



Artificial Neural Network Solution (ANS) Direct Pty Ltd

Machine Learning and Deep Learning: Design and Deployment

Dr. Tuan Nghia Nguyen, CLA
30 June 2021

Computer Vision Problem

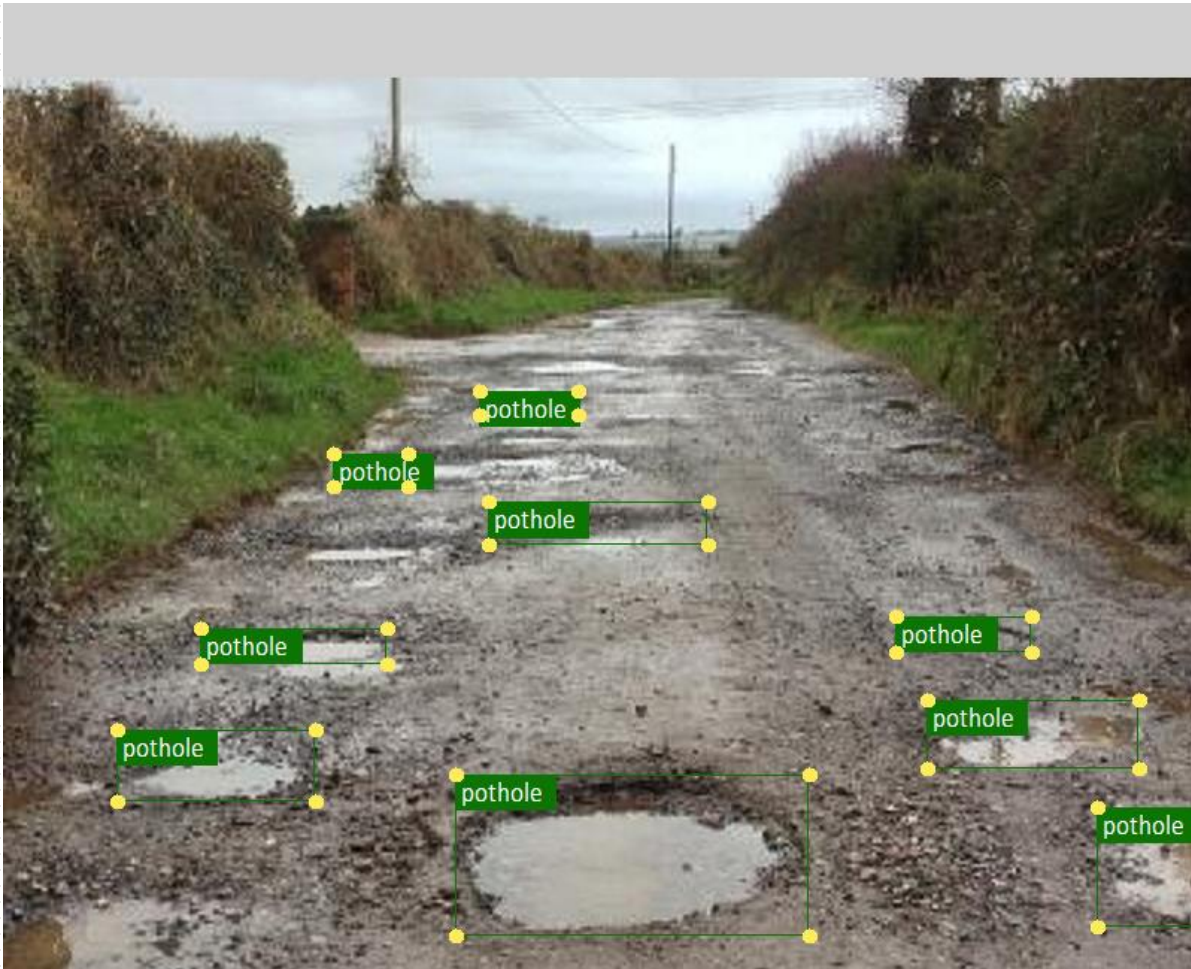
Question: How do we use normal computer vision technique to solve this defect detection problem?



Problem: Defects come from different shapes, sizes, and lighting conditions
→ How do we use a normal pattern matching to solve this issue?

Machine Learning Approach

Machine Learning and Deep Learning techniques can be used to resolve detect detection problems. The object detection technique involves in localize the objects in an image or a video stream and classify them.



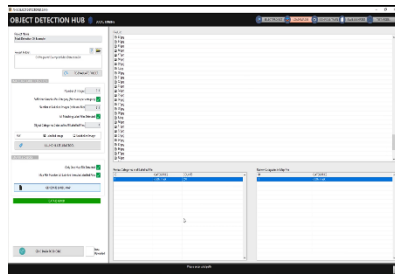
Problems:

- There are object detection frameworks for LabVIEW users.
- NI Vision Development Module does support inferencing Tensorflow models and OpenVINO models in CPU mode → does not meet real-time requirements (processing time is 4-5 seconds per image)
- Popular machine learning and deep learning frameworks (Tensorflow, Caffe, CNTK, MXNET, and Pytorch, etc) are in text based programming languages: Python, C, C++, Java, and C#.
- Require steep learning curve for LabVIEW users to properly design correct ML models.
 - How to design a neural network model and its architecture?
 - How to choose the best techniques for your tasks: SSD, Faster RCNN, Yolo, and EfficientDet architecture?
 - How to train a neural network model to meet requirements: high accuracy and no overfitting?
- It is difficult or impossible to integrate deep learning models into LabVIEW applications to meet real-time requirements (processing time is less than 100ms).
 - Use LabVIEW API natively (do not call Python script)
 - Support GPU graphic cards

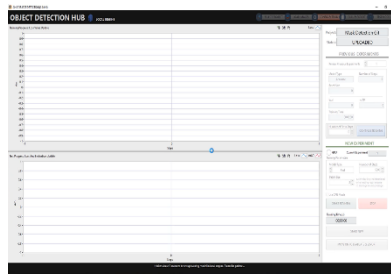
ODHUB – Object Detection Software

1. ODHUB DESIGN

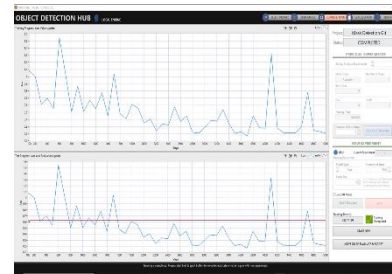
ODHUB is the graphical software that allows customers to design a custom object detection system without any coding. With ODHUB, you can do anything with your imagination for smart computer vision tasks, such as defect detection in manufacturing, license plate recognition/detection, people counting, food detection, etc.



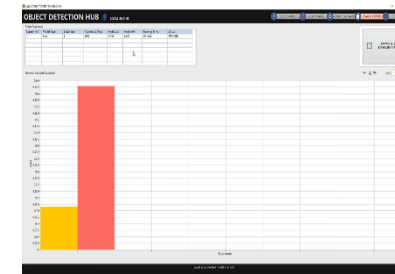
Step 1
Load data



Step 2
*Configure Neural
Network*



Step 3
Train Neural Network



Step 4
*Evaluate Neural
Network*



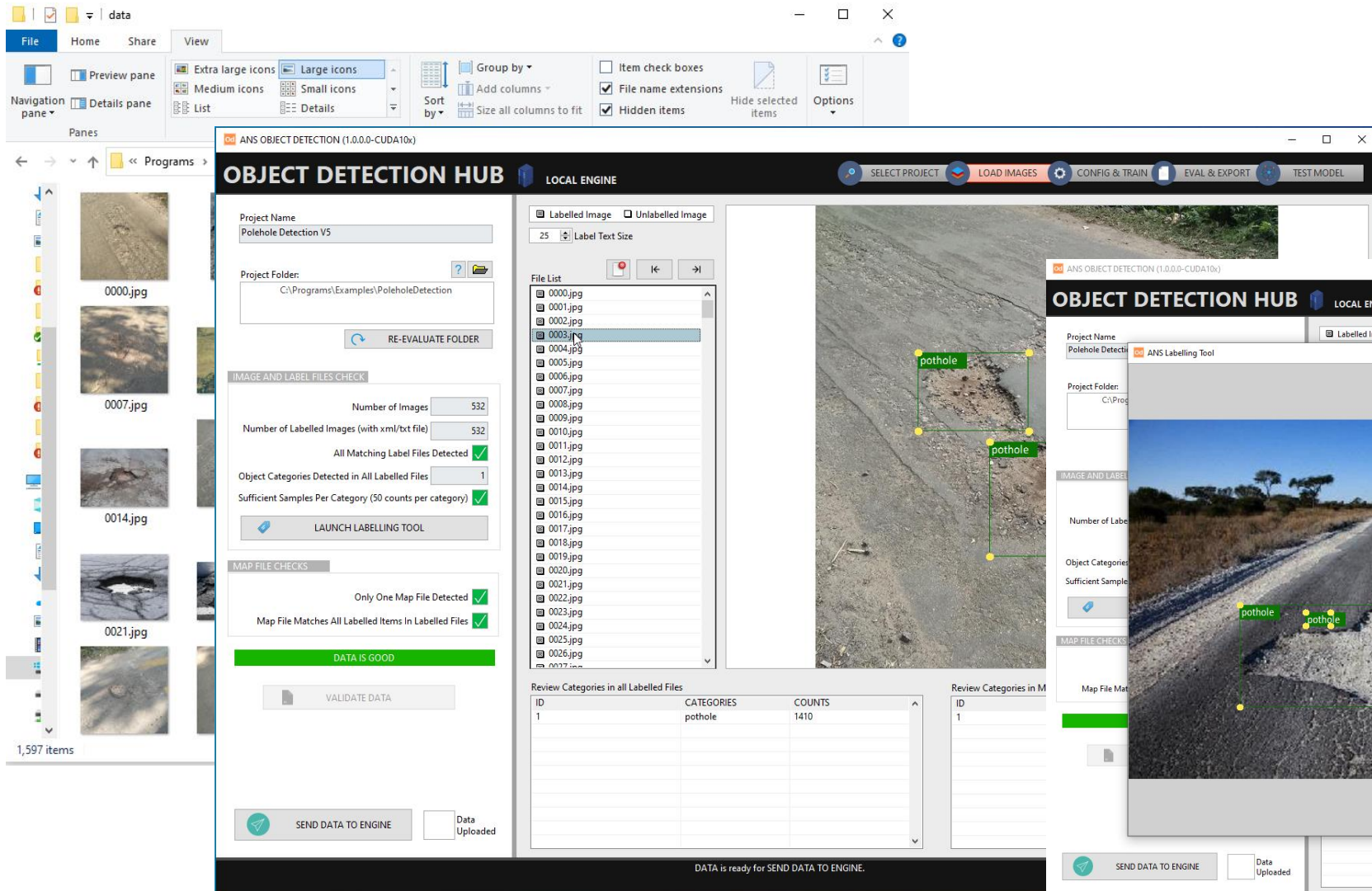
Step 5
*Test & Export Neural
Network Model*



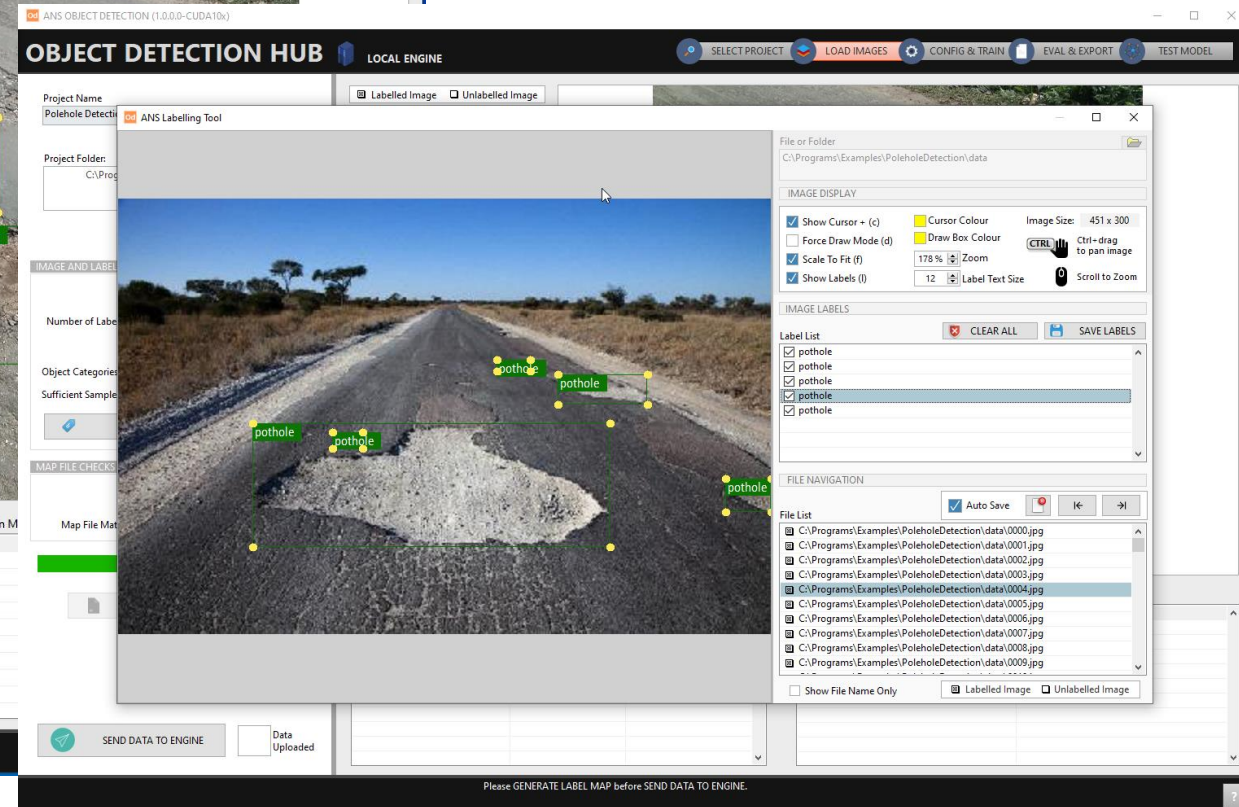
Please click on the play buttons to open demonstration video

ODHUB – Object Detection Software

Step 1: Load data



- Simple to organize data
- Support data checking and validation
- Support built-in label map tool



ODHUB – Object Detection Software

Step 2: Configure ML model

NEW EXPERIMENT

☐ GPU Current Experiment

Training Parameters

Model Type <input type="text" value="Fast"/>	Number of Steps <input type="text" value="2000"/>
Batch Size <input type="text" value="64"/>	Learning Rate <input type="text" value="0.001"/>

Training Time (s)

NEW EXPERIMENT

☐ GPU Current Experiment

Training Parameters

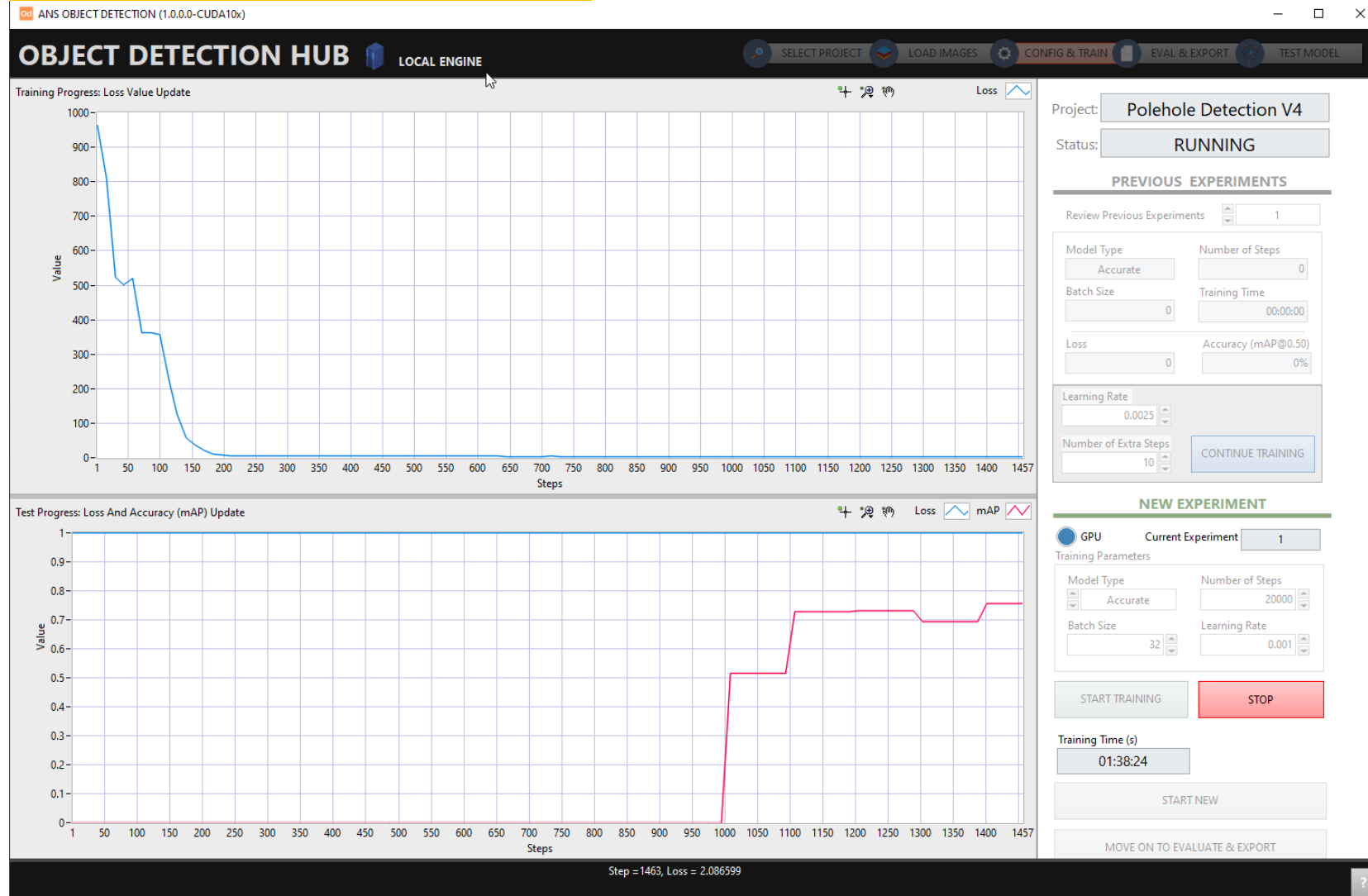
Model Type <div><div>✓ Fast</div><div>Accurate</div><div>Very Accurate</div><div>Custom</div></div>	Number of Steps <input type="text" value="2000"/>
Batch Size <input type="text" value=""/>	Learning Rate <input type="text" value="0.001"/>

Training Time (s)

- Easy to select ML: Fast, Accurate, Very Accurate, or Custom.
- Auto generate ML architecture to meet user requirements

ODHUB – Object Detection Software

Step 3: Load data



- Auto detect GPU to accelerate training process.
- Report model accuracy during training.
- Auto select the best model to avoid overfitting issue.

ODHUB – Object Detection Software

Step 4: Evaluate model

The screenshot displays the ODHUB Object Detection Software interface. At the top, the title bar reads "ANS OBJECT DETECTION (1.0.0.0-CUDA10x)". The main header is "OBJECT DETECTION HUB LOCAL ENGINE". Below this, there are navigation buttons: "SELECT PROJECT", "LOAD IMAGES", "CONFIG & TRAIN", "EVAL & EXPORT" (highlighted), and "TEST MODEL".

The "Result Summary" section contains a table with the following data:

Experiment	Model Type	Batch Size	Number of Steps	Final Loss	Final mAP (%)	Training Time	Status
1	Accurate	32	20000	1.501	78.870	03:18:43	TRAINING

To the right of the table is a button labeled "EXPORT SELECTED EXPERIMENT MODEL(S)".

The "Review Loss And Evaluation" section features a bar chart. The y-axis is labeled "Score" and ranges from 0 to 1.55. The x-axis is labeled "Experiment". There are two bars: a yellow bar for Experiment 1 with a score of approximately 1.5, and a red bar for Experiment 2 with a score of approximately 0.8.

At the bottom, there is a "Panels" section with a "Layout" tab. It shows a file explorer view for "PoleholeDetection > Polehole_Detection_V4 > 1". The file list includes:

Name	Date modified	Type	Size
model.zip	30/06/2021 1:13 PM	WinRAR ZIP archive	463,021 KB

At the bottom right, there is a search bar with the text "Search 1".

On the right side of the slide, there is a list of features:

- Intuitive UI to report model accuracy and its performance.
- Export model into raw format that can be used in any APIs
- Support exporting models into formats that are suitable for edge devices: Jetson series, Raspberry PI, etc.

ODHUB – Object Detection Software

Step 4: Test model

The screenshot displays the ODHUB Object Detection Software interface. The title bar reads 'ANS OBJECT DETECTION (1.0.0.0-CUDA10x)'. The main header is 'OBJECT DETECTION HUB LOCAL ENGINE'. The navigation bar includes 'SELECT PROJECT', 'LOAD IMAGES', 'CONFIG & TRAIN', 'EVAL & EXPORT', and 'TEST MODEL' (which is highlighted).

On the left, the 'TEST MODEL' panel shows a selection of an experimental model (labeled '1') and a model named 'Accurate'. It displays 'Final Loss' as 1.50062 and 'mAP' as 78.87. Below this, the 'SELECT IMAGE SOURCE' section has three options: 'CURRENT PROJECT IMAGES' (selected), 'OTHER IMAGES', and 'WEB CAM'. The 'CURRENT PROJECT IMAGES' option is further detailed with icons for 'Use current project evaluated images', 'Use still images from selected folder', and 'Use live stream from a web cam'. A 'Load Test Images...' button is present, with a note '-- USING CURRENT PROJECT TEST IMAGES--'.

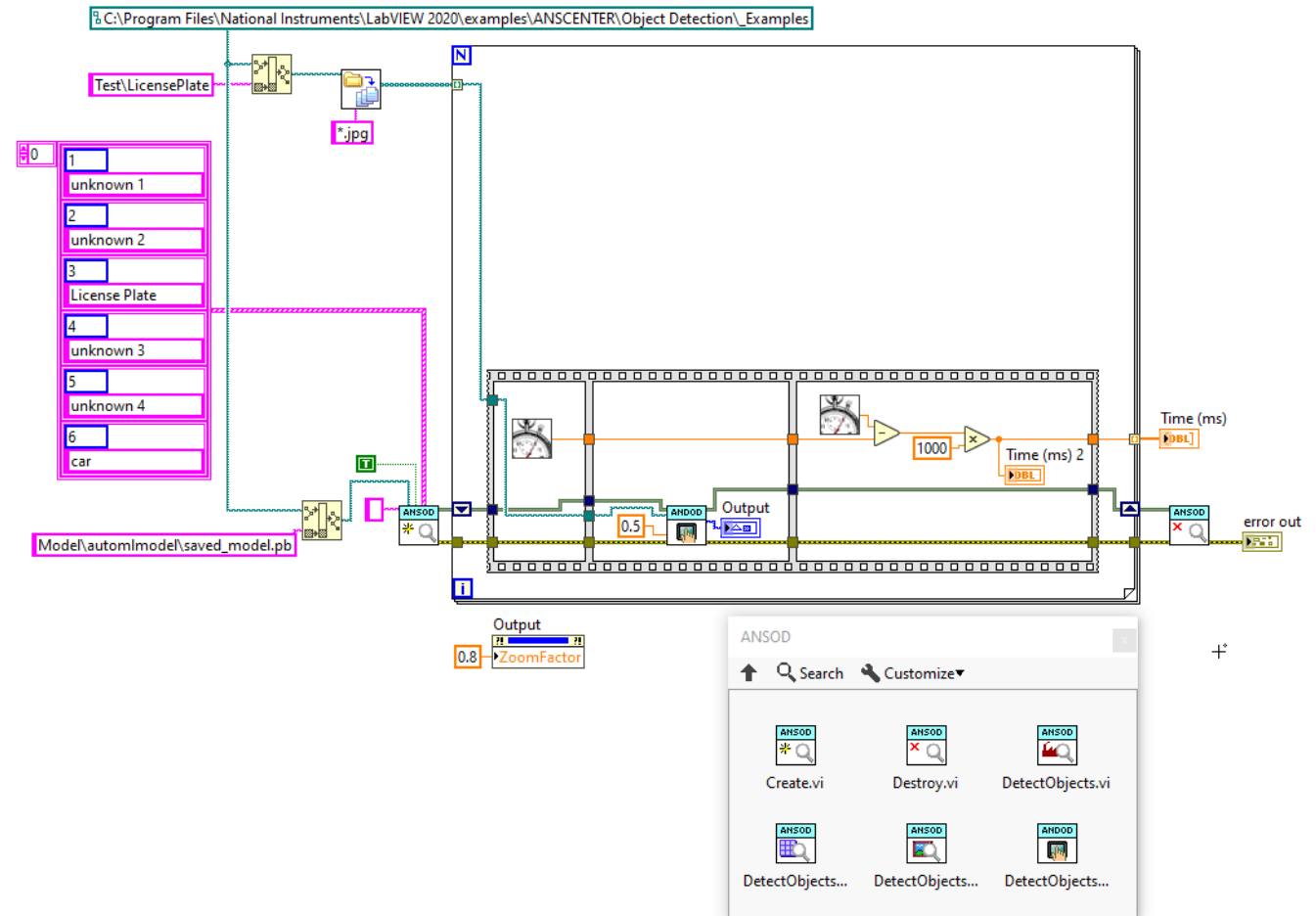
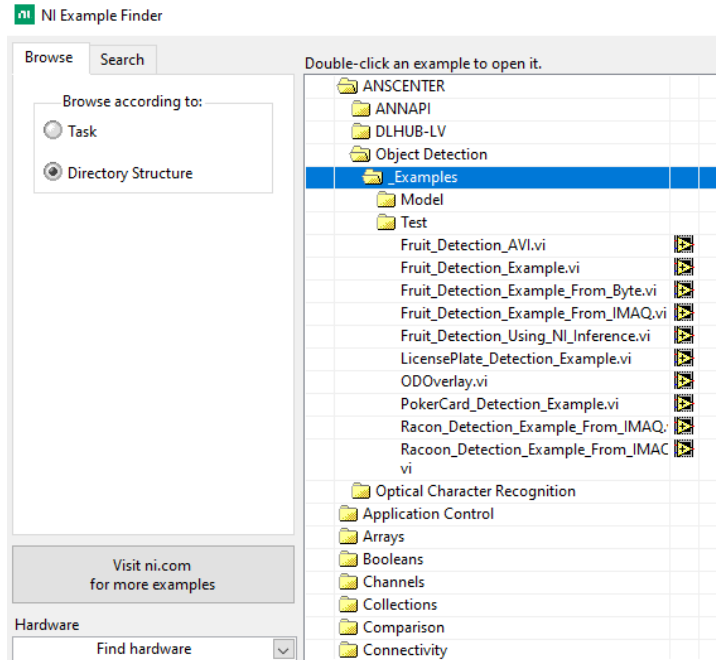
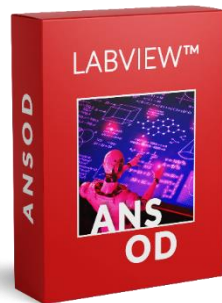
The central 'Image' panel shows a road surface with several potholes. Each pothole is enclosed in a green bounding box with yellow corner markers. The bottom status bar indicates 'Valid model selected.'

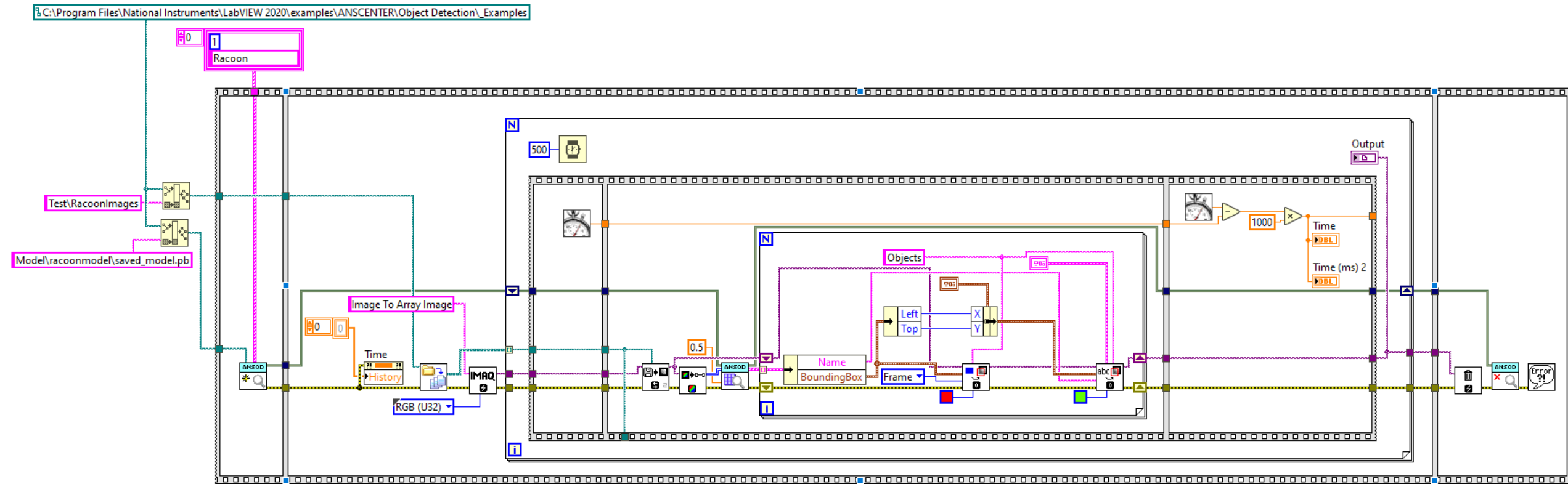
On the right, the 'Result' section displays a table of detection results. The table has columns for 'Object', 'Left', 'Top', 'Right', 'Bottom', and 'Score'. The 'Object Detection Time' is shown as 122 ms. The table lists six potholes with their respective bounding box coordinates and scores.

Object	Left	Top	Right	Bottom	Score
pothole	143	72	253	118	0.94
pothole	300	81	361	102	0.93
pothole	400	191	539	282	0.90
pothole	93	130	134	146	0.85
pothole	334	114	372	135	0.76
pothole	323	58	378	72	0.73

- Allow users to test model performance without other programming environments.
- Support IP cameras for real-time verification.
- Inspect model inference time.

2.1 ODHUB DEPLOYMENT: Native LabVIEW Integration





2.3 ODHUB DEPLOYMENT: Edge device deployment and LabVIEW integration via TCP/IP communication

Design



DESIGN & DEPLOY A CUSTOM MACHINE LEARNING MODEL (NO PROGRAMMING SKILLS)



DESIGN & DEPLOY A CUSTOM DEEP LEARNING MODEL (NO PROGRAMMING SKILLS)



DESIGN & DEPLOY A CUSTOM OBJECT DETECTION MODEL (NO PROGRAMMING SKILLS)



Deployment



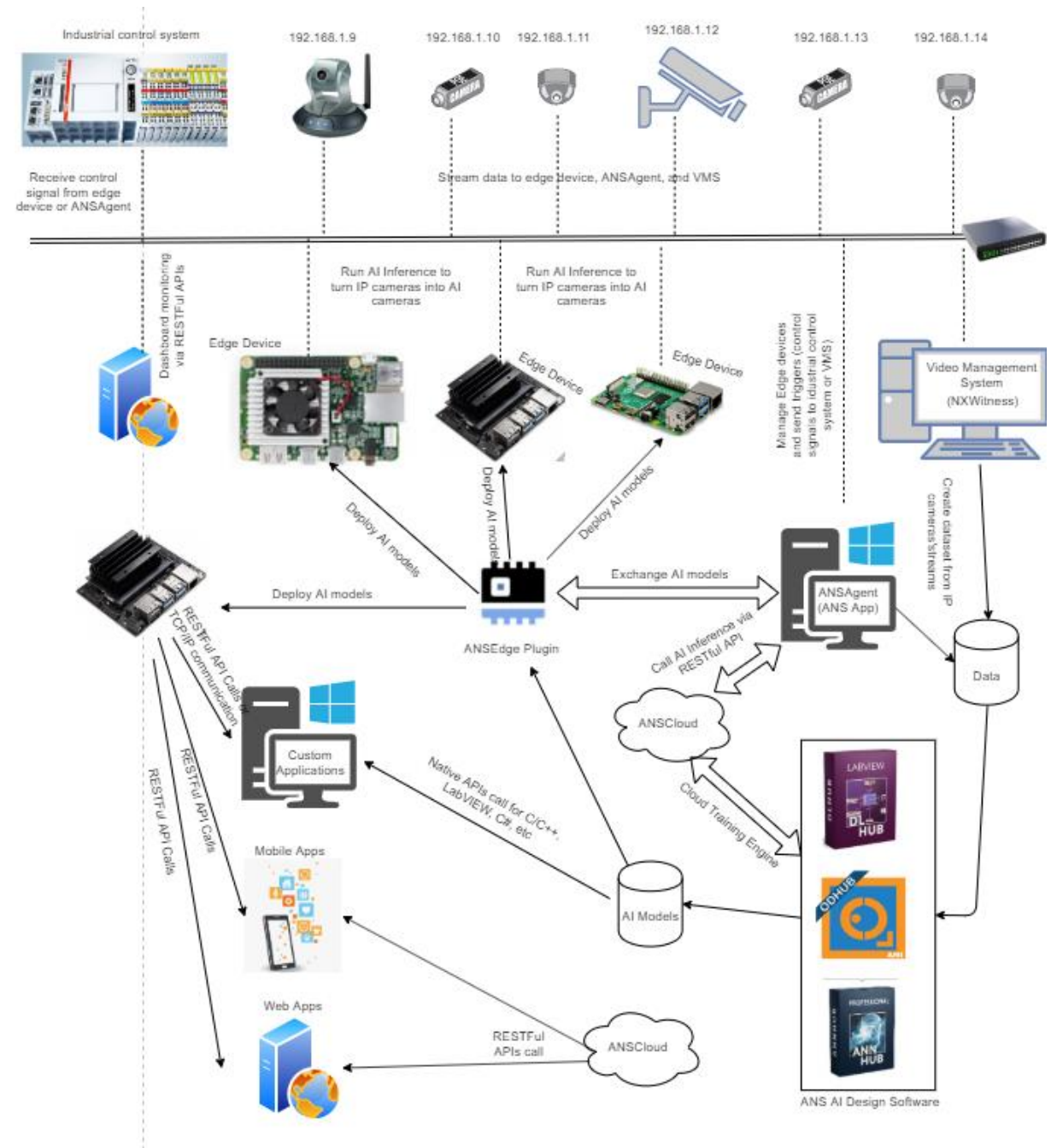
ANSAGENT: HOW TO ADD AI MODELS TO ANY SURVEILLANCE IP CAMERA SYSTEMS FOR INDUSTRIAL APPLICATIONS



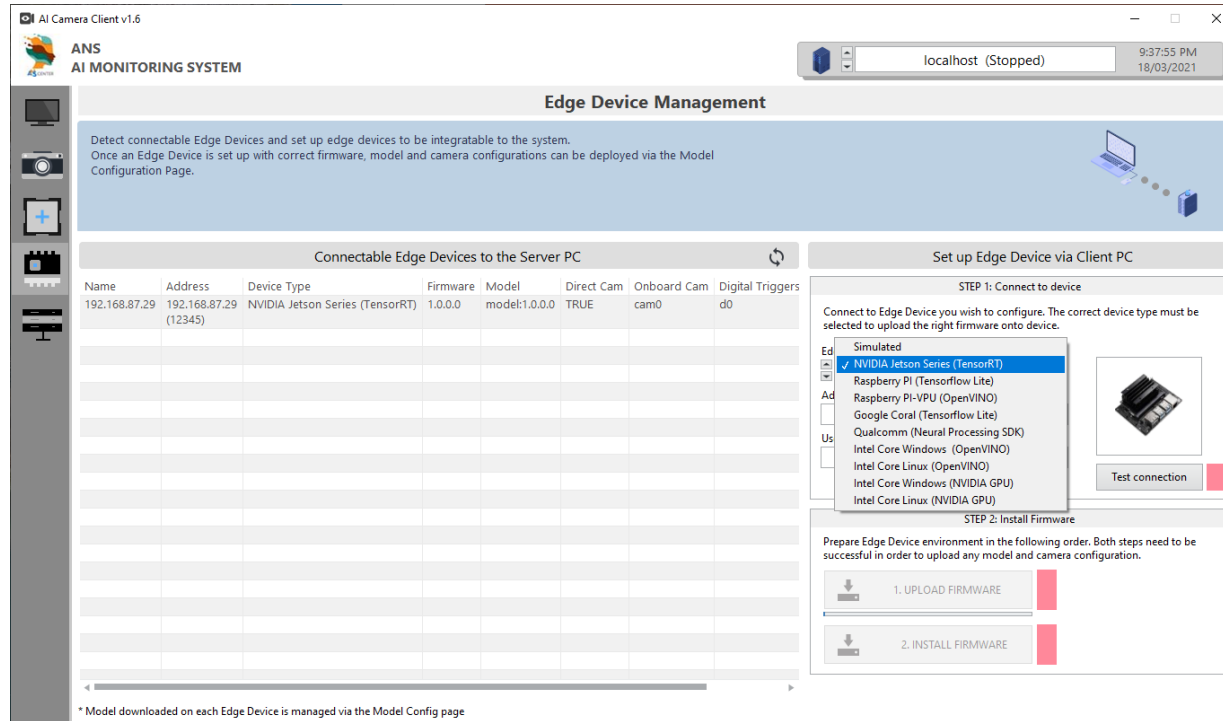
ANSEGE: HOW TO DEPLOY AI MODELS ON EDGE DEVICES FOR IOT APPLICATIONS



ANS CLOUD: TRAIN & HOST AI MODELS FOR SERVERLESS APPLICATIONS

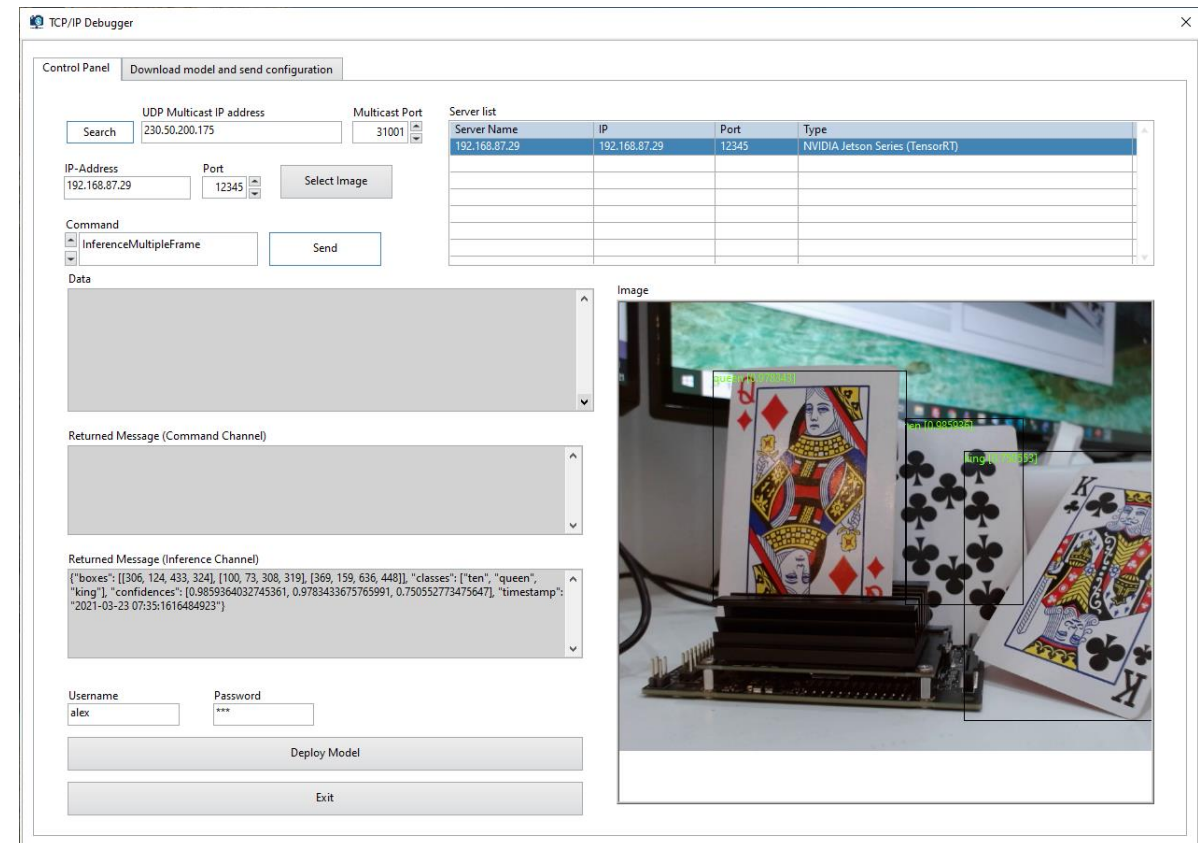


2.3 ODHUB DEPLOYMENT: Edge device deployment and LabVIEW integration via TCP/IP communication



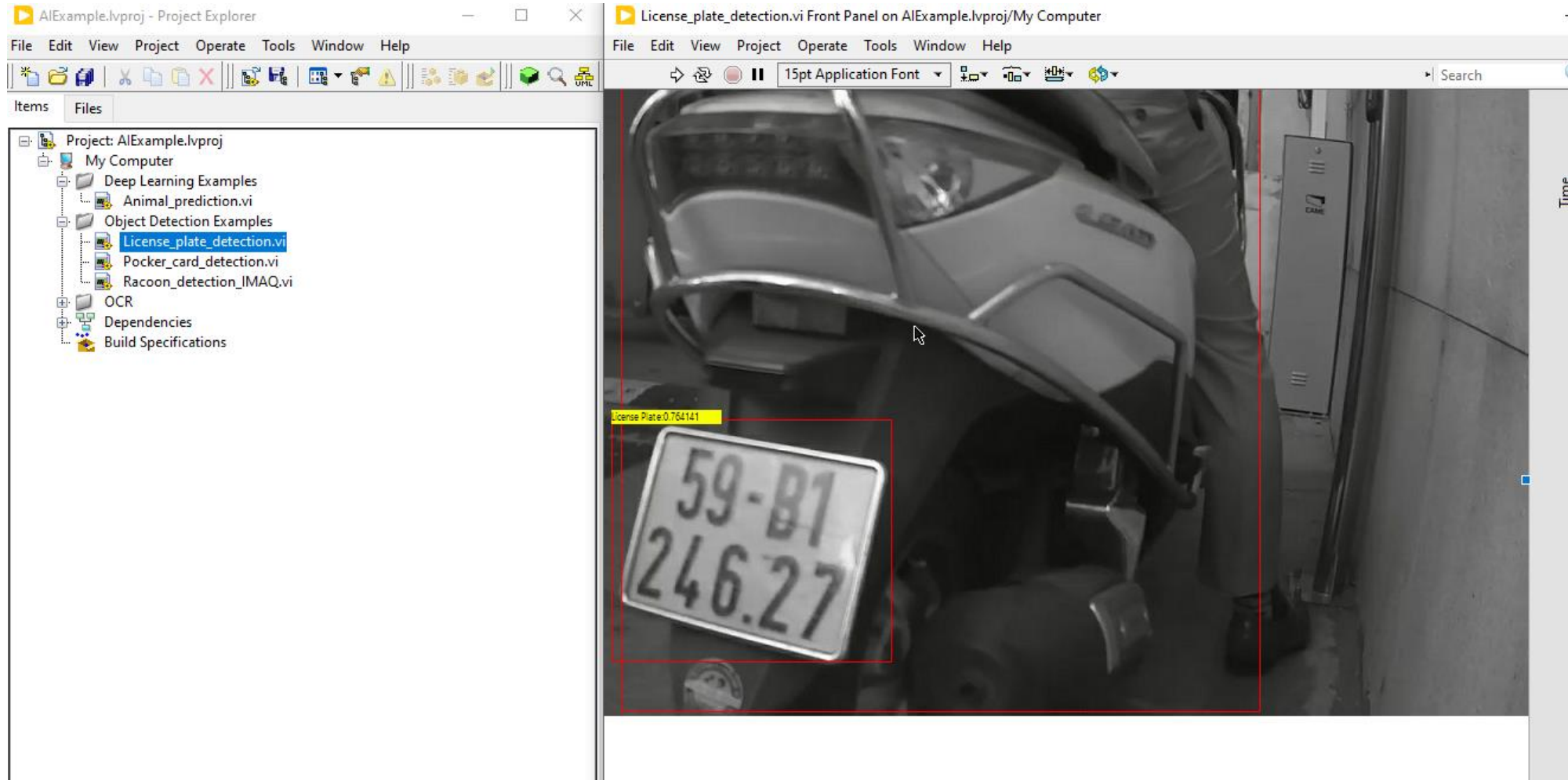
Deploy AI models into an edge device

Communication between LabVIEW app with edge device



ODHUB – Object Detection Software

3. Demo and questions





Artificial Neural Network Solution (ANS) Direct Pty Ltd

THANK YOU

FOR YOUR INTEREST IN OUR PRODUCTS



For more information, please visit us on:

Website: <https://www.anscenter.com>

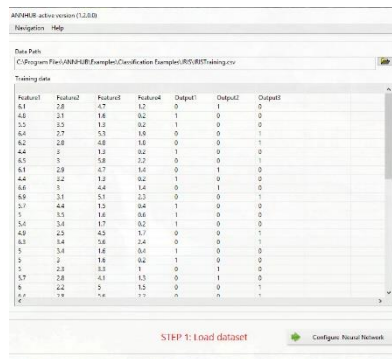
Youtube Channel: <https://www.youtube.com/c/ANSCENTER>

Facebook Page: <https://www.facebook.com/anscenter.official>

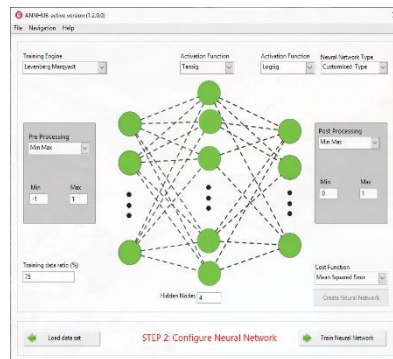
ANNHUB – Machine Learning Platform

1. MACHINE LEARNING DESIGN

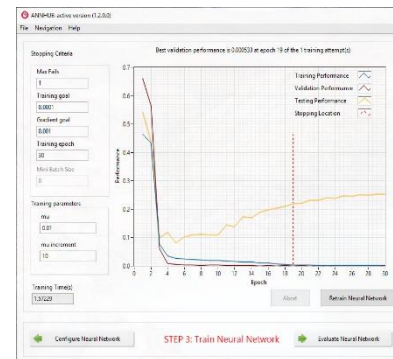
ANNHUB is a machine learning platform that allows machine learning design, train, and validation without any programming. Users can develop machine learning models to tackle real-life industrial solutions by merely clicking through the guided steps and utilizing auto-recommended training and validation features. It is ideal for engineers who do not have a profound knowledge of machine learning and programming skills to design a proper neural network in only 5 steps.



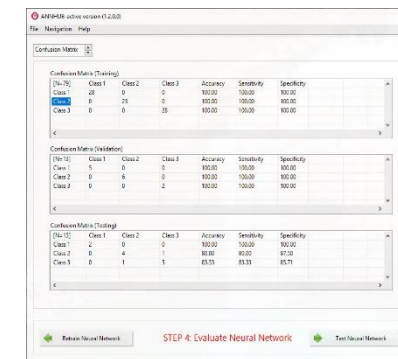
Step 1
Load data



Step 2
Design Neural Network



Step 3
Train Neural Network



Step 4
Evaluate Neural
Network



Step 5
Export Neural Network
Model

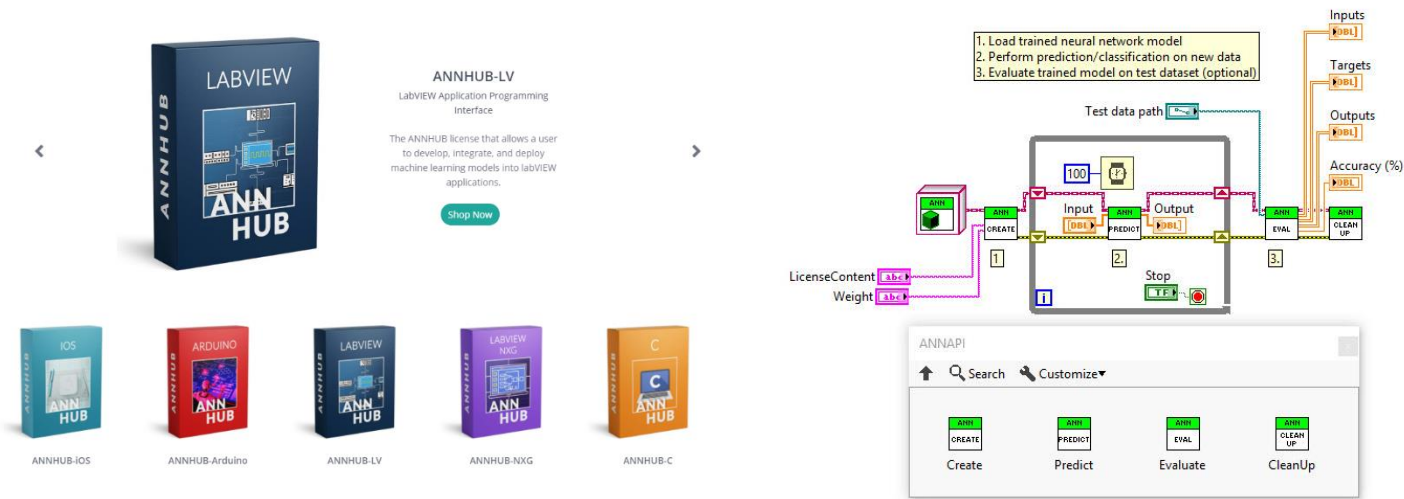


Please click on the play buttons to open demonstration video

ANNHUB – Machine Learning Platform

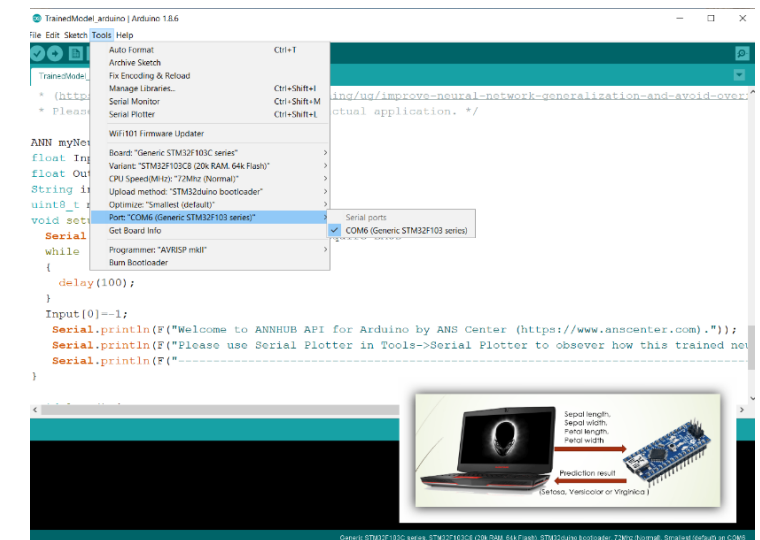
2. MACHINE LEARNING DEPLOYMENT/ INFERENCE

Machine learning deployment/inference is a difficult task as it involves hardware and software dependencies. ANSCENTER simplifies this task by providing Machine Learning Application Programming Interface (API) for various programming languages. Users can effortlessly develop the model in ANNHUB and integrate it into any applications with minimal API calls.



ANNHUB Application Programming Interface for various programming languages

Machine Learning Deployment/Inference in LabVIEW environment

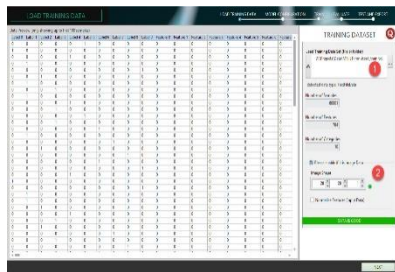


Machine Learning Deployment/Inference into IoT Arduino devices

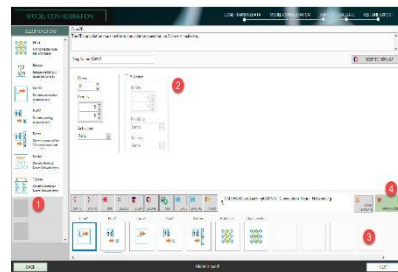
DLHUB – Deep Learning Platform

1. DEEP LEARNING DESIGN

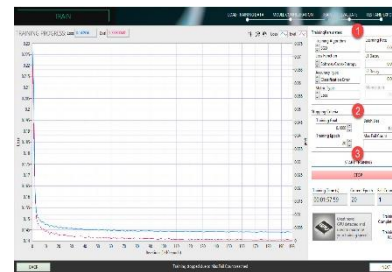
DLHUB is a graphical deep learning platform that enables deep learning design without requiring any programming skills. Users can develop, train, and evaluate deep learning models to tackle real-life industrial solutions by merely drag-and-drop technique and utilizing the model verification feature. It requires only five simple steps to design a deep learning model properly without any profound knowledge of deep learning and Python or C# programming language.



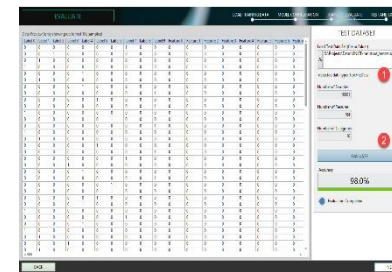
Step 1
Load data



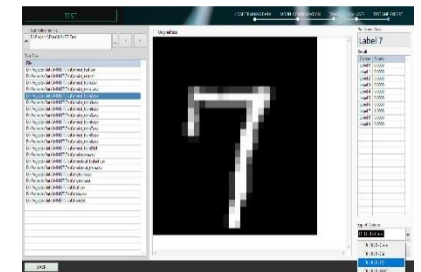
Step 2
*Design Deep Learning
Model*



Step 3
Train Deep Learning Model



Step 4
*Evaluate Deep Learning
Model*



Step 5
*Export Deep Learning
Model*



Please click on the play buttons to open demonstration video

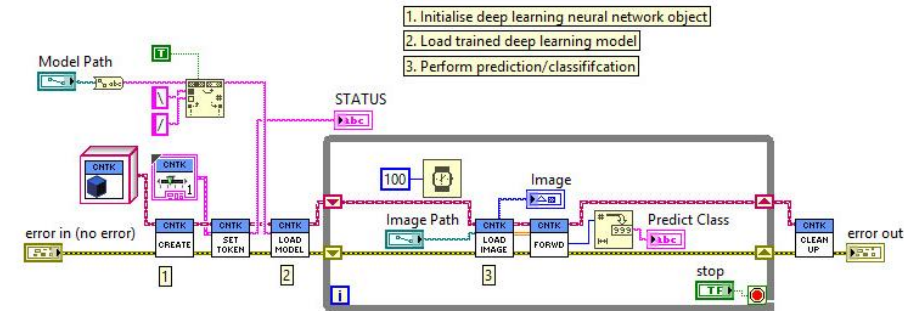
DLHUB – Deep Learning Platform

1. DEEP 2. DEEP LEARNING DEPLOYMENT/ INFERENCE

Deep learning deployment/inference is a difficult task as it involves hardware and software dependencies. ANSCENTER simplifies this task by providing Deep Learning Application Programming Interface (API) for various programming languages. Users can effortlessly develop the model in DLHUB and integrate it into any applications with minimal API calls.



ANNHUB Application Programming Interface
for various programming languages



Deep Learning Deployment/Inference in
LabVIEW environment