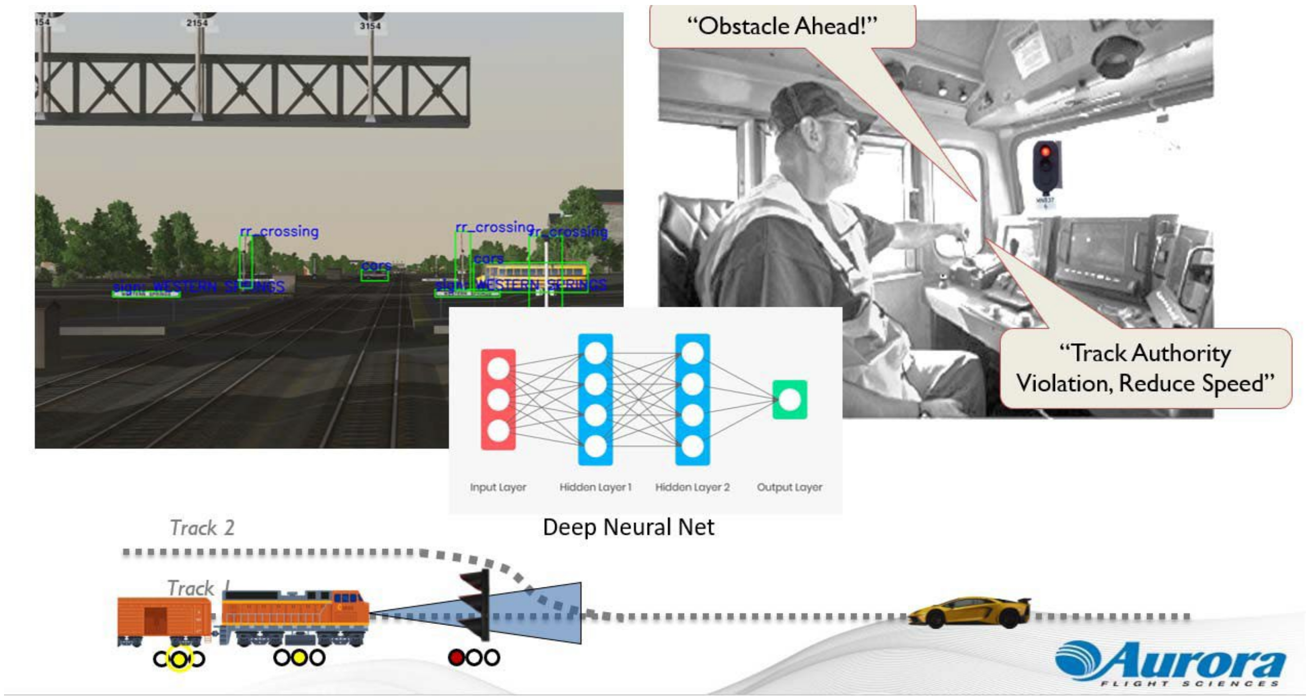
U.S. Department of Transportation

Federal Railroad Administration

Office of Research, Development and Technology
Washington, DC 20590

External Perception for Locomotives: Systems and Algorithm Development



DOT/FRA/ORD-24-13

Final Report | March 2024

REPORT DOCUMENTATION PAGE		*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.		
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE April 2020	3. REPORT TYPE AND DATES COVERED Technical Report 04/27/2018 to 03/27/2019
4. TITLE AND SUBTITLE External Perception for Locomotives: Final System and Algorithm Development Report		5. FUNDING NUMBERS CONTRACT NUMBER 693JJ618C000003
6. AUTHOR(S) Daniel Brown, Ben Jafek, Nathan Varney		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Aurora Flight Sciences 90 Broadway Cambridge, MA 02139		8. PERFORMING ORGANIZATION REPORT NUMBER AR19-170.
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Department of Transportation Federal Railroad Administration Office of Railroad Policy and Development Office of Research, Development, and Technology Washington, DC 20590		10. SPONSORING/MONITORING DOT/FRA/ORD-24-13
11. SUPPLEMENTARY NOTES COR: Michael E. Jones		
12a. DISTRIBUTION/AVAILABILITY STATEMENT This document is available to the public through the FRA eLibrary.		12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) The Federal Railroad Administration sponsored a research team from Aurora Flight Sciences to develop the External Perception for Locomotives (ExP-L) system, which uses the FRA Cab Technology Integration Lab (CTIL) to demonstrate prototype object operator augmentation. The system can detect and localize objects of interest in and around the track, interpret the state of the detected objects to determine if there is any urgent and actionable information, and communicate this information to the operator. Similar to adaptive cruise control in modern cars, the system provides information to the operator allowing them to actively monitor the environment, fight distraction and fatigue, detect potential obstructions, and determine the best course of action. This algorithm achieved 71 percent accuracy while running at 10 frames per second in FRA's CTIL. The approach and lessons learned from this program can be leveraged to transition the system for use in an operational locomotive. The existing CTIL-based system also can be used for human factors research to refine the human/machine interface to a point where it is most useful for engineers and conductors.		
14. SUBJECT TERMS Deep leaning, deep neural net, object detection, perception, cab technology integration lab, operator fatigue, operator distraction		15. NUMBER OF PAGES 66 16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

METRIC/ENGLISH CONVERSION FACTORS

ENGLISH TO METRIC

LENGTH (APPROXIMATE)

1 inch (in)	=	2.5 centimeters (cm)
1 foot (ft)	=	30 centimeters (cm)
1 yard (yd)	=	0.9 meter (m)
1 mile (mi)	=	1.6 kilometers (km)

AREA (APPROXIMATE)

1 square inch (sq in, in^2)	=	6.5 square centimeters (cm^2)
1 square foot (sq ft, ft^2)	=	0.09 square meter (m^2)
1 square yard (sq yd, yd^2)	=	0.8 square meter (m^2)
1 square mile (sq mi, mi^2)	=	2.6 square kilometers (km^2)
1 acre = 0.4 hectare (he)	=	4,000 square meters (m^2)

MASS - WEIGHT (APPROXIMATE)

1 ounce (oz)	=	28 grams (gm)
1 pound (lb)	=	0.45 kilogram (kg)
1 short ton = 2,000 pounds (lb)	=	0.9 tonne (t)

VOLUME (APPROXIMATE)

1 teaspoon (tsp)	=	5 milliliters (ml)
1 tablespoon (tbsp)	=	15 milliliters (ml)
1 fluid ounce (fl oz)	=	30 milliliters (ml)
1 cup (c)	=	0.24 liter (l)
1 pint (pt)	=	0.47 liter (l)
1 quart (qt)	=	0.96 liter (l)
1 gallon (gal)	=	3.8 liters (l)
1 cubic foot (cu ft, ft^3)	=	0.03 cubic meter (m^3)
1 cubic yard (cu yd, yd^3)	=	0.76 cubic meter (m^3)

TEMPERATURE (EXACT)

[(x-32)(5/9)] °F = y °C

METRIC TO ENGLISH

LENGTH (APPROXIMATE)

1 millimeter (mm)	=	0.04 inch (in)
1 centimeter (cm)	=	0.4 inch (in)
1 meter (m)	=	3.3 feet (ft)
1 meter (m)	=	1.1 yards (yd)
1 kilometer (km)	=	0.6 mile (mi)

AREA (APPROXIMATE)

1 square centimeter (cm^2)	=	0.16 square inch (sq in, in^2)
1 square meter (m^2)	=	1.2 square yards (sq yd, yd^2)
1 square kilometer (km^2)	=	0.4 square mile (sq mi, mi^2)
10,000 square meters (m^2)	=	1 hectare (ha) = 2.5 acres

MASS - WEIGHT (APPROXIMATE)

1 gram (gm)	=	0.036 ounce (oz)
1 kilogram (kg)	=	2.2 pounds (lb)
1 tonne (t)	=	1,000 kilograms (kg)
	=	1.1 short tons

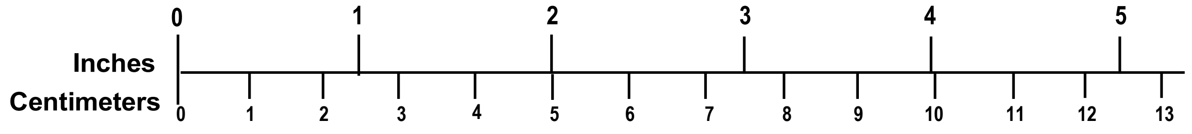
VOLUME (APPROXIMATE)

1 milliliter (ml)	=	0.03 fluid ounce (fl oz)
1 liter (l)	=	2.1 pints (pt)
1 liter (l)	=	1.06 quarts (qt)
1 liter (l)	=	0.26 gallon (gal)
1 cubic meter (m^3)	=	36 cubic feet (cu ft, ft^3)
1 cubic meter (m^3)	=	1.3 cubic yards (cu yd, yd^3)

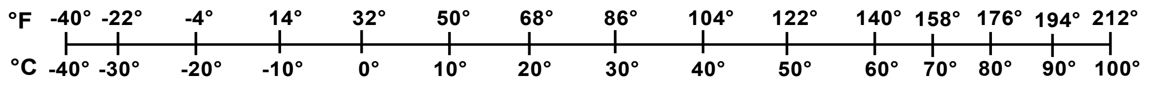
TEMPERATURE (EXACT)

[(9/5) y + 32] °C = x °F

QUICK INCH - CENTIMETER LENGTH CONVERSION



QUICK FAHRENHEIT - CELSIUS TEMPERATURE CONVERSIO



For more exact and or other conversion factors, see NIST Miscellaneous Publication 286, Units of Weights and Measures. Price $2.50 SD Catalog No. C13 10286

Updated 6/17/98

Contents

Illustrations

Tables

Executive Summary

The Federal Railroad Administration (FRA) sponsored a research team from Aurora Flight Sciences to develop the External Perception for Locomotives (ExP-L) system, which uses the FRA Cab Technology Integration Lab (CTIL) to demonstrate prototype object operator augmentation. The system can detect and localize objects of interest in and around the track, interpret the state of the detected objects to determine if there is any urgent and actionable information, and communicate this information to the operator. The research team selected four objective demonstrations to show the value of ExP-L, including 1) alerting the operator of a track obstruction by vehicles, 2) alerting the operator of potential signal violations, 3) alerting the operator of nonfunctional grade crossings, and 4) maintaining awareness of locomotive location through visual recognition of signs.

The team held three progressive ExP-L demonstrations. Demo I showcased the system's ability to detect and classify objects of interest in and around the track. The system detected cars, signals, railroad crossings, station signs, mile markers, and 2-mile warnings. Demo II incorporated state interpretation of the detected objects, including determining if a car was on the tracks, establishing the color of signals, reading station signs and mile markers, and evaluating if a railroad crossing was functioning properly. Demo III concluded this phase of development by integrating a heads-up display to communicate urgent and actionable information.

To detect objects of interest, the team developed an object detection algorithm to recognize important signs, signals, and objects (SSOs)[1] in the path of a moving train. Using a deep neural network (DNN) trained on 103,067 objects in 30,740 images, the algorithm achieved 71 percent per-frame accuracy while running near real-time at 10 frames per second in the CTIL. This accuracy is very promising considering two facts. First, most SSOs will be present in video for anywhere between 40 and 100 frames. Thus, 71 percent accuracy can be expected to correctly detect the object 29 to 71 times before it has passed. Second, this performance approaches state-of-the-art accuracy of algorithms trained on many hundreds of thousands of images. For example, the current top-performing, real-time object detector on the Microsoft COCO dataset (with over 330,000 images) achieves 78.5 percent accuracy. The research team's highest current accuracy levels using only 30,740 annotated images over 8 separate classes suggested that great improvements can be made by training on more images.

The team developed several more traditional computer vision techniques to interpret the state of these objects and determine if they represent an urgent and actionable response. These state interpretation algorithms include 1) evaluating if a car is on the tracks, 2) identifying signal colors, 3) reading station signs and mile markers, and 4) determining if a railroad crossing is operating correctly. Each of these traditional computer vision algorithms achieved greater than 90 percent accuracy, with most achieving 95 percent or greater when tested on a labeled ground truth data set. Researchers determined that incorrect state interpretations were reported due to erroneous results from the detection algorithm and were not a failure of the interpretation algorithm itself.

Object states that denote an urgent and actionable response from the operator are communicated using the CTIL's existing head-up display capabilities through easy-to-understand messages that

[1] Not to be confused with the term stop signal overruns (SSO) used in railroad operations.

appear on the engineer's "window." These situations include track authority violations, cars located on the tracks, and broken railroad crossings. The system will also use relevant CTIL webservice data to continuously display the name of the arriving station and the following station to help orient the operator.

The approach and lessons learned from this program can be leveraged to transition the system for use in an operational locomotive. The existing CTIL-based system also can be used for human factors research to refine the human/machine interface to a point where it is most useful for engineers and conductors. Studies evaluating such factors as whether operators find showing the next station to be helpful and other human factors-centric research can be conducted to help define and improve the requirements of the vision system for an operational locomotive.

This research was conducted to prove the capabilities of a vision-based operator augmentation system for locomotives. Further research can be conducted into integrating this type of system for an operational train in unconstrained outdoor environments. The research team believes that this phase of ExP-L represents an important first step toward perceptual enhancements in the cab to improve safety on our nations' railways.

1. Introduction

External Perception for Locomotives (ExP-L) uses deep learning to detect and identify objects external to a train and informs the operator of relevant information. Like adaptive cruise control in modern cars, the system helps the operator to actively monitor the environment, helping to fight distraction and fatigue while detecting potential obstructions and determining the best course of action.

To demonstrate the value of the system, the Federal Railroad Administration (FRA) sponsored a research team from Aurora Flight Sciences to develop a proof of concept system functioning in a simulated environment. The team used the Cab Technology Integration Lab (CTIL) at the Volpe National Transportation Systems Center for development and testing, as it is a comprehensive locomotive simulator that can depict a variety of environmental conditions and real-world situations. The low overhead required for operation, data collection, and testing makes the CTIL an ideal environment for proof of concept work.

The team trained a deep neural net to detect signs, signals, and objects in the CTIL simulation. The algorithm can detect cars, railroad crossings, station signs, mile markers, multiple types of signals, and split tracks. Additional computer vision techniques interpret the state of these objects. This information is used to inform the operator of potential objects on the track, track authority violations, signal drops, and other useful information using a head-up display (HUD) integrated into the CTIL. MIT's Human Systems Lab aided in designing the prototype human-machine interface (HMI) and provided the rail expertise necessary to establish the appropriate context for the work.

The key advancement of ExP-L beyond other similar train-based control systems such as Positive Train Control (PTC)[2] is twofold. First, ExP-L requires no infrastructure development between trains, since it relies only on sensors placed directly on the train engine. Second, it adds noncooperative capabilities to train situational awareness. This is most important in especially dangerous situations such as cars or pedestrians on the tracks or malfunctioned railroad crossing structures or track splits. Just as a car's cruise control is used to help the driver guide it safely to its destination, ExP-L provides vital supplemental information to train conductors and engineers to augment their operating experience.

1.1 Background

In August 2014, the collision of two Union Pacific trains in Hoxie, Arkansas, resulted in two deaths and over $5 million in damage. The root cause of this accident was a fatigued engineer who missed all three wayside signals instructing him to slow down, stop, and wait for the second train to move onto the adjacent track before proceeding. Similar accidents, classified as signal violations, often occur in both freight and shortline railroads across the U.S. FRA estimates that signal violations result in over 40 accidents per year, resulting in several deaths and millions of dollars in damage.[3] Rail conductors and engineers, especially extra board crews, often work

[2] Peters, J., and Frittelli, J. Positive Train Control (PTC): Overview and Policy Issues. *Congressional Research Service Report for Congress*. 30 July 2012.
[3] FRA Accident Database, Train Accidents By Type and Major Cause.

long, irregular shifts on short notice and travel long stretches of track with minimal changes in scenery, possibly causing them to experience fatigue and loss of situational awareness.

The ExP-L system uses machine vision and deep learning techniques to automatically detect and interpret railway signs, signals, and objects, and alert engineers and conductors about actionable information. ExP-L has the potential to help engineers and conductors maintain situational awareness, and to prevent deadly accidents such as occurred in Hoxie. This system also can be used in human factors research exploring the interaction between humans and autonomy.

1.2 Objectives

The research team sought to demonstrate how the ExP-L system uses a vision-based deep learning approach to help train operators improve safety by detecting and interpreting signs, signals, and objects (SSOs)[4] and alerting the operator to relevant information in a useful way.

1.3 Overall Approach

The ExP-L study was conducted in three phases, each comprised of three demonstrations. Phase I consisted of systems engineering and architecture selection as well as several trade studies to establish the feasibility of using deep learning for ExP-L. During this phase, researchers focused on defining system requirements and selecting the software architecture that would perform best for this application. During Phase II, researchers developed the software architecture to detect and classify signs, signals, and objects and interpret those objects in useful demonstration scenarios. Phase III focused on developing a prototype HMI to communicate that information to the operator. As the system matured, each milestone culminated in a demonstration, as described in Figure 1.

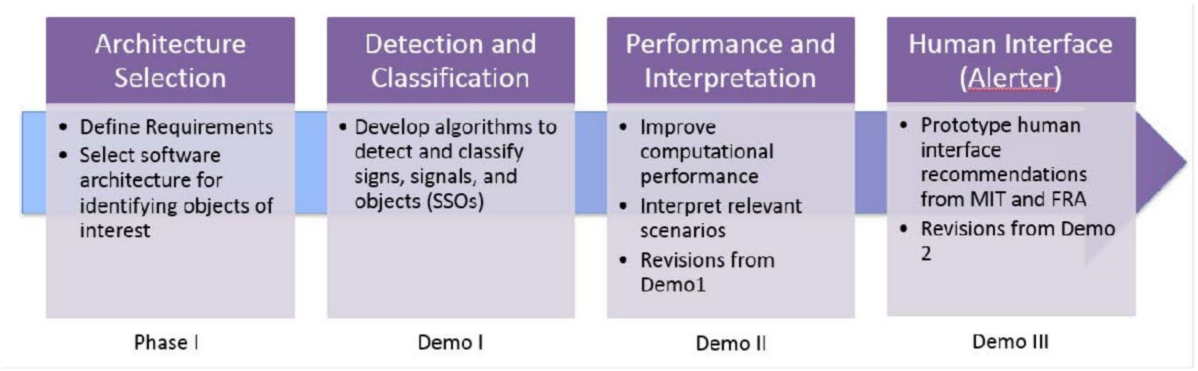

Architecture Selection
Define Requirements
Select software architecture for identifying objects of interest
Phase I
Detection and Classification
Develop algorithms to detect and classify signs, signals, and objects (SSOs)
Demo I
Performance and Interpretation
Improve computational performance
Interpret relevant scenarios
Revisions from Demo1
Demo II
Human Interface (Alerter)
Prototype human interface recommendations from MIT and FRA
Revisions from Demo 2
Demo III

Figure 1. Development Approach

1.4 Scope

This report details the ExP-L system at the conclusion of Phase III. It discusses the technical work completed to develop the technology, the current performance of the system, and the next steps needed to progress the technology toward a real-world application.

[4] Not to be confused with the term stop signal overruns (SSO) used in railroad operations.

1.5 Organization of the Report

Section 2 describes the ExP-L systems engineering. Section 3 provides details on the development of the algorithm for object detection. Section 4 describes the interpretation algorithms. Section 5 describes development of the HMI prototype. Section 6 discusses limitations of the simulation and the design. Section 7 presents possible next steps, and Section 8 presents the conclusions from this research.

Appendices A through G provide supplemental information related to this research.

2. Systems Engineering

2.1 System Objectives

This research comprised three separate milestones, represented as technology demonstrations, which build from each prior milestone's achievement in system design and functionality, representing a logical system development path. The research team developed the ExP-L system to:

1. Detect and localize objects of interest in and around the track, including cars, circular signals, elliptical signals, track splits, railroad crossings, station signs, mile markers, and 2-mile warnings
2. Interpret the state of the detected objects to determine if there is any urgent and actionable information
3. Communicate urgent and actionable information to the operator

Researchers outlined a set of seven objects for which the ExP-L system was required to detect, interpret, and communicate urgent and actionable information to achieve the four objectives discussed in Table 1 and Section 2.2. The team chose a deep neural net as the best approach to detect the seven objects. To interpret the objects, a variety of traditional computer vision techniques were evaluated and used. To communicate urgent and actionable information, researchers used CTIL's existing HUD functionality. A summary of the required objects of interest, the required interpretation, and an example output is shown in Table 1, which references the objectives listed in Section 2.2.

Table 1. ExP-L Required Objects of Interest

Object	Interpretation	Example Communication	Objective
Cars	On/Off Track	"CAR ON TRACK"	Objective 1
Circular Signals	Color	"REDUCE SPEED, PREPARE TO STOP"	Objective 2
Elliptical Signals	Color	"REDUCE SPEED, PREPARE TO STOP"	Objective 2
Railroad Crossing	Up/Down	"CROSSING NOT FUNCTIONING"	Objective 3
Station Signs	Text	"Arriving: HOLLYWOOD"	Objective 4
Mile Markers	Text	"16"	Objective 4
2-Mile Marker	N/A	"2-MILE WARNING"	Objective 4

To design a system that meets these requirements, the team performed a trade study of existing deep neural network architectures and selected YOLOv3 as the best candidate. Researchers examined the requirements needed to closely integrate with the CTIL environment and its web service, defined the overall architecture, and selected the required hardware.

2.2 Objective Demonstrations

ExP-L was designed to use CTIL to demonstrate the capabilities of a vision-based deep learning approach to aiding the operator by detecting, classifying, and interpreting SSOs and alerting the operator to relevant information in a useful way. To demonstrate the end use of this technology, a subset of potential vision-based scenarios requiring an engineer's alertness and intervention were selected:

Objective 1: Alert the operator of track obstructions by vehicles

Track obstruction is a common cause of accidents, resulting in enormous costs and potential loss of life. A system that can alert the operator to obstructions early enough for them to react would significantly reduce these accidents. Because the CTIL does not render images far enough out to fulfill this requirement, vehicles are detected as soon as they are visually distinct.

Objective 2: Alert the operator of potential signal violations

FRA estimates that signal violations cause over 40 accidents per year, resulting in several deaths and millions of dollars in damage.[5] PTC is a "cooperative" system that aims to solve this problem, relying on other trains and infrastructure (e.g., signs and signals) to continuously report their state so the system can understand if a locomotive is operating in violation of safety procedures. ExP-L solves this problem independent of infrastructure, allowing for implementation anywhere, regardless of infrastructure.

Objective 3: Alert the operator of potential maintenance issues on the track

Locomotive operators are instructed to inform dispatch of various maintenance-related issues in and around the track; however, these issues are often missed or go unreported. ExP-L can automatically report such occurrences using the existing perception system. Improperly functioning railroad crossings were selected as the representative case for this objective because CTIL renders crossings in both the up and down position.

Objective 4: Maintain awareness of location by reading signs and mile markers

Operators rely largely on memorizing the section of track on which they operate to understand their location and anticipate terrain changes, signals, and stations. ExP-L's perception-based interpretation of location adds another layer of awareness to the train location and state by reporting station signs.

To do so, many separate hardware, software, algorithmic development, and implementation items must be considered. For ExP-L to succeed, the system needs to seamlessly integrate into the CTIL data stream, both visually and by using available locomotive state data. This information would be processed though the ExP-L software system detecting objects such as cars, signal crossings, station signs, mile markers, and signals. This track state information, accompanied by situational awareness information, would output a range of detections, warnings, or advisory information for the locomotive engineer.

2.3 Deep Neural Network Architecture Trade Study

Deep learning was chosen as the main abstract method of enabling ExP-L to effectively and efficiently detect all objects required by the program though visual data. Figure 2 provides an example of the type of visual data objects. Deep learning has been used by virtually all self-driving car companies to detect, interpret, and classify objects from image sensor data, which makes a strong case for its use in ExP-L. Among the current most effective object detectors are

[5] FRA Accident Database, Train Accidents By Type and Major Cause.

YOLOv3[6], RetinaNet[7], and Faster RCNN.[8] After an extensive trade study comparing these three deep learning architectures (see Appendix A), researchers decided that YOLOv3 would be best-suited for ExP-L due to its high accuracy, generalizable training procedure, and near-real-time speed potential. The team implemented YOLOv3 in the Python programming language using the Keras package, which allowed more control over the training and testing aspects of development. The designing and training of the YOLOv3 implementation is discussed in this section.

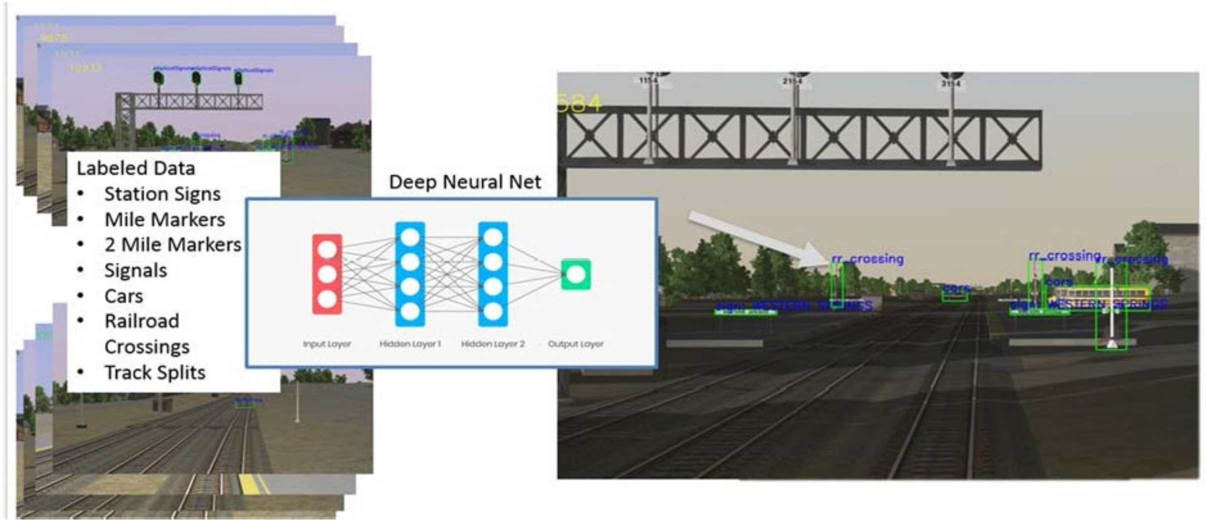


Figure 2. Applying Deep Learning to ExP-L

2.4 CTIL Requirements

CTIL has few requirements for interfacing. There are three main components that would require access for ExP-L. The first component is the CTIL video output to the engineer's main screen, which contains all visual information relating to the current scene. This is what the locomotive engineer sees when looking forward. To grab this image information, an HDMI splitter is required to split the image from the CTIL processor into a computer so the CTIL image can remain running unhindered. This split HDMI signal feeds into an HDMI capture card which acts as the video input device to the computer.

The second component is access to the CTIL's web service to capture data relevant to the state of the locomotive (e.g., speed, acceleration, state of hand controls, etc.). The CTIL runs on a separate internal network to pass information around the control console and locomotive console. The team needed the ability to join this network and retrieve webservice data over the wireless network and gather information using the web address API call to capture data.

Lastly, the CTIL needed to accept any information ExP-L wishes to pass on to the engineer. For example, ExP-L will warn engineers if they are approaching a signal indicating a limited approach section of track if they have not begun to reduce speed. ExP-L might warn the engineer

[6] Redmon, P., Farhadi, A. 2018. YOLOv3: An Incremental Improvement. Tech Report.

[7] Lin, T., Goyal, P., Girshick, R., He, K., Dollar, P. 2017. Focal Loss for Dense Object Detection. IEEE International Conference on Computer Vision.

[8] Ren, S., He, K., Girshick, R., Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Tech Report.

if a car is on the track ahead, or if grade crossing gates are broken and not lowered. This is done by way of a HUD image overlay on the engineer's main screen. Images to be displayed on this HUD require a RGB pure blue background (0, 0, 255) with images on top of the blue background. This image information is then fed into a video mixer where the blue is stripped out and the remaining images are overlaid onto the engineer's forward perspective display. The resolution of this input is 1920x1080 and fed via HDMI video connection.

2.5 CTIL Web Service

Access to the CTIL data stream is established by a web service running on HTTP 1.1, connecting via port 80. A data retrieval request is formatted by a specific web URL which returns the requested information. Any computer connected to the CTIL's network, either over wifi or Ethernet, has access to this data stream. The core URL request is: http://192.168.0.2/data.csv

This core request is augmented with the specific parameters being asked of the web service. The request is formatted using the standard URL parameter syntax, "?" to start the parameter list and "&" to separate parameters. Below is a list of the parameters:

- "v" – This required parameter allows the user to specify simulator variables to be received. Multiple variables can be listed, separated with commas. Any specified parameters that do not exist will be ignored.
- "r" – This optional parameter specifies the data rate (in hertz) with which the web service will respond. Its range is 1 to 20 Hz. If left unspecified, the default rate will be 20 Hz.
- "n" – This optional parameter specifies the number of responses with which the data stream will respond. If left unspecified, the web service will default to n = 1, returning only one result.
- "format=-h" – This is used to specify a data return without a header row.
- ":|" – Optionally specify delimiter type. In this example, the "|" delimiter is used. If left unspecified, the default value is a "," (comma). An "&" is not required to separate this variable but does need to be the final parameter specified. For example, to collect information on the throttle and dynamic brake handle at 5 Hz for 20 seconds without a header row, a "," would be used as the delimiter. The web URL request would be: http://192.168.0.2/data.csv?v=thtld,dbhld&n=100&r=5&format=-h:,

The URL library tool can be used to programmatically retrieve data via this web service, which transfers the data with the URL syntax. A user can request this data over the web service at any time to obtain the current state information about the CTIL.

2.6 Architecture Selection

ExP-L consists of both hardware and software implementations to produce the desired output. Many of these component selections were chosen based on a down select of commercial and academically available hardware and software components. Prior trade studies for a deep learning architecture can be found in Appendix A; computer and image capture trade studies can be found in Appendix C. The final architecture is discussed here.

This system takes in an image from the CTIL via an HDMI splitter so the CTIL can continue to function normally. At the same time, ExP-L is able to process the image, obtain web service data over the CTIL network, and determine if a car is on the tracks, the locomotive is in a signal violation, or if a crossing signal is broken. This information is then fed back to the CTIL via a HUD on the engineer's forward perspective display. See Figure 3 for an overview of the system flow.

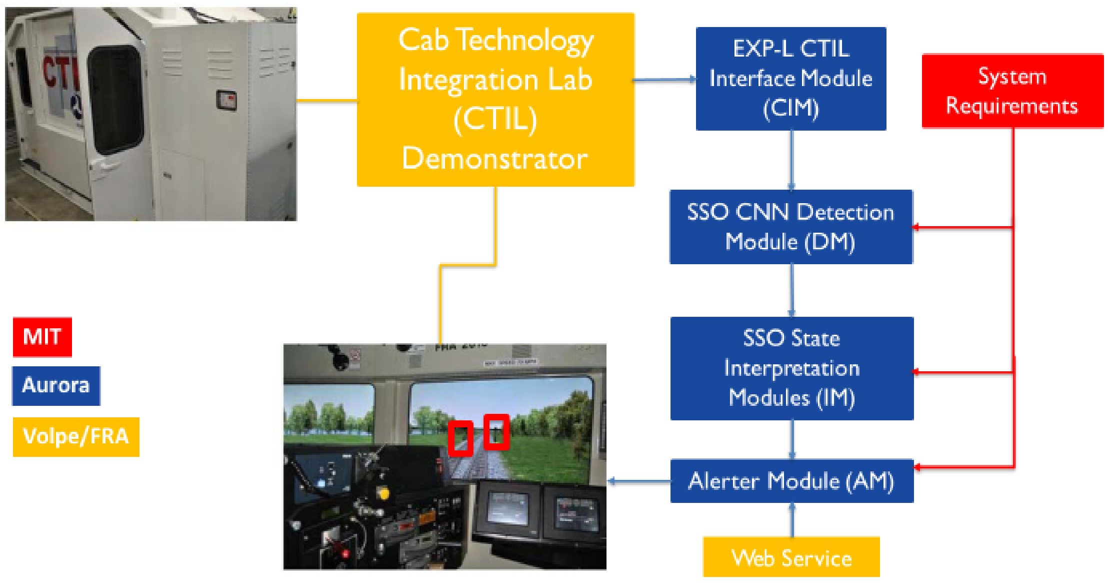

Cab Technology Integration Lab (CTIL) Demonstrator
EXP-L CTIL Interface Module (CIM)
System Requirements
SSO CNN Detection Module (DM)
SSO State Interpretation Modules (IM)
Alerter Module (AM)
Web Service
MIT
Aurora
Volpe/FRA

Figure 3. System Flow Diagram

1. CTIL Interface Module (CIM): This module uses an image capture card to stream imagery from the CTIL simulation into the EXP-L PC. The software written for the CIM uses OpenCV to capture the image and has inputs for image resolution.
2. SSO CNN Detection Module (DM): This module uses the convolutional neural network (CNN) YOLOv3 to detect the presence of trackside information. The structure and initial weights are open source; the AFS team has trained the network to detect trackside information found in the CTIL. The output of this module is the raw captured image and image bounding boxes of the detection results.
3. SSO State Interpretation Module (IM): There are several IM modules for EXP-L. Each module accepts the raw image and specific bounding box information from the DM. Each IM module crops the raw image based on the bounding box information and then runs a traditional computer vision algorithm on the cropped image. Examples include the signal color IM, which determines the color of the signal light, and the sign module, which determines the text displayed on detected signs.
4. Alerter Module (AM): This module receives input from all the IMs and the web service (i.e., locomotive state information) and then decides if the locomotive is operating within nominal range using the locomotive state information and results from the IMs. This module is also responsible with alerting the operator. This module will most likely hold GCOR and locomotive right of way rules.
5. Web Service: The CTIL transmits and receives data over its proprietary web service data stream. The web service allows data to be sent around the CTIL so any computer connected to the CTIL network has access to locomotive-related information. A few

examples of such information available are position, speed, acceleration, grade, positions of all cab controls, brake pipe pressures, cylinder pressures, ER pressures, ammeter, and fuel use variables. A list of these variables is included in Appendix B.

2.7 Hardware Selection

Due to the requirements of image capture and processing and based on the trade studies for neural network architectures and hardware research in Appendix A and Appendix C, a powerful, fast computer and quality image capture card are required. The image capture card should have native support to read in images using the OpenCV image processor and have a USB interface. Similarly, the core need of a computer resource is mainly driven by the neural network selection, which places high demand on a powerful graphics processing unit (GPU).

Simplistically, ExP-L's goal for integrating with the CTIL hardware would be to split the HDMI signal going out to the CTIL engineer's forward perspective display and capture that image while still displaying the original image through the splitter. The secondary output of the splitter would then be fed into the ExP-L computer via the HDMI to USB capture device. This representation is shown in Figure 4.

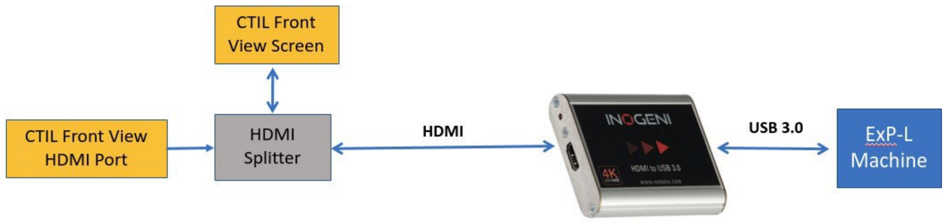


Figure 4. Splitting HDMI Out to CTIL Front Screen to ExP-L Machine

Image processing on a deep neural net can be computationally intensive, and a suitable framerate is required to capture useful information and communicate it to the operator. A target framerate of 4 frames per second was defined, requiring a high-end desktop with a powerful graphics card. The specific Alienware desktop mentioned in Appendix C was not selected. Instead, a Dell Precision 3630 with identical hardware specifications as the Alienware Aurora A7 was chosen. Of significance was ensuring the desktop would have a comparable and capable Nvidia GTX 1080 GPU. The system ultimately operated at close to 10 frames per second.

2.8 Outputting to CTIL

For testing, the ExP-L machine output its results, including bounding boxes and state information, onto a display connected to the machine, allowing for easy development and functionality verification. For demonstrations and later use in the CTIL, the ExP-L machine would have to output its detection results of annotations and alert information back to the CTIL. This was done in two phases. One method mirrored the CTIL's front main engineer's screen onto the left-most conductor screen, so the main display had the original CTIL output, while the screen to the left had the same original image with added annotations and information based on detections within the image.

The second mode of displaying the ExP-L output was to add this information on the CTIL's HUD. The CTIL HUD consists of a Roland V-1HD video switcher to combine two HDMI signals into one image which is then sent to the CTIL. For this HUD to function, the engineer's front view is combined with the output from the ExP-L machine. A chroma-key effect in the video switcher is used on the HUD machine's channel. This effect reads the signal coming in on its input (i.e., the ExP-L machine) channel and cuts out a specifically chosen color before combining it with the secondary (i.e., engineer's front view) channel. In collaboration with Volpe human factors engineers, it was recommended the HUD produce all its contents on a blue background that the switcher's chroma-key effect cuts out before combining the remainder with the engineer's front view. The resolution of the engineer's front view output is 1920x1080, and the output is set to be the same so the user can be certain any images or information drawn on the screen will appear on the CTIL front view with no distortion or issues.

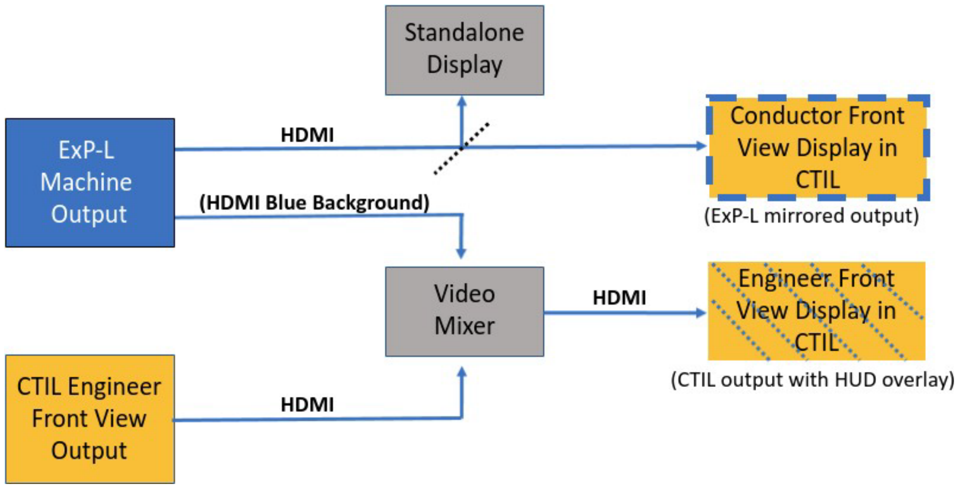

Standalone Display
ExP-L Machine Output
HDMI
(HDMI Blue Background)
Conductor Front View Display in CTIL
(ExP-L mirrored output)
Video Mixer
HDMI
Engineer Front View Display in CTIL
(CTIL output with HUD overlay)
CTIL Engineer Front View Output
HDMI

Figure 5. HUD Output Mixed with Engineer Screen and Standalone Monitoring Configuration

2.9 ExP-L Software Flow

The software process flow of ExP-L is shown in Figure 6. ExP-L obtains real-time image data of CTIL's main engineer screen via the HDMI to USB capture card mentioned Section 2.7. This imagery data is then fed into the custom Aurora-trained YOLOv3 neural network (see Section 3.1 for more information). This custom neural network detection module outputs bounding box information of the locations and labels for signals, signs, and objects found from within the input CTIL image.

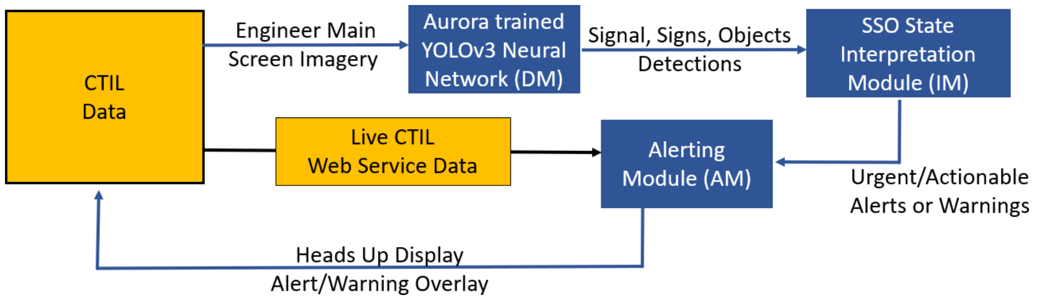

CTIL Data
Engineer Main Screen Imagery
Aurora trained YOLOv3 Neural Network (DM)
Signal, Signs, Objects Detections
SSO State Interpretation Module (IM)
Live CTIL Web Service Data
Alerting Module (AM)
Urgent/Actionable Alerts or Warnings
Heads Up Display Alert/Warning Overlay

Figure 6. ExP-L Software Flow Diagram

The bounding boxes obtained from the detector module are sent to the interpretation module, which discerns signals or objects seen within the CTIL image. If a given SSO is determined to contain urgent or actionable information, its data is fed into the altering module which fuses the relevant live web service data from the locomotive to be displayed onto the HUD, thus alerting the operator.

For example, if the locomotive approaches a red signal, ExP-L will detect the signals ahead and pass this information on to the IM. Once received, the IM interprets the location and detailed information (i.e., that the control signal is red). It is the job of the IM to understand that a red signal means the engineer has lost authority to operate on the current track and must stop. Once the interpretation of signals, signs, or objects is completed, the alerting module determines whether the information warrants an alert or warning to be sent over the HUD to alert and inform the operator.

3. Algorithm Development for Object Detection

ExP-L is required to not only detect which objects are present in each image but also to localize the objects. In computer vision, this task is known as *object detection.* The current leading method for visual object detection is *supervised deep learning,* a technique which detects objects by capturing complex relationships between pixels in an image. This technique comprises three parts: 1) data gathering to create a labeled dataset, 2) training the algorithm on the labeled dataset to learn to discriminate objects of interest, and 3) testing the algorithm on images that are novel but similar to the training set to quantify how well it can detect objects of interest. Researchers used current, state-of-the-art methods in supervised deep learning to accomplish this goal. The supervised deep learning method achieved 71.46 percent test accuracy in detecting objects of interest while running at 7.5 frames per second on average.

3.1 Training YOLOv3

The "out-of-the-box" Keras-YOLOv3 network was tested on the CTIL data and performed well on some objects (e.g., signals and cars) but failed to detect other CTIL objects (e.g., signs). This is because the out-of-the-box network was never trained to detect unique signs and other objects within the CTIL simulated environment. Therefore, the team had to re-train YOLOv3 to detect CTIL specific objects. Results of this test can be found in Appendix A.

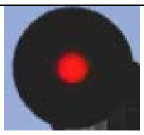
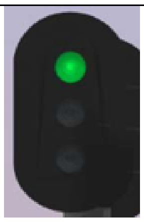
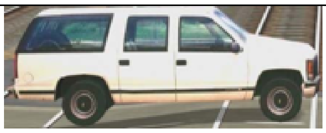
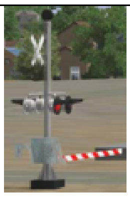
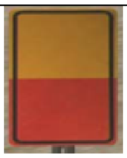
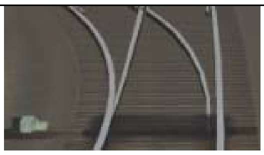
3.1.1 Data Collection

While the CTIL is quite realistic to the real-world operation of a locomotive in many respects, the imagery within the CTIL is not as representative of the real world as the imagery within the COCO dataset. For this reason, the team had to create its own dataset of CTIL images and annotations to train its own YOLOv3 network. The team created an image dataset using CTIL imagery to train the out-of-the-box Keras-YOLOv3 network on the open-source Common Objects in Context image dataset[9] (COCO) with over 80 object categories.

Proper network training requires a very large number of images. The more annotated images and training data, the better the network should perform when trained properly. The team's used between 4,500 and 20,000 individual images of each SSO (the goal was to capture approximately 5,000 images per class, but datasets such as COCO contain hundreds of thousands of labeled images). The team accomplished its own dataset collection by visiting the CTIL to record video of the main screen while the locomotive was under operation. This method allowed researchers to collect many images of each class of object very quickly. The team also collected some imagery during varying lighting and weather conditions to help give its dataset added variation. Table 2 summarizes the dataset, which was generated through annotation.

[9] Lin, T. et al. 2014. Microsoft COCO: Common Objects in Context. European Conference on Computer Vision.

Table 2. Annotation Classes and Number of Labeled Images per Class

Class	Example Image	Number of Annotations
Circular Signal		18,918
Elliptical Signal		20,174
Car		16,164
Railroad Crossing		19,168
Station Sign	CLARENDON HILLS	13,282
Mile Marker Sign		5,890
Two-mile Marker Sign		4,604
Track Split		4,866

3.1.2 Data Annotation

After preliminary research during Phase I (see Appendix D), the research team decided to annotate its relatively small collection of data on their own. At first, the team explored the possibility of using crowdsourcing marketplaces for image annotation (e.g., Mechanical Turk of Figure Eight) but found that it was too difficult to ensure the high-precision object annotation necessary for training accurate deep learning models. The red boxes in Figure 7 show the items the annotators were asked to label, and the white box shows how they labeled the data.

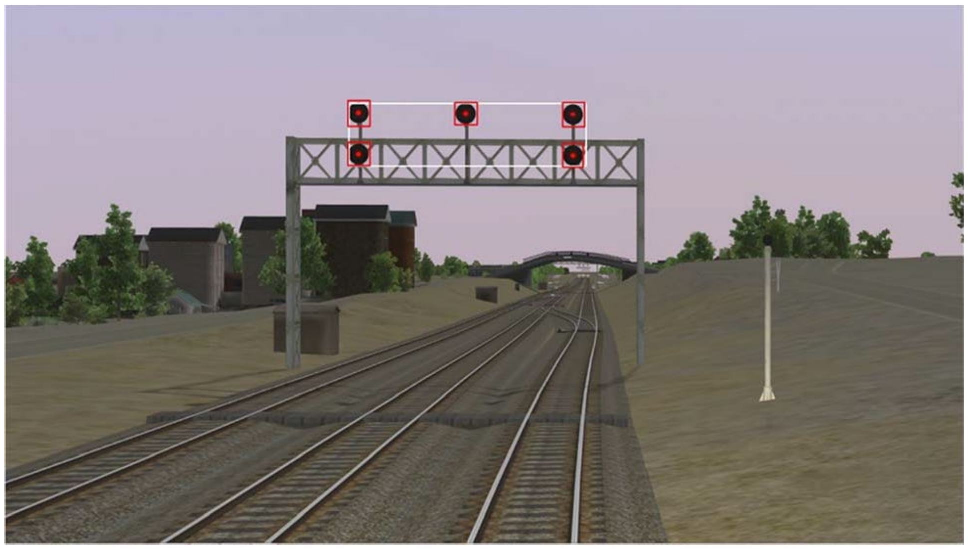


Figure 7. Correct and Incorrect Labeling of Signals

The team built a custom annotating software tool intended to make the task of annotating as efficient as possible. This tool reads a directory full of images and shows each image one at a time, allowing the user to draw bounding boxes around objects of interest. Since the labeled images were captured from a video stream, the objects' positions remained relatively constant across frames. Thus, this tool tracked objects from one frame to the next using a discriminative correlation filter.[10] In this way, the research team could annotate every image without having to redraw bounding boxes between each frame. Researchers ensured quality by adjusting the tracking algorithm when necessary. The team manually labeled over 100,000 objects in over 30,000 images. The complete count of all the objects labeled is shown in Figure 8.

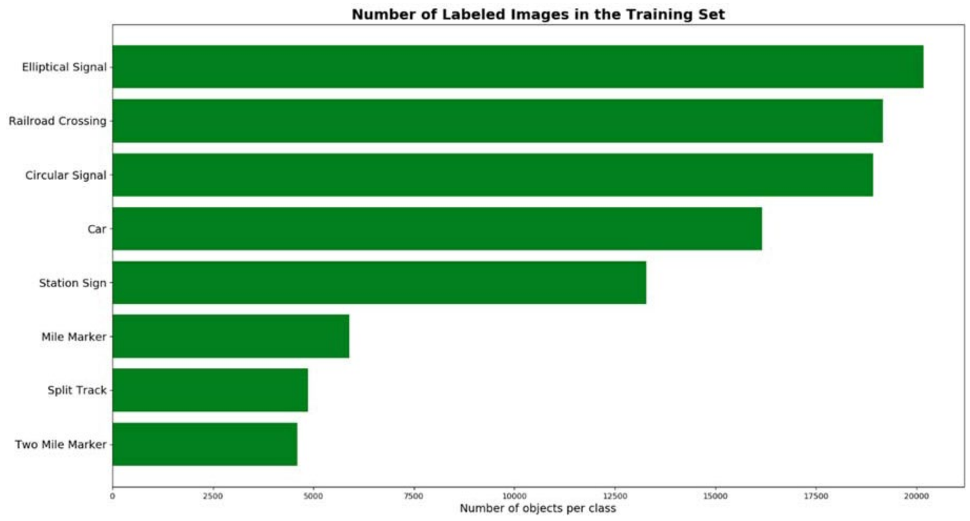


Figure 8. Number of Labeled Images per Class

[10] Lukezic, A., Vojir, T., Zajc, L., Matas, J., Kristan, M. 2018. Discriminative Correlation Filter with Channel and Spatial Reliability. International Journal of Computer Vision.

The team was also interested in the state of certain objects, and manually annotated these states. A complete explanation of which states were labeled and how those labels were used can be found in Section 4, Interpretation Algorithms.

3.1.3 Training Process

Deep learning is a machine learning method. This means that it "learn[s] from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."[11] In this case, it learns by observing the labeled dataset of objects and learning the discriminating features of the objects of interest.

A deep learning architecture for object detection is defined by a set of weights W, so that for any input image I, the transformation I*W generates an image which indicates the locations and identities of all objects of interest within the image I. This transformation is composed of a huge number of mathematical operations, usually consisting mainly of convolutions and activation functions. In the case of YOLOv3, the network performs 140.69 billion operations per iteration. Its accuracy is determined by a loss function which compares the predicted output $\overline{y} = I*W$ to the ground-truth labeled images y and quantifies the accuracy of the guess.

The network learns by comparing its predicted outputs $\overline{y}$ to the labeled ground-truth data y and adjusting its weights W in a way that will minimize the loss function, in a process known as backpropagation. The network begins its learning process by randomly assigning weight values. Thus, the predicted locations of the objects in the image will also be random. After this first step, the loss function will have a very high value. In the training phase, the network is programmed to change its weights after every prediction to minimize its loss function. However, since the weights are huge matrices composed of tens of thousands of values, this loss function cannot be minimized analytically using simple differentiation techniques. Instead, the network minimizes the loss function iteratively, using gradient-based techniques to decrease the value of the loss function after each step.

On the YOLOv3 architecture, this training process usually takes anywhere from a few days to a few weeks, depending on the processing power of the computer it is using to train. Since the exploration space of all possible weights values is so huge, it is computationally infeasible to solve for the global minimum of the loss function, so training continues until the value of the loss function is sufficiently low and its progress slows (See Figure 9).

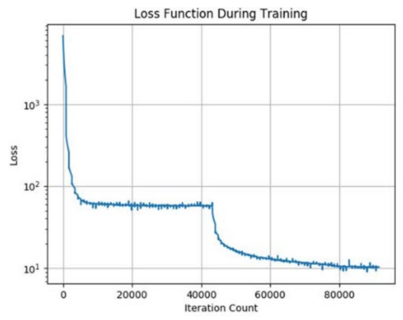


Figure 9. Loss Function of Training Algorithm over Time

[11] Mitchell, T. 1997. Machine Learning. McGraw-Hill Science/Engineering/Math, ISBN 0070428077.

Once training is complete, it is important to test the network on test images on which the network has not yet been trained to evaluate how well the network has learned to identify the objects.

3.2 Evaluation of Object Detection Algorithms

3.2.1 Detection Accuracy

When using a novel test set (i.e., one on which the network has not been trained) to evaluate the performance of the trained YOLOv3 network, ExP-L achieves 71.46 mean average precision (mAP) (See Figure 10). mAP is a commonly used metric in object detection research and refers to the average precision across all the classes in the test set. Since this value ranges between 0 and 100, it can intuitively be thought of as percent accuracy. For reference, YOLOv3's accuracy on its original training set, Microsoft's COCO, was 57.9.[12] The COCO dataset is a much more difficult dataset than the constrained simulated environment of the CTIL, which explains the increase in mAP that was observed in ExP-L's performance.

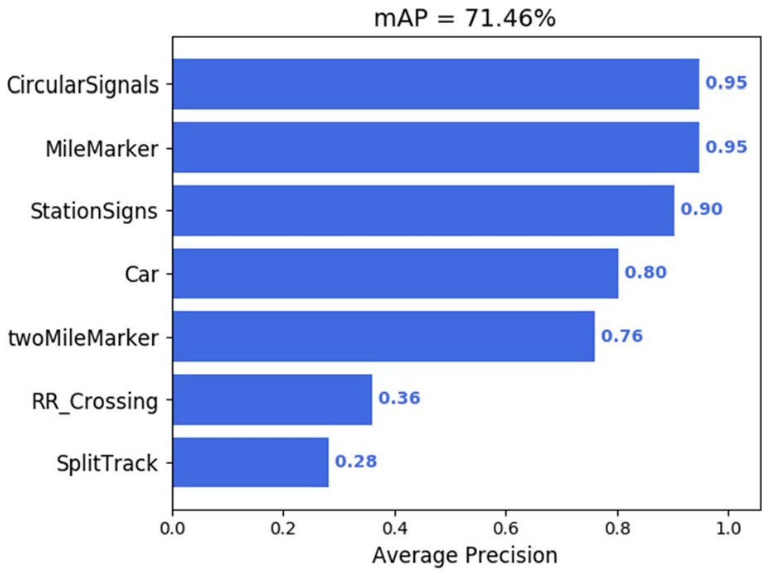


Figure 10. Detection Accuracy Over Training Data

It is difficult to visualize exactly what a mAP of 71.46 percent means in terms of qualitative performance. One problem is that objects appear in view for many frames, usually 40–200 frames, so if around 70 percent of those objects are correctly detected then it can be certain that the object will be correctly identified; however, if the network detects false positives, this can be misleading for the user. The second and most salient problem is that the accuracy reported can only be as correct as the accuracy of the ground-truth labels.

Most of the error in this metric is due not to human error (though there was a small amount of human error in labeling), but due to intentional design choices about which objects to label and how. Figure 11 shows how the system can be correct, but stylistic choices result in a "false positive" reported in the dataset (the cyan boxes show the ground-truth label of the objects present in the image, and the red boxes show false positives detected in the image). For example, researchers made the decision that they would only label cars that were on or near the track,

[12] Redmon, P., Farhadi, A. 2018. YOLOv3: An Incremental Improvement. Tech Report.

because those were the only vehicles that could be manually included in the CTIL. Thus, the car that the trained network detected near the left edge of the image was counted as a false positive, even though it was certainly correct that a car was present at that location. Also, the railroad crossing was recorded as a false positive because its overlap with the ground truth box was 37.58 percent, which is less than the required 50.0 percent to be considered a true positive. However, in looking at the image it appears that the red box makes a satisfactory prediction of the location of the railroad crossing. Overall, a qualitative study of the predicted bounding boxes appearing in a video showed confidence that this ExP-L visual system could be used to supplement an engineer's situational awareness in the train cab. See Figure 12 for more output image examples from ExP-L.

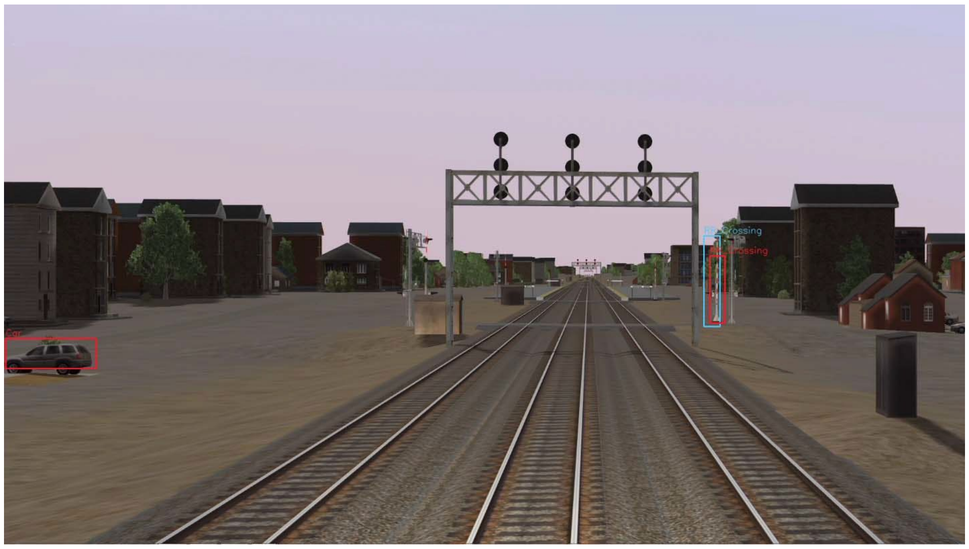
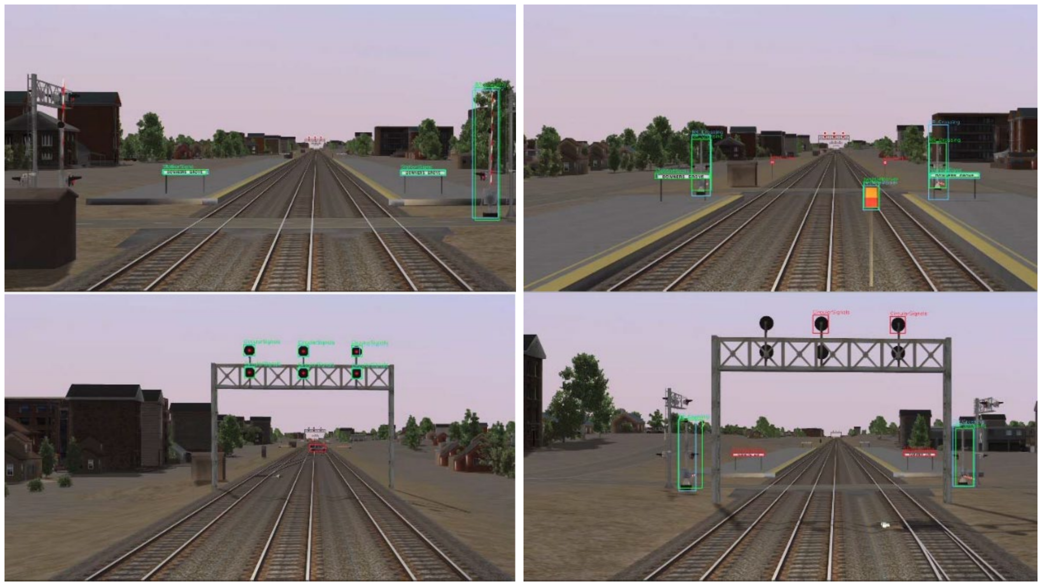


Figure 11. Disparity between Labeled and Predicted Object Locations



Key: ground truth - true positive - false positive

Figure 12. False Positive Example Object Detection Output

There are many improvements that could increase the overall accuracy of a deep neural network, but the most dependable and straightforward method of improving the results is almost always to increase the amount and quality of training data. However, in this case, a possible limitation for achieving higher accuracy was that the CTIL's generic imagery limited the overall increases to be gained from neural networks when fed additional data.

3.2.2 Detection Speed

Speeds of the trained algorithm differed depending on the central processing unit (CPU) or GPU on which the network ran. From a purely CPU-based approach, the YOLOv3 network trained on the team's annotated data (running on a Dell Precision 5810 desktop with an Intel Xeon E5-1630 v4 processor) processes the YOLOv3 detections at approximately 0.8 frames per second. In this scenario, the YOLOv3 neural network is heavily bottlenecked by the CPU due to lack of computational cores. In this case, GPU-accelerated computing excels due to the orders of magnitude more computational cores onboard GPUs. When running the same neural network on a Dell Precision 3630 with a Nvidia GTX 1080 GPU, the team recorded a mean processing speed of 7.5 frames per second (see Figure 13 below).

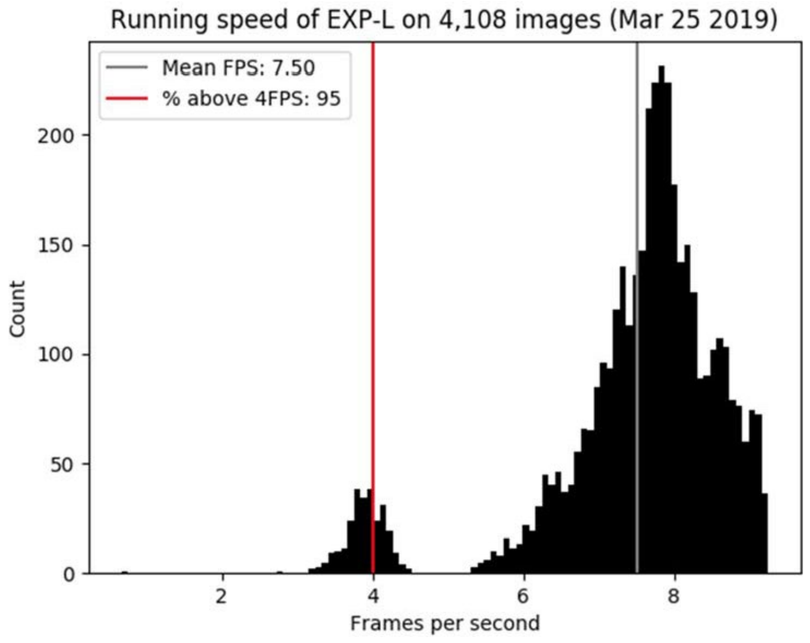


Figure 13. YOLO Inference Speed

4. Interpretation Algorithms

ExP-L is designed to help conductors and engineers in the cab to be more aware of their surroundings. The first step in this process is to detect SSOs within the image, but this is not sufficient. For a vision-based algorithm to be valuable, it must be able to not only detect objects within the image, but also interpret important information about those objects. For example, knowing the location of an overhead signal is meaningless without also knowing if that signal is displaying a red, yellow, or green light. The specific SSOs which required closer analysis were determined in a collaborative effort between the research team, FRA, and MIT researchers. The results are summarized in Table 3.

Table 3. Interpretation Algorithm Performance on Ground Truth Bounding Boxes

Class	State Information	Example State	Aggregated Accuracy (%)
Circular Signals	Signal color	"GREEN"	100%
Elliptical Signals	Signal color	"RED"	99.7%
Cars	Location relative to train tracks	"ON TRACK"	95.6%
Railroad Crossings	Gate state	"DOWN"	98.5%
Station Signs	Name of station	"CLARENDON HILLS"	95.0%
Mile Marker Signs	Mile number	"14"	90.0%
Track Splits	Split Direction	"LEFT"	99.4%

4.1 Interpreted States

Phase II of ExP-L was focused on first determining which objects need further attention and then developing algorithms to classify those objects into groups. Section 2.1 shows what further information, if any, was needed to provide actionable feedback to the CTIL user. The next sections describe the algorithms used to determine the states of each object (an example of object states is shown in Table 4). Each algorithm takes as input the region of interest (ROI) within the image, which is determined using the object detector. The accuracies of all the state classifiers are summarized above in Table 3. Since none of these state classifier methods rely on supervised learning, the reported accuracy reflects the accuracy achieved on all the available labeled images.

Note that the state interpretation algorithms were tested using ground-truth bounding boxes for the ROI instead of the predicted bounding boxes of the object detector. This is a fair way to examine these algorithms in isolation, but the perceived accuracy may be slightly lower than reported since interpretation accuracy will naturally suffer from bounding box mistakes from the object detector.

Table 4. Example Object States

Object State	Example Image	Example Output
Text on station signs		"WESTMONT"
Numbers on mile markers		"19"
State of signals		"YELLOW"

4.1.1 Signal Color

Circular and elliptical signals provide information about the required behavior of the train for certain lengths of track. Different configurations could signal to the engineer that they should speed up, slow down, prepare to slow, or switch tracks. It is very important for the engineer to be aware of the displayed color of the image. Railway governing officials are particularly concerned about situations in which the color of the signals changes while it is in view of the engineer. ExP-L's capability to continuously monitor the color of railway signals will be particularly valuable in these scenarios. See Figure 14 for a visual representation of this algorithm.

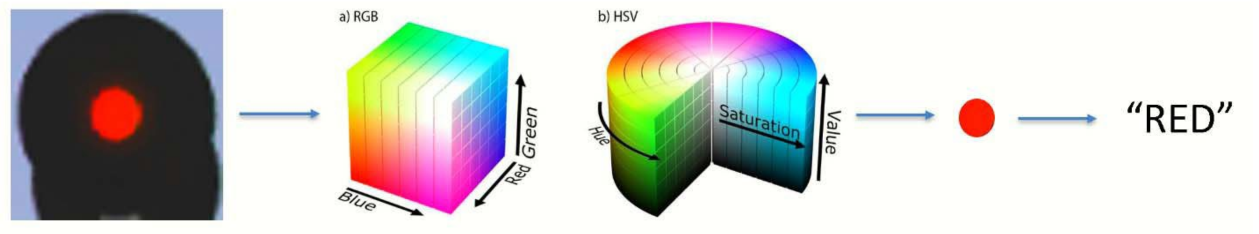


Figure 14. Signal Color Classification

It is assumed that at any given time, a signal (circular or elliptical) can be in one of four states: red, yellow, green, or off. ExP-L uses low-level color information to determine the signal's state, and then stores the color of the signal between captured frames. If any color configuration requires action from the engineer or conductor (e.g., red signals, yellow signals, or signals changing colors), an icon is displayed on the HUD.

After the location of the signal has been determined, ExP-L uses the hue, saturation, value (HSV) color space to determine its color. HSV color space is a very common tool for discerning color because it allows each pixel to be explicitly described by its hue and saturation, rather than as a mixture of red, green, and blue. In a cropped image around a signal, the only pixels of high saturation are those displaying the signal color. Thus, ExP-L extracts these pixels from the ROI, then determines the mode of the hues of these pixels. In practice, it was determined that the mode was a more accurate descriptor of color than the mean or median. If this "quintessential pixel" falls within the red, yellow, or green range of the color wheel, then the signal is classified to be of the corresponding color. If the pixel falls outside all these ranges, or if no sufficiently saturated pixels are present, then the signal is classified as "OFF." This method achieved near perfect accuracy for the team's labeled images (see Table 3). In practice, this method fails only when YOLOv3 outputs an erroneous signal bounding box.

4.1.2 Car on Track Detection

Whenever a train passes through a residential area, cars will be present in its surroundings. It is important for the engineer to be aware of the locations of cars near the train, but the only situation in which it is critical to be aware of a car's location is if the car is on the tracks ahead of the train. Thus, any perception supplement to the engineer must be able to detect when cars are located on the tracks ahead of the train.

ExP-L accomplishes this task in two steps. First, it must determine the location of its own tracks and the location of the car. Second, it must determine whether these objects intersect. Each of these steps is discussed in detail below.

The location of the car within the frame is determined from YOLOv3, reported as a bounding box around the car. The most important insight in determining the location of the tracks is to recognize that any image captured from a camera mounted rigidly on the front of the train will always display the closest parts of the track in the same location. If the camera is upright and centered on the front of the train, for example, then the tracks directly under the train will always be at the bottom of the frame, with the left-most track just left of center and the right-most track just right of center. Since a straight track ahead of the train engine will appear as a long, straight line in the image, ExP-L uses the Hough transform with a high accumulator threshold to find all long straight lines in the image. It then filters out all but the two lines which fall closest to the expected location of the track. It then extends these detected lines out to the horizon in the image (which remains constant in the CTIL environment), and labels these two lines as the left and right track. This method works very well under a range of environmental and lighting conditions and is much more robust to changing lighting conditions than color-based track detection methods. The most obvious drawback of this method is that it cannot detect curving tracks ahead of the train. This weakness was deemed acceptable for the proof-of-concept phase of ExP-L but will need to be improved as it moves more toward real-world application. For straight-ahead situations, this method has proven to be near-perfect, even in the darkest environments that the CTIL supports.

The second step in classifying a car as on-track or off-track is to determine whether the detected tracks intersect the detected car. The bounding areas of the tracks and the car are represented as convex quadrilaterals. For the car, this is the bounding rectangle detected by YOLOv3. For the tracks, this quadrilateral represents the area between the lines representing the left-most and right-most tracks. Since these quadrilaterals are convex, straightforward methods of intersection can be used. ExP-L uses Python's Shapely package to implement this method quickly. The overall accuracy of this method is 95.6 percent, as shown in Table 3.

4.1.3 Railroad Crossing Gate State

Crossing gates are frequently located at vehicle intersections with the train track. These gates close 15 to 20 seconds before the train arrives at the intersection, as a warning to car drivers and pedestrians to stay off the track. It is important for these gates to remain operational to ensure the safety of all users of the train track and the road near the track.

After detecting the location of railway crossing gates at intersections, ExP-L also reports the state of the gate as either "UP" or "DOWN." If the gates are functioning properly, it should always report that the gate is down as it is crossing the intersection. However, sometimes these gates malfunction and remain in the "up" position even when the train passes by. Since the gate bars open and close across the street, perpendicular to the train tracks and away from the view of the train, it will not be possible to detect the state of the gate from a train-mounted camera until the train is very close to the gate. Thus, ExP-L is not able to inform the engineer of a malfunctioning gate with enough forewarning to slow the train down before arriving at the intersection. However, the classification of gate state from a train-mounted camera is still valuable for at least two reasons. First, knowledge of malfunctioning gates allows the engineer to slow the train and take other appropriate actions needed. Second, while it may not be possible to recognize the state of the gate from a train-mounted camera, similar vision technology could be used in a crossing-mounted camera to provide constant surveillance of the crossing gate. It could then relay this information to oncoming trains. Thus, the engineer of the train could be notified of the malfunction with enough forewarning to slow down the train in anticipation of cars or pedestrians on the track.

ExP-L uses simple low-level information about the image to determine gate state (See Figure 15). After the whole crossing structure is cropped out of the image, the important piece of information is the red-and-white striped gate. The algorithm to determine whether this gate is up or down is shown in Figure 15. First, the ROI is converted to HSV color space. Second, the sections of the ROI containing strong red pixels are extracted. Third, morphological closing is used to close the gaps between the extracted red sections. After this step, ExP-L has extracted the crossing gate, and the width and height of this crossing gate are determined. If the width is greater than the height, that means that the crossing bar is mostly horizontal, and it is classified as "DOWN." Otherwise, it is classified as "UP." The per-frame accuracy of this method is 90.2 percent.

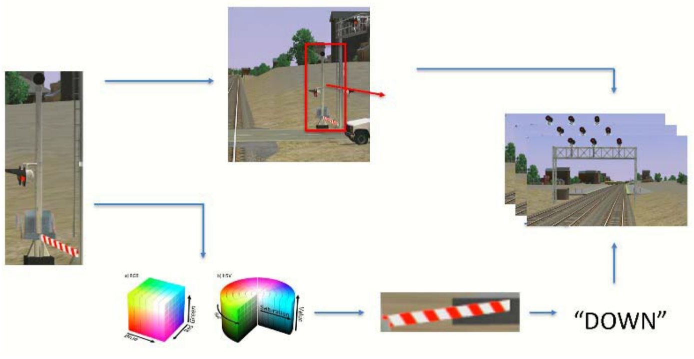


Figure 15. Method to Classify Rail Road Crossing Gate Position

ExP-L also uses a voting mechanism to aggregate votes across frames. By Condorcet's jury theorem and since the per-frame accuracy is well above 50 percent, a voting algorithm will perform far better than detecting the state of each frame.[13] The voting mechanism first uses a "decision variable" of zero in the first frame in which the railroad crossing is detected. Then, for every frame that the gate state is detected, the decision variable increases by one if the state is predicted to be "UP" and decreases by one is the state is predicted to be "DOWN." If the decision variable reaches a sufficiently negative value, then the gate is classified as "DOWN." If the decision variable reaches a sufficiently positive value, then it is classified as "UP." This voting mechanism provided a dramatic increase in gate state prediction accuracy, up to the 98.5 percent reported in Table 3.

4.1.4 Text Recognition

Mile markers and text on signs are decomposed using optical character recognition (OCR), which is a modern type of feature matching. The team used Google's Tesseract OCR library comprised of a long short-term memory recurrent neural network which has been pre-trained on over 100 languages and different font types to recognize lines and shapes quickly and accurately and then match those shapes to known characters. This algorithm was chosen based on research completed in Phase I (Appendix E). The team's implementation of this OCR algorithm functions

[13] Condorcet, Marquis de. "Essay on the Application of Analysis to the Probability of Majority Decisions." Print. 1785.

by its YOLOv3 network supplying the bounding box of the detected station sign or mile marker to the Tesseract algorithm (Figure 16).

However, even under the controlled environment of the CTIL simulator, OCR is very difficult, and Tesseract does not perform perfectly. Researchers implemented two methods to increase text recognition accuracy. First, they leveraged their knowledge of the expected text on the signs and signals to improve accuracy. In the case of mile markers, the team post-processed the predicted text output of Tesseract, filtering out all predictions that were not numbers or decimal points. In the case of station signs, the predicted text was compared against a known list of all possible station sign names between Chicago, IL, and Aurora, IL (i.e., the route of the CTIL). Using the Damerau-Levenshtein distance, ExP-L output the sign name most like the predicted text from Tesseract.

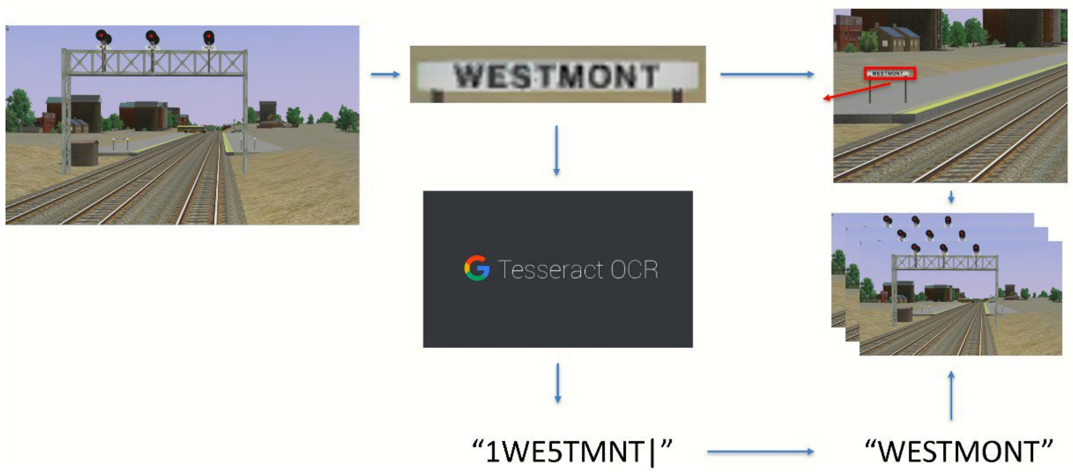


Figure 16. Station Sign Text Classification

Second, the team detected signs across multiple frames, and only output the text once the same text had been predicted for a given sign for five frames in a row. This helped reduce noisy guesses from Tesseract when the sign was still far away. These two methods greatly increased ExP- L's accuracy. Thus, the observed accuracy on a live video was higher than the per-frame accuracy reported in Table 3.

Using no vote aggregation, the per-frame accuracy of this method was 82.0 percent for station signs and 74.7 percent for mile markers, as shown in Table 3. ExP-L also included an option to return no prediction of the text on the sign or marker. On the test set of over 4,000 objects, ExP-L achieved 90.0 percent recall and 100.0 percent precision for mile markers, and 95.0 percent recall and 100 percent precision for station signs when using vote aggregation. ExP-L outputs its interpretation as soon as it is confident enough in its prediction. This means that although the optical character recognition does not work at the same distance that the team chose to annotate the objects, the train operator can trust in what ExP-L outputs as soon as it is confident enough in its prediction. Since station signs and mile markers do not convey time-sensitive or safety-critical information (as does a car on the tracks), the engineer does not need to be immediately informed of its text, making this a satisfactory result.

4.1.5 Track Split Direction

Railroad track splits allow multiple trains to travel along similar lengths of track at the same time. They are often located near train stations to allow trains to switch between nearby tracks for a short time and then diverge to different locations.

Engineers must be aware when the train is approaching track splits so that they can make sure that their train remains on the scheduled track. When more than two tracks are lined up in parallel, the engineer must know not only that a track split is coming up, but also to which adjacent track the split is going to transfer the train. Appendix F reports findings related to the visual detection of railroad tracks within images. ExP-L reports not only the location of track splits but also whether the track is splitting to the right or to the left (see Figure 17).

The method for determining the direction of the track split also depends on the location of the track. The Hough transform is used to localize the tracks within the frame, just as it was used in determining whether a car is on the tracks. Then, the location of these tracks is compared to the location of the bounding box around the track split. The key is to recognize that the bounding box around the track split contains part of the track that heads toward the new track. Thus, this bounding box will be offset to the left if the track splits to the left and to the right if the track splits to the right. Therefore, if the midpoint of the bounding box is left of the midpoint of the tracks, then the split is classified as "LEFT;" otherwise, it is classified as "RIGHT." If the bounding box does not intersect the tracks at all, then the split is classified as "OFF TRACK," meaning that the train is not currently on the track that will be split.

The team noted that the intended direction of travel for the train at the split would be a valuable piece of knowledge. Researchers explored the potential of providing this information as an element of ExP-L; however, upon close inspection of the track splits within the CTIL, it was discovered that they were not rendered with fine enough detail to discern this information. Further research into this application should be conducted on real-world, train-mounted imagery to determine the feasibility of reporting this state.

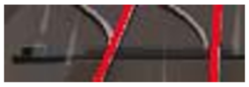
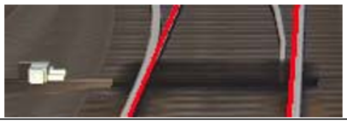
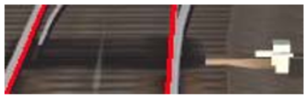
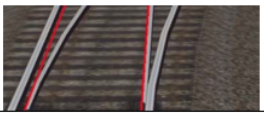
Example Track Split Images		
LEFT		
RIGHT		

Figure 17. Example of Track Split Images

5. Human-Machine Interface Prototype

After detecting important SSOs and determining their current state, the next step for ExP-L is to communicate urgent and actionable information in a meaningful way. Presentation of this information can be broken down into two steps. First, determine what information is urgent and actionable (i.e., what information merits being displayed to the engineer). Second, determine how that urgent and actionable information will be presented to the engineer in an HMI. Figure 18 shows an example of how human-machine interfaces can quickly become overwhelmingly complex.

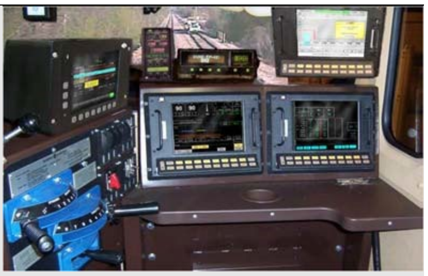
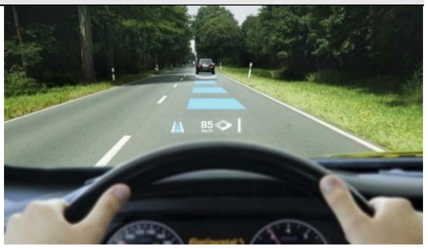
Cluttered HMI	
Simple HMI https://www.autoevolution.com/news/continental-shows-its-augmented-reality-head-up-display-for-2017-84307.html	

Figure 18. Human-Machine Interface Examples

5.1 Urgent and Actionable Information

Not all information about every object in every state will be urgent or actionable. For example, a green signal requires no action on the part of the engineer; an engineer notified by the HMI every time a green signal is detected would soon come to ignore that HMI. On the other hand, if an engineer were to fail to notice when that light changed to red, the proper notification of that red-light change could prevent a costly or deadly accident.

Recall from Section 2.1 that the SSOs which the team determined could convey urgent and actionable information are cars, circular and elliptical signals, station signs, mile markers, railroad crossings, split tracks, and two-mile markers. From these, the team determined four particularly important situations to display in the HMI. These situations are track authority violations, a car on the tracks, broken railroad crossings, and current and next station.

Depending on the severity of the situation, either an alert or warning will be issued. Warnings are reserved for less actionable information (e.g., informing an engineer of a malfunctioning railroad crossing) and are presented to the engineer as warning text displayed on a yellow background. Alerts are more severe and require the immediate attention of the engineer. Alerts for scenarios such as cars on the track or track authority violation are presented with the relevant alert text displayed on a red background to capture the engineer's attention.

These situations certainly do not constitute an exhaustive list of all the information of which the engineer should be aware. A list of that type would require a more in-depth study into the frequency and severity of particularly dangerous train situations. However, these scenarios can be extrapolated to other scenarios. Thus, these four situations serve as an appropriate prototype.

5.1.1 *Track Authority Violations*

There are some situations where the detected signal state above the current traveled track would indicate some type of track authority violation. This indication could mean one of many things. For instance, a flashing red signal over green could indicate to an engineer that a limited approach is in effect on the current track and they have been given authority to move onto a specified adjacent track. While there are hundreds of possible combinations of signal states across all track signals, ExP-L focused on interpreting the state of the current and (if applicable) adjacent track authority. Table 5, Table 6, and Table 7 below represents all signal states ExP-L can interpret and present to the engineer.

Table 5. Single Signal State Interpretation

State	Name
Green	none
Yellow	APPROACH
Red	STOP
Blinking Yellow	APPROACH MEDIUM
Blinking Red	RESTRICTING

Table 6. Double Signal State Interpretation

Top Signal	Bottom Signal	Name
Green	Red	None
Yellow	Flashing Green	APPROACH LIMITED
Yellow	Green	ADVANCE APPROACH
Flashing Yellow	Red	APPROACH MEDIUM
Yellow	Flashing Red	APPROACH RESTRICTING
Yellow	Red	APPROACH
Red	Green	DIVERGING CLEAR
Red	Flashing Yellow	DIVERGING APPROACH MEDIUM
Red	Yellow	DIVERGING APPROACH
Red	Flashing Red	RESTRICTING
Red	Red	STOP

Table 7. Signal State Change Interpretation

Original Signal State	New Signal State	Name
Green	Red	SIGNAL CHANGE TO RED
Red	Green	SIGNAL CHANGE TO GREEN
Green	Yellow	SIGNAL CHANGE TO YELLOW
Yellow	Green	SIGNAL CHANGE TO GREEN
Yellow	Red	SIGNAL CHANGE TO RED

ExP-L has been designed to extract significant signal state information rather than extract all possible signal states. This is done for simplicity so the engineer is not overwhelmed with extraneous information. For instance, an engineer does not need to be alerted to situations such as a green signal on their own, all other tracks, a limited clear approach on a secondary track, or if they have signal and right-of-way authority on their own current track. An example of an appropriate signal state alert can be seen in Figure 19.

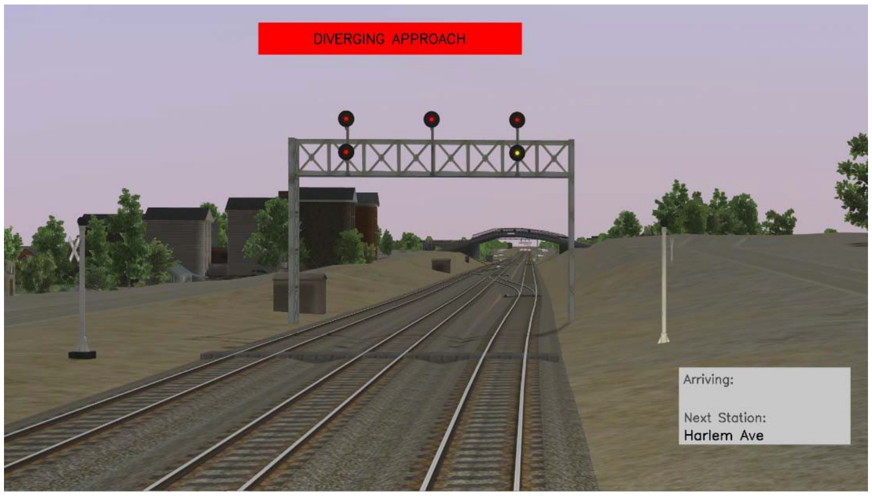


Figure 19. Red over Yellow Signal, Diverging Approach Alert

5.1.2 Car on the Track

As described in Section 4.1.2, cars, buses, or trucks determined within the bounds of the current track alert ExP-L to display a warning in the upper-left of the engineer's front view. An example of this warning can be seen in Figure 20. Alert information of a car on the track is ranked as the most important alert provided by ExP-L because of possibility of loss of life and significant damage.

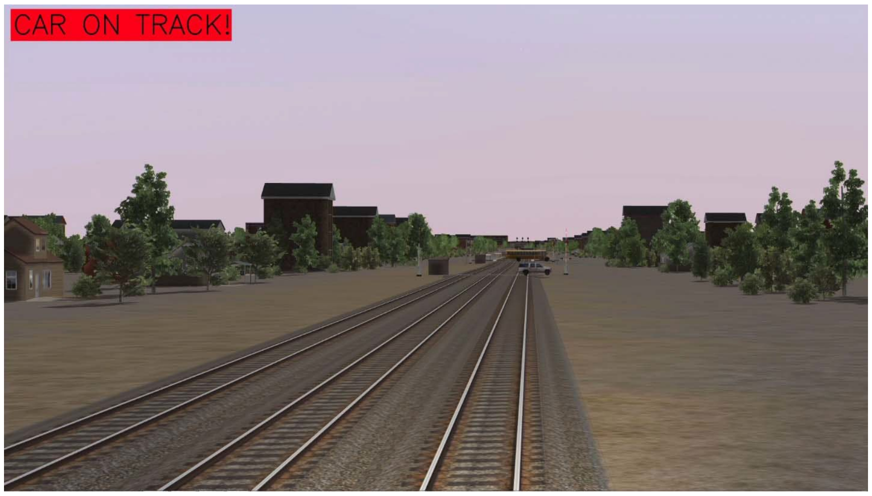


Figure 20. Car on Track Alert

5.1.3 Railroad Crossings

Section 4.1.3 explains how ExP-L determines whether a railroad crossing gate is in the up or down position when a vehicle is present. If determined to be up, an alert is presented to the engineer in the upper-right corner. An example of this warning can be seen in Figure 21.

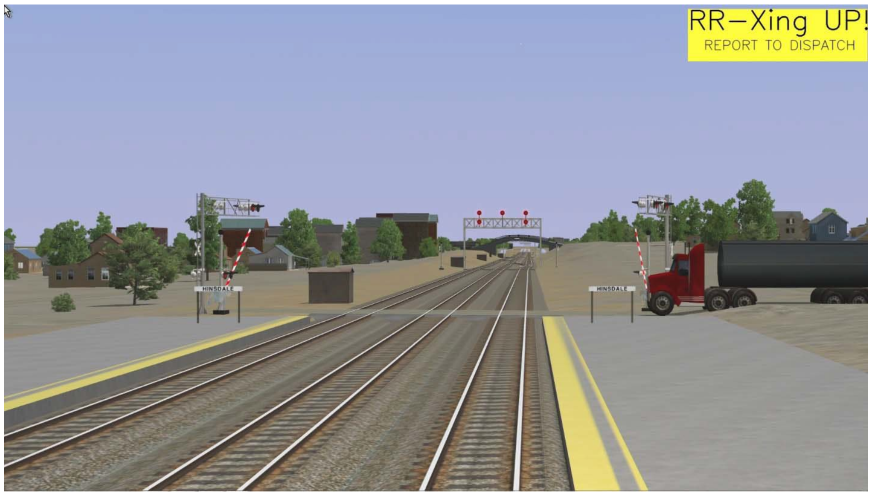


Figure 21. Railroad Crossing Warning

5.1.4 Current and Next Station

The final piece of information presented in the HUD is the next station along the track. This last display is less urgent than the other three, because the train's location does not necessarily constitute an emergency. However, engineers must be aware of their location along the track, and prepare for what is coming ahead. In fact, according to FRA certification standards, each engineer is only certified along certain sections of track[14], and engineers are required to memorize the segments of track upon which they will be operating the train. Thus, a supplemental system with the same capacity may be a useful addition.

Since the CTIL only simulates a known section of track between Aurora, IL, and Chicago, IL, all the possible signs that ExP-L will observe can be stored in the order that they will be observed. (In a real-world scenario, it is reasonable to assume that this information will be available, as well.) Thus, when ExP-L recognizes the text on a given sign, it can also output the name of the next sign that it will see. This information will allow the engineer to be aware of not only the current surroundings but also of the surroundings that are ahead.

[14] https://www.federalregister.gov/documents/2011/11/09/2011-28175/conductor-certification

6. Limitations

There remain some limitations in the ExP-L system that must be resolved before it can be used on a real train on real track. The most obvious limitation is that ExP-L has only been demonstrated in a simulated environment. While a simulated environment offers an opportunity to demonstrate a novel system with relatively low overhead, it carries with it a few inherent constraints that cannot be overcome without moving to a real-world environment. There are design limitations which could not be resolved under the current system requirements.

Some design limitations are a result of requirements evolving as the team learned lessons throughout the program. Because deep learning is heavily dependent on data and how the data is labeled, evolving requirements require different data. This program offered the opportunity to refine the requirements of the vision system with relatively low overhead and provide insight into efforts toward defining the requirements of a system installed in operational locomotives in future work. Lessons learned during this research will prove invaluable as the system is progressed into the real world.

6.1 Simulation Limitations

6.1.1 Distance to Time-Critical Objects

In the early phases of ExP-L, it was determined that eight SSOs in the CTIL environment would be of particular importance in providing situational awareness. Four of these eight SSOs – circular and elliptical signals, cars, and track splits – require a quick reaction from the train operator. Given that trains can require a mile or more to stop, a valid ExP-L system must be able to detect these objects at that distance. In preliminary experiments using the CTIL, it was shown that objects are not fully rendered graphically beyond approximately one half-mile down the track. Thus, no visual system would be able to detect or classify these objects in time for the operator to stop the car. The operational sensor payload must be designed to meet the long-distance detection requirement to identify at risk cars far enough down the track.

6.2 Design Limitations

6.2.1 Unlabeled Objects

The state of most of the SSOs the ExP-L system can detect can be determined by either analyzing their location or a prominent factor in that object. For example, the color of a circular signal is determined by looking at the light, the most prominent element of that object. However, track splits are more difficult. In this example, the most important factor an engineer would want to know about a track split is its setting (i.e., if the train will remain on its own track or be shifted to an adjoining track when it reaches the split). However, to discern the split's setting, a visual observer must look carefully for a small space between the current track and the beginning of the secondary track to see if the train will be diverted to the adjoining track (see Figure 22). The CTIL only renders this information when the track split is in the very near field. Additionally, the number of track split images used for training was one of the smallest datasets due to the limited number available in the CTIL. This led to the relatively low performance of the detection algorithm in comparison to other detection classes. With the CTIL being a relatively rough representation of track switches in the real world, it was decided that the requirement of determining switch position should be a requirement explored for the real-world operation, but not of this demonstration system.

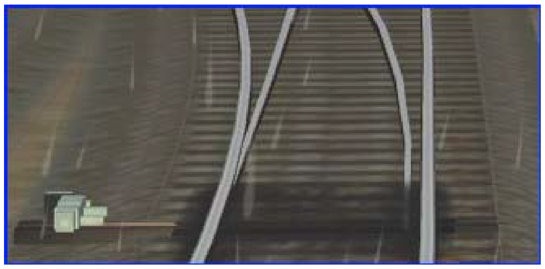


Figure 22. Close-up of Track Split Setting

A second object which is difficult to discern in the CTIL environment is the difference between a signal which is off and one which is simply facing the other direction. As shown in Figure 23, there are only subtle differences between a switched-off signal and a signal facing the other direction. The importance of signals in the "off" state was not identified during the data collection and labeling phase of the effort, so the training data set did not contain any signals in their off state. When the object detector was trained to detect signals, it "learned" to ignore any that were facing backwards, and ultimately extended that learning to signals that were off when tested in the new scenario. However, because all signals should always display either a red, yellow, or green light, a switched-off signal could indicate a potentially hazardous malfunction. Thus, ExP-L should be able to detect this, and future developments should incorporate this requirement into the design.

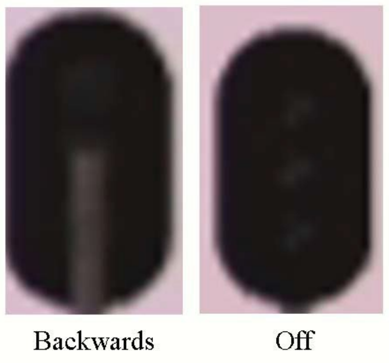


Figure 23. Backwards and Off Signals

6.2.2 *Determining Appropriate Signals*

ExP-L has been shown to be effective in alerting the train operator of urgent and actionable visual information regarding the environment. However, one situation which will require further development is its ability to discern which signals correspond to which track. In Figure 24, for example, ExP-L should be able to output that there are five possible tracks, and that the four red elliptical signals correspond to the two rightmost tracks. Currently, ExP-L just outputs recommendations based on the right-most column of signals. In this case, this would mean telling the train to come to a stop even though that red signal corresponds to a track that is two tracks over.

In the example shown in Figure 24, ExP-L is unaware of additional context required to determine that the signals shown do not apply to the track the train is currently on; specifically, the software must recognize that the signal stanchion is located on the right side of the track and therefore the two columns of signals refer to the two rightmost tracks. This requires an additional set of data be collected and labeled and the system retrained to recognize left versus right standing stanchions. Additionally, the system must be aware of how many tracks are running parallel to the current track. Currently, the algorithm detects the track by leveraging the fact that the relative position between the camera and the rails will always be the same; in other words, the track always extends away from the train from the same spot on the frame. Additional work should be completed to detect all the tracks in the field of view and discern which signal corresponds to that track; this requirement must be included in the operational development.

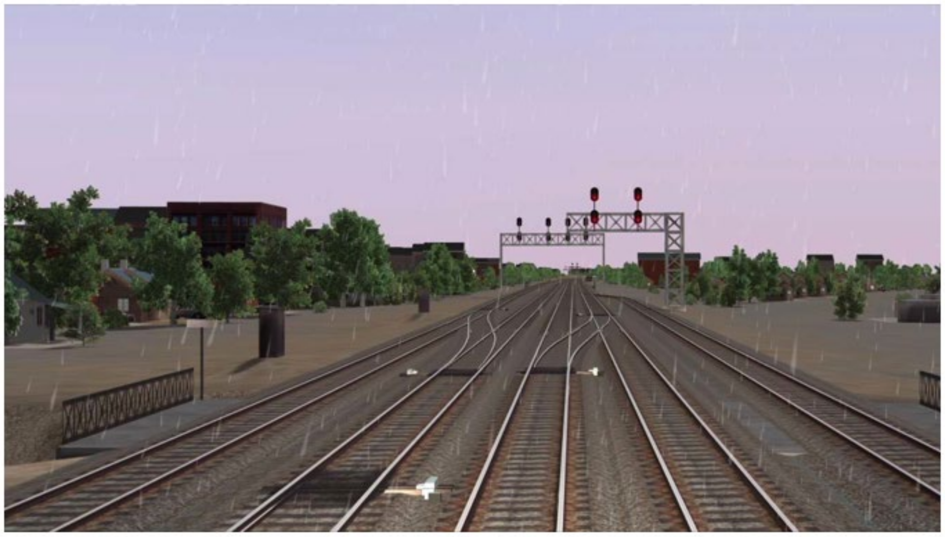


Figure 24. Multiple Signals over Multiple Tracks

7. Next Steps

Having successfully demonstrated the ExP-L system, the approach and lessons learned from this program can be leveraged to transition the system to an operational locomotive. In parallel, the existing CTIL-based system can be used for human factors research to refine HMI to a point where it is most useful for operators. This section considers what future work is required for each of these opportunities.

7.1 Implementation into Operational Locomotives

While the CTIL provides an excellent platform for the rapid development of a prototype perception system, operational locomotives present a much different set of requirements. Lessons learned from the CTIL demonstration will help define these requirements; however, to transition the system into rail operations, a detailed study establishing the top-level requirements of an operational external perception system would need to be conducted through discussions with industry representatives to examine the most useful capabilities and determine sensor requirements. Regulatory approval/certification requirements also must be examined early on to understand the constraints regulations put on the system.

The sensor payload can be selected from these requirements, considering items such as detection range, lighting conditions, and other operational requirements. A significant amount of labeled data from the sensor payload is required to train a neural net that can detect signs, signals, and objects with a high level of confidence. For example, Microsoft's Common Objects in Context contains millions of images with tens to hundreds of thousands of unique instances of each object of interest. A similar order of magnitude would be required from this dataset.

After the system is trained, developers must conduct extensive testing on representative test sets to determine the efficacy of the system and identify areas that may require improvement. As acceptable performance is reached, development of the logic that integrates with the HMI to provide contextual information could culminate in a test on board an operational locomotive.

7.2 Human-Machine Interface

The development of an appropriate HMI similarly starts with defining high-level requirements. These requirements will overlap significantly with the development of the sensing system. However, requirements that incorporate human factors can be more difficult to define. Using the existing capabilities in the CTIL, studies with engineers to see how they interact with various revisions of the HMI can be conducted to help establish these requirements and the initial design. The information that is found to be most useful will flow back to the requirements of the operational system detection payload requirements.

The MIT Human Systems Laboratory has provided initial thoughts on the various factors that would go into designing a production level HMI for the vision system. This is included in Appendix G.

8. Conclusion

A research team from Aurora created an end-to-end perception system capable of 1) capturing synthetic images from CTIL, 2) detecting, locating and interpreting objects of interest within the image, and 3) communicating the resulting urgent and actionable information to the engineer in an accurate and efficient manner. This technology has only been demonstrated in a simulated environment, but similar efforts could be used to extend these capabilities into rail operations. The research team anticipates that a visual system of this kind will be integral to any system seeking to enhance the decision-making capabilities of an engineer.

The biggest contribution of ExP-L is its noncooperative perception ability. Other operator-supplementing systems, such as PTC, connect the train to a network of other trains and signals, but ExP-L actively monitors its immediate surroundings. This assists the train engineer in reacting to unforeseen and potentially hazardous situations, such as a car on the track. Any system which seeks to support human perception capability must be aware of both planned and unplanned scenarios.

One major result of this research was a clearer idea of which signs, signals, and objects present the most urgent and actionable information to the engineer in a noncooperative environment. The objects determined to be most important to detect were cars, station signs, mile markers, circular signals, elliptical signals, railroad crossings, and track splits. Current state-of-the-art computer vision techniques require many labeled objects of interest to be successful, so this knowledge will be valuable in creating labeled datasets when the capabilities of this system are generalized to the real world.

This technology represents a valid proof-of-concept for the ability of visual perception to aid engineer situational awareness. Localization accuracies upwards of 95 percent for many important objects and interpretation accuracies upwards of 99 percent prove that a computer vision system can provide valuable and reliable information to human train operators. Qualitative reviews of ExP-L running on live CTIL video demonstrate its usefulness in noticing small but important changes, such as a signal changing color while in view.

The biggest obstacle foreseen for ExP-L moving forward is the difficulty in generating real-world, labeled datasets large enough to create a reliable and robust computer vision system. The CTIL provides a great starting point for qualifying visual perception capabilities, but the real world is much less constrained than the images that can be generated in a simulated environment. Weather, lighting conditions, and human activity are examples of variations that ExP-L must be trained to handle before it can be trusted to provide a visual supplement to train operators. Another important factor in all areas of autonomy is the problem of trust. How can a system be created that is reliable and efficient enough so that the human operators in the cabin will trust the displays that appear in the HUD? This is a difficult question that can be answered as researchers from the fields of artificial intelligence and human factors collaborate to create trustworthy perception systems on trains. The research team believes that ExP-L represents an important first step toward perceptual enhancements in the cab to improve the safety of American railways.

Appendix A.
Deep Neural Network Selection

Deep Neural Network Selection

For this work, the research team proposed to use a deep learning approach for object detection. Since this is an initial prototyping phase, major emphasis was on using commercial off-the-shelf (COTS) or open source products to show system feasibility with the eventual goal of 80 percent threshold, 90 percent objective detection accuracy. For deep neural network (DNN) selection, the team conducted trade studies of start-of-the-art deep learning algorithms and an open source, pre-trained network was selected. Transfer learning techniques were then used to fine-tune the pre-trained network to detect the SSOs found in the CTIL imagery. This appendix details the selection and testing process for the DNN that will be used for ExP-L.

Deep Neural Network Framework Selection

Several deep learning frameworks exist for developing and training neural network architectures. Some are supported by large companies such as Google (officially or via funding), while others are maintained by the open source community. For ExP-L, the chosen framework must be easy to use and implement, state-of-the-art, and include support for Python and NVIDIA GPUs. A thorough review of all the available frameworks (Table 8) evaluated against a variety of criteria helped determine that Keras was the best framework to use for ExP-L.

Keras is easy to use, sits on top of TensorFlow, and supports Python. Keras supports recurrent neural networks (RNNs), convolutional neural networks (CNNs), and parallel execution. Keras is also heavily supported online, with implementations of popular DNNs readily available for use.

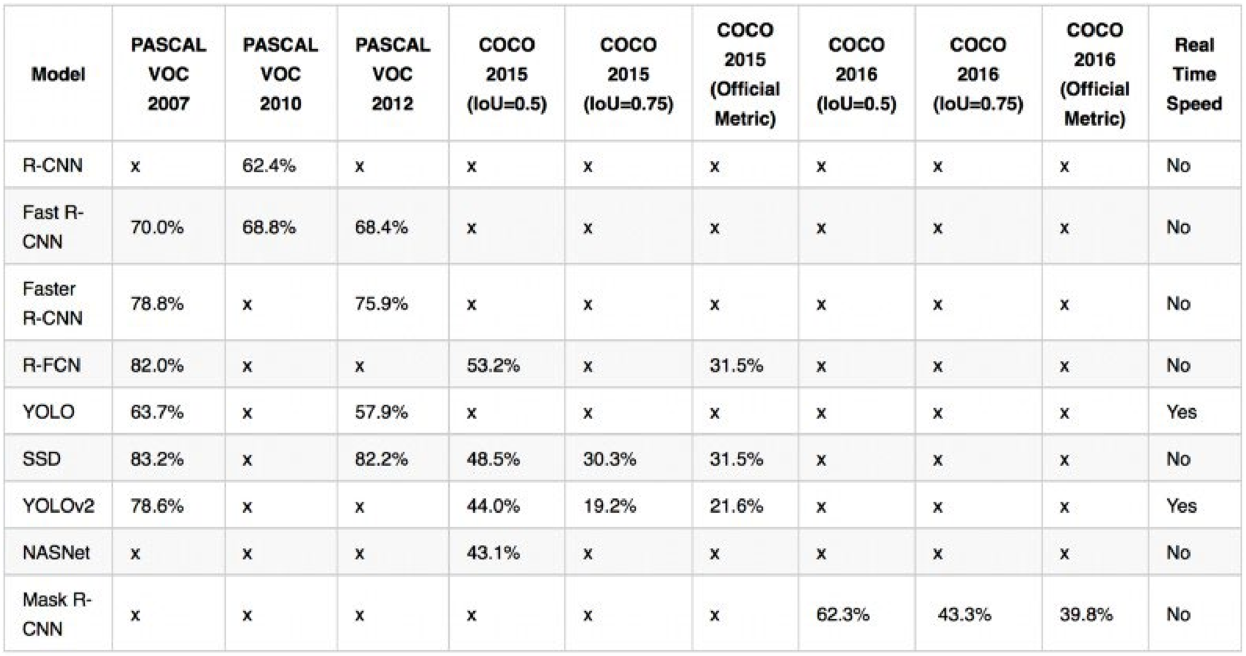
Table 8. Deep Learning Frameworks Available via Open Source

roNNie.ai	Darknet	Intel Math Kernel Library	Neural Designer	TensorFlow
BigDL	Dlib	Keras	OpenNN	Theano
Caffe	DataMelt	MATLAB + Neural Network Toolbox	PlaidML	Torch
Deeplearning4j	DyNet	Microsoft Cognitive Toolkit	PyTorch	Wolfram Mathematica
Chainer	Intel Data Analytics Acceleration Library	Apache MXNet	Apache SINGA	VerAI

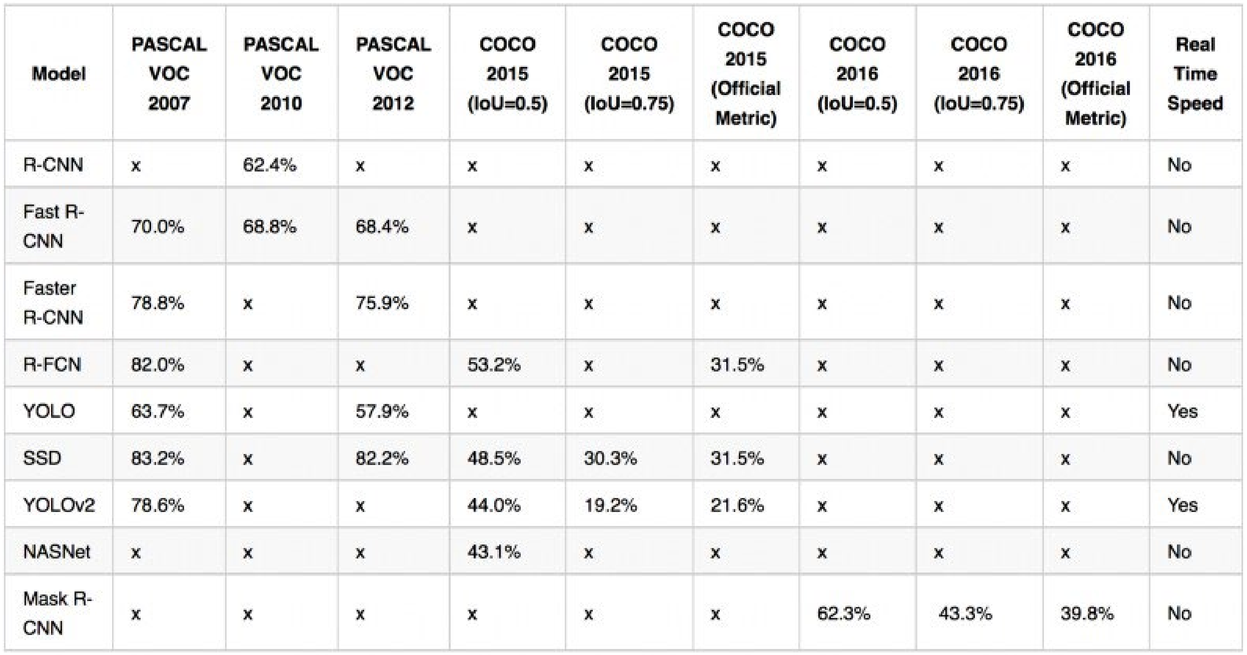
Deep Neural Network Trade Study

The team's approach to this trade study was to survey several state-of-the-art DNNs that have been evaluated against large datasets, and then down select several networks for evaluation against CTIL imagery. The final down select network was based on detection accuracy, real-time speed, and availability of transfer learning implementations. The DNNs shown were evaluated against large datasets such as COCO and PASCAL Visual Object Classification (VOC). YOLO stood out as an attractive option due to its high mAP, which is a measure of accuracy, and real time speed. Faster RCNN10 and Mask RCNN also stood out for their performance, although real time implementations were not readily available. RetinaNet is not shown here, but it was also down selected due to its performance.

Table 9. mAP Evaluation of DNNs against Large Datasets



Model	PASCAL VOC 2007	PASCAL VOC 2010	PASCAL VOC 2012	COCO 2015 (IoU=0.5)	COCO 2015 (IoU=0.75)	COCO 2015 (Official Metric)	COCO 2016 (IoU=0.5)	COCO 2016 (IoU=0.75)	COCO 2016 (Official Metric)	Real Time Speed
R-CNN	x	62.4%	x	x	x	x	x	x	x	No
Fast R-CNN	70.0%	68.8%	68.4%	x	x	x	x	x	x	No
Faster R-CNN	78.8%	x	75.9%	x	x	x	x	x	x	No
R-FCN	82.0%	x	x	53.2%	x	31.5%	x	x	x	No
YOLO	63.7%	x	57.9%	x	x	x	x	x	x	Yes
SSD	83.2%	x	82.2%	48.5%	30.3%	31.5%	x	x	x	No
YOLOv2	78.6%	x	x	44.0%	19.2%	21.6%	x	x	x	Yes
NASNet	x	x	x	43.1%	x	x	x	x	x	No
Mask R-CNN	x	x	x	x	x	x	62.3%	43.3%	39.8%	No



Model	PASCAL VOC 2007	PASCAL VOC 2010	PASCAL VOC 2012	COCO 2015 (IoU=0.5)	COCO 2015 (IoU=0.75)	COCO 2015 (Official Metric)	COCO 2016 (IoU=0.5)	COCO 2016 (IoU=0.75)	COCO 2016 (Official Metric)	Real Time Speed
R-CNN	x	62.4%	x	x	x	x	x	x	x	No
Fast R-CNN	70.0%	68.8%	68.4%	x	x	x	x	x	x	No
Faster R-CNN	78.8%	x	75.9%	x	x	x	x	x	x	No
R-FCN	82.0%	x	x	53.2%	x	31.5%	x	x	x	No
YOLO	63.7%	x	57.9%	x	x	x	x	x	x	Yes
SSD	83.2%	x	82.2%	48.5%	30.3%	31.5%	x	x	x	No
YOLOv2	78.6%	x	x	44.0%	19.2%	21.6%	x	x	x	Yes
NASNet	x	x	x	43.1%	x	x	x	x	x	No
Mask R-CNN	x	x	x	x	x	x	62.3%	43.3%	39.8%	No

Table 10 is a summary of down selected networks with high-rated mAP or accuracy. These DNNs were then evaluated based on CPU and/or GPU processing time, availability of implementation on the GPU, and ease of use. The dataset used for this evaluation was gathered in the CTIL. At the time of this trade study, the latest version of YOLO was YOLOv3, so this was the version used. Based on processing or inference time and ease of use, YOLOv3 and RetinaNet were selected as the top two pre-trained networks.

Table 10. Pre-Trained DNNs Considered for ExP-L

DNN	Creator	Open/Closed Source	Year created	Framework	CPU/GPU Processing Time	Ease of use/availability of software
YOLOv3	Joseph Redmon, Ali Farhadi	Open Source	2018	Darknet, PyTorch, Keras, and Tensorflow	1 sec/0.3 5 sec	Very easy. Keras implementation incl. transfer learning available.
RetinaNet	FAIR (Facebook AI Research Group)	Open Source	2018	Keras, PyTorch	9 sec/not available	Very easy. Keras implementation available. GPU implementation not available.
Mask R-CNN	Microsoft	Open Source	2017	Keras, PyTorch	14 sec/not available	Very easy. Keras implementation available. GPU implementation not available.
Faster R-CNN	FAIR (Facebook AI Research Group)	Open Source	2015	Keras, PyTorch	15 sec/not available	Very easy. Keras implementation available. GPU implementation not available.

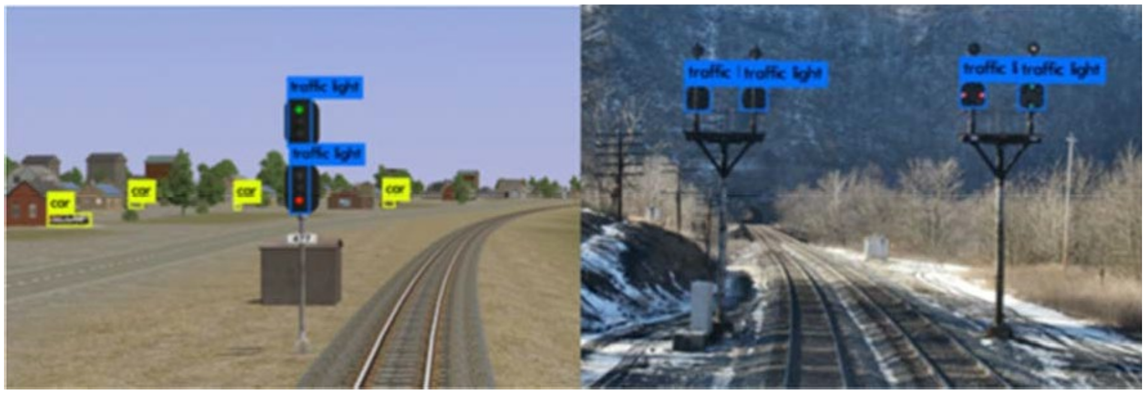


Figure 25. YOLOv3 Detections on CTIL and Real-Life Imagery

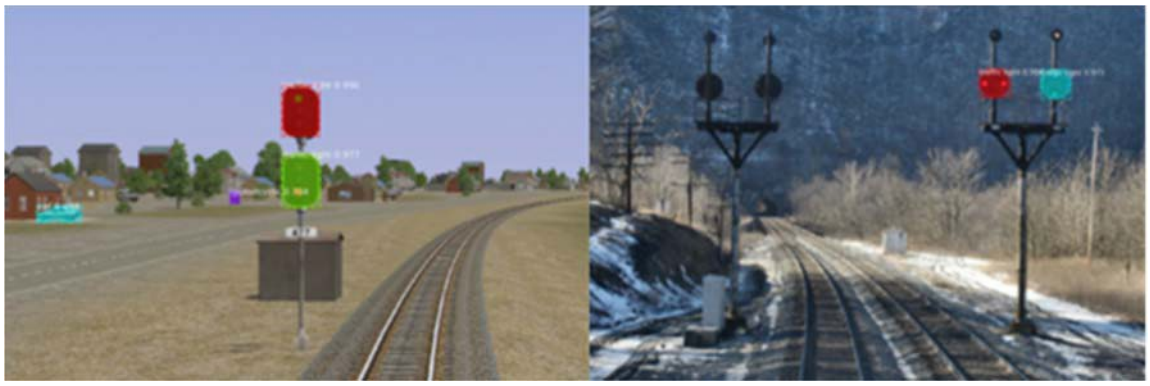


Figure 26. Mask R-CNN Detections on CTIL and Real-Life Imagery

To further evaluate performance of YOLOv3 over RetinaNet, a precision/recall curve was generated on their detection results using CTIL imagery. For neural networks, precision and recall[15] are two closely related, but slightly different, quantitative metrics that determine (in general terms) the accuracy and precision of the network on a given evaluation dataset. This is somewhat analogous to accuracy and precision calculations for making measurements. Precision is a measure of how likely every prediction is to be correct. Recall is a measure of how many of the ground-truth objects to be detected are detected by the neural network. Although it can be easy to mix up precision and recall, it is helpful to evaluate how both are affected as the probability threshold of the network is adjusted. Every detection that a network generates has an associated class probability value directly indicating how confident the network is in the prediction.

The probability threshold of a neural network is the confidence value below which a prediction is eliminated. It can be easily reasoned that a high confidence threshold would mean that a network is more selective in filtering out detections, thus eliminating low confidence detections that might have been right in favor of detections that are more certain to be right. This means that precision increases while recall decreases as confidence threshold increases. Conversely, a low confidence value indicates being less selective in filtering detections, thus accepting lower confidence detections that might not be correct but correctly detecting more objects overall. This means that precision decreases and recall increases as confidence threshold decreases.

Figure 27 shows that the precision/recall curves for YOLOv3 and RetinaNet are very similar, so expected performance will be similar. In terms of real time operation, YOLOv3 is the obvious, optimal choice. Therefore, based on these criteria and the fact that Keras implementations (both the network and transfer learning) are readily available, the YOLOv3 pre-trained network was selected for ExP-L.

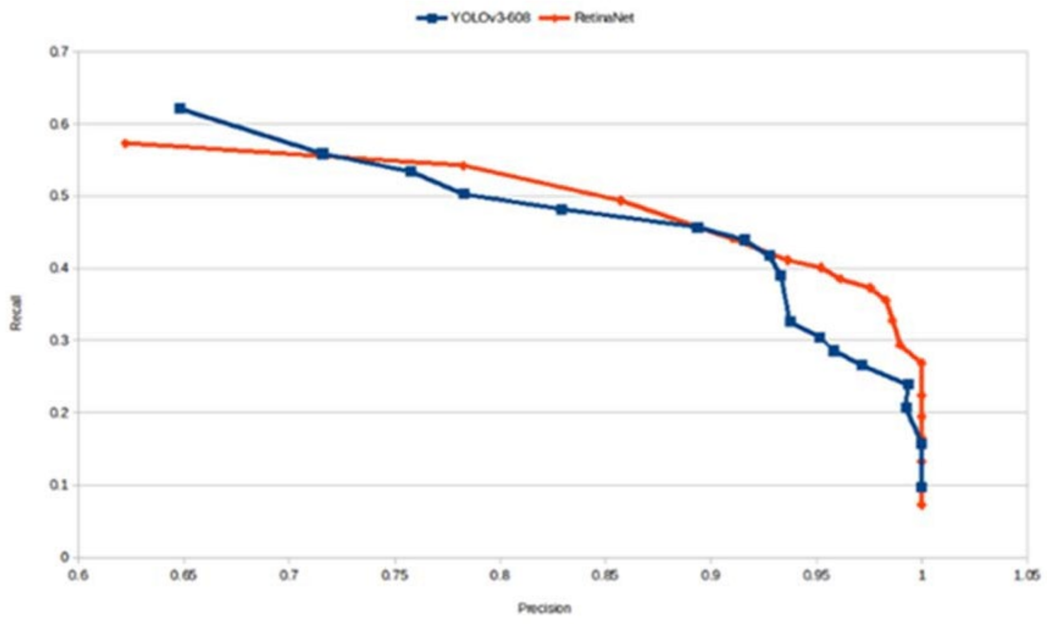


Figure 27. Precision vs. Recall for YOLOv3 and RetinaNet

[15] Peters, J., and Frittelli, J. Positive Train Control (PTC): Overview and Policy Issues. *Congressional Research Service Report for Congress*. 30 July 2012.

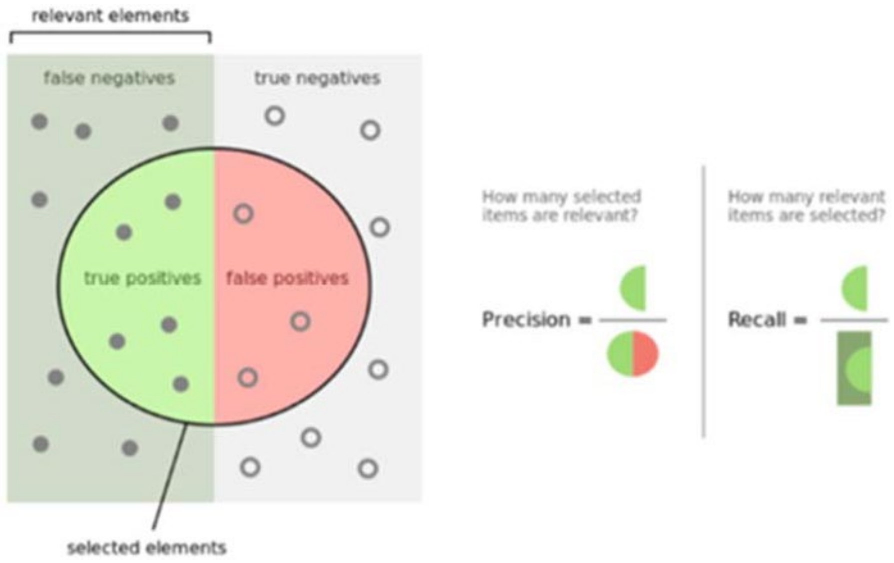


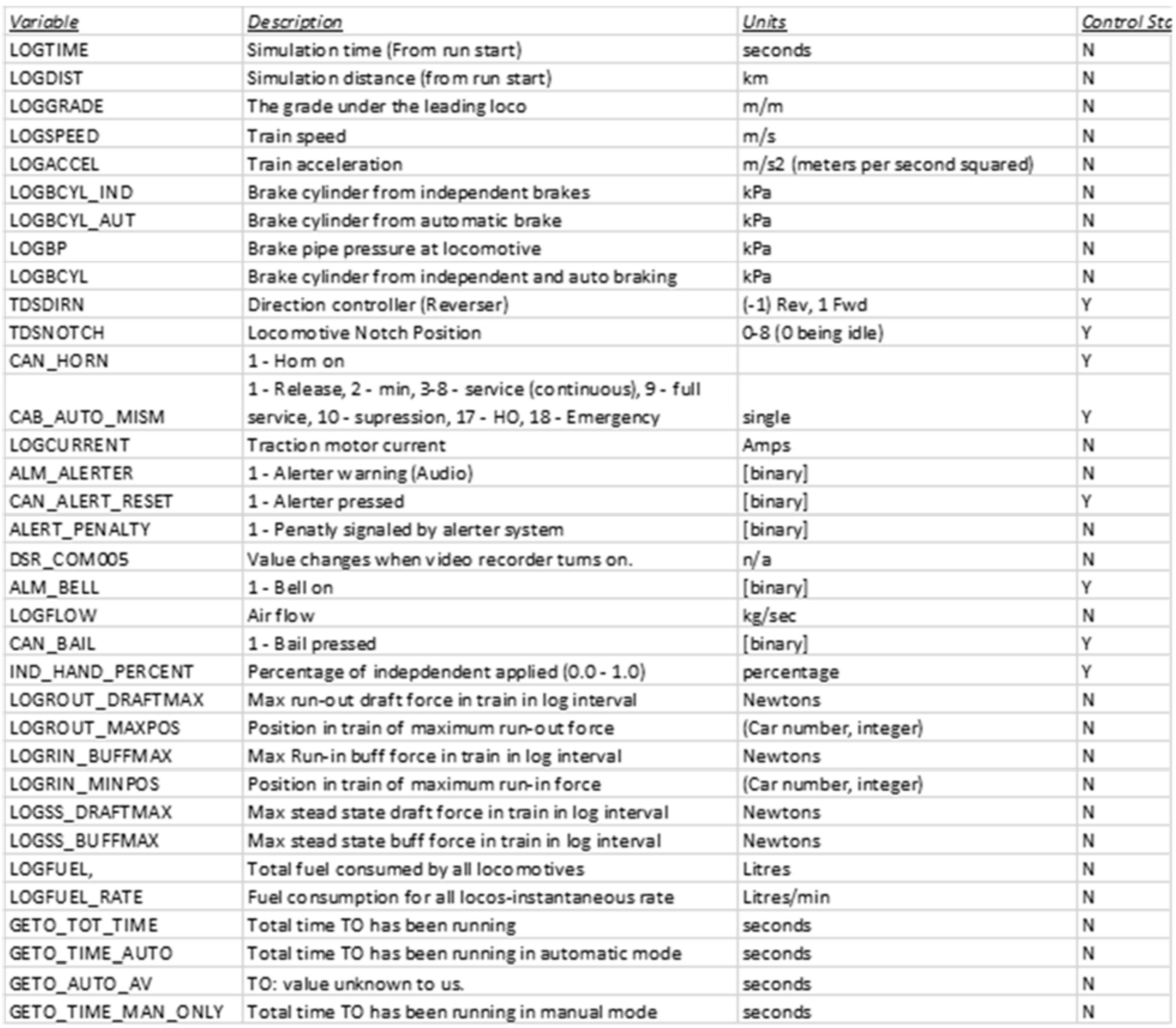
Figure 28. Precision vs. Recall Diagram. Detections would be selected based on the chosen confidence value

Open Source Implementation of YOLOv3

The rapidly growing field of deep learning has generated substantial interest from various technical communities. The result is that newer algorithms and technologies are quickly replicated and open-sourced across various programming domains. With YOLO and its variants, a similar trend has occurred. For ExP-L, a trade study was conducted to determine the best open source version of YOLOv3. The requirements for the selected version were that it 1) was implemented using the Keras framework, 2) supports transfer learning techniques, and 3) is well supported and documented.

The open source implementation Keras-YOLOv3 was found to meet the above requirements. Most importantly, it was found that the transfer learning implementation was straightforward and well documented. Some various experiments with transfer learning and data augmentation were carried out relatively easily and helped facilitate the decision to use this implementation. Currently, a copy of Keras-YOLOv3 is being used for ExP-L on Aurora's internal Git repository.

Appendix B.
List of CTIL Web Service Variables

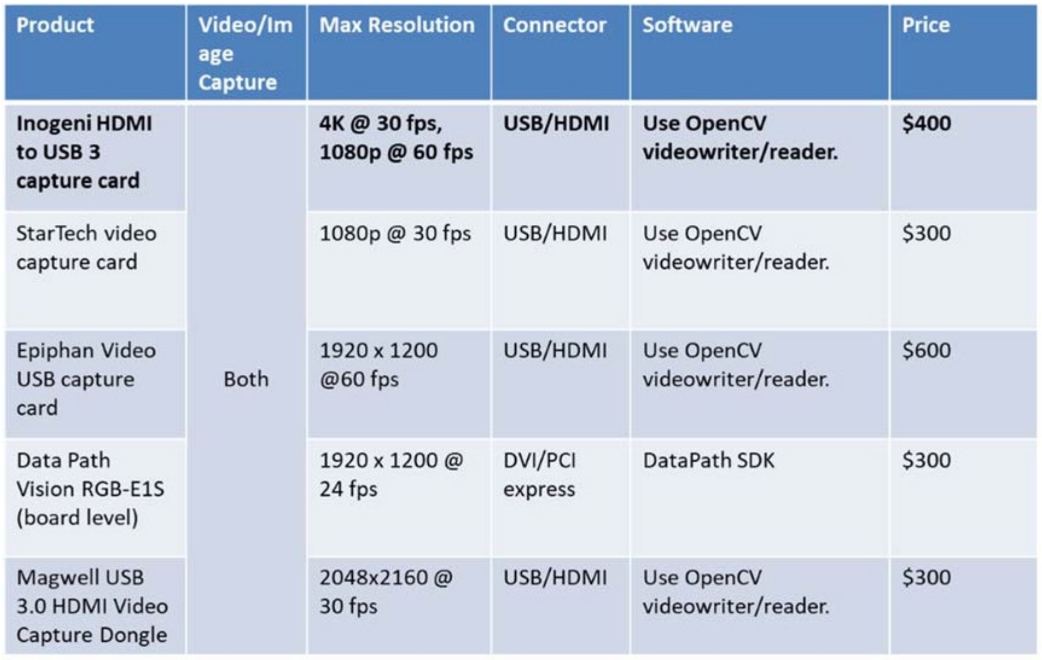


Variable	*Description*	*Units*	*Control Sta*
LOGTIME	Simulation time (From run start)	seconds	N
LOGDIST	Simulation distance (from run start)	km	N
LOGGRADE	The grade under the leading loco	m/m	N
LOGSPEED	Train speed	m/s	N
LOGACCEL	Train acceleration	m/s2 (meters per second squared)	N
LOGBCYL_IND	Brake cylinder from independent brakes	kPa	N
LOGBCYL_AUT	Brake cylinder from automatic brake	kPa	N
LOGBP	Brake pipe pressure at locomotive	kPa	N
LOGBCYL	Brake cylinder from independent and auto braking	kPa	N
TDSDIRN	Direction controller (Reverser)	(-1) Rev, 1 Fwd	Y
TDSNOTCH	Locomotive Notch Position	0-8 (0 being idle)	Y
CAN_HORN	1 - Horn on		Y
CAB_AUTO_MISM	1 - Release, 2 - min, 3-8 - service (continuous), 9 - full service, 10 - supression, 17 - HO, 18 - Emergency	single	Y
LOGCURRENT	Traction motor current	Amps	N
ALM_ALERTER	1 - Alerter warning (Audio)	[binary]	N
CAN_ALERT_RESET	1 - Alerter pressed	[binary]	Y
ALERT_PENALTY	1 - Penatly signaled by alerter system	[binary]	N
DSR_COM005	Value changes when video recorder turns on.	n/a	N
ALM_BELL	1 - Bell on	[binary]	Y
LOGFLOW	Air flow	kg/sec	N
CAN_BAIL	1 - Bail pressed	[binary]	Y
IND_HAND_PERCENT	Percentage of indepdendent applied (0.0 - 1.0)	percentage	Y
LOGROUT_DRAFTMAX	Max run-out draft force in train in log interval	Newtons	N
LOGROUT_MAXPOS	Position in train of maximum run-out force	(Car number, integer)	N
LOGRIN_BUFFMAX	Max Run-in buff force in train in log interval	Newtons	N
LOGRIN_MINPOS	Position in train of maximum run-in force	(Car number, integer)	N
LOGSS_DRAFTMAX	Max stead state draft force in train in log interval	Newtons	N
LOGSS_BUFFMAX	Max stead state buff force in train in log interval	Newtons	N
LOGFUEL,	Total fuel consumed by all locomotives	Litres	N
LOGFUEL_RATE	Fuel consumption for all locos-instantaneous rate	Litres/min	N
GETO_TOT_TIME	Total time TO has been running	seconds	N
GETO_TIME_AUTO	Total time TO has been running in automatic mode	seconds	N
GETO_AUTO_AV	TO: value unknown to us.	seconds	N
GETO_TIME_MAN_ONLY	Total time TO has been running in manual mode	seconds	N

Appendix C.
Hardware Trade Study

Two image capture methods were originally proposed for ExP-L. The first used cameras and the second extracted the imagery directly from the CTIL. The first method was quickly ruled out because it was determined that captured imagery would contain significant screen artifacts. For example, the refresh rate of the screen would introduce lighting inconsistencies and improper sampling effects in the imagery, and screen pixilation would introduce errors at higher magnifications. Finally, cameras would need to be placed inside the CTIL cab to capture the outdoor environment. This is not realistic, because in practice, external perception cameras would be placed on the exterior of the locomotive facing forward toward the track. Cameras in the CTIL would only produce heavily cropped images.

To work effectively with the CTIL, it was determined that imagery must be captured directly from the CTIL simulation. An initial investigation determined that an HDMI splitter could be included as part of the A/V equipment of the CTIL, which meant that the simulation input to the screens could be bypassed for potential image capture. It was determined that this could be accomplished using an image capture or screen capture card. A trade study of available image capture cards was conducted, as shown in Figure 29. The image capture card must be able to capture 1080p at minimum at 30 fps so that the computer vision results can be presented to the user with minimal latency (i.e., around 40 ms). Furthermore, a capture card should be selected that allows for fast setup and testing. Several options from plug-and-play to board level cards were reviewed. Ultimately, the team selected the Inogeni capture card, which is a plug-and-play device that has the highest resolution/frame rate combination. Furthermore, in terms of software setup, the Inogeni card is recognized by OpenCV as a USB capture device, which makes setup and deployment easy and customizable.



Product	Video/Image Capture	Max Resolution	Connector	Software	Price
Inogeni HDMI to USB 3 capture card	Both	**4K @ 30 fps, 1080p @ 60 fps**	**USB/HDMI**	**Use OpenCV videowriter/reader.**	**$400**
StarTech video capture card		1080p @ 30 fps	USB/HDMI	Use OpenCV videowriter/reader.	$300
Epiphan Video USB capture card		1920 x 1200 @60 fps	USB/HDMI	Use OpenCV videowriter/reader.	$600
Data Path Vision RGB-E1S (board level)		1920 x 1200 @ 24 fps	DVI/PCI express	DataPath SDK	$300
Magwell USB 3.0 HDMI Video Capture Dongle		2048x2160 @ 30 fps	USB/HDMI	Use OpenCV videowriter/reader.	$300

Figure 29. Image Capture Card Trade Study

Figure 30. Inogeni HDMI to USB 3.0 Capture Card Selected for Use with EXP-L

The ExP-L system is intended to be a real-time system that uses state-of-the-art deep learning algorithms. Training and deployment of deep learning algorithms typically requires a high-performance CPU or GPU. To achieve this performance, ExP-L requires higher-end CPU/GPU processing with at least 8 GB of RAM each for the GPU and CPU. The CPU should be at least an Intel i7 with a GPU that supports the chosen DNN. This selection criteria will allow for potential future tradeoffs between algorithm deployment on the CPU or GPU, depending on the needs of the final ExP-L system.

Although most data will be stored offline on a server or through a service such as Amazon Web Services (AWS), local data storage will be important for data collection during test events. To determine the amount of data storage required, the following calculation can be completed using the specs on the Inogeni hardware:

Image Data Rate = 1920 x 1080 x 8 bits/(8 bits/1 byte) x 60 frames/sec. = 124 MB/sec

Using the chosen hardware, ExP-L has the potential to capture frames at 124 MB/sec. In 30 minutes, this will be 225,000 MB (225 GB) of data. Thirty minutes of imagery from the CTIL comprises approximately 216,000 images, which is more than enough for training purposes. Therefore, local data storage of at least 250 GB is required.

Finally, there is a tradeoff between laptops and PCs. Laptops are limited in terms of available processing power and can be limited on the number and types of ports available; this mean a desktop PC is required. In general, gaming PCs are the best choice for intense graphics and image processing, as these high functioning computers come with plenty of ports and data storage. Therefore, a trade study of available gaming PCs was performed and a Dell Alienware Aurora R7 gaming PC was selected, as shown in Figure 31.

Processor	Intel Core i7 8700K, 4.7 GHz
Video Card	NVIDIA GeForce GTX 1080 w/8 GB GDDR5X
Memory	16 GB, 2666 MHz, DDR4 up to 64 GB
Hard Drive	256 GB SSD + 1 TB SATA
Wireless	Qualcomm DW1810 1x1 802.11ac Wi-Fi Wireless LAN and Bluetooth 4.1
Cost	$2000

Figure 31. Computing Platform Selected for EXP-L

The Dell Alienware PC has a state-of-the-art CPU/GPU and will be able to support DNN training and deployment for all phases. It should also be enough platform for initial data collection on locomotives. For longer-term deployment situations, the PC will need to be custom built with similar specifications and ruggedized for the locomotive environment.

Appendix D. Methods of Labeling Data

Labeling Dataset from the CTIL

Accurate data labeling is extremely important for training neural networks. For computer vision, the label for an image is the bounding box (i.e., subset of the image) that contains the object of interest and a class name. The bounding box location is given by a corner pixel location and length/width of the bounding box. The label should be something simple and informative; for example, the label for a dog in an image would be "dog."

However, as the training dataset grows, labeling can become time-consuming. For ExP-L, the estimated dataset size is several thousand images, possibly higher. To overcome lengthy labeling sessions, three different options were conceived. Ultimately, it was determined that manually labeling data was the best option and that it will not be as lengthy a process as originally expected.

Automatic Labeling of Data

It was originally proposed to label the imagery gathered from the CTIL automatically using the locomotive state information and railway map information for the CTIL simulation. Working with Corys, it was determined that the CTIL simulation follows the Geographic Information System (GIS) format, which is a system for capturing, storing, and displaying data related to positions on Earth's surface. The CTIL simulation stores geometry information using .shp, .dbf, and .prj files. Furthermore, there is a web service that broadcasts real-time sim and locomotive data over a network connection. Ideally, GIS data and real time information could be used to accurately predict trackside information encountered by the locomotive. This would allow for predictions of image content, which could be used to label the image itself.

The .dbf files contain locations of simulation landmarks (e.g., signals, signs) measured in terms of chainage from Aurora, IL. The .dbf file is also human-readable in MS Excel, which means that this file could be used as a reference location file for real time operations in the CTIL. Combined with the real time chainage value from the web service, the .dbf file can be referenced to make a prediction of the landmark to be encountered. However, it was determined that this prediction would not be accurate enough for image labeling because the .dbf file does not indicate the ending location of a landmark accurately. In practice, ExP-L could start labeling data, but not know when to stop. Ad-hoc methods could be used to determine the stopping point, but this would introduce the risk of incorrectly labeled data, which would affect the accuracy of the DNN. Finally, there is no accurate way to determine the image bounding box automatically, therefore automatically labeling was ruled out as a feasible option.

Amazon Mechanical Turk and Manual Data Labeling

Amazon Mechanical Turk (MTurk) is a standard option for data labeling. MTurk is a pay service where data is posted to Amazon and a collection of remote workers manually label the data. Typically, these workers are paid several cents per image to ensure quality image labeling of data. Additionally, using MTurk would enable the research team to use AWS for data storage and potentially processing power, as well. As an example, labeling 4,000 images would cost around $200 – $300, plus storage fees. An application that would allow the workers to label the data would also need to be developed, which amounts to about 2 weeks of engineering time and testing.

Finally, experiments were performed to understand how to manually label data using internal labeling software. The initial dataset gathered from the CTIL took several hours to label, which is approximately 500-1,000 images/hour depending on the experience of the labeler. In total, this means that it would take 16 hours (2 engineers x 8 hours a day x 1 day) to label all the CTIL data gathered, assuming the data set is 10,000 images. This estimate is reasonable, so the team decided that going forward all data will be labeled manually.

Appendix E.
Tesseract and Optical Character Recognition

Interpreting trackside signs is a critical component for a system like ExP-L. Correctly correlating sign content with the current state of the locomotive will enable ExP-L to understand whether the operator has the proper track authority. From a computer vision standpoint, this is a two-stage problem, consisting of object detection followed by text interpretation. This can be accomplished using YOLOv3 and optical character recognition (OCR) techniques. OCR has been in use for decades to convert hand-written or copied texts into computer-readable formats using basic computer vision techniques. One type of OCR is a pattern recognition-based system that decomposes text information into lines, intersections, and loops. Feature extraction can then be performed to compare image features with stored features and choose the closest match.

A second type of OCR, Google's Tesseract OCR library, was used in this analysis. The Tesseract OCR library uses a more modern type of feature-matching. It uses a long short-term memory (LSTM) RNN that has been pre- trained on over 100 languages and different font types to recognize lines and shapes quickly, and then accurately match those shapes to known characters. These LSTMs excel in learning long term dependencies that perform very well when applied to classifying and making predictions over arbitrary time intervals and text.

Tesseract and Image Processing

While the goal of OCR might appear straightforward, there are several aspects that make OCR challenging. Specifically, there are certain considerations that must be made when working with OCR libraries. Tesseract is currently the state-of-the-art in OCR detection, but the algorithm does not work well with too much background clutter. In general, the location of the text must be pre-defined for Tesseract to be successful. For ExP-L, the input to the Tesseract OCR algorithm will be a cropped image containing the milepost or sign only. The location of the sign in the image can be provided by the YOLOv3 network to the Tesseract algorithm or through other image processing methods. If required, further refinement can be accomplished through image preprocessing and other computer vision techniques to narrow the ROI to sections of text only. Most of the experiments for sign detection were heavily focused on generating the best image for the Tesseract algorithm. These experiments are discussed below.

The sign reading module (SRM) (Figure 32) starts with an input image into YOLOv3, which then outputs a bounding box location of a detected sign within the image.

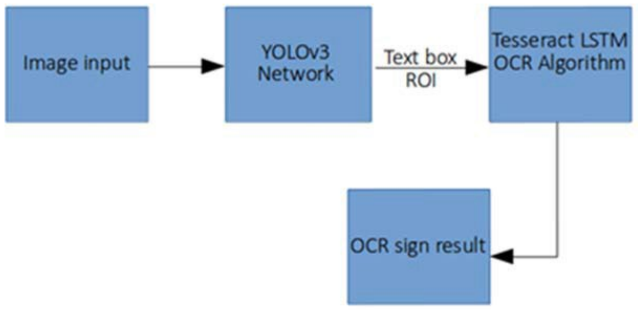


Figure 32. Sign Reading Module (SRM)

OCR detection begins with passing the image and the bounding box information to the SRM, which crops down the image to the bounding box size and adds some extra padding around the detected sign.

Figure 33 shows a raw output image from YOLOv3, cropped based on the bounding box information, and the final resized image used by the SRM.

The top row of Figure 33 are images from the YOLOv3 milepost detection output with added padding. The lower row of images shows the cropped output with annotated OCR results. The distances and resolution of the milepost images play a large role in the success of performing OCR and text detection on these images.

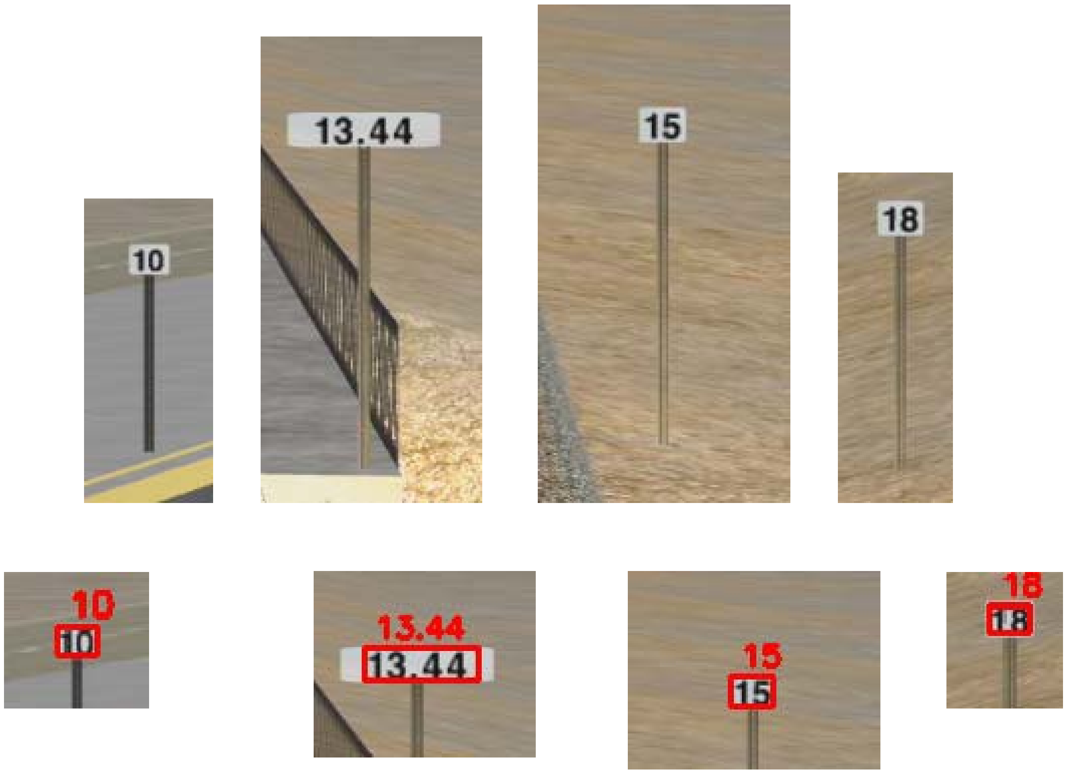


Figure 33. Sample Tesseract Sign Reading Results

In summary, OCR methods make it possible to perform text detection and recognition of milepost markers within the CTIL imagery. Mileposts detected via the YOLOv3 neural network, and their respective bounding boxes are output. Once the text image is localized, it is then passed to Tesseract to detect and interpret the text within the bounding box. Over a small sample dataset of 50 images, this method proved to have an accuracy of 75 percent when padding and cropping images during preprocessing and removing images whose overall resolution is too low to detect text. Due to the low renderings of distance objects and overall fidelity in the CTIL, this OCR pipeline could have significant forward-looking capabilities if the overall quality of the input imagery could be improved. Accuracy of this method could be improved with higher resolution images and imagery obtained closer to the milepost.

Appendix F. Railroad Track Detection

Estimating the position of the railroad track is important for proper operation of the locomotive. As in autonomous driving, systems such as ExP-L should be able to detect railroad track in near-real-time to extract track location and current track occupancy and determine whether the locomotive is traveling safely. Two approaches to feature extraction were explored – a DNN approach and an approach using conventional computer vision techniques. Both were investigated to show feasibility, but future work will require more investigation into outlier removal and possible Kalman filtering to increase the recall of railroad tracking to 80 percent or higher. The proposed railroad track detection algorithm is shown in Figure 34.

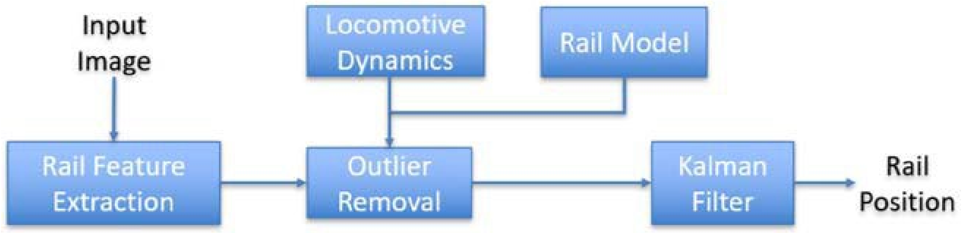


Figure 34. Rail Position Estimation Approach

Performance requirements such as precision and recall will be used to measure object detection accuracy, but a true positive (TP) will be defined as:

$$\|P_{GT} - P_{LE}\| < \delta$$

where P_{GT} is the ground truth position and P_{LE} is the Lane estimate position. If the difference between these two values is less than the threshold, then the detection will be considered a TP. This approach to the TP metric is more useful than simply counting the number of TPs and comparing to the ground truth, because it also captures the positional accuracy of the detection.

The DNN approach for feature extraction was to label sections of the track and train YOLOv3 to estimate the position of the track and splits in the track in real time. An example of track split detection is shown in Figure 35. Consistent stretches of track and track splits were detected by the DNN, and achieved low detection confidence levels and around 50 percent recall values. As more training data is gathered from the CTIL, the DNN will become better at detecting these track features with higher detection confidence and recall levels.

A second approach taken for feature extraction was a non-DNN technique similar to methods used for autonomous driving. In most autonomous driving cases, the successful approach has been to capture the image, correct for perspective, threshold the image, and use simple techniques such as blob or Hough lines to extract lanes or lane markers. This study's approach was similar, except that railroad tracks were being detected instead of car lanes. As shown in Figure 35, a conventional approach was demonstrated using CTIL data. First, the image was captured and then warped to enable a top down perspective of the track itself. Strong levels of thresholding and filtering were used to remove the background and segment out the track just in front of the locomotive. This approach works well except for regions where there are multiple tracks crossing. For example, this approach has a high confusion rate around track splits. For these types of events, the DNN approach performs very well. A DNN could also be used to detect the track at specific intervals and then conventional techniques used to investigate the state of the individual rails. Therefore, in general, future work will involve exploring the combination of DNN and conventional feature extraction techniques, the implementation of outlier removals, and Kalman filtering.

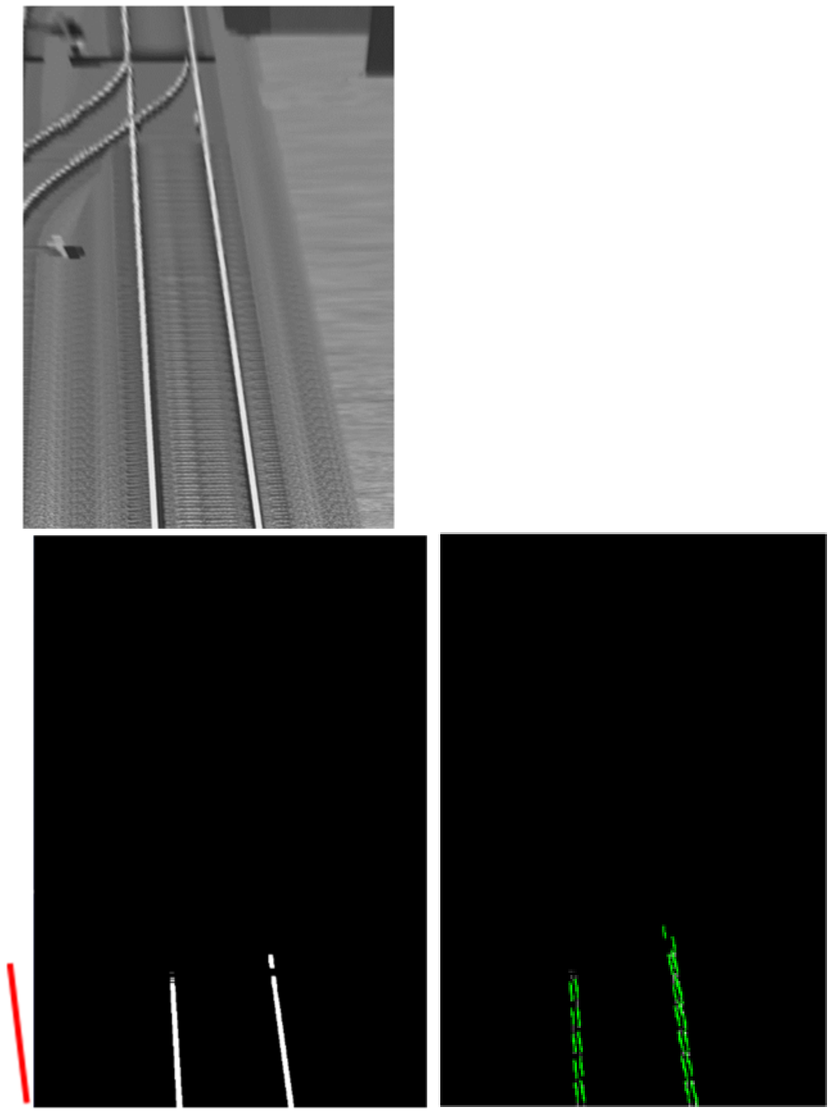


Figure 35. Top left – raw image; Lower Left – Thresholding and filtered to capture the rails only; Lower Right – Hough lines used to detect and model the rail; Red line represents spline generated to track the rail position over time

Appendix G. Human Machine Interface to Support Engineer Situational Awareness

Introduction

The capabilities of the current ExP-L system developed in the CTIL include detection of the following objects and contexts:

- Vehicles obstructing the current right-of-way
- Signals that control the movement authority on the current track.
- Trackside objects that provide location information including milepost markers and station signs
- Warning flags before speed restrictions
- Railroad crossings and faulty equipment (e.g., gates in the up position)
- The position of track switches to ensure correct alignment

To use the ExP-L system to support engineer situation awareness, notifications and warnings based on ExP-L recognition will need to be integrated into locomotive operating displays. Most modern freight and passenger locomotives use digital display screens to provide primary train information for controlling the vehicle (Figure 36), although there are undoubtedly many older locomotives still in service outfitted with steam gauges. Two display panels, one of which is the primary operating display, are located on a console in front of the engineer and below the main windscreen. Additional displays for PTC or other supplemental systems can be displayed in different locations, including atop the AAR 105 control stand (Figure 36, left), between the front windscreens (Figure 36, right), or to one side or even above the engineer's windscreen.

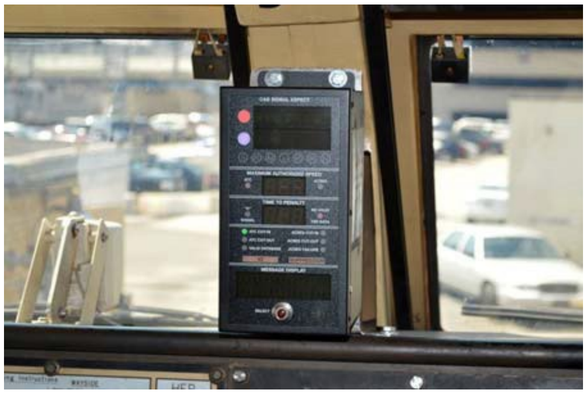
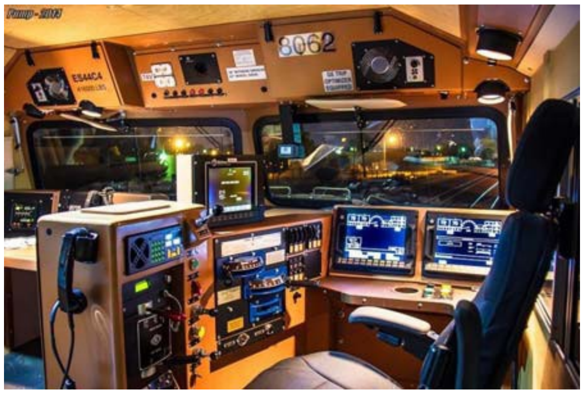



Figure 36. Left: A display for the ACSES positive train control system is mounted between the two windscreens; Right: Cab layout for a GE Evolution locomotive: Note the display atop the AAR control stand in the middle of the cabin

The primary driving display (Figure 37, left) provides information about train state used by the engineer to control the train, including speed, throttle and brake settings, brake line pressures, and engine output. The displays for the secondary locomotive systems also show some of the primary information (e.g., speed) and provide additional information related to their function, such as the train location and the speed plan for Trip Optimizer (i.e., an automated driving system), the upcoming section of track (Figure 37, center), or train location relative to an upcoming speed restriction on the PTC display (e.g., the red cross-hatched area overlaid on the moving map in Figure 37, right).

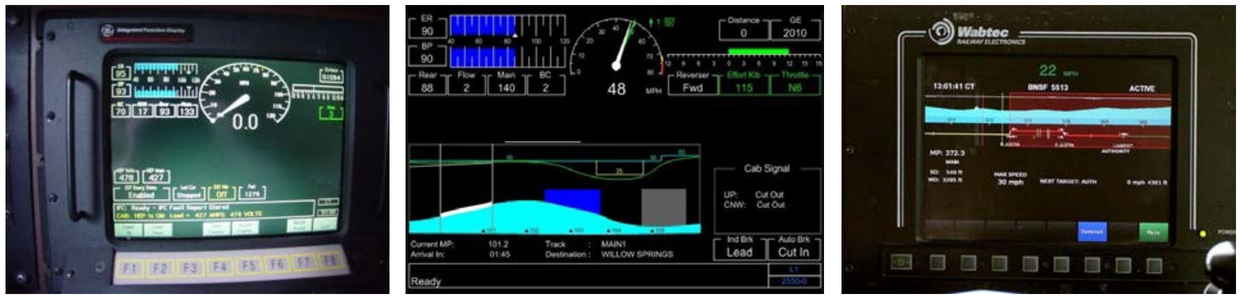


Figure 37. Examples of typical digital locomotive displays. Left: Example of a primary driving display. Center: Trip Optimizer display screen (Wabtec); Right: i-ETMS (Wabtec) PTC display screen. Note that the secondary system displays (TO and i-ETMS) also show primary driving information along with unique system information.

General Human Factors Considerations

The main human factors consideration in integrating ExP-L information into current displays would be to ensure that the display modifications do not impose an additional visual workload that would interfere with the primary task of controlling the train or secondary task of monitoring the environment for potential hazards. The current train operating displays described above are already information-dense, so careful consideration of information display is needed. Overloading the visual attention system may detract from the engineer's ability to perform the secondary vigilance tasks or even lose situational awareness, which could lead to severe consequences further along the track.[16]

The level of authority delegated to the ExP-L system to automatically control the train will also shape the interaction model and display of information. If ExP-L operates as an automatic safety overlay system like PTC, then the system may simply display information to help explain impending or current system actions. If the engineer is the final authority on controlling the train, then more detailed information, such as the nature of a potential track hazard, might need to be displayed to enable the engineer to take appropriate action. The salience of ExP-L information to the current train operating situation could be used to determine the level of display. For example, confirmatory information, such as identifying a resume flag at the end of a permanent speed restriction, might not require any notification if it conforms to the known track configuration. However, any conflict between what is identified in the environment by EXP-L and the known trip plan would need to be indicated immediately to the engineer.

Implementation with Current Displays

This section considers a few possible implementations of ExP-L into current freight locomotives configured with typical cab operating displays (e.g., Trip Optimizer and i-ETMS). Researchers assumed that ExP-L will be used as a semi-autonomous advisory system with no authority to

[16] National Transportation Safety Board, "Derailment of Amtrak Passenger Train 188 Philadelphia, Pennsylvania, May 12, 2015," National Transportation Safety Board, Washington, DC, Railroad Accident Report. PB2016-103218, May 17, 2016.

control the throttle or brakes but with the capability to automatically activate non-vital train systems such as the horn or bell. An auditory alarm should accompany any alerts signaled by the EXP-L system that are obviously different from other systems (e.g., the alerter). Forward-facing cameras mounted on the locomotive would capture imagery that can be analyzed by the machine vision processing pipeline. Outputs of the recognition system would have to be added to a locomotive information bus so that they could be included in the existing cab displays (e.g., i-ETMS). These display modifications would have to be made in cooperation with the locomotive or third-party display manufacturers.

As noted above, the current operating displays have limited space for any additional ExP-L related symbology. Figure 38 shows one example of how ExP-L output (in this case, identification of potential track obstacles) could appear as an alert to the engineer. The red box indicates a critical alert showing that ExP-L has identified trespassers ahead of the train (one or more people on the track ahead and to the left of the track). A set of icons could be used to differentiate the type of track obstacle (e.g., vehicles or equipment). The color-coding of the trespasser icon indicates the proximity of the trespassers to the train track (e.g., red could indicate direct interference with the train, while yellow might represent a potential hazard). Similarly, the color of the cross-hatched area represents the distance- or time-to-contact to the trespassers (e.g., red could represent 5 seconds to contact). This display is based on typical automotive forward collision avoidance displays which indicate distance to vehicles ahead (see Figure 38). Automatic sounding of the horn could occur at the same time as the appearance of the alert. An additional warning message could direct the engineer to apply the train brakes to reduce the severity of impact.

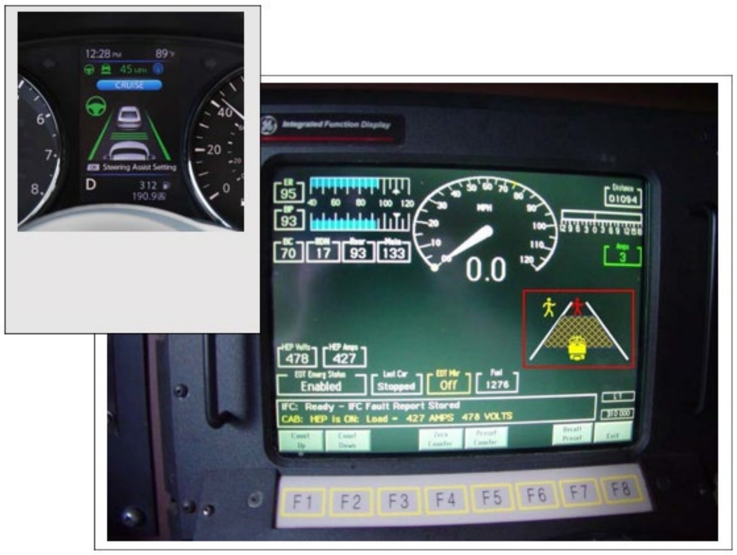


Figure 38. Example of ExP-L symbology on a typical primary operating display; this symbology indicates an obstruction on the tracks (red figure) and another close to the tracks on the left side (yellow figure)

Similarly, ExP-L identification of signal aspects could be displayed with icons representing the identified aspect. With sufficient camera capability, the ExP-L system could provide a preview of the upcoming signal before it can be visually confirmed by the engineer. This could be critical if the upcoming signal shows an unexpected restrictive aspect or has failed and does not display a signal. Furthermore, the signal could be tracked by the ExP-L system until it has been passed to ensure that any late signal changes are detected and indicated to the engineer. Figure 39 shows one possible symbol to indicate that the signal has changed, or "signal dropped" since it was initially identified by the ExP-L system. The original signal aspect is overlaid with a red "X" to clearly indicate that it is no longer valid and the red arrow indicates a downgrade to a more restrictive signal.

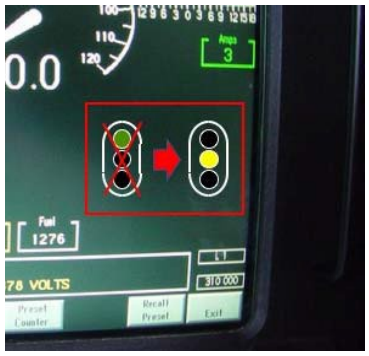


Figure 39. Icon representing a signal change that occurred since the signal was identified

When approaching a switch or crossover, the engineer must visually ascertain that the switch is correctly lined for the intended travel path. Automatic identification of switch alignment by ExP-L could be indicated by the symbols shown in Figure 40. In this example, the intended travel path is to cross over to another main line track. The switch is incorrectly lined to continue along the current track as indicated by the red path. Proper alignment could be indicated with a green outline showing the correct intended route over the switch. This would require the ExP-L system to be coupled with the trip plan, which can be provided by the PTC system.

by EXP-L

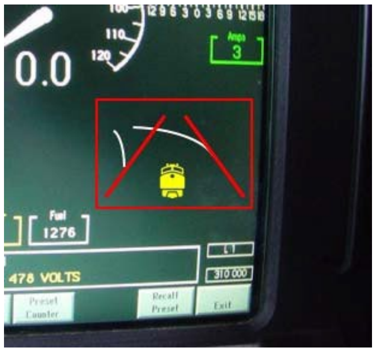


Figure 40. Icon representing an incorrectly lined switch; the intended travel path is to cross over to a new track

Implementation Using a Separate Augmented Reality Display

A second implementation would be to create an entirely separate display that overlays ExP-L derived information over a video stream of the forward view from the locomotive. This approach would use one of the ExP-L external camera views to show the forward view, then overlay symbology that presents the ExP-L information seemingly embedded in the external environment. This approach is currently available commercially for aviation, as illustrated by the screenshot from FlyQ Insight, an iPad/iPhone-based app developed by Seattle Avionics Software

(Figure 41).[17] In this platform, the device camera captures the forward view from the airplane and then uses the device's GPS and orientation information in combination with a digital map databases to overlay approach and landing information on the augmented view.

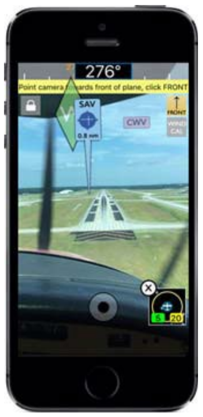


Figure 41. Screenshot of the FlyQ Insight app from Seattle Avionics

For rail operations, a similar head-down display could be created that showed the video feed from the ExP-L camera capturing the forward view overlaid with symbology indicating the type of object detected. This scene augmentation could be implemented without any connection to the locomotive data infrastructure, since location information can be obtained through GPS and machine vision would be responsible for labelling identified objects from the scene. An example of automatic switch alignment indication is shown in Figure 42 (left). While the enhancements are very similar to the symbology previously described, this implementation provides better clarity in the operating context, especially if there were a series of switches in succession (Figure 43, right).

Similar symbology could be used for highlighting obstacles on or near the tracks. Unlike the simple icons displayed on the primary driving display, the engineer can now use their own perception to better evaluate the situation and consider possible actions. Depending on the camera capabilities, ExP-L could even identify the type of obstacle much earlier than the engineer, allowing for a longer window to blow the horn or even slow the train.

There are a few practical implementation considerations with this approach. First, an additional physical display device needs to be placed somewhere in an already-crowded cab. Ideally, it would be placed on the forward console below the windshield; this would allow the engineer the shortest distance to shift gaze between display and real-world views. Using a tablet computer as a self-contained augmented reality (AR) display would add another constraint on its placement by requiring access to the forward view. Second, the camera on a commercial tablet may not have the dynamic range to satisfactorily handle the wide range of possible lighting conditions, (i.e., low light/nighttime to rain/snow/mist to extremely bright conditions at dawn or dusk, depending on the direction of travel). Dedicated camera systems would have better performance capabilities for this application. Third, it is not clear that commercial tablet systems would have sufficient computing power to perform the image processing and graphics rendering at real-time rates.

[17] Information can be found at https://www.seattleavionics.com/FlyQInSight.aspx

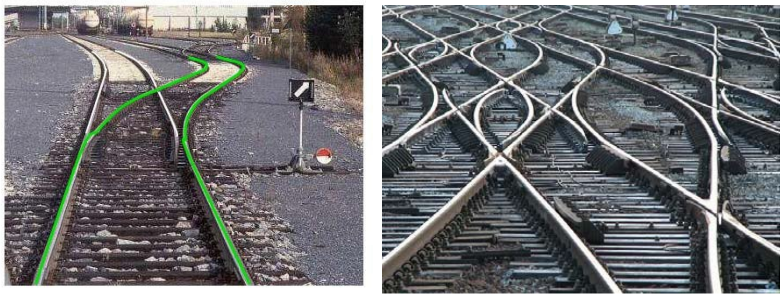
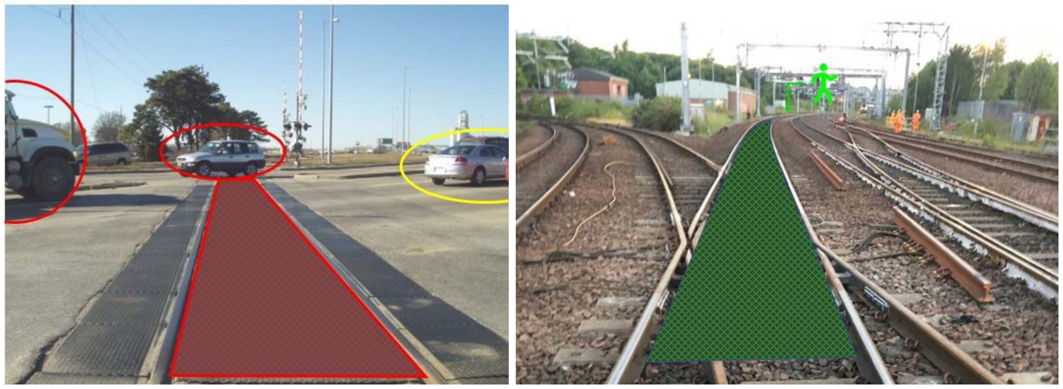


Figure 42. Left: AR overlays indicating that a switch is correctly lined fort the planned path; Right: AR overlays could be especially helpful through complicated switches in yards



The color of the cross-hatching again conveys the distance- or time-to-contact with the obstacle. Left: For closer obstacles, the salience of different obstacles is shown through the coloring of the bounding boxes. The vehicle on the right side is exiting the grade crossing, so it is less salient than vehicles in or approaching the crossing. Right: For distant obstacles, ExP-L could potentially determine the type of obstacle well before the engineer could make a visual confirmation, thus providing a longer window to take appropriate action.

Figure 43. AR symbology identifying obstacles

Integration with the Current FRA Sponsored AR-HUD Project

The tablet-based approach would display the EXP-L information on a full, windscreen-sized AR-HUD. Integrating the ExP-L capabilities into the current AR-HUD project would allow researchers to investigate several additional questions related to the use of AR in rail operations including 1) determining the appropriate display methods to inform the engineer of infrequent but potentially important information and to provide guidance on appropriate action, 2) Determining how the addition of ExP-L information affects the vigilance and situation awareness of the locomotive crew over the course of a long trip, and 3) determining how the crew's trust in ExP-L information changes with repeated use of the system. These questions could be incorporated into the planned human-in-the-loop studies with a prototype AR-HUD.

Abbreviations and Acronyms

ACRONYM	DEFINITION
AM	Alerted Module
CIM	CTIL Interface Module
CNN	Convolutional Neural Network
COCO	Microsoft Common Objects in Context dataset
COTS	Commercial, Off-The-Shelf
CPU	Central Processing Unit
CTIL	Cab Technology Integration Laboratory
DM	SSO CNN Detection Module
DNN	Deep Neural Network
ExP-L	External Perception for Locomotives
FRA	Federal Railroad Administration
GPU	Graphics Processing Unit
HDMI	High-Definition Multimedia Interface
HMI	Human-Machine Interface
HSV	Hue, Saturation, Value
HUD	Head-Up Display
IM	Interpretation Module
mAP	Mean Average Precision
OCR	Optical Character Recognition
RGB	Red, Green, Blue
ROI	Region of Interest
SSO	Signs, Signals, and Objects
YOLOv3	Redmon's You Only Look Once deep learning architecture