the results obtained from generating the general-domain Level Two Summaries, followed by an in-depth analysis of the results of computing the person-domain Level Two Summaries. This chapter concludes with a summary of the Level Two Summary evaluation, where the second research question and hypothesis posed in Chapter 1 are revisited.

5.6.1 General-Domain Level Two Summaries

Table 5.25

General-Domain Level Two Summary Results

Total Images	506
Total General-Domain Level Two Summaries	3833
Accurate General-Domain Level Two Summaries	3779
Incorrect General-Domain Level Two Summaries	54
Incorrect - Depth Problem	54
Incorrect - Occlusion Problem	0
Incorrect - Bad Localization Result	0
Incorrect	0

Section 5.3.2 provided the implementation details for the general-domain Level Two Summary FIS. This FIS completes the last component of the S2T system for the general-domain. Section 5.3.3 presented the general-domain Level Two Summaries, used as output for the S2T system in

the general domain, for the original input images first presented in Chapter 3. A manual validation process performed on the general-domain Level Two Summary implementation was presented in Section 5.3.4 and the remainder of this subsection is devoted to analyzing the results of the validation.

Table 5.25 shows that the general-domain Level Two Summaries were computed for the 506 input images with Level One Summaries deemed valid for use with Level Two Summary computation. The metrics used to consider a Level One Summary as valid input for Level Two Summaries was discussed in Section 4.8.1. Table 5.25 shows that for the 506 images, there were 3833 object tuple Level One Summaries used as input to the general-domain Level Two Summary computation. As mentioned previously, there was no pruning process for the general-domain as there was for the person-domain, thus the general-domain Level Two Summaries were computed for every object tuple in an image.

Table 5.25 shows that of the 3833 computed general-domain Level Two Summaries, 3779 of the summaries were accurate. The table also shows that of the 54 inaccurate results, all 54 of the incorrect results were attributed to the missing depth problem. The missing depth problem is discussed in extensive detail in Section 5.5. Table 5.25 shows that the general domain summaries provided an accuracy of 99% for the COCO [35] data set. It is important to note that "correctness" of the system was a subjective measure, as the general-domain Level Two Summaries themselves were inherently subjective, and was measured by the author's agreement with the label generated by the S2T system. The results presented in Section 5.3.3 provide evidence to support that the general-domain Level Two FIS evaluation satisfied the second research question and associated hypothesis posed in Chapter1:

RQ2: How can the Level One Summaries provide meaningful information to the creation of informative, high-level, natural language descriptions (Level Two Summaries) of the object interactions in a scene?

H2: By utilizing a fuzzy inferencing approach in the generation of scene labels, the uncertainty in spatial relationship reasoning can be effectively modeled and incorporated into scene descriptions generated by the S2T system.

5.6.2 Person-Domain Level Two Summaries

Table 5.26

Person-Domain Level Two Summary Results

Total Images	506
Images with Person - Object Interactions	335
Total Person - Object Interactions	602
Correct Level Two Summaries	559
Incorrect - Depth Problem	22
Incorrect - Occlusion Problem	4
Incorrect - Bad Localization Result	16
Incorrect	1

Section 5.4.2 provided the implementation details for the person-domain Level Two Summary FIS. Construction of the person-domain Level Two Summary FIS was the last component of the S2T system for the person-domain. Section 5.4.3 shows the results used as S2T system output in the person-domain for the original raw input images first presented in Chapter 3. Section 5.4.4 detailed the manual validation strategy used to validate the Level Two Summaries computed in the person-domain. The remainder of this subsection is devoted to an analysis of the metrics obtained by applying the person-domain Level Two Summary FIS to the COCO [35] data set.

Table 5.26 shows that of the 506 input images with Level Two usable Level One Summaries, 335 images contained person - object interactions. From the 335 images, there were 602 person - object Level One Summaries for which a person-domain Level Two Summary was computed. Of the 602 computed person-domain Level Two Summaries, Table 5.26 shows that 559 of these were deemed accurate. All but one of the 43 inaccurate results fell into one of three categories: depth problem, occlusion problem, and bad localization problem. These problems were discussed extensively in Section 5.5. Table 5.26 also shows that of the 43 inaccurate results, 22 were due to the depth limitation problem, 4 were due to the occlusion problem, and 16 were due to the bad localization result problem.

In total, there was only one result from the person-domain Level Two Summaries that was not a result of any of the problems discussed. Not withholding the missing depth, occlusion, and bad detection inaccurate results, the system performed with an accuracy of 93%. When withholding the inaccurate Level Two Summaries due to known limitations, the system performed with an accuracy of 99%. It is important to point out once again, that "correctness" was a subjective measure for this system because the Level Two Summaries themselves were subjective, and correctness was primarily a function of the system designer agreeing with the output Level Two Summary for a person - object tuple. Based on the accuracy of the S2T system and the results presented in Section 5.4.3, the person-domain Level Two Summary FIS evaluation provided positive evidence for the second research question and supported the associated hypothesis posed in Chapter 1 for the person domain:

RQ2: How can the Level One Summaries provide meaningful information to the task of creating informative, high-level, natural language descriptions (Level Two Summaries) of the object interactions in a scene?

H2: By taking a fuzzy inferencing approach in generating scene labels, the uncertainty in spatial relationship reasoning can be effectively modeled and incorporated into scene descriptions generated by the S2T system.

5.7 Potential System Usage

This section briefly presents one potential usage of the S2T system developed by this research. At the time of writing, deep neural networks were the most widely used machine learning technique in the field. Training these networks is not only expensive in terms of resources and time consumption, but training also requires large amounts of data to adequately train the larger models that are popular. To date, the data sets that exist that do incorporate scene labels contain labels that are required to be manually curated. These labels also do not incorporate spatial relationship information. The S2T system constructed by this research could potentially be used as an automated process to generate scene description labels that incorporate spatial relationship information between each of the object tuples in an image.

During evaluation, one idea was that the final S2T system constructed in this research could be used to apply S2T labels to an image data set to use for training a deep neural network. The S2T system could be used alongside any existing object detection algorithm to process the object localization results. Doing this would provide the following information for each object localization tuple in an image. The Level One Summaries for each object tuple would provide the overlap and spatial relationship information for each object tuple in an image. The general-domain Level Two Summaries would provide information about which objects were or were not interacting for each object tuple in an image. The person-domain Level Two Summaries would provide refined information about the person - object interactions in a scene. Combining all of this information with the object localization results would provide much more informative information in terms of training a large scale deep learning algorithm. In theory, the constructed S2T system can provide all of this information with minimal effort, as the system is agnostic to the localization algorithm used. With this in mind, the S2T system developed would be able to generate training labels alongside the object localization results by simply attaching the S2T system to the output of the object localization algorithm. Chapter 6 revisits this topic as the sensitivity analysis validation performed effectively acts as a proof of concept for applying the S2T system to an unlabeled data set to generate informative labels.

CHAPTER 6

SYSTEM SENSITIVITY ANALYSIS VALIDATION

This chapter presents a sensitivity analysis validation that was performed in order to ensure that the S2T system constructed via the methods of Chapters 3 - 5 could generalize well to other data sets. Unlike the evaluations in Chapters 3 - 5 where each evaluation and the corresponding results were generated individually for the purposes of validating each stage of the S2T system, the validations in this chapter were all performed sequentially, where the result of each stage of the S2T system becomes the input of the stage that follows.

First, this chapter discusses the data set and which images were used for the sensitivity analysis validation. The validation design and setup is discussed after the data set discussion, followed by a discussion of the methods used to perform the sensitivity analysis validation. Lastly, the results obtained from the validation are discussed in-depth and the remaining research questions and hypotheses from Chapter 1 are revisited.

6.1 Data Set

This validation made use of the 2017 version of the COCO [35] data set. The 2017 data set is an updated version of the 2014 COCO data set. The COCO 2014 data set was used to construct the S2T pipeline in Chapters 3 - 5. The COCO 2017 data set contains images not available in the 2014 data set, and as such the images in the 2017 data set were not utilized in development of the system.

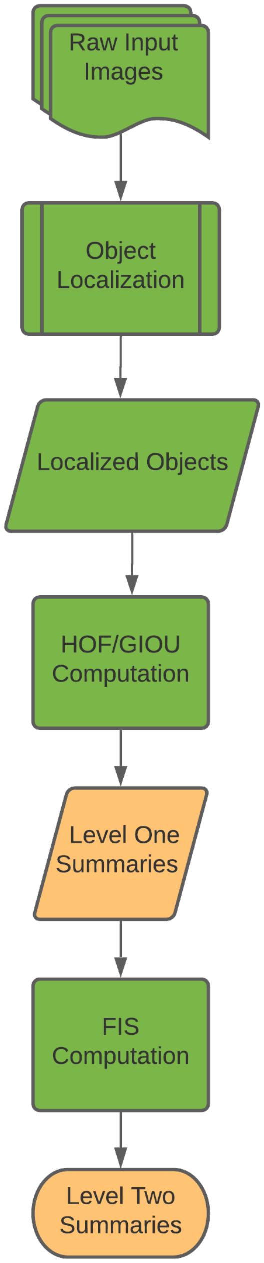


Figure 6.1

Completed S2T Pipeline

In other words, the COCO 14 data set can be viewed as the training set, and the COCO 17 data set can be viewed as the validation/testing data set. The FiftyOne [8] library was used in a Python [45] script to download the COCO 17 data set. During downloading, the FiftyOne library allowed filtering of data set images to only download specific object categories. As the S2T system was developed to generate Level Two Summaries in the person-domain, image downloads were filtered to only those which likely contained a person detection. For brevity, the download was limited to 2000 images from the COCO 17 data set.

The FiftyOne [8] library also allowed the download of segmentation results and labels for the COCO 17 data set, but for completeness of the validation, a stand alone object detection algorithm was used to apply the full S2T pipeline developed by the research of this dissertation, shown in Figure 6.1. The following section describes the validation design that processed the COCO 17 [35] images through each stage of the S2T pipeline in detail.

6.2 Validation Design

This section discusses how each stage of the S2T pipeline of Figure 6.1 was applied to the COCO 17 [35] data set. When developing the S2T pipeline, each phase was separated into individual validations that mimicked the diagram in Figure 5.1. For this validation, a single “driver” script was used to perform all of the stages of the pipeline sequentially. The remainder of this section discusses how each stage of the S2T pipeline was implemented for this validation.

6.2.1 Object Localization

Figure 6.1 shows that object localization was the first step performed on the set of raw input images. For this validation, the YOLOv3 [48] object detection model was used to compute the object localization results for the subset of 2000 images for the COCO 17 [35] data set. YOLOv3 was chosen for a few reasons. First, the GIOU [51] computations require only bounding boxes, and the HOF [40] computations work on either bounding boxes or masks. YOLOv3 provided the necessary input for both of the algorithms used to compute the Level One and Level Two Summaries of Sections 6.2.2 and 6.2.3 respectively. Secondly, the entirety of the YOLOv3 object localization results can be stored in CSV or JSON files. This allowed for easily sharing the code and results where the object localizations for an image did not have to be recomputed. Mask localization results, such as those computed by Mask R-CNN [21], were not chosen because in order to share the results, the binary mask images, which were quite large, would also have to be shared. Lastly, the YOLOv3 library demonstrated better object localization results compared to the SSD [37] library during the evaluations of Chapter 3.

To begin, the OpenCV [5] library was used to load the YOLOv3 [48] model from disk. Each of the 2000 raw input images from the COCO 17 [35] data set was then processed through the YOLOv3 model to generate potential object localization results with corresponding confidence scores. Each potential result was then subjected to non-maxima suppression (NMS) [24] thresholding, where the threshold value was chosen as 0.4. NMS thresholding served to eliminate multiple overlapping object localization results. The NMS threshold value was chosen identical to the methods discussed in Section 3.3.1.2. After NMS thresholding, the object localization results were stored in a Pandas [62] data frame that contained the following information for each image:

- relative path of the image
- bounding box localization results for each object
- class label for each localization result
- number of object localization results

Table 6.1

Object Localization Results

Original Input Images	2000
Images with two or more object localization results	1637
Images with at least one person detection	1243

The candidate object localization results were then further refined by first eliminating any image where there were not two or more objects detected in the scene. This refinement was performed because both Level One and Level Two Summaries required at least one object tuple interaction. Table 6.1 shows that of the 2000 original raw input images, 1637 images contained at least two object localization bounding boxes. The second refinement performed was to remove any images that did not contain a "person" localization result. While Level One Summaries and general-domain Level Two Summaries can both operate on any object tuple in an image, the person-domain Level Two Summaries require a person - object interaction. As such, this additional refinement was performed such that each raw input image contained object localization results that could be processed by all stages of the S2T pipeline shown in Figure 6.1. Table 6.1 shows that of the 1637 images that contained at least two object localization results, 1243 of those images contained a "person" object detection. The entire set of localization results was stored as a Pandas [62] data frame and the data frame was written to disk for inspection if so desired.

Each of the 1243 images with at least two object localization results, one of which being a "person" localization, was then processed using the Inception [61] model in order to generate meta data for each image. Chapter 5 showed that meta data was necessary in some instances in order to allow the S2T system to determine the appropriate fuzzy rule base to apply given the object tuple.

The PyTorch [44] library was used to load the Inception model and also used to process each of the 1243 images. For each image, the Inception model output a list of class labels with corresponding confidence scores. Each of these class labels described the overall scene, as opposed to object segmentations. The class labels were sorted according to their respective confidence scores, and the top five labels were retained to be used as meta data for input to the Level Two Summaries computed in Section 6.2.3. The meta data generated by the Inception model for the entire image set was stored as a Pandas [62] data frame and the data frame was written to disk such that it could be loaded and inspected if the user desired to do so.

The object localization results for the 1243 images with a "person" object localization result were used as input to the Level One Summary computation discussed next in Section 6.2.2. These same object localization results, along with the meta data generated by the Inception [61] model for each image, were used as input to the Level Two Summary computation discussed in Section 6.2.3.

6.2.2 Level One Summaries

Figure 6.1 shows that the second phase of the S2T pipeline is the generation of Level One Summaries. To recap, the Level One Summaries provide spatial relationship information about each object tuple in an image. As mentioned previously, the 1243 images that contained more than one object localization result, one of which was a "person" detection, were used as input to the Level One Summary phase of the S2T pipeline. Similar to the evaluations of Chapter 4, the GIOU [51] and HOF [40] algorithms were used to construct the Level One Summaries for the data set.

The object localization results were loaded from the Pandas [62] data frame constructed in Section 6.2.1. For each input image's object localization results, the object tuples were organized by first taking the two-permutations of the objects in the image. Afterwards, the tuples were ordered hierarchically where persons took precedence over animate objects and animate objects took precedence over inanimate objects. After the object tuple permutations were constructed, the GIOU [51] and HOF [40] algorithms were applied to each tuple.

The GIOU [51] algorithm was implemented using the Python [45] programming language and the Numpy [43] Python library. Section 4.4.1 discusses the GIOU algorithm implementation in detail and the same process detailed in that section was used to implement the GIOU algorithm for this validation. The result of the GIOU algorithm were two values: intersection over union (IOU) and generalized intersection over union (GIOU), both of which were floating point values. The IOU value was used to compute the "overlap" relationship, whereas the GIOU value was used to compute the "proximity" relationship. Both the IOU and GIOU numerical values were used as input to the Level Two Summary computation detailed in Section 6.2.3. The GIOU value was not used for constructing Level One Summaries as proximity information was not incorporated into the Level One Summaries. The IOU value was used to compute the "overlap" relationship and that relationship was incorporated into the Level One Summaries.

The overlap relationship was represented as a fuzzy variable, using the fuzzy membership function shown in 4.2. The IOU value was used as the fuzzy input value and the centroid [60] defuzzification method was used to generate the crisp output value. The crisp output value was one of "Overlap" or "No Overlap" for each object tuple in the image's localization results. The membership functions and defuzzification methods were all constructed using the SKFuzzy [69] library within a Python [45] script.

The HOF [40] algorithm implementation made use of a combination of Matlab [39] and Python [45] scripts. The underlying code for the HOF algorithm was generated by the authors of the original HOF work, and was graciously provided for use within this research. The HOF algorithm implementation is detailed extensively in Section 4.5.1 and a similar implementation process was used for the purposes of this validation. After applying the GIOU [51] algorithm to an object tuple, the HOF algorithm was applied to generate the spatial relationship value for the tuple. This value fell in the range $0 - 360$ where each value represented an angle. There were three types of HOF values computed: constant force (F0), gravitational force (F2), and hybrid force (FH) histograms. Each of these values were stored for the object tuple for use when constructing Level One and Level Two Summaries.

The spatial relationship value was used to compute the cardinal direction in which the argument object was with respect to the referrant object. To achieve this, the consensus value of the three force histogram outputs was chosen, shown in Algorithm 4.3. The spatial relationships were represented as a fuzzy membership function, shown in Figure 4.4, and the consensus angle was used as the fuzzy input value to compute the crisp output spatial relationship label. Centroid [60] defuzzification was used to determine the crisp output spatial relationship label given the fuzzy HOF [40] consensus value as input. As can be seen in Table 4.1, the crisp output for spatial relationships fell into one of nine cardinal directions. All code used to develop the fuzzy membership functions and to perform defuzzification was developed using the SKFuzzy [69] library inside of a Python [45] script. The final result of centroid defuzzification was an output label that represented the direction of the argument object with respect to the referrant object.

After applying the GIOU [51] and HOF [40] algorithms, identical to the methods shown in Algorithms 4.2 and 4.3 respectively, the numerical values for GIOU, IOU, and the three force histogram types were stored in a Pandas [62] data frame. Unlike the object localization results that used one row per image, the Level One Summary data frame contained a row for each object tuple. After these values were generated for an object tuple, the centroid [60] defuzzification method was applied to the IOU and consensus force histogram to generate the "overlap" and "spatial relationship" labels respectively. Using these labels, the Level One Summary for the object tuple was constructed and stored as an additional field in the Pandas row. Examples of the Level One Summaries computed during this validation are provided in Section 6.2.2.1. The resulting Pandas data frame after constructing the Level One Summaries for each object tuple contained the following information for each object tuple in an image:

- result key, represented as the image path plus the argument and referrant object
- the relative path of the image
- the class label of the argument object
- the bounding box that corresponds to the argument object
- the class label of the referrant object
- the bounding box that corresponds to the referrant object
- the IOU numerical value used to compute the overlap relationship
- the GIOU numerical value used to compute the proximity relationship
- the F0, F2, and FH force histogram values used to compute spatial relationships
- the Level One Summary for the object tuple

Table 6.2

Total Level One Summaries

Object Localization Result Images	1243
Level One Summaries	10318
Level One Summaries that contain a “person”	8466

Table 6.2 shows that for the 1243 images with at least two object localization bounding boxes for which one was a “person” detection, there were a total of 10,318 Level One Summaries computed. In other words, there were 10,318 object tuples the GIOU [51] and HOF [40] algorithms processed for the 1243 images. Each of these Level One Summary results could be used to generate a Level Two Summary in the general-domain. However, only Level One Summary results that contain a “person” detection were processed for the person-domain Level Two Summaries. Table 6.2 shows that 8466 Level One Summaries contained a person - object interaction and as such only 8466 person-domain Level Two Summaries were computed. Both the general-domain and person-domain Level Two Summary computations are discussed in detail in Section 6.2.3. In the following section, results for the Level One Summary computation are presented to the user and discussed.

6.2.2.1 Results

Table 6.3

“Umbrella” Level One Summary Results

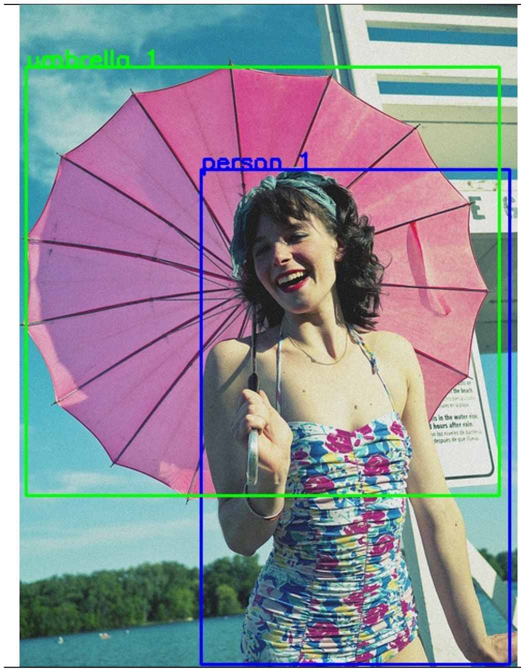
Localization Result	Level One Summaries
	• person_1 overlaps and is below and to the right of umbrella_1

Table 6.4

“Cake” Level One Summary Results

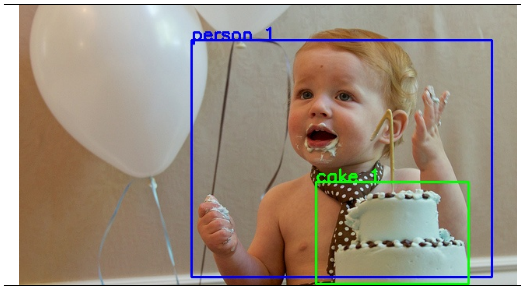
Localization Result	Level One Summaries
	• person_1 overlaps and is above cake_1

Table 6.5

"Broccoli" Level One Summary Results

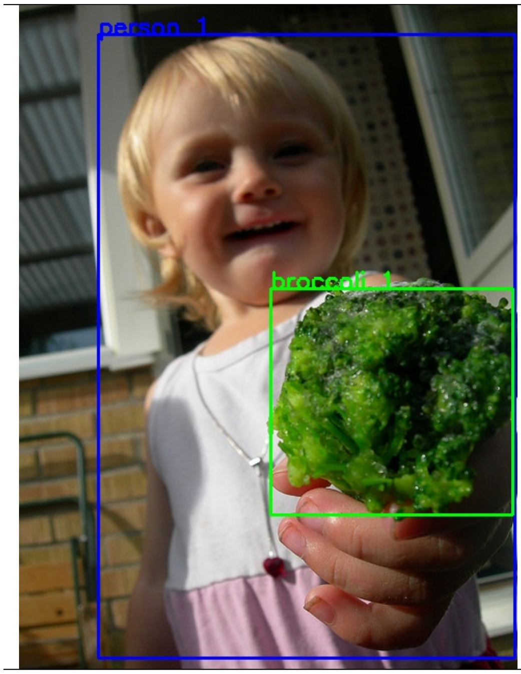
Localization Result	Level One Summaries
	• person_1 overlaps and is above and to the left of broccoli_1

Table 6.6

"Bench" Level One Summary Results

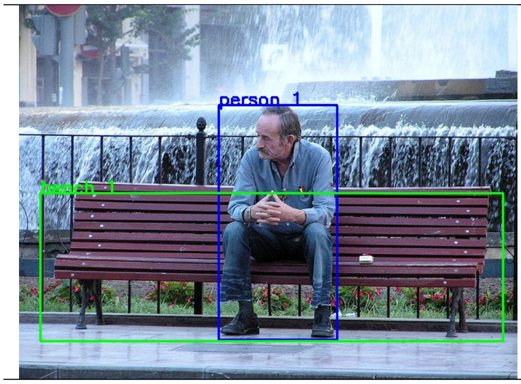
Localization Result	Level One Summaries
	• person_1 overlaps and is above and to the right of bench_1

Table 6.7

"Tie" Level One Summary Results

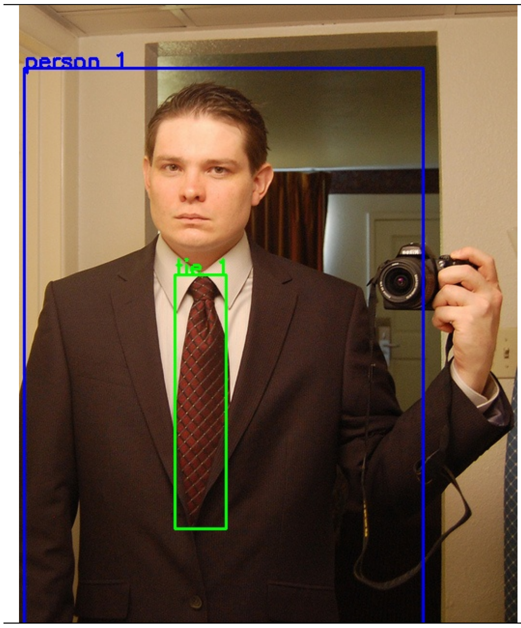
Localization Result	Level One Summaries
	- person_1 overlaps and is above and to the right of tie_1

Table 6.8

"Dog" Level One Summary Results

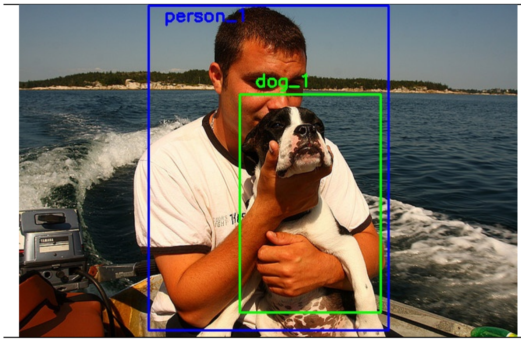
Localization Result	Level One Summaries
	- person_1 overlaps and is above dog_1

Table 6.9

“Cell Phone” Level One Summary Results

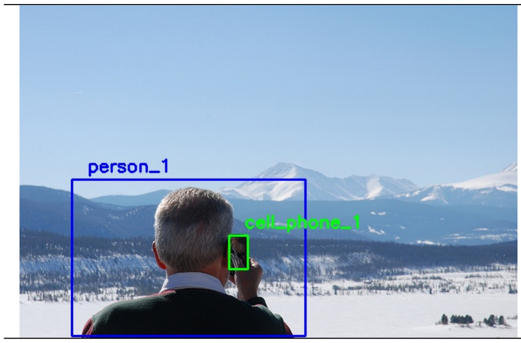
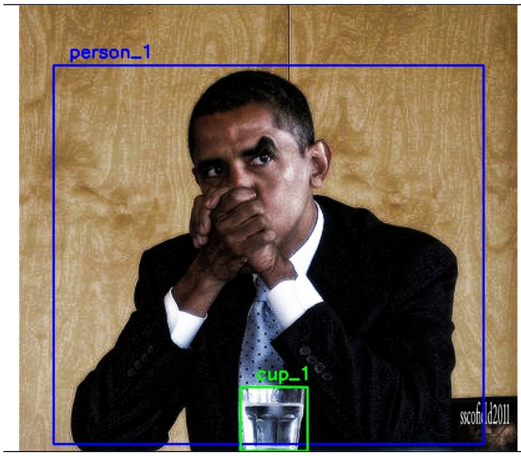
Localization Result	Level One Summaries
	• person_1 overlaps and is to the left of cell_phone_1

Table 6.10

“Cup” Level One Summary Results

Localization Result	Level One Summaries
	• person_1 overlaps and is above cup_1

Tables 6.3 - 6.10 shows examples of Level One Summary results for the COCO 17 [35] data set. In each table, the image’s object localization results are shown on the left with the corresponding Level One Summary shown on the right. Each of these results were chosen because each image has

a person - object interaction and as such, both general- and person-domain Level Two Summaries can be computed for them.

Table 6.3 shows a correct Level One Summary of a person that overlaps and is below and to the right of an umbrella. Table 6.4 shows a Level One Summary for a person overlapping and above a cake. Table 6.5 shows that a person is overlapping and is above and to the left of a piece of broccoli. Table 6.6 shows that the person overlaps and is above and to the right of the bench. Each consensus angle indicated that the cardinal direction computed was in the "above right" direction, likely due to the center of mass of the detection result. Table 6.7 shows that the person overlaps and is above and to the right of the tie. Table 6.8 shows that the person is overlapping and above the dog. Table 6.9 shows that the person and cell phone overlap and the person is to the left of the cell phone. Lastly, Table 6.10 shows that the person overlaps and is above the cup.

Each of the Level One Summaries sufficiently reflect the spatial relationships between the person and the object they are interacting with in the scene. Additionally, each of these Level One Summaries show that the use of bounding box localization results for computing Level One Summaries provide less refined information as opposed to using mask localization results. For example, in Table 6.6, it can be argued that the person is simply "above" the bench and not "above and to the right," a relationship that would likely be reflected with mask localization results due to the tighter bounds. These Level One Summary results, as shown in Figure 6.1, can be used as a final output of the system if so desired. Figure 6.1 also shows that the Level One Summary results serve as input to the Level Two Summary computations. The computations of Level Two Summaries for both the general- and person-domain are discussed in the next section.

6.2.3 Level Two Summaries

Figure 6.1 shows that the final phase of the S2T pipeline is the computation of the Level Two Summaries. The general-domain Level Two Summaries provide information describing whether or not two objects are interacting. As the name suggests, they are a general description of the interactions occurring in the scene. The person-domain Level Two Summaries provide refined detail toward whether or not a person is performing a specific action with regards to an object. This validation used the Level One Summary results, computed in Section 6.2.2, as input to the general- and person-domain Level Two Summary computations. The methods developed in Chapter 5 were used for this validation to compute these summaries. Specifically, Algorithms 5.1 and 5.2 were used to compute the general-domain and person-domain Level Two Summaries respectively. The Level One Summaries were loaded from the Pandas [62] data frame constructed in Section 6.2.2. For each Level One Summary, the GIOU [51], IOU, and three HOF [40] angles were loaded from the data frame. The GIOU value was converted into a fuzzy input value for proximity using the fuzzy membership functions shown in Figure 4.3 from Chapter 4. The IOU value was converted into a fuzzy input value for the overlap relationship using the fuzzy membership functions shown in Figure 4.2 from Chapter 4. The consensus HOF angle was converted into a fuzzy input value using the fuzzy membership functions shown in Figure 4.4, also from Chapter 4. Once these values had been fuzzified, they were set as input to the general-domain and person-domain FIS in order to compute the general-domain and person-domain Level Two Summaries.

The general-domain Level Two Summaries were computed identical to the algorithm shown in Algorithm 5.1 of Chapter 5. The rule base used for the general-domain was the same rule base as is shown in Table 5.2 from Chapter 5. Construction of the rule base, membership functions, and FIS made use of the SKFuzzy [69] library and all code for this validation was written in a Python [45] script. Section 5.3.2 discusses the implementation of the general-domain FIS in further detail and an identical process was followed to construct the general-domain FIS for this validation. The result of the general-domain FIS was a label that indicated whether or not two objects were interacting. Results were computed for each Level One Summary, and as such each object tuple in the set of Level One Summaries had a corresponding general-domain Level Two Summary. The general-domain Level Two Summaries were stored in a Pandas [62] data frame that contained all information stored in the Level One Summaries with an additional field for the general-domain Level Two Summaries. Specifically, the general-domain Level Two Summary data frame contained the following information for each object tuple in an image:

- result key, represented as the image path plus the argument and referrant object
- the relative path of the image
- the class label of the argument object
- the bounding box that corresponds to the argument object
- the class label of the referrant object
- the bounding box that corresponds to the referrant object
- the IOU numerical value used to compute the overlap relationship
- the GIOU numerical value used to compute the proximity relationship
- the F0, F2, and Hybrid force histogram values used to compute spatial relationships
- the Level One Summary for the object two tuple
- the general-domain Level Two Summary for the object tuple

As mentioned previously, each Level One Summary has a corresponding Level Two Summary. Table 6.2 shows that 10,318 Level One Summaries were computed for the input data set, therefore there were 10,318 general-domain Level Two Summaries generated for the input data set. The Pandas data frame was written to a CSV file for retrieval if the user wished to manually inspect the results as well.

The person-domain Level Two Summaries were computed identical to the algorithm shown in Algorithm 5.2 in Chapter 5. The rule base used for the person-domain was the same as the rule base used for the evaluations in Section 5.4.1. The entire rule base for the person-domain Level Two Summaries is shown in Appendix A. The rule base, membership functions and person domain FIS were all implemented using the SKFuzzy [69] library and all code was written in the Python [45] programming language. The process used to construct person-domain Level Two Summaries was discussed extensively in Section 5.4.2 and an identical process was used to construct the person-domain Level Two Summaries for this validation. The output of the person-domain FIS was a label indicating a specific interaction that a person was or was not performing given a referrant object. These interactions were based on the specific category of the object in question. This category was determined based an object hierarchy and an extensive discussion of the object hierarchy used for this validation was given in Section 5.2. As with the evaluations of Chapter 5, the object category determines which rules to apply to the person - object tuple Level Two Summary computation. Person-domain Level Two Summaries required a person - object interaction, and thus any object tuple that did not contain a person localization result were not processed using the person-domain Level Two FIS. The person-domain Level Two Summaries were stored in a Pandas [62] data frame that contained the same information stored in the general-domain Level Two Summaries data frame with the exception that the person-domain Level Two Summaries are stored as opposed to the general-domain Level Two Summaries.

Table 6.2 shows that for the 10,318 Level One Summaries, 8466 of those Level One Summaries contained a person object localization. As such, person-domain Level Two Summaries were computed for those 8466 Level One Summaries. After computation, the resulting Pandas data frame was written to a CSV file for easy retrieval. The following section discusses results obtained during validation for the general- and person-domain Level Two Summaries.

6.2.3.1 Results

Table 6.11

“Umbrella” Level Two Summary Results

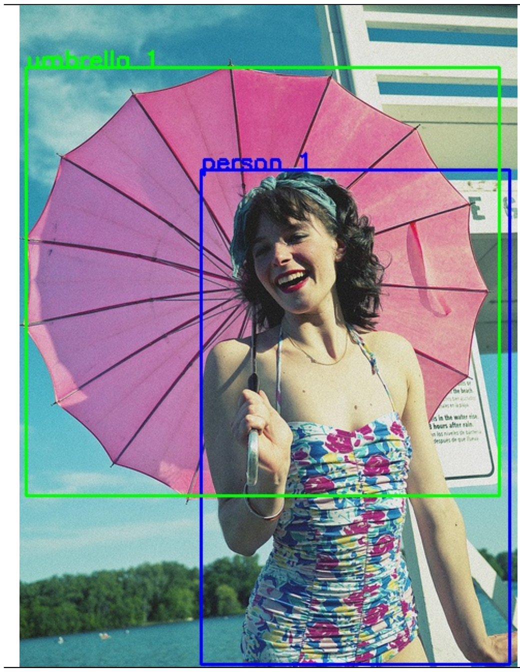
Localization Result	Level Two Summaries
	• **General**: person_1 interacting with umbrella_1 • **Person**: person_1 carrying umbrella_1

Table 6.12

“Cake” Level Two Summary Results

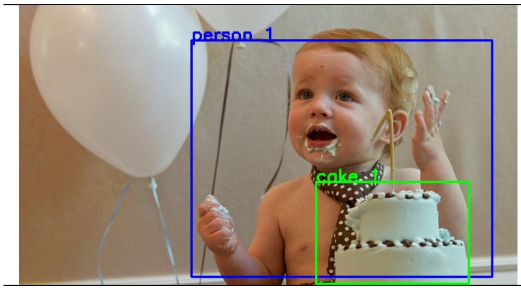
Localization Result	Level Two Summaries
	• **General**: person_1 interacting with cake_1 • **Person**: person_1 eating cake_1

Table 6.13

“Broccoli” Level Two Summary Results

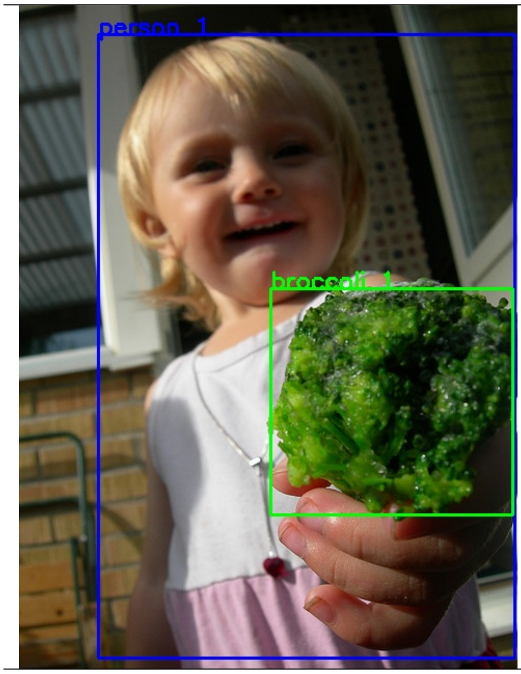
Localization Result	Level Two Summaries
	• **General**: person_1 interacting with broccoli_1 • **Person**: person_1 eating broccoli_1

Table 6.14

"Bench" Level Two Summary Results

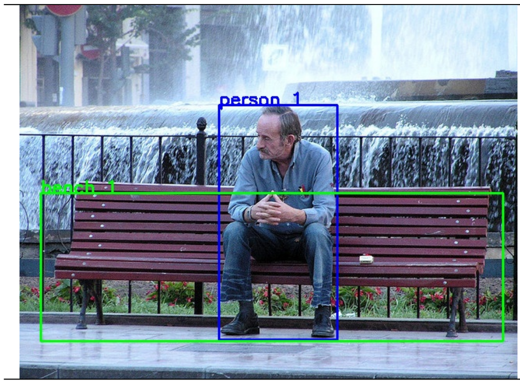
Localization Result	Level Two Summaries
	• **General**: person_1 interacting with bench_1 • **Person**: person_1 sitting on bench_1

Table 6.15

"Tie" Level Two Summary Results

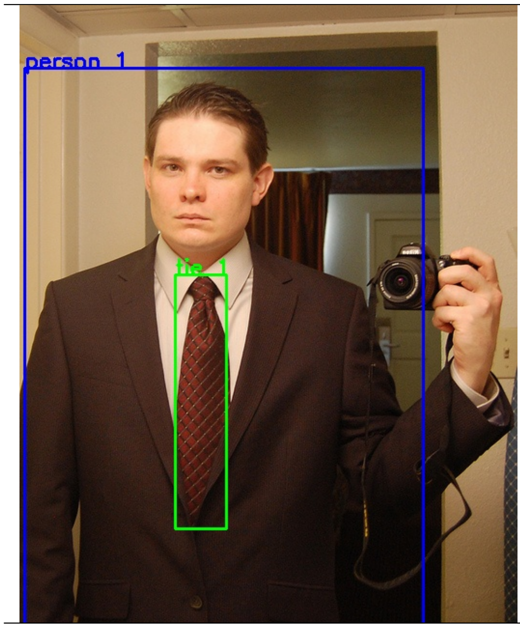
Localization Result	Level Two Summaries
	• **General**: person_1 interacting with tie_1 • **Person**: person_1 wearing tie_1

Table 6.16

“Dog” Level Two Summary Results

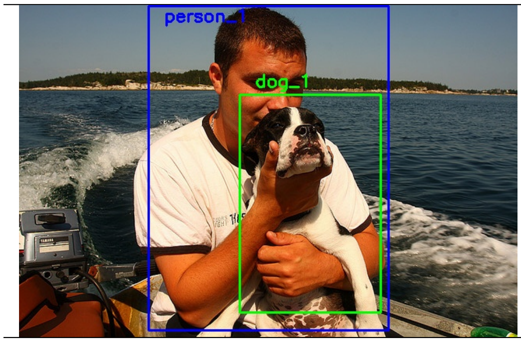
Localization Result	Level Two Summaries
	• **General**: person_1 interacting with dog_1 • **Person**: person_1 petting dog_1

Table 6.17

“Cell Phone” Level Two Summary Results

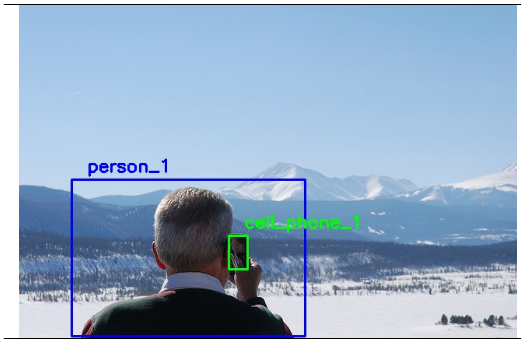
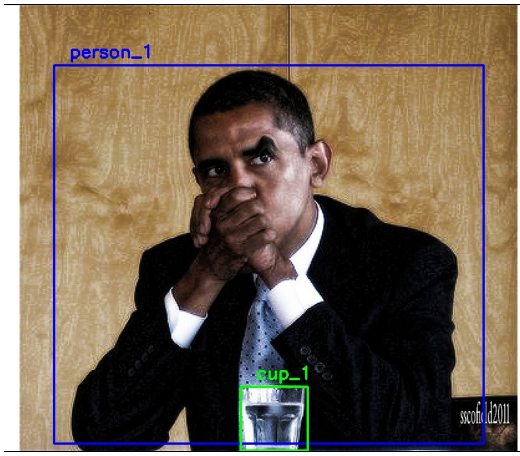
Localization Result	Level Two Summaries
	• **General**: person_1 interacting with cell_phone_1 • **Person**: person_1 talking on cell_phone_1

Table 6.18

“Cup” Level Two Summary Results

Localization Result	Level Two Summaries
	• **General**: person_1 interacting with cup_1 • **Person**: person_1 drinking cup_1

Tables 6.11 - 6.18 show the Level Two Summary results for the corresponding Level One Summaries of Tables 6.3 - 6.10. Each table presents the object localization results on the left with the general-domain Level Two Summaries, denoted by **General:**, and person-domain Level Two Summaries, denoted by **Person:**, on the right of the table.

Table 6.11 shows that the general-domain Level Two Summary indicates that the person is interacting with the umbrella. The person-domain Level Two Summary further refines the description and indicates that the person is carrying the umbrella. Table 6.12 shows the general description of a person interacting with the cake. The person-domain Level Two Summary further refines this information by indicating the person is likely eating the cake. Table 6.13 shows the general label of a person interacting with a piece of broccoli, which is further refined by the person-domain summary of the person eating broccoli. Table 6.14 describes the general interaction of the person interacting with the bench. The person-domain Level Two Summary shows the more detailed description of the person sitting on the bench. Table 6.15 provides the general domain interaction of person interacting with a tie and the refined person-domain interaction of the person wearing the tie. Table 6.16 shows that the person is interacting with the dog in the general-domain. In the person-domain, the table shows that the person is petting the dog in the image. Table 6.17 shows that in general, the person is interacting with a cell phone. Using the refined rules for the person-domain, the label becomes the person is talking on a cell phone. Lastly, Table 6.18 presents the general domain interaction of a person interacting with a cup. The person-domain refinement generated the label indicating that the person is drinking from the cup. The following section presents the methods used to verify the validations conducted in this chapter.

6.3 Validation

The validation process followed a similar process to the Level One Summary and Level Two Summary validations described in Sections 4.8.2, 5.3.4, and 5.4.4. The validation processes discussed in those sections were a manual process as there were no baseline Level One or Level Two Summaries to use for comparison. The validation processes discussed served to verify that the generated summaries were an accurate reflection of what could be inferred given the object localization results in a scene. Furthermore, since S2T labels were already subjective in a sense, the manual validation process did not introduce any additional subjectivity to the S2T scene labeling process.

Validation was performed on a subset of the person-domain Level Two Summaries. Person-domain Level Two Summaries were chosen because they contained a label for Level One Summaries, general-domain Level Two Summaries, and person-domain Level Two Summaries. As such, selecting this subset of results verified that the same amount of results were analyzed for each type of Level Two Summary. To select the image subset, the person-domain Level Two Summary result file, generated from the validations of Section 6.2.3, was randomly sampled. To accomplish this, the person-domain Level Two Summary file was loaded into a Pandas [62] data frame and the Python [45] programming language's built in random functionality was used to randomly select 500 image paths. For the selected 500 image paths, the Pandas library was used to filter the person-domain Level Two Summary file to just the rows that corresponded to the randomly selected image paths.

The validation process was performed in two stages. The first stage of validation was the inspection of the general-domain Level Two Summaries. The second stage of validation performed was the inspection of person-domain Level Two Summaries. As shown in Section 4.8.2, the Level One Summaries were reliably accurate during construction of the system, and since both of the Level Two Summary types incorporated the Level One Summary information, explicit validation of the Level One Summaries was omitted from this validation process.

Though there were two validation procedures, both were implemented using almost identical code. First, the Level Two Summary CSV file was loaded into a Pandas [62] data frame, where each row was inspected individually. It is important to note that inspecting each row individually inherently implies that each general-domain Level Two Summary and person-domain Level Two Summary was inspected individually because of how the output Level Two Summary data frame was constructed in Section 6.2.3. After loading a row, the argument and referrant bounding boxes, along with their corresponding class labels, were drawn on the original image with the appropriate class label. This process was implemented using the OpenCV [5] library. After drawing the localization results for the row, the general-domain Level Two Summary, or the person-domain Level Two Summary depending on which validation procedure was being performed, was displayed to the console at which point the user was prompted to “accept” or “reject” the result. Accepted and rejected results were stored in individual data frames and written to CSV files for inspection after the validation process was completed. The following tables show the result for the sensitivity analysis validation. It is important to note that person - person interactions were computed only for the general-domain. For the person-domain, only person - animate object and person - inanimate object interactions were computed.

Table 6.19

Validation Data Set Breakdown

Sampled Images	500
General-Domain Level Two Summaries	4179
Person-Domain Level Two Summaries	2029

Table 6.20

Incorrect Image Totals

Sampled Images	500
Images With Incorrect General Results	9
Images With Incorrect Person Results	58

Table 6.19 shows for the 500 sampled images, there were 4179 corresponding general-domain Level Two Summaries and there were 2029 corresponding person-domain Level Two Summaries. As mentioned previously, only person - animate and person - inanimate person-domain Level Two

Summaries were inspected, which explains the large discrepancy in the numbers. Table 6.20 shows that nine images account for all of the incorrect general-domain Level Two Summaries and that 58 images account for all of the incorrect person-domain Level Two Summaries. This implies that a small percentage of the images were responsible for the inaccurate labels.

Table 6.21

General-Domain Results

Total Interactions	1027
Accepted Interactions	998
Rejected Interactions	29
Total Non Interactions	3152
Accepted Non Interactions	3152
Rejected Non Interactions	0

Table 6.21 shows the results obtained for the general-domain Level Two Summaries computed for the set of 500 randomly sampled images. The general-domain Level Two Summaries were broken into two categories: summaries that contained an interaction and summaries that did not. The table shows that of the 4179 general-domain Level One Summaries, 1027 of those summaries contain labels that show two objects are interacting. Conversely, the table shows that of the 4179 total summaries, 3152 of those summaries are summaries that indicated the object tuple was not interacting. The results were separated in this manner to provide further insight to the system's strengths and weaknesses. The table shows that of the 1027 computed positively labeled interactions, only 29 of those interactions were incorrect. The table also shows that all 3152 negatively labeled interactions were accurate. These results are similar to the results presented in Section 5.6.1. Table 6.23 breaks down the results of the inaccurate labels in further details,

but when taking the 29 invalid general-domain Level Two Summaries into consideration, the S2T system was able to achieve an accuracy of 99% on the COCO 17 [35] data set, a number on par with the results of Section 5.6.1 during initial development of the system which used the COCO 14 data set.

Table 6.22

Person Domain Results

Total Actions	544
Accepted Actions	496
Rejected Actions	46
Total Non Actions	1485
Accepted Non Actions	1461
Rejected Non Actions	24

Table 6.22 shows the results obtained for the person-domain Level Two Summaries computed for the 500 randomly sampled images of the COCO 17 [35] Level One Summaries. Identical to Table 6.21, the person-domain Level Two Summaries were separated by those that contained a person interacting with an object and those that contained a person not interacting with an object. As mentioned previously, only object tuples that contained a person potentially interacting with an object were considered for the person-domain Level Two Summary computation. Table 6.22 shows that of the 2029 total summaries, 544 of them contained a positive interaction system output and 1485 contained a negative interaction system output. Of the 544 person-domain Level Two Summaries labeled as a positive person - object interaction, 496 were deemed accurate. Of the 1485 person-domain Level Two Summaries labeled as a negative person - object interaction, 1461 of the summaries were deemed correct. Not withholding the incorrect results due to the known

limitations, discussed in detail in Section 5.5, the S2T system was able to achieve an accuracy of 96% on the COCO 17 data set. Again, these were images not presented to the system during development, so these results provide preliminary evidence that the system generalizes well to new data sets.

Table 6.23

Incorrect Result Breakdown

	General Domain	Person Domain
Total Incorrect	29	72
Bad Detection	20	7
Depth Problem	9	34
Occlusion Problem	0	5
Bad Result	0	26

As a final analysis, Table 6.23 provides additional details about where the system falls short. As can be seen in the table, of the 29 total incorrect general-domain Level One Summaries, all 29 of them are due to either the bad detection problem or the depth problem. These two known limitations are discussed in great detail in Section 5.5. Unlike the results in Chapter 5, the general-domain Level Two Summaries were affected by the bad localization problem. The results subject to the bad localization problem occurred when the YOLOv3 [48] algorithm generated a bounding box that was too large for an object, and the subsequent overlapping resulted in the generation of an invalid general-domain Level Two Summary. The table also shows that none of the general-domain summary invalid labels were due to the occlusion problem or a bad result in general. Based on the results shown in Tables 6.23, the general-domain Level Two Summaries were 98% accurate on the sampled validation set. This is similar to the results shown in Section 5.6.1. The accuracy of the general-domain Level Two Summaries will likely always be fairly high because the general-domain Level Two Summary computations are based on a very simple rule base, as shown in Table 5.2 and the simplicity of design allows for the general-domain computations to generalize easily.

Table 6.23 shows that of the 72 person-domain Level Two Summaries: seven were due to the bad detection problem, 34 were due to the depth problem, five were due to the occlusion problem, and 26 were an actual bad person-domain Level Two Summary. Possible remedies for the known limitations were discussed in Section 5.5. During validation, the 26 Level Two Summaries that were inaccurate because of system output were further analyzed. During analysis, it was observed that all of the inaccurate labels were in the "grey area" of the FIS output. In other words, the proper fuzzy output and improper fuzzy output both had a degree of membership but the improper output was the eventual output generated during centroid defuzzification [60]. Withholding the 46 results due to known system limitations, the person-domain S2T system performed with an accuracy of 98% on the COCO 17 [35] data set. As mentioned previously, the COCO 17 data set samples used during validation were all images the system did not encounter during development, discussed in Section 5.4.2, and these results provide positive evidence to support the person-domain Level Two Summary FIS does generalize well to additional data sets. This chapter concludes by revisiting the remaining research questions and hypotheses from Chapter 1.

6.4 Discussion

This chapter described a sensitivity analysis validation performed by applying the S2T system developed during this research to the COCO 17 [35] data set, a data set which the system had not yet been applied to. The validation made use of the YOLOv3 [48] model for object localization, the results of which were used to construct Level One Summaries using the GIOU [51] and HOF [40] methods. These Level One Summaries were then used as input to the computation of the general-domain and person-domain Level Two Summaries. Unlike the evaluations performed in Chapters 3 - 5 where each phase of the S2T pipeline was constructed individually, the entire S2T pipeline was implemented in a streamlined manner for the sensitivity analysis validation. As such, given the raw input images of the COCO 17 data set, the output of the system were CSV files for each of: object localization results, Level One Summaries for each object tuple, general-domain Level Two Summaries for each Level One Summary, and person-domain Level Two Summaries for each appropriate Level One Summary. The YOLOv3 model was decided upon so that each resulting CSV output file would contain all necessary information to replicate the system results, as opposed to requiring binary mask images to replicate the object localization results.

The primary focus of the sensitivity analysis was verification that Level Two Summary computation could be performed on a data set not used to develop the initial system and retain accuracy. Two types of Level Two Summaries were computed, the general-domain Level Two Summaries and the person-domain Level Two Summaries. Section 6.2.3 presented the implementation details of the methods used to construct both types of Level Two Summaries for this validation. The validations of this chapter built off of work developed in Chapter 5, where the general-domain FIS and person-domain FIS were both implemented used the COCO 14 [35] data set as baseline data to construct the system. The rule base for the general-domain FIS was visualized in Table 5.2 and the entire rule based used to construct the person-domain FIS is reserved to Appendix A.

The results in Section 6.2.3.1, specifically Tables 6.11 - 6.18 show examples of general- and person-domain Level Two Summary results for sample images of the COCO 17 [35] data set.

These results, combined with the metrics presented in Section 6.3, provide positive evidence to answer the third research question and support the associated hypothesis (H3) from Chapter 1:

RQ3: When applying the S2T framework to a data set other than the data set used to construct the system, will the results be as reliable as the initial data set results?

H3: Since the S2T system was modeled as a fuzzy inferencing system at its core, the rule-base will be data set agnostic and as such the developed S2T system will be as reliable on an additional data set as it was on the training data set.

Importantly, none of the images used in the validation process had been incorporated during the S2T system development, and this implies that the S2T system does generalize well to other data sets in both the general- and person-domains, in support of H3.

The last outstanding research question and hypothesis from Chapter 1 that needs to be addressed is:

RQ4: How can the S2T framework developed in this research be used as a mechanism for applying scene interaction labels that incorporate spatial reasoning information to a previously unlabeled data set?

H4: The developed S2T system requires no a priori *information regarding scene labels, thus applying the framework to a previously unlabeled data set will not limit the effectiveness of the system in regards to generating accurate scene descriptions.*

Based on the results shown in Tables 6.11 - 6.18 coupled with the computed validation metrics, it is a reasonable conclusion that the S2T framework can indeed be used to label a raw image data set reliably, supporting hypothesis H4. Section 5.7 discussed this as a potential system usage, and using the S2T system in this manner could prove to be an invaluable resource when training larger scale machine learning models that require massive data sets for input. In the final chapter, the conclusions drawn from this research will be discussed.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

The research, evaluations, and validations presented in this research constructed a S2T system that incorporated spatial relationship information when deriving natural language scene descriptions. The goal of the constructed S2T system was the ability to generate more informative scene descriptions given an unlabeled image as input. The S2T pipeline, shown in Figure 7 was constructed in three phases that corresponded to the three evaluations performed throughout the research. A final validation was performed on the S2T system on a data set the system had not yet encountered.

Generation of the object localization results for the set of raw input images was the focus of Chapter 3. These localization results were generated as one of either bounding box segmentation results or pixel-wise mask segmentation results for each object detected in an image. The localization results were generated using pre-trained off the shelf object detection models and since the goal of the research of this dissertation was not improving object detection models, all localization results were used as is from the model. Additionally, each image was processed using the Inception [61] model in order to generate image meta data for each image in the data set. This meta data was used solely as latent input to the computation of Level Two Summaries, detailed in Chapter 5. The

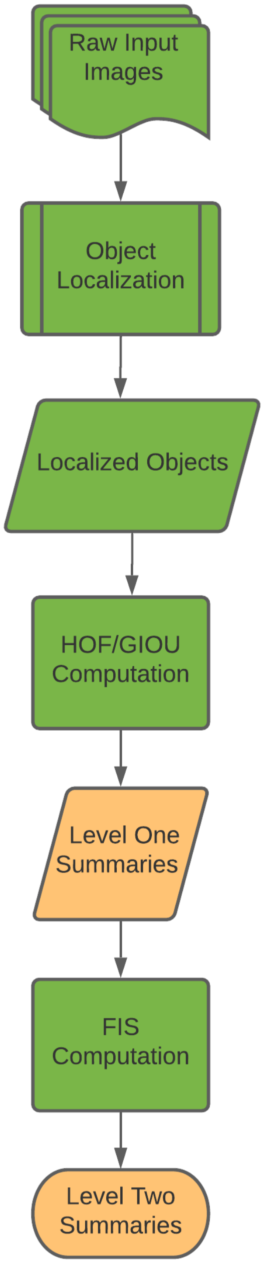


Figure 7.1

Level Two Summaries in the S2T Pipeline

object localization results for each input image were then passed along to the second phase of the S2T pipeline.

The second phase of the constructed S2T pipeline introduced the concept of Level One Summaries. Chapter 4 focused on the generation and refinement of Level One Summaries. The Level One Summaries described the spatial relationships that exist between each object tuple in an image. The GIOU [51] and HOF [40] methods were used to construct Level One Summaries. The GIOU algorithm provided information regarding the overlap and proximity relationship between object tuples in an image. The HOF algorithm provided information pertaining to the cardinal directionality relationship between the object tuples in an image. The result of each of these algorithms were numerical values that were inherently fuzzy. These fuzzy values were defuzzified to obtain crisp values, and the crisp values for the overlap and spatial relationship were used in order to construct a Level One Summary for each object tuple in the set of object tuples for an input image. The Level One Summary results and validation of Chapter 4 provided evidence to answer the first research question and and support the associated hypothesis (H1) posed in Chapter 1:

RQ1: How can concise and meaningful natural language descriptions (Level One Summaries) of spatial relationships between all object tuples in an image be derived?

H1: Histogram of Forces (HOF) [40] and Generalized Intersection Over Union [51], both rigorously validated methods with strong mathematical proofs, will provide concise natural language descriptions of spatial relationship information between object tuples in a scene.

The resulting output of the Level One Summaries, when viewed as fuzzy variables, were the proximity, overlap, and spatial relationship values. These fuzzy values were used alongside the object localization results and image meta data generated in Chapter 3 as input to the final phase of the S2T pipeline.

The final phase of the S2T pipeline introduced the concept of Level Two Summaries. The evaluations and research detailed in Chapter 5 were focused on the construction and refinement of Level Two Summaries. Level two Summaries were generated in two domains. The first domain, the general domain, constructed Level Two Summaries that describe the general interactions of object tuples in a scene. The second domain, the person domain, generated a refined set of labels that pertained to interactions between a person and object tuple in an image. For each domain, a FIS was constructed that was responsible for the generation of Level Two Summaries. These FIS both took as input the fuzzy values for proximity, overlap and spatial relationships, which were in essence the information contained in Level One Summaries as well as the object localization results computed in Chapter 3. One important caveat was that the meta data generated during the object localization evaluation was also used as input to the computation of the person-domain Level Two Summaries.

As the general-domain consisted of high-level descriptions of interactions, a simple rule base was designed for the general-domain FIS and the entire rule base consisted of a handful of rules, shown in Table 5.25 of Chapter 5. Arguably, the more interesting scene descriptions were generated by the FIS responsible for constructing person-domain Level Two Summaries. The person-domain FIS relied on an object hierarchy and a very detailed set of domain specific rules, shown in completion in Appendix A, that were manually crafted to represent how a person would interact with each of the objects in the set of possible object classes in the COCO [35] data set. Construction of the person-domain Level Two FIS highlighted the requirement that in order to generate refined labels for a specific domain, domain expert knowledge was required to construct the rule base. While these evaluations focused on the person - object interaction domain, any number of domain-specific Level Two Summary rule bases can be constructed given an expert with enough knowledge about the desired system outputs. The Level Two Summary results and validation metrics computed for both domains provided evidence to support the second hypothesis (H2) and answer the second research question asked in Chapter 1:

RQ2: How can the Level One Summaries provide meaningful information to the creation of informative, high-level, natural language descriptions (Level Two Summaries) of the object interactions in a scene?

H2: By utilizing a fuzzy inferencing approach in the generation of scene labels, the uncertainty in spatial relationship reasoning can be effectively modeled and incorporated into scene descriptions generated by the S2T system.

Chapter 5 effectively completed the S2T pipeline construction and finalized the system shown in the pipeline diagram of Figure 6.1.

The validation outlined in this research was geared toward verification of the S2T system constructed in Chapters 3 - 5, and was the focus of Chapter 6. During construction of the S2T system, the COCO 14 [35] data set was used. For this validation, the COCO 17 data set was used such that images the S2T system had not yet encountered could be processed and inspected. Unlike the object localization, Level One Summary generation, and Level Two Summary generation evaluations, which were all implemented as standalone evaluations in order to validate each phase of the system, the sensitivity analysis validation showed that the pipeline could be implemented in a streamlined manner. For this validation, the same methodology in the corresponding chapters was used to generate the object localization results (Chapter 3), which fed into Level One Summaries (Chapter 4), which then fed into Level Two Summaries (Chapter 5). This validation also was designed to be easily shareable so that the code could be transmitted without the need for large downloads. The sensitivity analysis validation of Chapter 6 provided Level One Summaries and both general- and person-domain Level Two Summaries for the COCO 17 data set. A detailed inspection of the generated general- and person-domain Level Two Summaries was performed on a subset of the entire set of results in order to verify that the system performed at the same standard as the results generated by the S2T system on the COCO 14 data set.

The results and computed metrics obtained during the sensitivity analysis validation performed in Chapter 6 provided positive evidence to answer the third research question and support the third hypothesis (H3) originally stated in Chapter 1:

RQ3: When applying the S2T framework to a data set other than the data set used to construct the system, will the results be as reliable as the initial data set results?

H3: Since the S2T system was modeled as a fuzzy inferencing system at its core, the rule-base will be data set agnostic and as such the developed S2T system will be as reliable on an additional data set as it was on the training data set.

Additionally, the results and computed metrics provided sufficient evidence to support the final research hypothesis (H4) and answer the fourth research question of Chapter 1:

RQ4: How can the S2T framework developed in this research be used as a mechanism for applying scene interaction labels that incorporate spatial reasoning information to a previously unlabeled data set?

H4: The developed S2T system requires no a priori *information regarding scene labels, thus applying the framework to a previously unlabeled data set will not limit the effectiveness of the system in regards to generating accurate scene descriptions.*

As stated previously, the goal of this research was to construct a S2T system that successfully incorporated spatial relationship information that provided more accurate information than existing S2T systems. The evaluations and validations in Chapters 3 - 6 demonstrate that the constructed S2T system can generate informed scene descriptions which do inherently incorporate spatial relationship information. Additionally, these labels do provide accurate descriptions of scene interactions that can not be generated without the inclusion of spatial relationships. Currently, no S2T models generate such labels, and the only existing data sets that incorporate this information and provide scene descriptions are those which were manually labeled.

7.1 Research Contributions

As demonstrated in Chapter 4, Level One Summaries provide additional information to image data sets in the form of spatial reasoning information between object tuples. This research applied that information to the task of constructing more informative scene descriptions, however these Level One Summaries themselves can be used as an additional feature of a data set if so desired.

Chapters 5 and 6 demonstrated the ability of the system to be used as a standalone image annotation tool. These chapters show that the S2T system incorporates spatial reasoning information into image annotations, a property not readily available for current image annotation data sets. Additionally, Chapter 6 showed that the S2T system can be applied to a previously non-annotated data set with virtually no additional work.

In addition, Chapter 6 also demonstrated the ability to use the S2T system as an image annotation tool for labeling training data sets for deep learning applications. Validations performed in Chapter 6 provide a proof of concept of how the S2T system can be used in this manner. As the S2T system was applied to a previously unseen data set, and generated labels with no required *a priori* information, it follows that the S2T system can accomplish the task of image training data set annotation.

7.2 Future Work

Initial future work will focus on the incorporation of depth information into the S2T system to alleviate erroneous results. Depth estimation appears to be the primary bottleneck of the current system and the hypothesis is that incorporating the appropriate depth information will alleviate the "depth problem" discussed in Section 5.5. Additionally, the incorporation of depth information also opens the door for applying the system to any three-dimensional data set, a very interesting problem space. The work by Kaur et al in [29] presents one potential avenue of incorporating depth information into the current S2T system.

In addition to incorporating depth information, the system can be extended such that n-tuples as opposed to tuples are processed by the S2T system. Current work, such as the work by Scott et al. in [56], is a potential method for extending the current system to n-tuple processing.

An additional avenue of future work involves applying the S2T system to a domain other than the "person domain." For this work, a subject matter expert from the domain will be necessary to provide input toward constructing the rule base for the domain. However, applying the S2T system in this manner would provide supporting evidence that the system is extensible and can work across multiple domains with the only leg work involved being the construction of the rule base.

Deep learning models are the current hot button topic in the field of computer vision and machine learning in general [7], [71], [27], [12], [30], and [13]. Training such models requires a substantial amount of data. Current computer vision data sets do not contain spatial reasoning information like the person-domain Level Two Summaries provided by this research. An interesting future work would include using the S2T system as an image annotation tool, as discussed in Chapter 6, and determining if enough data can be labeled via the S2T system to determine if state-of-the-art deep learning techniques could be "taught" to infer spatial reasoning.

Additional work for the S2T system will be focused on streamlining the process such that it works in real-time. If the S2T system can achieve real-time execution speeds, it can very easily be applied to video as opposed to still images. As such, future work will also include applying the S2T system to video sources, with the caveat the system can achieve real-time execution speeds. Lastly, an ambitious future work, then, is to apply the S2T system to video sources with three-dimensional information, a very interesting problem space as this involves autonomous vehicles and robotics.

Further additional work will involve exploring the capabilities of using the degree of membership of the Level Two summaries. As the system currently stands, only the crisp outputs are used, which results in at least some information loss with regards to object-object interactions. By using the fuzzy output values of the system, Level Two summaries can be constructed in a manner that reflects not only the action being performed, but also the confidence the system has in the associated label. Adding in this functionality would provide more information, specifically the system's confidence of the assigned label, to the user.

7.3 Plan of Publication

Currently, the S2T system is the subject of an International Joint Conference on Neural Networks (IJCNN) conference paper pending acceptance. As this conference is focused heavily on neural networks, the ability to use the S2T system as a data set annotation tool is the primary focus of the submitted paper.

Future publications will also center around the completion of the future work listed above. Primarily, extending the S2T system to three-dimensions and application of the S2T system to video sources will each be the topic of future publications.

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,", 2015, Software available from tensorflow.org.

[2] Anaconda Core Team, *Anaconda Software Distribution*, Anaconda, 2016.

[3] D. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz, and M. Aud, "Linguistic summarization of video for fall detection using voxel person and fuzzy logic," *Computer vision and image understanding : CVIU*, vol. 113 1, 2009, pp. 80–89.

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection,", 2020.

[5] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[6] N. Buch, S. A. Velastin, and J. Orwell, "A Review of Computer Vision Techniques for the Analysis of Urban Traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, 2011, pp. 920–939.

[7] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers,", 2020.

[8] J. Corso and B. Moore, "Voxel51 // developer tools for ML,".

[9] J. Dai, K. He, and J. Sun, "Instance-aware Semantic Segmentation via Multi-task Network Cascades," *CoRR*, vol. abs/1512.04412, 2015.

[10] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds., Curran Associates, Inc., 2016, pp. 379–387.

[11] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, Washington, DC, USA, 2005, CVPR '05, pp. 886–893, IEEE Computer Society.

[12] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets Great Again,", 2021.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,", 2021.

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, June 2010, pp. 303–338.

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, September 2010, pp. 1627–1645.

[16] K.-P. Gapp, "Basic Meanings of Spatial Relations: Computation and Evaluation in 3D Space," *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2)*, USA, 1994, AAAI'94, p. 1393âĂŞ1398, American Association for Artificial Intelligence.

[17] R. Girshick, "Fast R-CNN," *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2014, CVPR '14, pp. 580–587, IEEE Computer Society.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, January 2016, pp. 142–158.

[20] M. Haldekar, A. Ganesan, and T. Oates, "Identifying Spatial Relations in Images using Convolutional Neural Networks," *CoRR*, vol. abs/1706.04215, 2017.

[21] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition,", 2015.

[23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, November 1997, pp. 1735–1780.

[24] J. H. Hosang, R. Benenson, and B. Schiele, “Learning non-maximum suppression,” *CoRR*, vol. abs/1705.02950, 2017.

[25] S. Huang, Y. Chen, T. Yuan, S. Qi, Y. Zhu, and S. Zhu, “PerspectiveNet: 3D Object Detection from a Single RGB Image via Perspective Points,” *CoRR*, vol. abs/1912.07744, 2019.

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional Architecture for Fast Feature Embedding,” *Proceedings of the 22Nd ACM International Conference on Multimedia*, New York, NY, USA, 2014, MM ’14, pp. 675–678, ACM.

[27] G. Jocher, A. Stoken, A. Chaurasia, J. Borovec, NanoCode012, TaoXie, Y. Kwon, K. Michael, L. Changyu, J. Fang, A. V, Laughing, tkianai, yxNONG, P. Skalski, A. Hogan, J. Nadar, imyhxy, L. Mammana, AlexWang1900, C. Fati, D. Montes, J. Hajek, L. Diaconu, M. T. Minh, Marc, albinxavi, fatih, oleg, and wanghaoyang0106, “ultralytics/yolov5: v6.0 - YOLOv5n ’Nano’ models, Roboflow integration, TensorFlow export, OpenCV DNN support,”, October 2021.

[28] A. Karpathy and L. Fei-Fei, “Deep Visual-Semantic Alignments for Generating Image Descriptions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, April 2017, pp. 664–676.

[29] J. Kaur, T. Laforet, and P. Matsakis, “Fast Fourier Transform based Force Histogram Computation for 3D Raster Data,” *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2020, Valletta, Malta, February 22-24, 2020*, M. D. Marsico, G. S. di Baja, and A. L. N. Fred, eds. 2020, pp. 69–74, SCITEPRESS.

[30] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in Vision: A Survey,” *ACM Computing Surveys*, Jan 2022.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Proceedings of the 25th International Conference on Neural Information Processing Systems*, USA, 2012, NIPS’12, pp. 1097–1105, Curran Associates Inc.

[32] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, 5 2015, pp. 436–444.

[33] L.-J. Li, R. Socher, and F.-F. Li, “Towards total scene understanding: Classification, annotation and segmentation in an automatic framework.,” *CVPR*. 2009, pp. 2036–2043, IEEE Computer Society.

[34] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully Convolutional Instance-aware Semantic Segmentation,” *CoRR*, vol. abs/1611.07709, 2016.

[35] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” *CoRR*, vol. abs/1405.0312, 2014.

[36] T.-Y. Lin, P. DollÃąr, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,”, 2017.

[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed, “SSD: Single Shot MultiBox Detector.,” *CoRR*, vol. abs/1512.02325, 2015.

[38] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, November 2004, pp. 91–110.

[39] *MATLAB version 9.3.0.713579 (R2017b)*, The Mathworks, Inc., Natick, Massachusetts, 2017.

[40] P. Matsakis and L. Wendling, “A New Way to Represent the Relative Position between Areal Objects,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 7, 1999, pp. 634–643.

[41] K. Miyajima and A. Ralescu, “Spatial organization in 2D images,” *IEEE World Congress on Computational Intelligence*, 07 1994, pp. 100 – 105 vol.1.

[42] K. Miyajima and A. Ralescu, “Spatial Organization in 2D Segmented Images: Representation and Recognition of Primitive Spatial Relations,” *Fuzzy Sets Syst.*, vol. 65, no. 2-3, August 1994, pp. 225–236.

[43] T. Oliphant, “NumPy: A guide to NumPy,”, USA: Trelgol Publishing, 2006–.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds., Curran Associates, Inc., 2019, pp. 8024–8035.

[45] Python Core Team, *Python: A dynamic, open source programming language*, Python Software Foundation, 2015.

[46] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *CoRR*, vol. abs/1506.02640, 2015.

[47] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” *arXiv preprint arXiv:1612.08242*, 2016.

[48] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv*, 2018.

[49] J. Reed, M. Naeem, and P. Matsakis, "A First Algorithm to Calculate Force Histograms in the Case of 3D Vector Objects," *ICPRAM 2014 - Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods, ESEO, Angers, Loire Valley, France, 6-8 March, 2014*, M. D. Marsico, A. Tabbone, and A. L. N. Fred, eds. 2014, pp. 104–112, SciTePress.

[50] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *CoRR*, vol. abs/1506.01497, 2015.

[51] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression,", 2019.

[52] A. Rosebrock, "Face Alignment with OpenCV and Python,", 2017.

[53] T. J. Ross, *Fuzzy Logic with Engineering Applications*, 3rd edition, Wiley, New York City, NY, USA, 2010.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, 2015, pp. 211–252.

[55] M. A. Sadeghi and D. Forsyth, "30Hz Object Detection with DPM V5," *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., Cham, 2014, pp. 65–79, Springer International Publishing.

[56] G. Scott, M. Klaric, and C.-R. Shyu, "Modeling Multi-object Spatial Relationships for Satellite Image Database Indexing and Retrieval," *Image and Video Retrieval*, W.-K. Leow, M. S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, and E. M. Bakker, eds., Berlin, Heidelberg, 2005, pp. 247–256, Springer Berlin Heidelberg.

[57] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks.," *CoRR*, vol. abs/1312.6229, 2013.

[58] A. Shrivastava, A. Gupta, and R. B. Girshick, "Training Region-based Object Detectors with Online Hard Example Mining," *CoRR*, vol. abs/1604.03540, 2016.

[59] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2014.

[60] M. Sugeno, "An introductory survey of fuzzy control," *Information Sciences*, vol. 36, no. 1, 1985, pp. 59–83.

[61] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[62] The Pandas Development Team, *Pandas*, February 2020.

[63] J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H. Scha, “Real-Time Visual Concept Classification,” *IEEE Transactions on Multimedia*, vol. 12, no. 7, 2010, pp. 665–681.

[64] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, “Segmentation as selective search for object recognition,” *2011 International Conference on Computer Vision*, 2011, pp. 1879–1886.

[65] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3156–3164.

[66] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge,” *CoRR*, vol. abs/1609.06647, 2016.

[67] P. Viola and M. Jones, “Robust Real-time Object Detection,” *International Journal of Computer Vision*, 2001.

[68] Y. Wang and J. Ye, “An Overview Of 3D Object Detection,” *CoRR*, vol. abs/2010.15614, 2020.

[69] J. Warner, J. Sexauer, scikit fuzzy, twmeggs, A. M. S., A. Unnikrishnan, G. CastelÃčo, F. A. Pontes, T. Uelwer, pd2f, laurazh, F. Batista, alexbuy, W. Song, T. G. Badger, R. A. M. PÃľrez, J. F. Power, H. Mishra, G. O. Trullols, A. HÃűrteborn, and 99991, “scikit-fuzzy/scikit-fuzzy: Scikit-Fuzzy version 0.4.1,”, March 2019.

[70] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,” *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, eds., Lille, France, 07–09 Jul 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 2048–2057, PMLR.

[71] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, “TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios,”, 2021.

APPENDIX A

PERSON DOMAIN FUZZY RULE BASE

This appendix contains the rule base used to construct the person domain level two summaries of Chapter 5. The rules are separated by category for ease of readability. While the SKFuzzy [69] library requires enumeration of all negative values, only the positive interaction rules are shown in this appendix for brevity.

A.1 Person - Animal Rules

- IF pet AND very close OR close AND above THEN petting
- IF ridden animal AND very close AND above THEN riding
- IF large animal AND very close OR close THEN interacting

A.2 Person - Appliance Rules

- IF overlap AND very close OR close THEN using
- IF no overlap AND very close THEN using

A.3 Person - Clothing Rules

- IF worn clothing AND overlap AND very close OR close THEN wearing
- IF carried clothing AND overlap AND very close OR close THEN carrying

A.4 Person - Electronics Rules

- IF tv AND overlap AND very close OR close OR medium THEN watching
- IF tv AND no overlap AND very close OR close THEN watching
- IF cell phone AND overlap AND very close THEN talking on
- IF other device AND very close OR close THEN using

A.5 Person - Food Rules

- IF utensil AND overlap AND very close OR close THEN using
- IF beverage AND overlap AND very close THEN drinking
- IF food AND overlap And very close OR close THEN eating

A.6 Person - Furniture Rules

- IF couch AND overlap AND very close THEN sitting on
- IF bed AND overlap AND very close THEN laying on
- IF chair AND overlap AND very close AND above THEN sitting in
- IF other furniture AND overlap AND very close OR close THEN sitting at
- IF decoration AND overlap AND very close THEN interacting with

A.7 Person - Household Item Rules

- IF overlap AND very close THEN interacting

A.8 Person - Sports Rules

- IF ridden board AND overlap AND very close OR close OR medium AND above THEN riding
- IF ridden board AND no overlap AND very close OR close AND above THEN riding
- IF baseball bat AND very close OR close AND below THEN playing baseball
- IF tennis racket AND overlap AND very close OR close THEN playing tennis
- IF frisbee AND very close OR close OR medium THEN throwing frisbee
- IF kite AND no overlap AND medium OR far OR very far THEN flying kite
- IF sports ball IS soccer ball AND very close OR close OR medium THEN playing soccer
- IF sports ball IS baseball AND overlap AND very close OR close OR medium THEN playing baseball
- IF sports ball IS tennis ball AND very close OR close AND not below THEN playing tennis

A.9 Person - Urban Rules

- IF bench AND overlap AND very close OR close AND above THEN sitting
- IF traffic light AND no overlap AND medium OR far OR very far AND below THEN at intersection
- IF parking meter AND overlap AND very close THEN using
- IF other urban object AND overlap AND very close OR close THEN interacting

A.10 Person - Vehicle Rules

- IF cycle AND overlap AND very close AND above THEN riding
- IF passenger vehicle AND overlap AND very close THEN riding in
- IF personal vehicle AND overlap AND very close AND not below THEN driving