Knowing When You Don't Know

Quantifying, utilizing, and reasoning about uncertainty in complex environments

Eric Heim, Senior Machine Learning Research Scientist
AI Division, Software Engineering Institute, Carnegie Mellon University

[Distribution Statement A] Approved for public release and unlimited distribution.

Copyright 2022 Carnegie Mellon University.
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.
The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.
[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.
This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.
Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
DM22-0883

The AI Division at the Software Engineering Institute

Shaping the future of AI, with a focus on AI Engineering, for a better world

We are helping to make AI as strategic advantage for defense and national security through the identification, creation, and proliferation of leading processes, practices, and tools for AI Engineering.

Within AI Engineering, we focus on

Scalable AI
Accommodate the size, speed, and complexity of mission needs

Robust and Secure AI
Operate reliably when faced with uncertainty or threat

Human-Centered AI
Designed with the goal of working with, and for, people

We do this by

Pushing forwards in strategic areas of AI, creating new areas of focus and research

Connecting academia, industry and government

Supporting key programs as trusted experts

Facilitating the building of an AI Engineering discipline

[Distribution Statement A] Approved for public release and unlimited distribution.

Building Robustness into AI is an intentional activity

Without robust approaches, the results can be brittle and overconfident. Utilizing robust AI is important in achieving reliable outcomes

Some things to consider

Models trained on accuracy are generally *not robust*

Utilizing accuracy as a sole metric may result in being overconfident on seen samples, and unpredictable results on unseen samples

Confidence estimates and Human-Centered Design can be a key part of establishing trust

Black-box approaches make it difficult for Humans to trust AI

The data used for training is generally not what you see at runtime

Not only is training data is difficult to get, but in live situations, data is unpredictable, unreliable and contains unknown-unknowns

False-positives and false-negatives are key issues in AI adoption

Unreliable performance in novel situations and missed classifications are top-of-mind concerns for AI

[Distribution Statement A] Approved for public release and unlimited distribution.

Our work in Robust AI focuses on these key challenges

We work across industry, government, and academia to explore Robust AI

We are actively working on

What are the right metrics for building robust AI?

We integrate well-known approaches, such as requirements engineering, and create metrics to introduce robustness at train-time

Calibrating AI models to represent true probabilities

Scores from AI models tend to be over-optimistic, and uncalibrated to what humans understand as probabilities. Calibration makes the outputs of the AI into metrics humans can understand

Building Robust AI that provides accurate information around (Uncertainty Quantification)

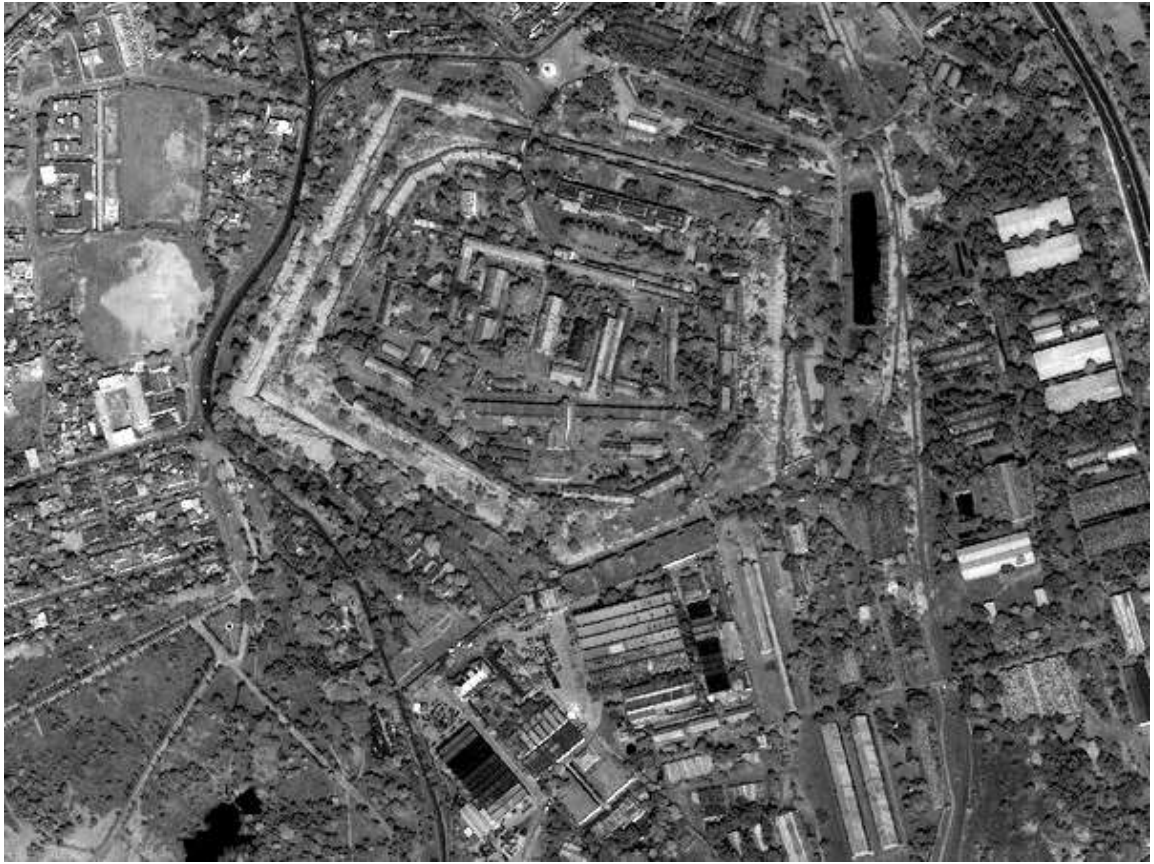
Creating approaches to build into the AI the ability to detect lower-confidence, and work in unknown-unknown environments with dataset drift.

Testing and Evaluation for Robustness

We perform tests to assess how robust AI systems are, and identify approaches to mitigate deficiencies

[Distribution Statement A] Approved for public release and unlimited distribution.

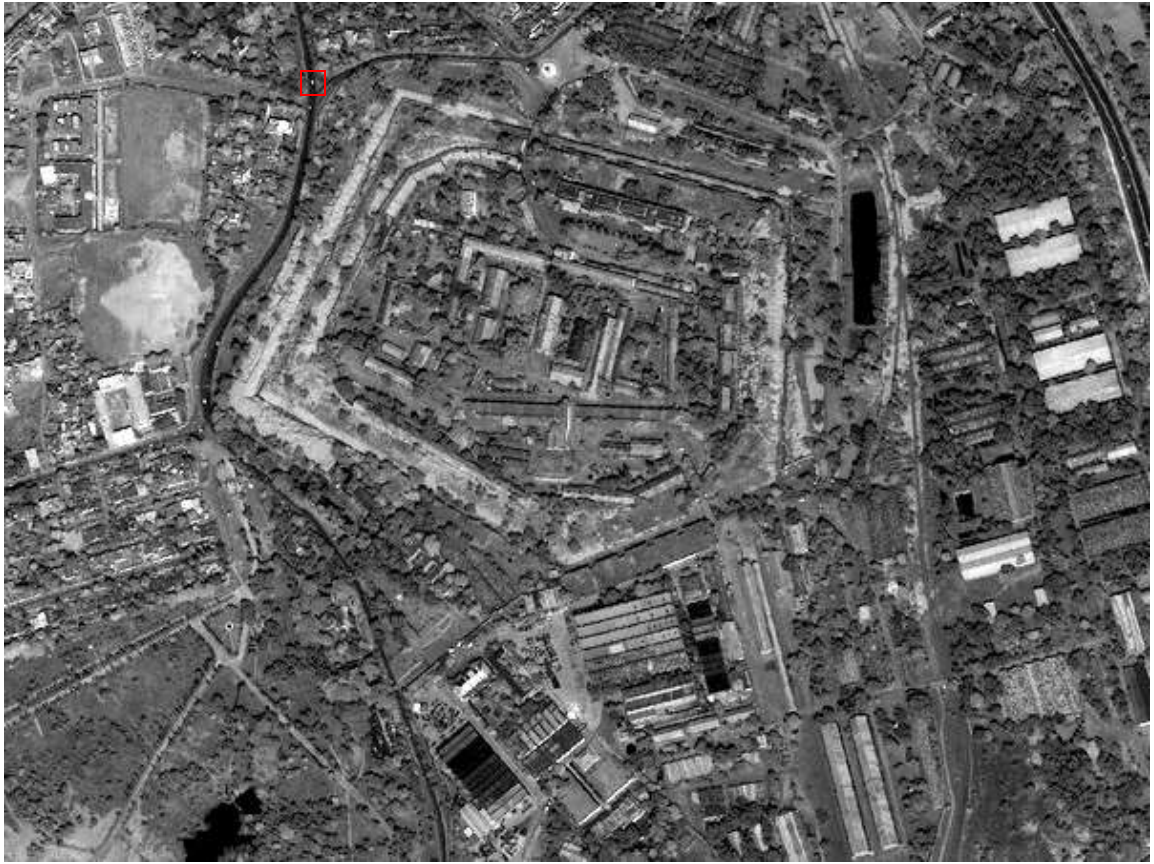
An Argument for Uncertainty in Machine Learned Models



[Distribution Statement A] Approved for public release and unlimited distribution.

An Argument for Uncertainty in Machine Learned Models

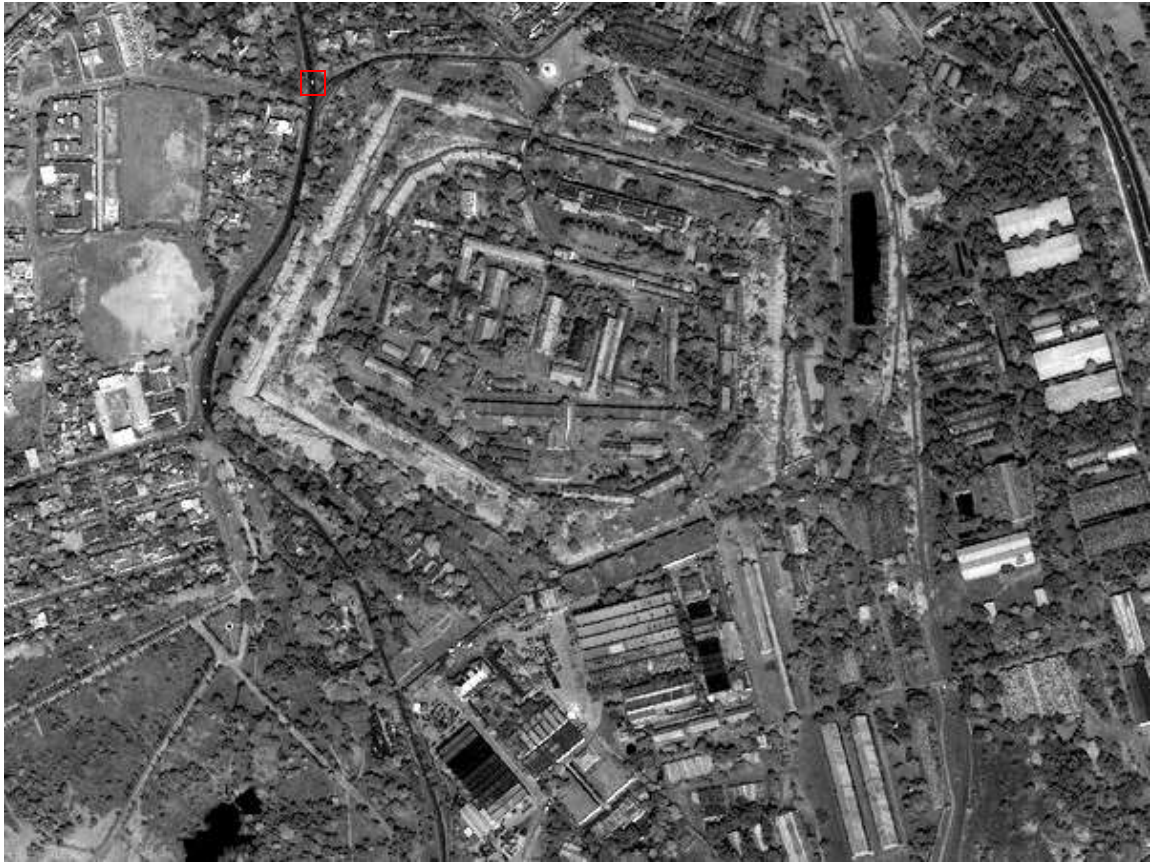
Predicted Class
Civilian Vehicle



[Distribution Statement A] Approved for public release and unlimited distribution.

An Argument for Uncertainty in Machine Learned Models

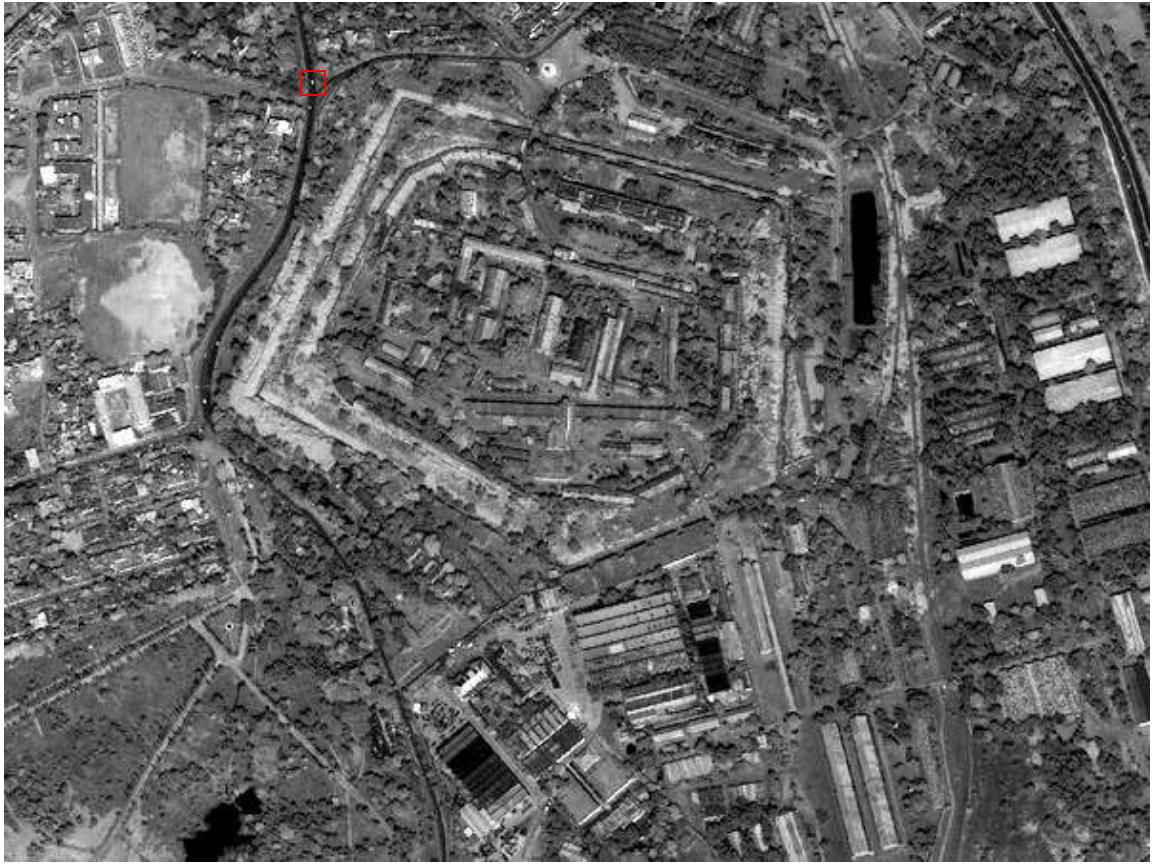
Predicted Class	Confidence
Civilian Vehicle	0.95



[Distribution Statement A] Approved for public release and unlimited distribution.

An Argument for Uncertainty in Machine Learned Models

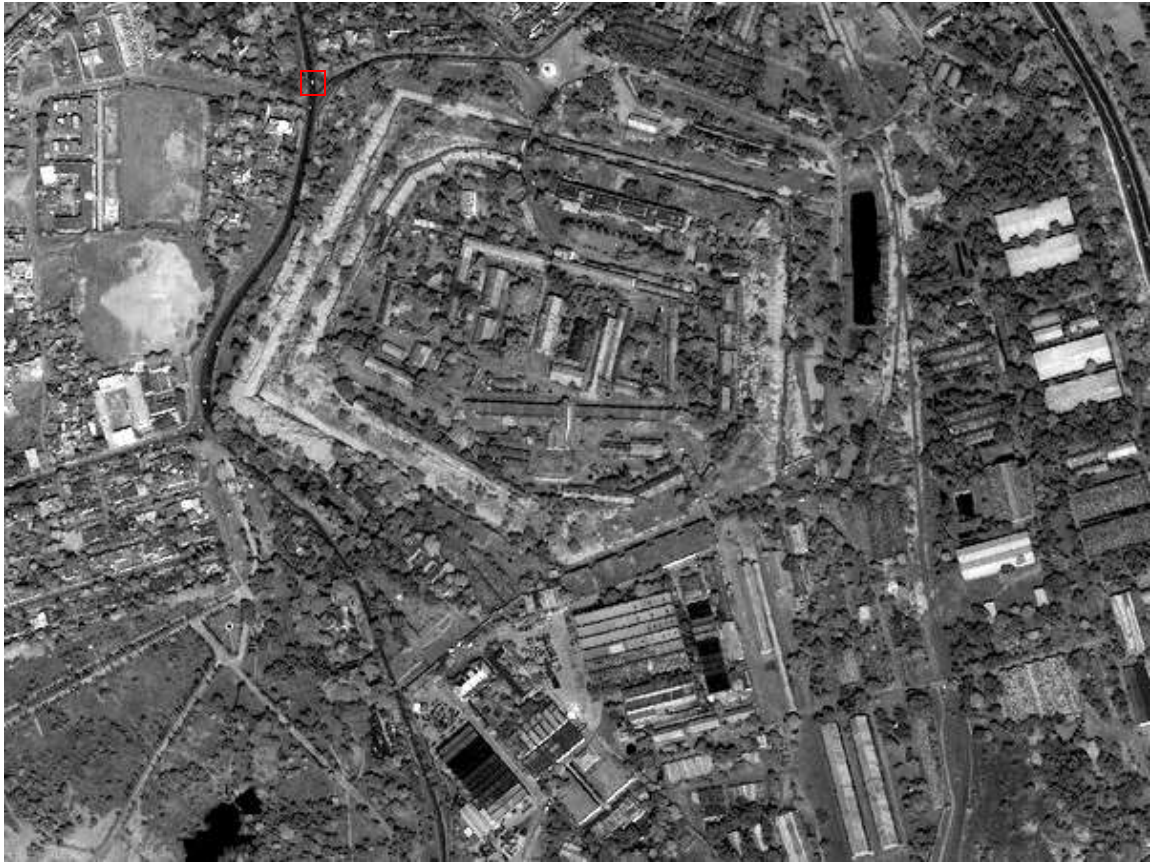
Predicted Class	Confidence
Civilian Vehicle	0.6



[Distribution Statement A] Approved for public release and unlimited distribution.

An Argument for Uncertainty in Machine Learned Models

Class	Confidence
Civilian Vehicle	0.6
Enemy Tank	0.35
…	…

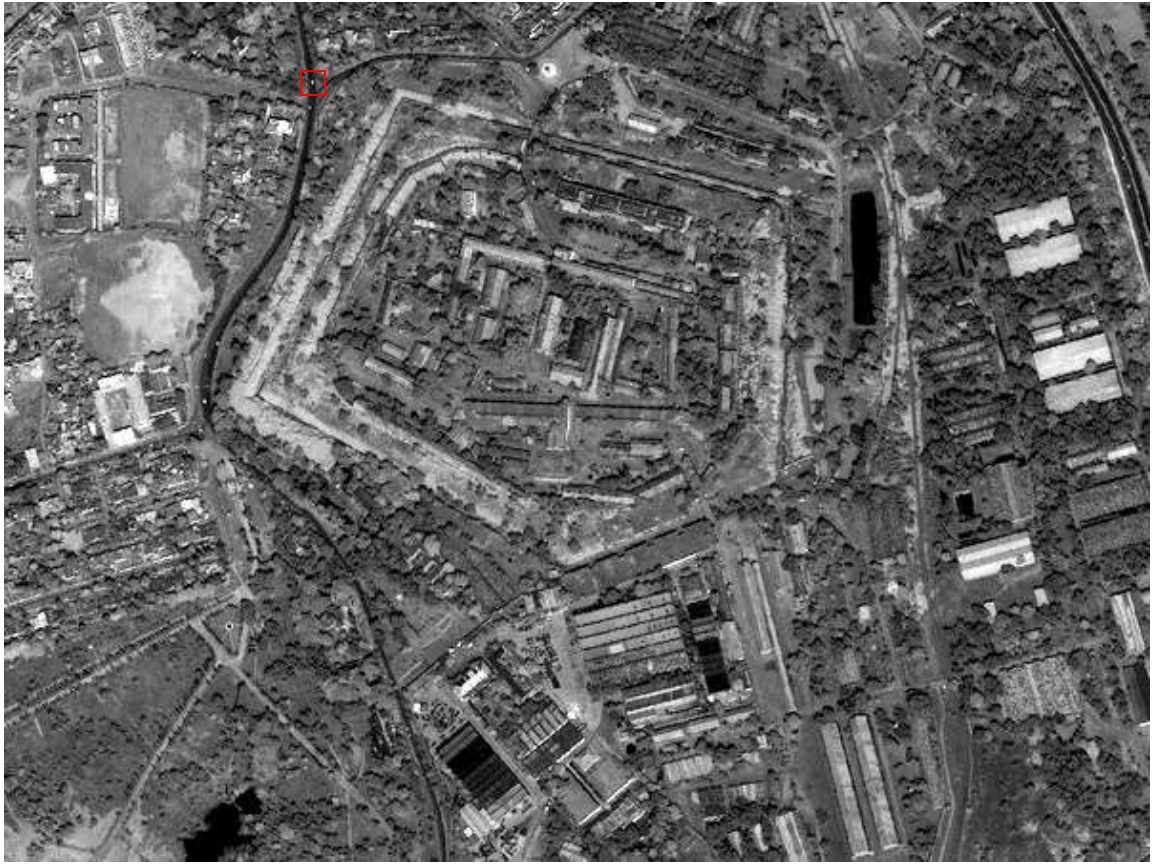


[Distribution Statement A] Approved for public release and unlimited distribution.

An Argument for Uncertainty in Machine Learned Models

Class	Confidence
Civilian Vehicle	0.6
Enemy Tank	0.35
…	…

Confidence/Uncertainty can lead to more informed decision-making.



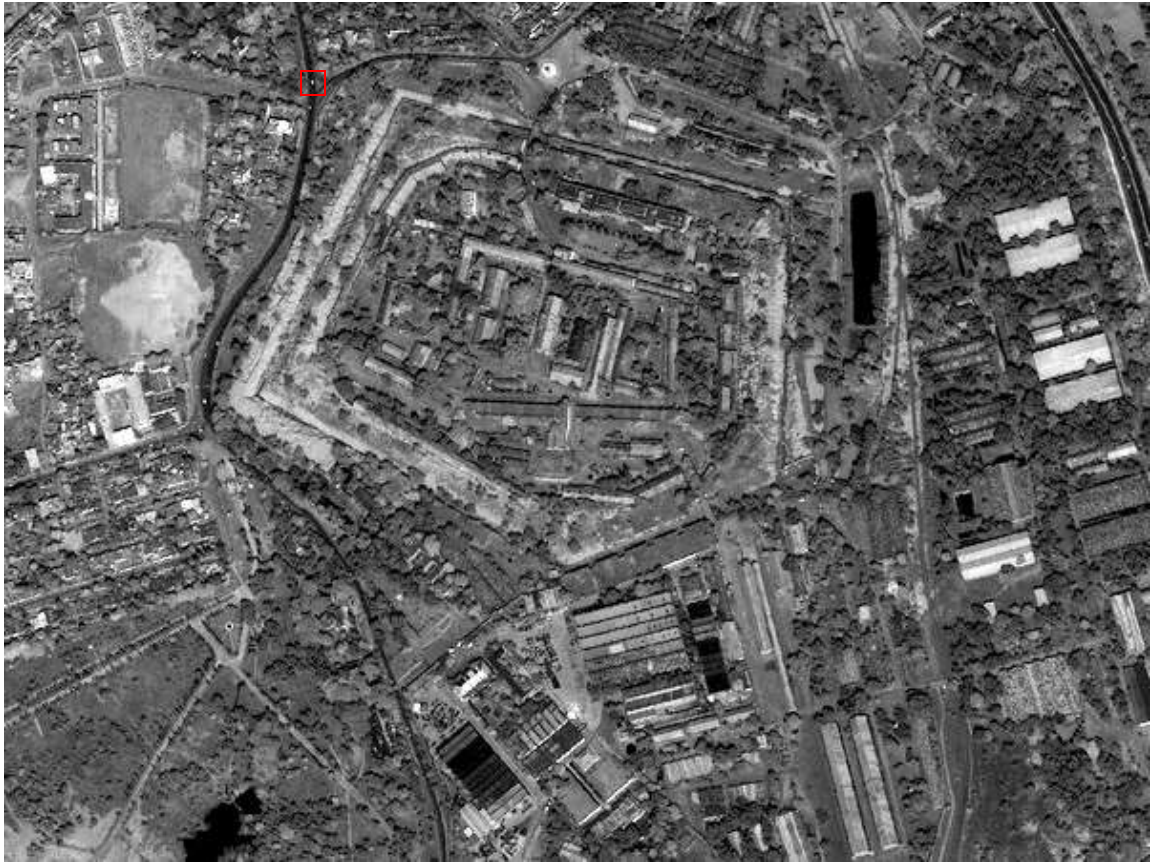
[Distribution Statement A] Approved for public release and unlimited distribution.

An Argument for Uncertainty in Machine Learned Models

Class	Confidence
Civilian Vehicle	0.6
Enemy Tank	0.35
…	…

```
if conf("Enemy Tank")
        > 0.25
```

Alert!



[Distribution Statement A] Approved for public release and unlimited distribution.

An Argument for Uncertainty in Machine Learned Models

Class	Confidence
Civilian Vehicle	0.6
Enemy Tank	0.35
…	…

```
if conf("Enemy Tank")
        > 0.25
```

Alert!

Confidence/Uncertainty can be used to make AI Systems more reliable.

[Distribution Statement A] Approved for public release and unlimited distribution.

Topics for Discussion

A couple questions for possible discussion:

- What definition of "uncertainty" is important for your problems?
- How do you make uncertainty estimates from a model accurate?
- How do you measure the quality of uncertainty estimates?

Roadmap:

1. Different (informal) definitions of calibration
2. Top-1 Calibration
   a) Is it a common problem?
   b) How to calibrate uncalibrated models
3. Metrics for evaluating calibration

[Distribution Statement A] Approved for public release and unlimited distribution.

Calibration: Intuition and Different Forms

One way to define the quality of predictive uncertainty estimates is through the assessment of a model's *calibration*.

Most works focus on *Top-1 calibration*

"Of all the times my model outputs maximum confidence 0.6, it should be right 60% percent of the time."

Class	Confidence
Civilian Vehicle	0.6
Enemy Tank	???
…	???

Other interpretations of classifier outputs are more appropriate to facilitate decision making in different contexts or focused evaluation.

Class	Confidence
Civilian Vehicle	0.6
Enemy Tank	0.35
…	…

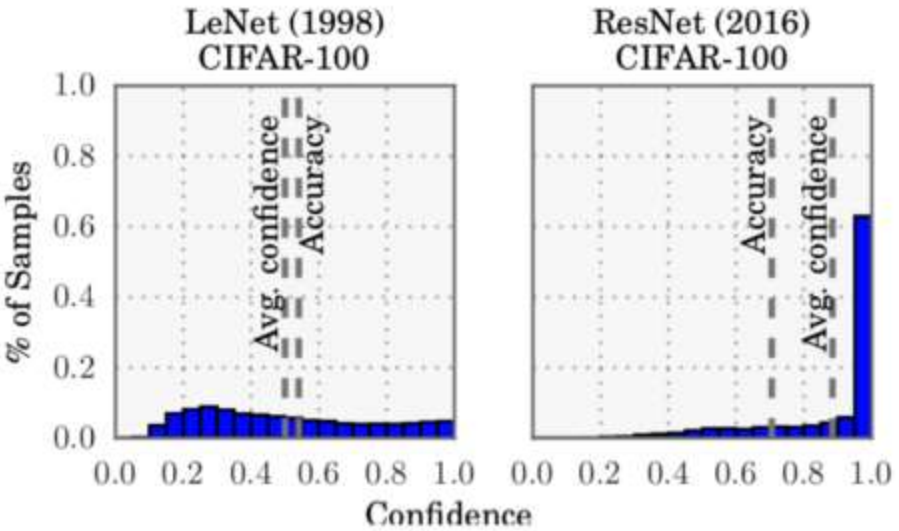
VS.

Confidence in class
Very Low
Low
Medium
High
Very High

[Distribution Statement A] Approved for public release and unlimited distribution.

Modern Neural Networks are Often Trained to be Miscalibrated

Problem: Many modern machine learning methods fail at accurately quantifying their confidence in predictions.

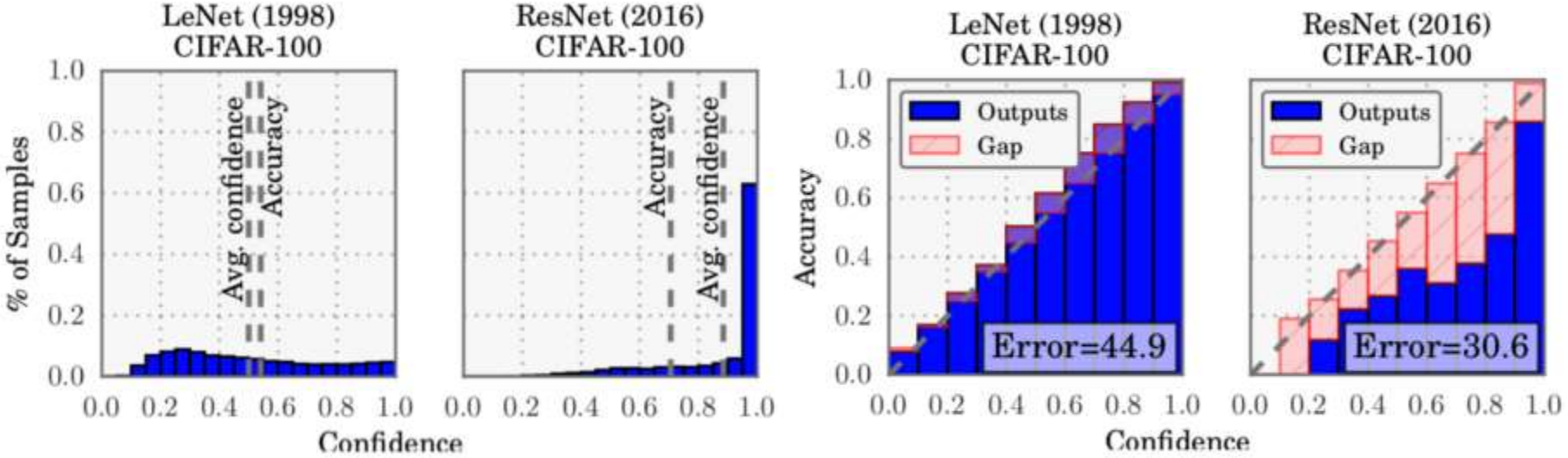


(Guo, et al; 2017)

[Distribution Statement A] Approved for public release and unlimited distribution.

Modern Neural Networks are Often Trained to be Miscalibrated

Problem: Many modern machine learning methods fail at accurately quantifying their confidence in predictions.



(Guo, et al; 2017)

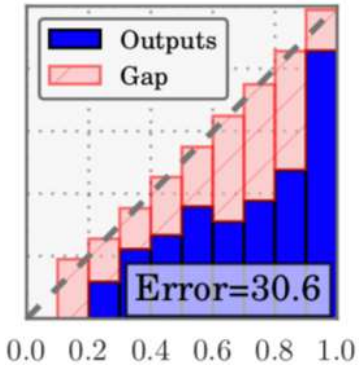
[Distribution Statement A] Approved for public release and unlimited distribution.

Techniques to Improve Top-1 Calibration

Generally, calibration techniques fall into one of two categories:

1. Train Time – Techniques used to augment standard training procedures
   - Examples: Data augmentation (AugMix), model definition (Bayesian Neural Networks), objective functions (Focal loss)
   - Positive: May already be part of your training pipeline
   - Negative: If they are not, require you to retrain you model
2. Post-training – Techniques that can be used on black box models
   - Examples: Temperature scaling, histogram binning
   - Positive: No need to retrain model if you are satisfied with it's accuracy
   - Negative: Requires additional validation data

$$\hat{q}_i = \max_k \sigma_{\mathrm{SM}}(\mathbf{z}_i/T)^{(k)}.$$

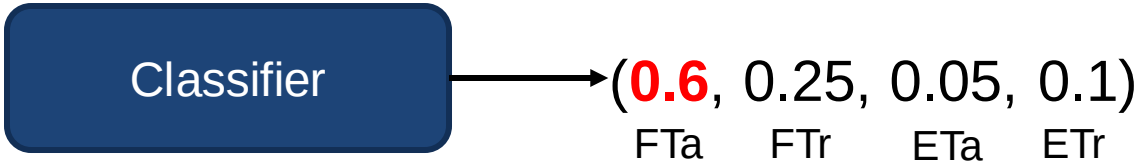


[Distribution Statement A] Approved for public release and unlimited distribution.

Evaluating Calibration Performance

Modern machine learning literature has focused on evaluating classifier calibration according to their **Top-1 Expected Calibration Error (ECE)** (Guo et al; 2017)

Classes = {Friendly Tank, Friendly Truck, Enemy Tank, Enemy Truck}



Top-1 Expected Calibration Error (ECE)

Considers only the most confident class in evaluating for calibration

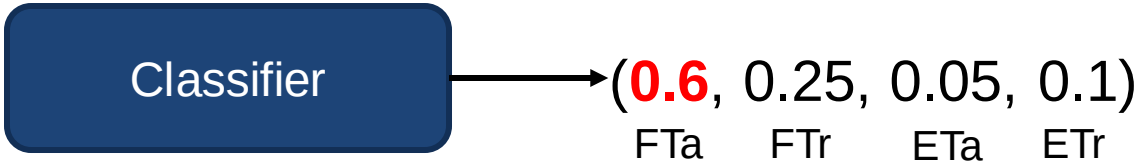
For all possible inputs that the classifier outputs 0.6 as the most confident class… 60% of the those inputs should be that class.

[Distribution Statement A] Approved for public release and unlimited distribution.

Evaluating Calibration Performance

Modern machine learning literature has focused on evaluating classifier calibration according to their **Top-1 Expected Calibration Error (ECE)** (Guo et al; 2017)

Classes = {Friendly Tank, Friendly Truck, Enemy Tank, Enemy Truck}



For all possible inputs that the classifier outputs 0.6 as the most confident class… 60% of the those inputs should be that class.

Top-1 Reliability Condition

$$\mathbb{P}[Y = \operatorname{argmax} g(X) \mid \max g(X)] = \max g(X)$$

[Distribution Statement A] Approved for public release and unlimited distribution.

Evaluating Calibration Performance

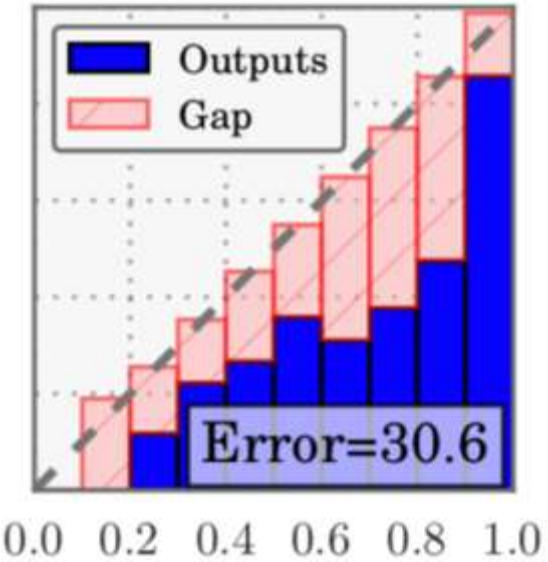
Modern machine learning literature has focused on evaluating classifier calibration according to their **Top-1 Expected Calibration Error (ECE)** (Guo et al; 2017)

Top-1 Reliability Condition

$$\mathbb{P}[Y = \operatorname{argmax} g(X) \mid \max g(X)] = \max g(X)$$

To measure how from from "calibrated" a model is, you derive an error metric:

1. Get the output probabilities from the model for a validation set.
2. Take the maximum value for all output probabilities
3. Bin those probabilities
4. Find the difference between the average maximum probability and the proportion of correct class predictions (error)
5. Take the average error over all bins.

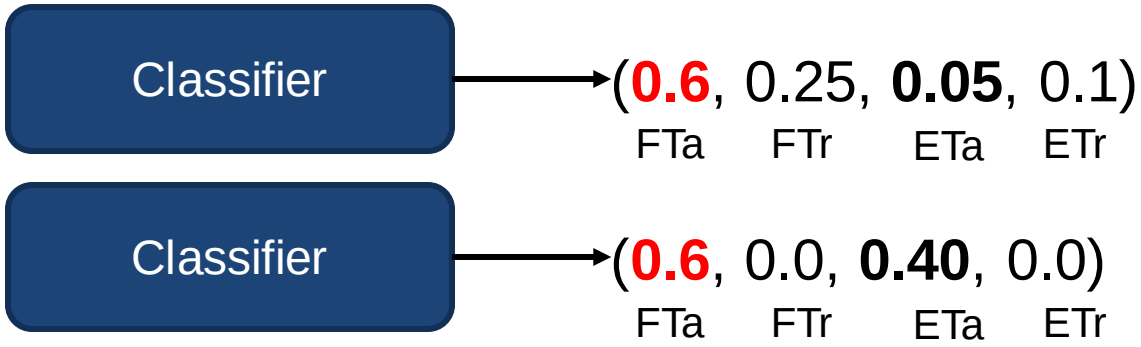


[Distribution Statement A] Approved for public release and unlimited distribution.

When Top-1 ECE Fails

Modern machine learning literature has focused on evaluating classifier calibration according to their **Top-1 Expected Calibration Error (ECE)** (Guo et al; 2017)

Classes = {Friendly Tank, Friendly Truck, Enemy Tank, Enemy Truck}



According to Top-1 ECE, these two classifiers *are considered the same*.
However, the two outputs can mean very different things with *mission context*.

This motivates the need for calibration metrics that are able to map to contexts and use cases that match realistic usage of deployment settings.

[Distribution Statement A] Approved for public release and unlimited distribution.

A statistical framework for context-specific ECE (Kirchenbauer, Oaks and Heim; 2022)

We adopt the framing of ECE from (Vaicenavicius et al; 2019), and adapt it to a generalized version of calibration error

Top-1 Reliability Condition

$$\mathbb{P}[Y = \operatorname{argmax} g(X) \mid \max g(X)] = \max g(X)$$

Generalized Reliability Condition

$$\mathbb{P}[t(Y) \mid o(g(X)] = o(g(X)$$

To measure how from from "calibrated" a model is, you derive an error metric:

1. Get the output probabilities from the model for a validation set.
2. **Apply a lens** to all output probabilities
3. Bin those probabilities
4. Find the difference between the average **lensed** probability and the **lensed** class predictions (error)
5. Take the average error over all bins.

[Distribution Statement A] Approved for public release and unlimited distribution.

Empirical Results

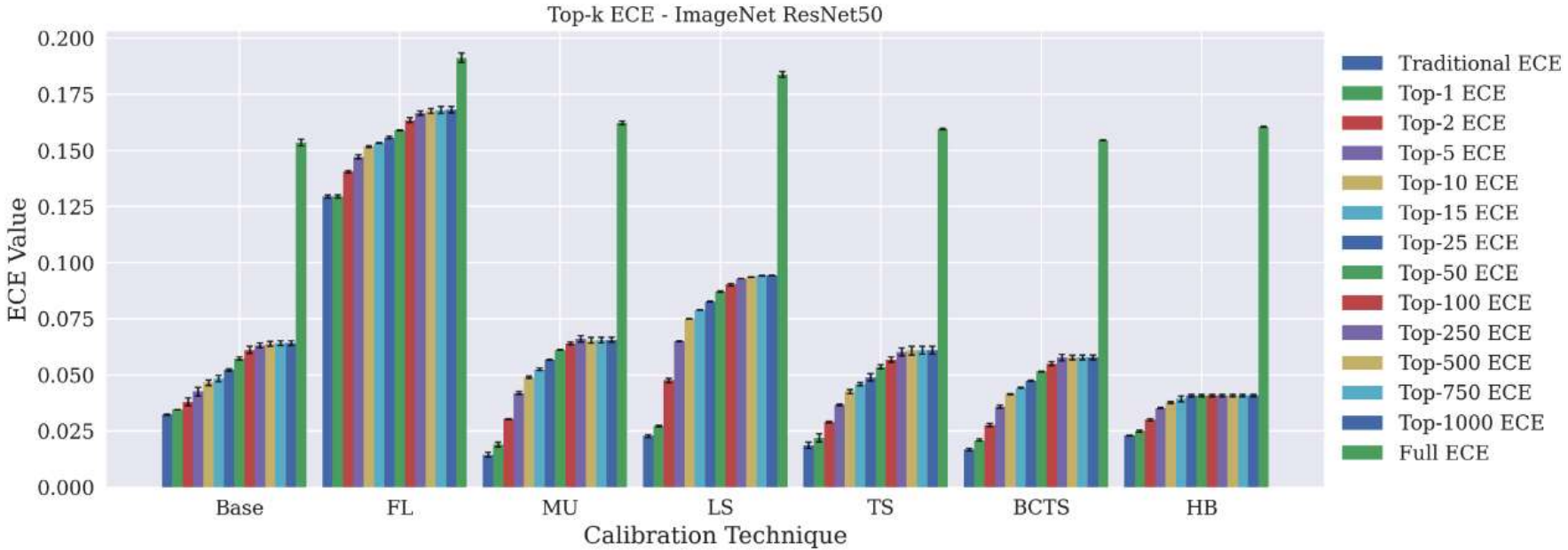
Goal: Evaluate a sampling of the current state-of-the-art in classifier calibration with respect to traditional Top-1 ECE and ECE metrics tailored to particular contexts.

[Distribution Statement A] Approved for public release and unlimited distribution.

Empirical Results

Goal: Evaluate a sampling of the current state-of-the-art in classifier calibration with respect to traditional Top-1 ECE and ECE metrics tailored to particular contexts.

Experiment #1: Top-k ECE

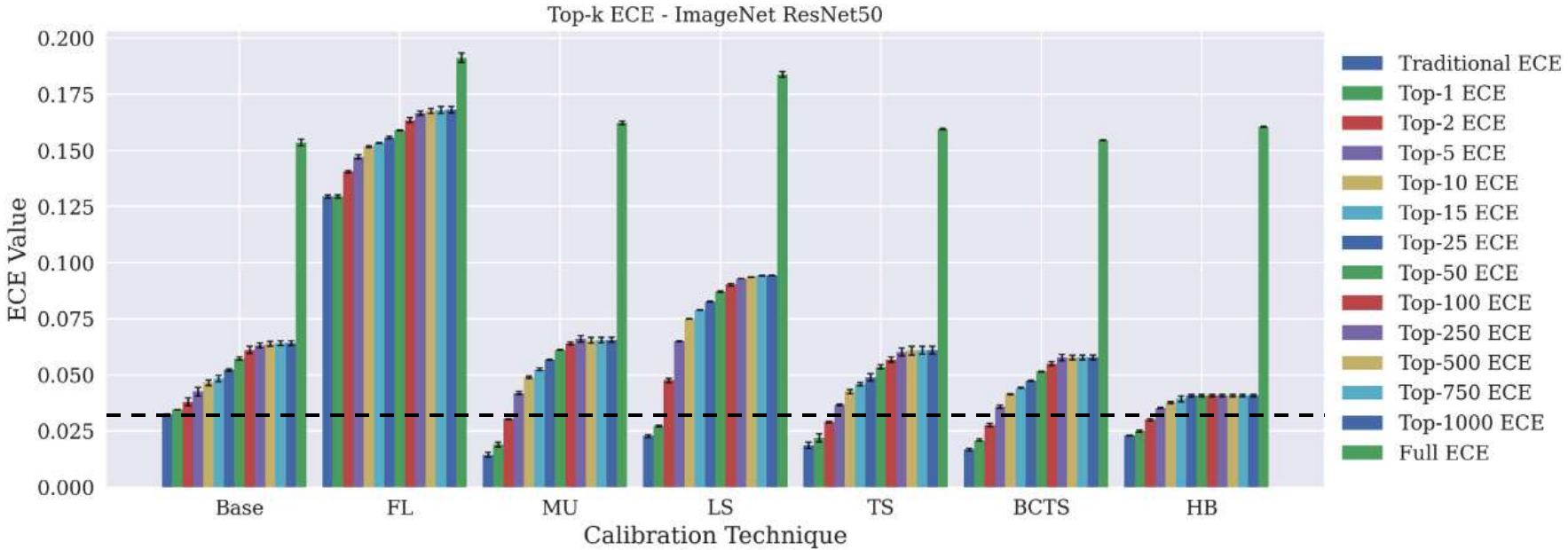


[Distribution Statement A] Approved for public release and unlimited distribution.

Empirical Results

Goal: Evaluate a sampling of the current state-of-the-art in classifier calibration with respect to traditional Top-1 ECE and ECE metrics tailored to particular contexts.

Experiment #1: Top-k ECE

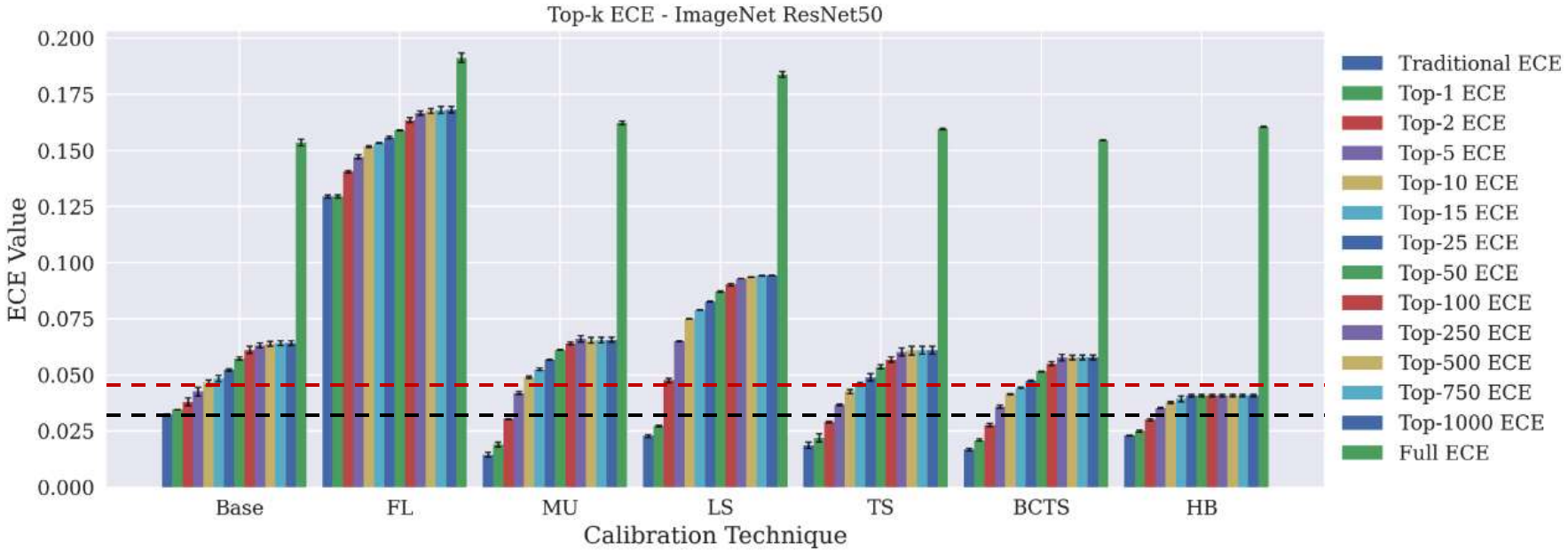


[Distribution Statement A] Approved for public release and unlimited distribution.

Empirical Results

Goal: Evaluate a sampling of the current state-of-the-art in classifier calibration with respect to traditional Top-1 ECE and ECE metrics tailored to particular contexts.

Experiment #1: Top-k ECE

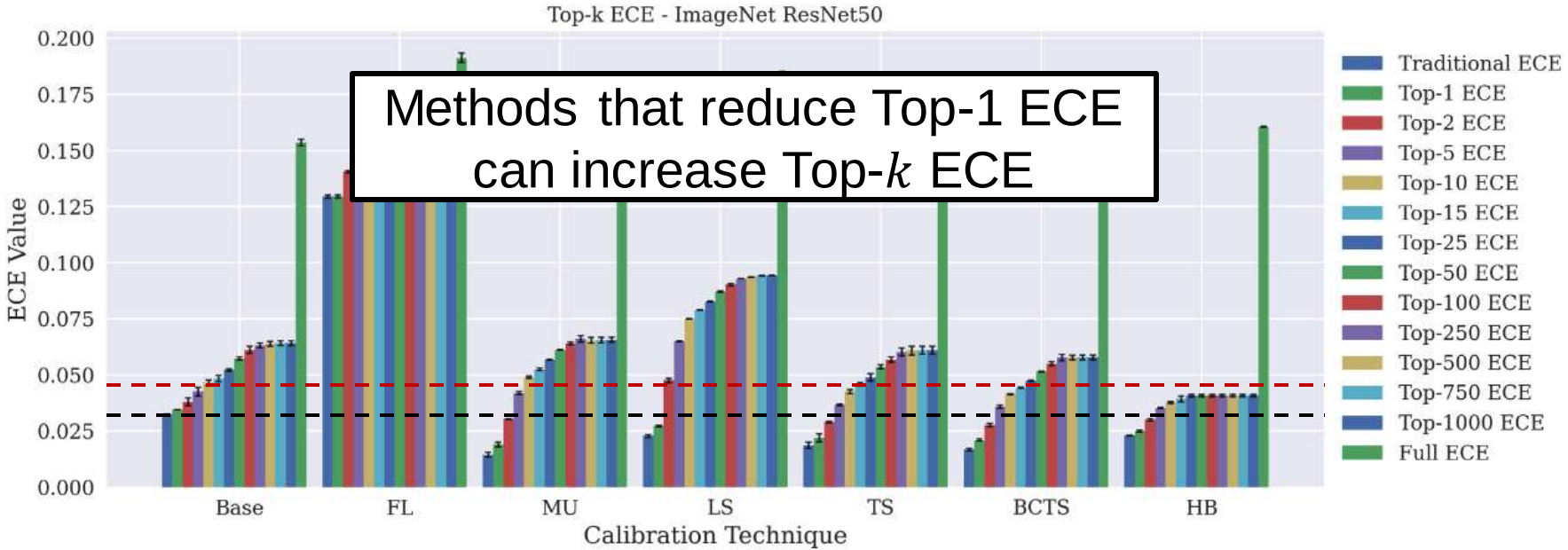


[Distribution Statement A] Approved for public release and unlimited distribution.

Empirical Results

Goal: Evaluate a sampling of the current state-of-the-art in classifier calibration with respect to traditional Top-1 ECE and ECE metrics tailored to particular contexts.

Experiment #1: Top-k ECE



[Distribution Statement A] Approved for public release and unlimited distribution.

Summary

- Ensuring that the confidence values that are output by an ML model match the empirical probability of events is called *calibration*
- For classification there are number of techniques to calibrate the *highest confidence output* of a model.
  - (Not covered) Also techniques for regression.
- For other definitions of calibration, it is less clear how to calibrate a classifier.
- However, there are metrics to measure different definitions of classifier calibration.

[Distribution Statement A] Approved for public release and unlimited distribution.

Begin Backup Slides

[Distribution Statement A] Approved for public release and unlimited distribution.

A Quick Crash Course on Classifier Calibration

What does it mean for a class confidence to be “good”

[Distribution Statement A] Approved for public release and unlimited distribution.

A Quick Crash Course on Classifier Calibration

What does it mean for a class confidence outputs to be “good”

How do we understand these values?

[Distribution Statement A] Approved for public release and unlimited distribution.

A Quick Crash Course on Classifier Calibration

What does it mean for a class confidence to be “good”

Classifier Calibration: Classifier outputs match the frequency of class labels.

[Distribution Statement A] Approved for public release and unlimited distribution.

A Quick Crash Course on Classifier Calibration

What does it mean for a class confidence to be “good”

For all possible inputs that the classifier outputs (0.6,0.4)…
60% of the inputs should be a Civilian Vehicle,
40% of the inputs should be an Enemy Tank.

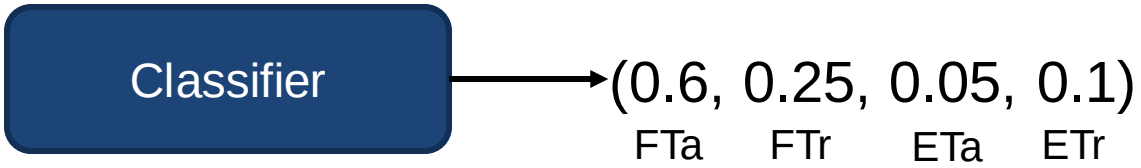
Classifier Calibration: Classifier outputs match the frequency of class labels.

[Distribution Statement A] Approved for public release and unlimited distribution.

This work: Evaluating classifiers for context-specific calibration

Modern machine learning literature has focused on evaluating classifier calibration according to their **Top-1 Expected Calibration Error (ECE)** (Guo et al; 2017)

Classes = {Friendly Tank, Friendly Truck, Enemy Tank, Enemy Truck}



[Distribution Statement A] Approved for public release and unlimited distribution.

A statistical framework for context-specific ECE

We adopt the framing of ECE from (Vaicenavicius et al; 2019). Here calibration metrics start with a *reliability (calibration) condition*.

Canonical Reliability (Calibration) Condition

$$\mathbb{P}[Y \in \cdot \mid g(X)] = g(X)$$

Top-1 Reliability Condition

$$\mathbb{P}[Y = \operatorname{argmax} g(X) \mid \max g(X)] = \max g(X)$$

Generalized Reliability Condition

$$\mathbb{P}[t(Y) \mid o(g(X)] = o(g(X)$$

Calibration "Lens"

[Distribution Statement A] Approved for public release and unlimited distribution.

A statistical framework for context-specific ECE

We adopt the framing of ECE from (Vaicenavicius et al; 2019). Here calibration metrics start with a *reliability (calibration) condition*.

Generalized Reliability Condition

$$\mathbb{P}[t(Y)|\ o(g(X)] = o(g(X)$$

Calibration "Lens"

Informally, a lens is a transformation of 1) The classifier's output 2) The probability simplex over classes that results in an induced classification problem.

Starting from a lensed reliability condition, one can derive specific metrics of *calibration error*.

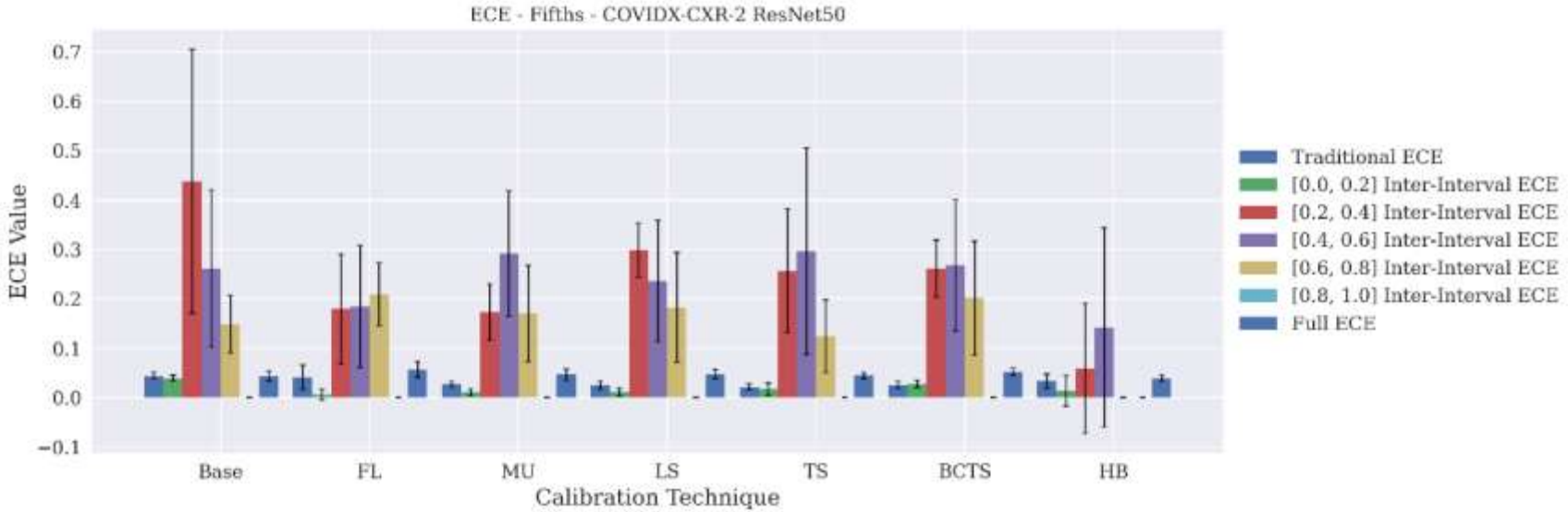
By doing so calibration metrics can be created that map to specific contexts, enabling evaluation of classifier calibration in more application-specific ways.

[Distribution Statement A] Approved for public release and unlimited distribution.

Empirical Results

Goal: Evaluate a sampling of the current state-of-the-art in classifier calibration with respect to traditional Top-1 ECE and ECE metrics tailored to particular contexts.

Experiment #2: Likert Category ECE

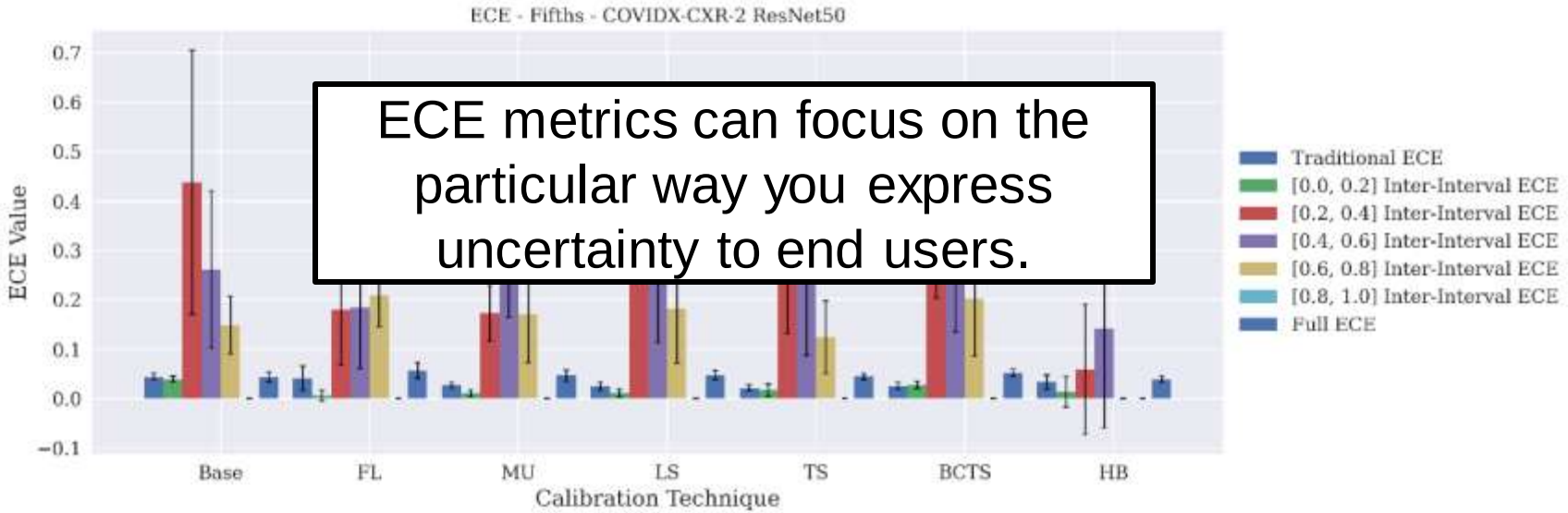


[Distribution Statement A] Approved for public release and unlimited distribution.

Empirical Results

Goal: Evaluate a sampling of the current state-of-the-art in classifier calibration with respect to traditional Top-1 ECE and ECE metrics tailored to particular contexts.

Experiment #2: Likert Category ECE

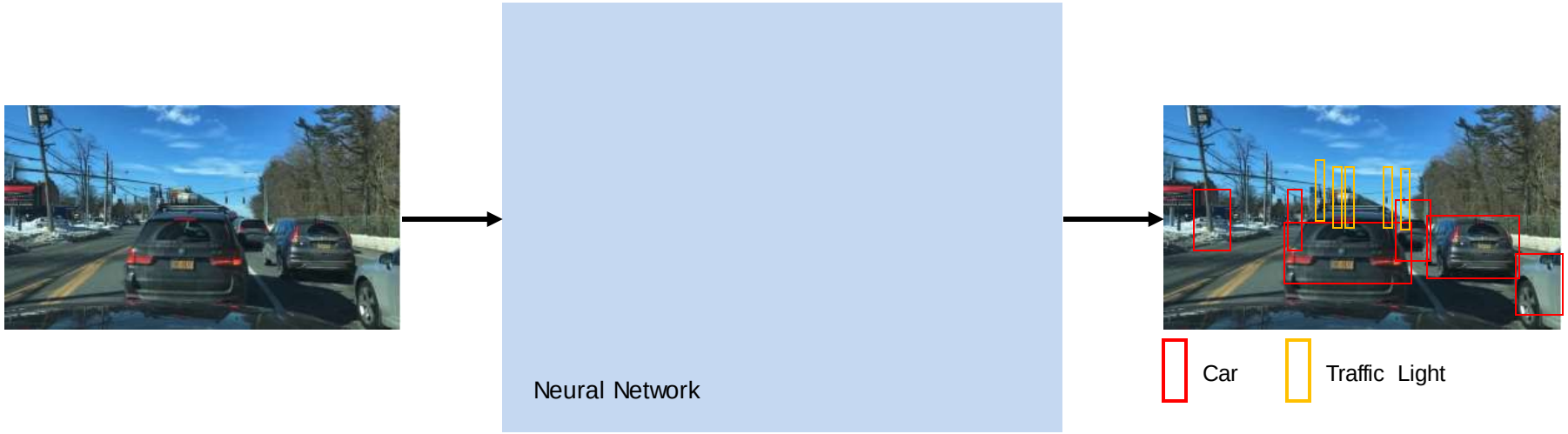


ECE metrics can focus on the particular way you express uncertainty to end users.

Introduction to Modern Object Detection

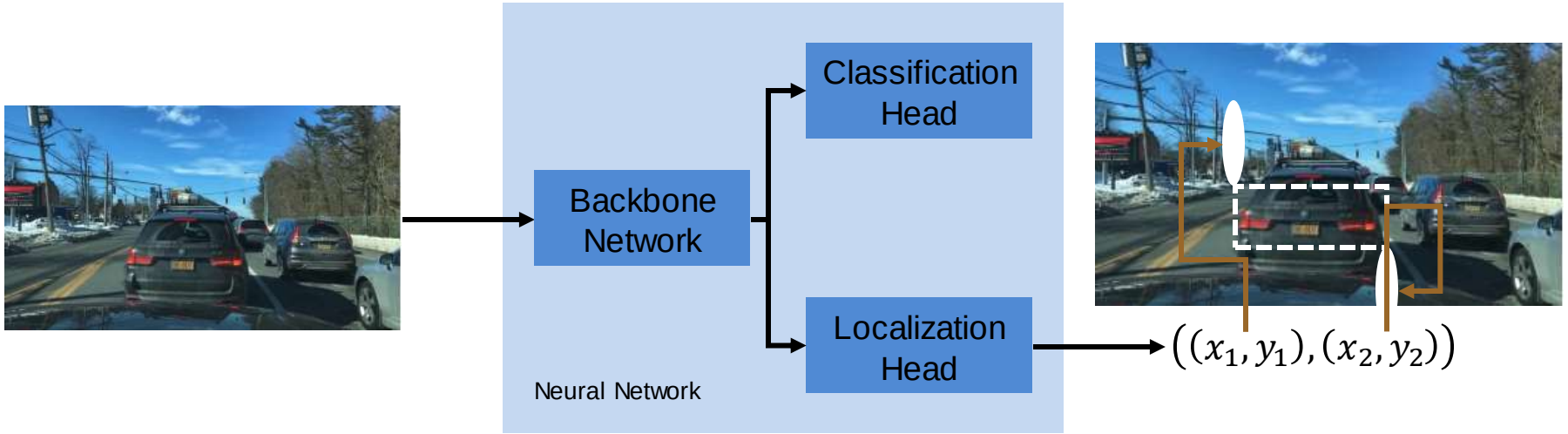
Object detection is really two tasks done in tandem:

1. **Localization**: Identifying *where* in the image objects are
2. **Classification:** Identifying *what* those objects are



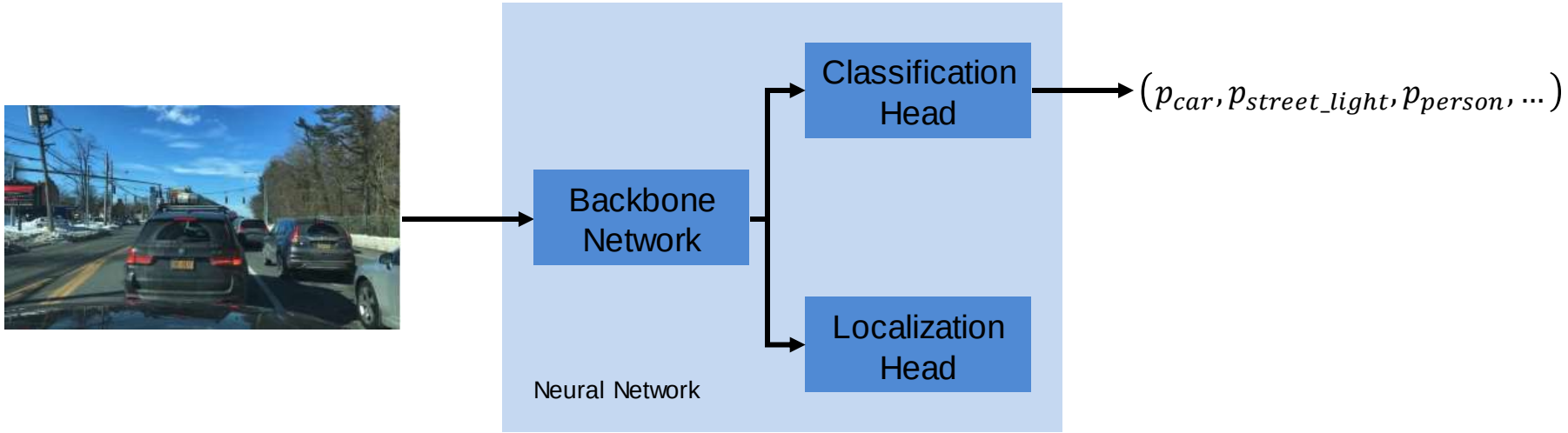
[Distribution Statement A] Approved for public release and unlimited distribution.

Introduction to Modern Object Detection



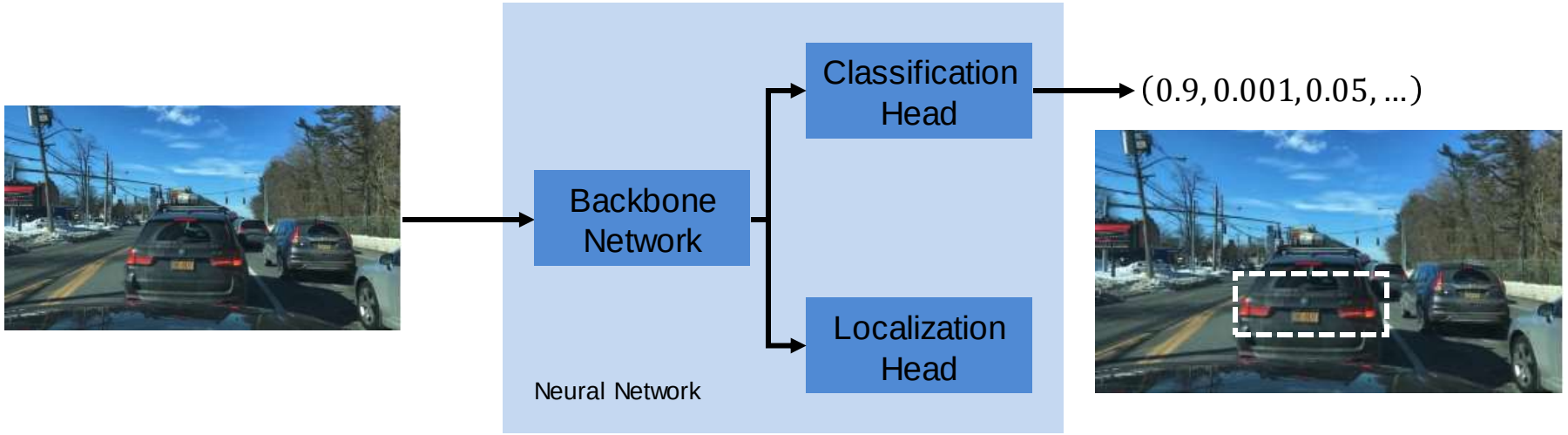
[Distribution Statement A] Approved for public release and unlimited distribution.

Introduction to Modern Object Detection



[Distribution Statement A] Approved for public release and unlimited distribution.

Introduction to Modern Object Detection



[Distribution Statement A] Approved for public release and unlimited distribution.

Introduction to Modern Object Detection

Maximum value corresponds to “ Car” class

Backbone Network

Classification Head

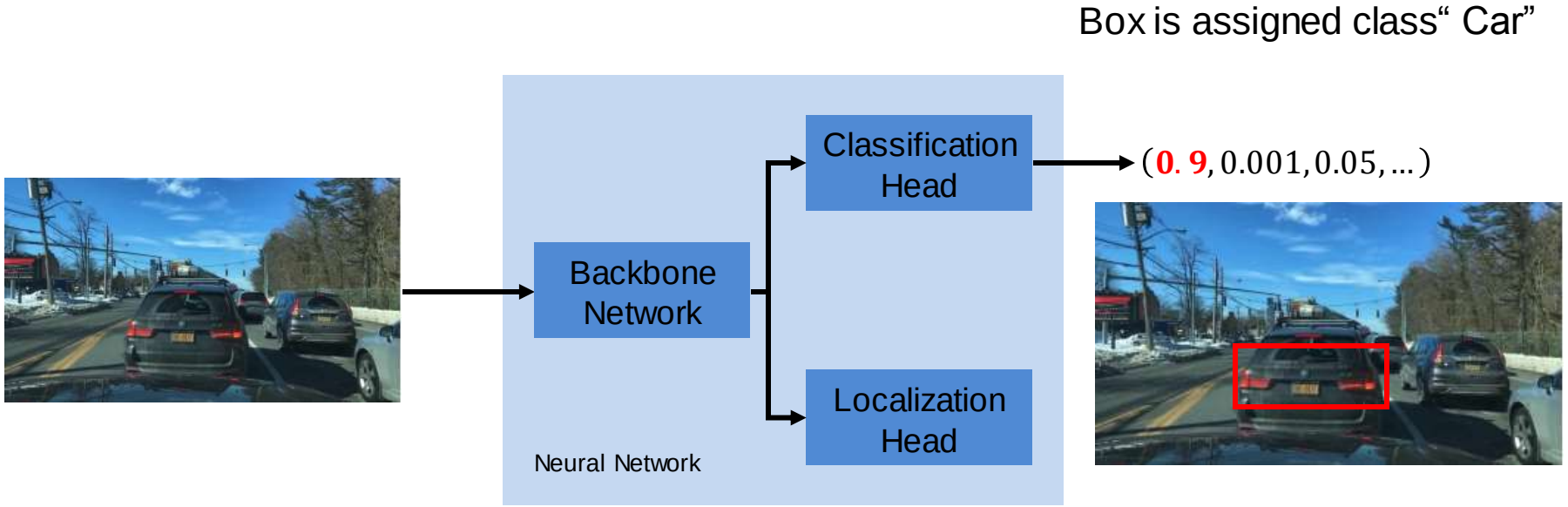
Localization Head

Neural Network

$(\mathbf{0.9}, 0.001, 0.05, \ldots)$

[Distribution Statement A] Approved for public release and unlimited distribution.

Introduction to Modern Object Detection



[Distribution Statement A] Approved for public release and unlimited distribution.