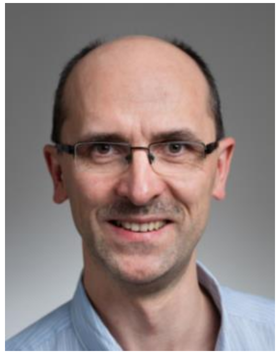
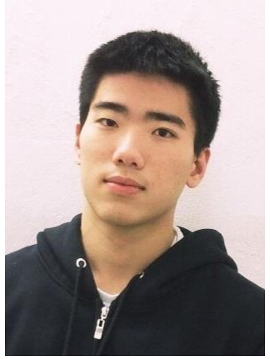
If a Human Can See It, So Should Your System: Reliability Requirements for Machine Vision Components

ICSE 2022








Boyue Caroline Hu Lina Marsso Krzysztof Czarnecki Rick Salay Huakun Shen Marsha Chechik

UNIVERSITY OF TORONTO

UNIVERSITY OF WATERLOO

UNIVERSITY OF TORONTO UNIVERSITY OF WATERLOO

Introduction

- **M**achine **V**ision **C**omponents (MVCs) in safety-critical systems
- Undesired behaviors can lead to fatal accidents
- Vision tasks are performed using machine learning (ML), since vision tasks are hard to specify

Towards safe MVCs, one needs to define what it means for an MVC to be correct and then check its correctness prior to system deployment

- In SE, reliability is the ability of a system or component to perform its required functions in a specified environment [IEEE-90]
- **Reliability** of MVC

  Whether the performance of an MVC remains reliably **unaffected by** image transformations that commonly occur in **real-world scenarios**

Uber SUV accident 2018

"IEEE Standard Glossary of Software Engineering Terminology" , 1990

Related Work in MVC Reliability

Specifying reliability of MVCs

- Set of qualities of the training dataset [Kohli-et-al-17]
- High-level MVC requirements [Gauerhof-et-al-20]
- …

Assessing reliability

- Adversarial robustness (e.g., [Serban-et-al-20])
- Using metamorphic testing (e.g., [Zhang-et-al-18])
- …

Unclear how to test the satisfaction of requirements

Only a small range of changes considered

Only changes that are close enough to the training data

Kohli Marc D et al. "Medical image data and datasets in the era of machine learning—whitepaper from the 2016 C-MIMI meeting dataset session". Journal of Digital Imaging, 2017

Gauerhof, Lydia et al. "Assuring the Safety of Machine Learning for Pedestrian Detection at Crossings". In: Proc. of SAFECOMP'20

Zhang Mengshi et al."DeepRoad: GAN-based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems". In: Proc. of ASE'18

Serban Alex et al. "Adversarial Examples on Object Recognition: A Comprehensive Survey". In: Proc. of CSUR'20

Dan Hendrycks et. al. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: Proc. of ICLR'19

Problem

- Lack of detailed and **machine-verifiable** reliability requirements limits the ability to assess MVC reliability
- Reliability should be studied with changes that can occur in real-world scenarios
- MVC are developed to automate human vision, thus it should be at least as reliable as humans.

Need: A method to establish human performance as a reference for defining and checking MVC reliability

Our Solution

Use human performance as a baseline to define reliability of MVCs against realistic changes in the real-world deployment:

if the changes do not affect humans, they shouldn't affect MVC either

1. Specify **two reliability requirements classes** for MVCs, with parameters representing human performance
2. A method to instantiate the requirement classes into machine–verifiable requirements

Reliability Requirements

Visual Change (Δ_v) Using IQA

A generic metric to measure changes of different transformations?

Different parameter domains

Gaussian Blur: (kernel size, sigma)

Gaussian Noise: (mu, sigma)

Different visual effects

Definition:

A measure for visual changes in images Δ_v, using established Image Quality Assessment (**IQA**) metrics [e.g., Sheikh-et-al]

Original image:
IQA value: 1 $\Delta_v = 0$

Minimal changes:
IQA value: 0.995 $\Delta_v = 0.005$

Reasonable changes:
IQA value: 0.29 $\Delta_v = 0.71$

Unreasonable changes:
IQA value: 0.004 $\Delta_v = 0.96$

Hamid Sheikh et al. *"Image Information and Visual Quality"* , IEEE Transactions on Image Processing, 2006

Reliability Requirements: Correctness-Preservation Class

Intuitively: For the range of changes in images that do not affect human performance, correctness of MVC should not be affected as well

Transformation T_x

Threshold (measured with Δ_v) t_c

Performance metric ***m*** that measures correctness of MVC output compared to ground truth

Ground truth

The MVC's performance ***m*** must not degrade for images transformed with visual changes within the threshold t_c

Note: ground truth **is** required to measure correctness.

m should be chosen according to the type of MVCs, e.g., prediction accuracy for image classification MVCs.

Reliability Requirements: Prediction-Preservation Class

Intuitively: For the range of changes in images that do not affect human predictions, the predictions of MVC should stay unaffected as well

Transformation T_x

Threshold (measured with Δ_v) t_p

Prediction similarity metric s that compares MVC outputs on both original and transformed images

→

The MVC's prediction similarity s must not degrade for images transformed with visual changes within the threshold t_p

Note: ground truth **is not** required

s should also be chosen according to the type of MVC, e.g., for image classification MVCs:
0 if the two labels are the same and 1 otherwise

Comparing the Reliability Requirement Classes

Correctness-Preservation

Checks for the correctness of decisions after transformation

Requires ground truth which is costly to obtain

If an MVC satisfies only the correctness-preservation requirement, it may correctly recognize different objects before and after transformation

Prediction-Preservation

Checks for the preservation of decisions after transformation

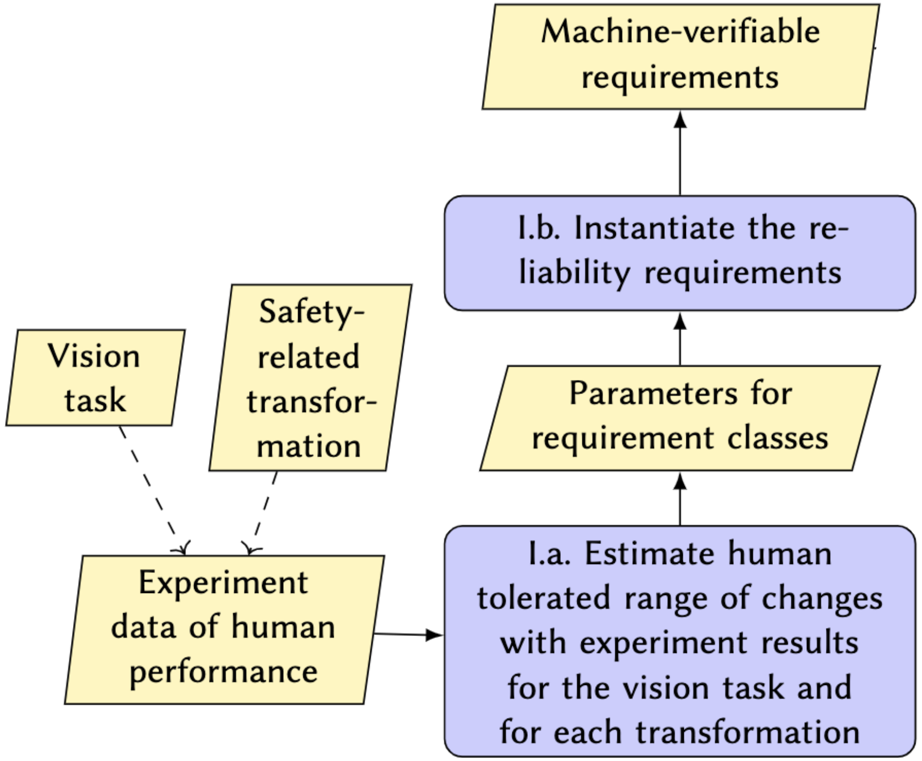
Can be checked on unlabeled images which are easier to obtain

If only the prediction-preservation requirement is satisfied, the MVC might preserve incorrect decisions and change correct ones

Neither requirement subsumes the other.

Obtaining Machine-Verifiable Requirements

Requirement Instantiation



<u>Parameters of the requirement classes:</u>

Image transformations

Thresholds

Metrics m and s

Obtaining Thresholds t_c and t_p

Can we require MVCs to remain reliable subject to any range of changes in the environment? **NO!**

Example: adding frost

Estimate the thresholds (t_c/t_p) of visual changes that do not affect humans through experiments with human participants.

Experiments with Human Participants

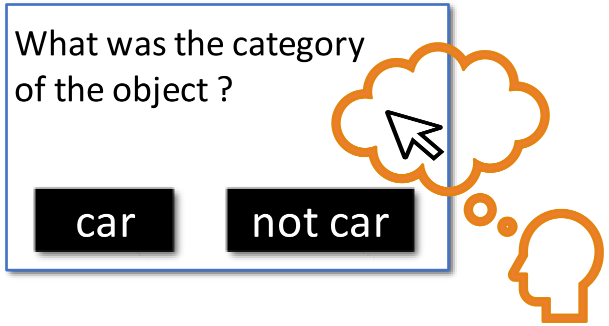
<u>Objective</u>: obtain human predictions on original and transformed images

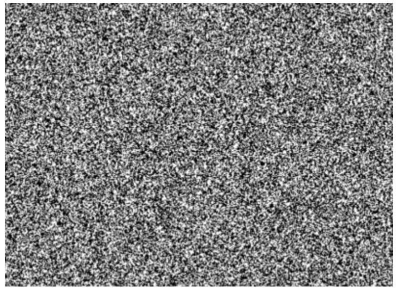
Forced-choice image categorization task:

Humans are presented with the images with transformations applied, for **200 ms**

Asked to **choose one of the presented categories** (e.g., car or not car)

Between images, shown **noise mask** to minimize feedback influence in the brain





Noise mask

Conducted experiments:

- Amazon Mechanical Turk platform (2,000 human participants)
- 8 safety-related transformations: RGB, contrast, defocus blur, brightness, **frost**, color jitter, jpeg compression, and Gaussian noise

Instantiated Requirements: Example

Transformation: artificial frost addition

(Correctness-preservation) The recognition accuracy (*m*) of an MVC should not decrease if the visual change in the images is within the range **Δ_v <= 0.84**
(Prediction-preservation) The percentage of labels an MVC can preserve (*s*) after adding frost should not decrease if visual change in the images is within the range **Δ_v <= 0.91**

Original

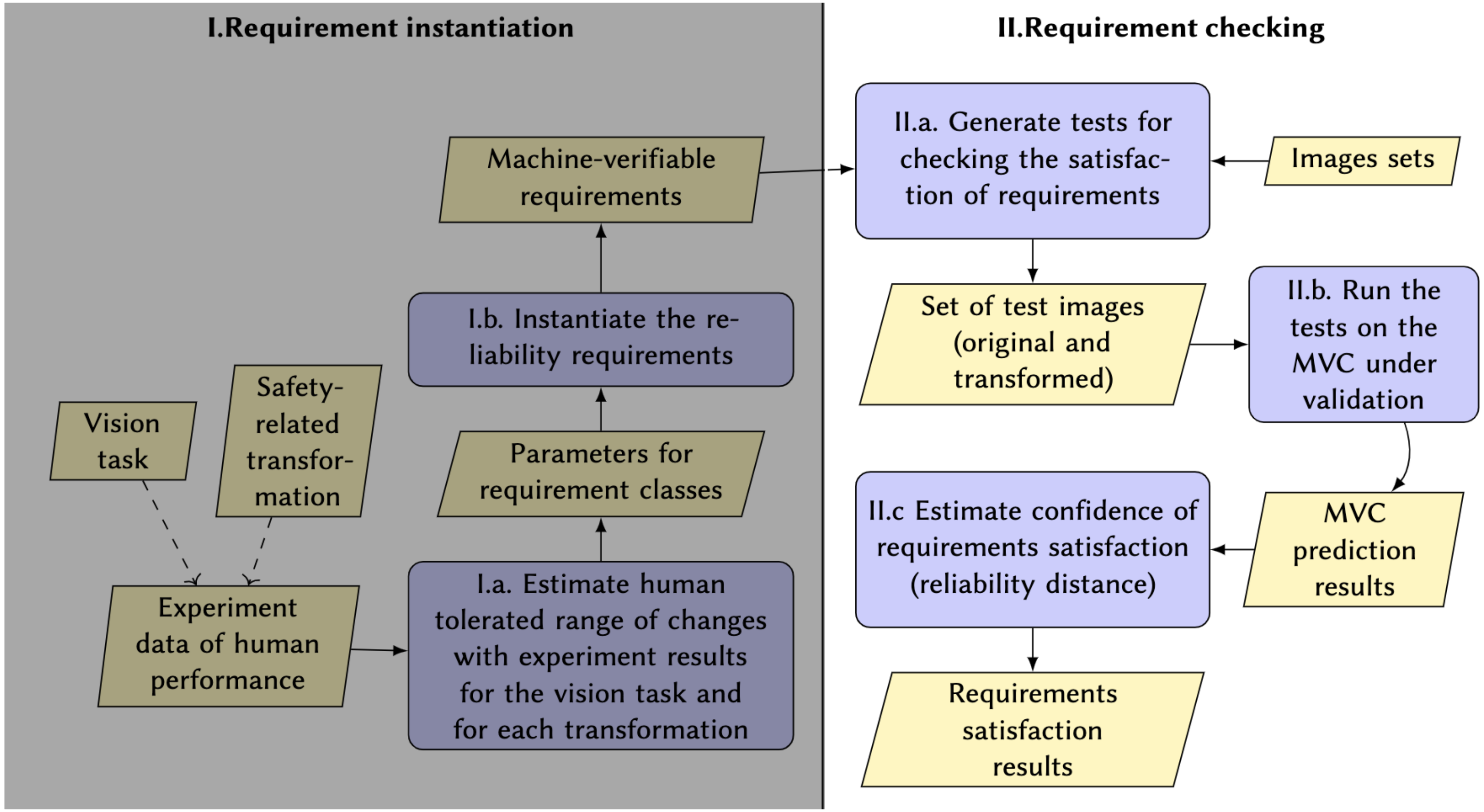
Within range

Within range

Outside of range

Checking Satisfaction of Requirements

Requirement Checking



Requirement Checking

Does an MVC satisfy our requirements for a given transformation?

1. Generate test cases: transformed images under the specified thresholds t_c/t_p (uniformly sampling in the parameter domain)

original

…

2. Execute the test cases on the MVC

Ground truth: car

not car

…

car

car

not car

not car

MVC outputs

Requirement Checking

3. Evaluate the test case execution results

Correctness-Preservation

Ground truth: car

not car

...

not car ✗

car ✓

car ✓

not car ✗

not car ✗

Same as ground truth ✓

Different from ground truth label ✗

Correctness-Preservation: "as correct as the original" (0 VS 40%)

m of original image: 0/1 (0%)

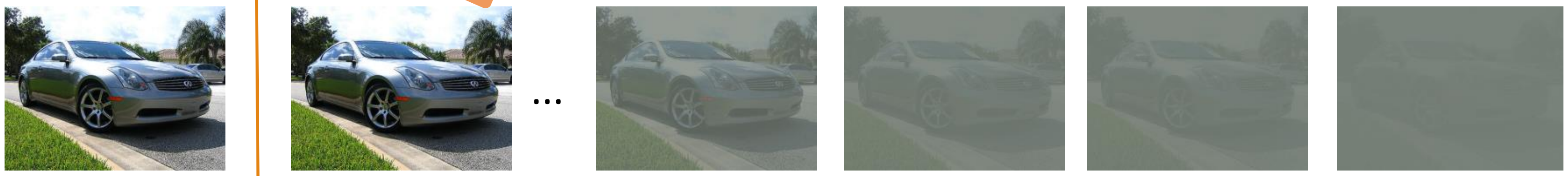
m of transformed images: 2/5 (40%)

Requirement Checking

3. Evaluate the test case execution results

Minimal changes

Prediction-Preservation



Original output label

not car | not car | car | car | not car | not car

Same as original output label

Different from original output label

Prediction-Preservation: "same prediction for minimal vs. significant changes as for the original" (100% VS 60%)

s of original image: estimated using minimal changes of the original image: 1/1 (100%)

s of transformed images: 3/5 (60%)

Requirement Checking

3. Evaluate the test case execution results

Ground truth: car

Correctness-Preservation	not car	not car ✗	car ✓	car ✓	not car ✗	not car ✗
Prediction-Preservation	not car	not car ✓	car ✗	car ✗	not car ✓	not car ✓

4. Requirements considered satisfied if values of the metric on transformed images are “close enough" to values of the metric on original images

Evaluation

Research Questions

1. **Evaluate our ranges**

How well do the existing reliability evaluation methods cover the human-tolerated range of changes?

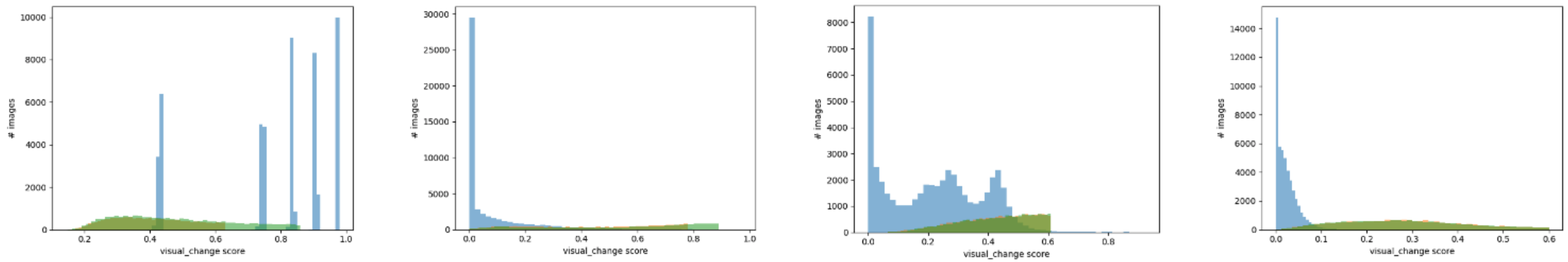
2. **Evaluate usefulness of our requirements**

How effective is our requirement checking method in identifying reliability gaps compared to existing approaches?

Evaluating Our Ranges

How well do the existing reliability evaluation methods cover the human-tolerated range of changes?

Here we show the comparison of distribution of test images with state-of-the-art dataset for benchmarking robustness against common transformations: CIFAR-10-c.



(a) Contrast with CIFAR-10 images

(b) Brightness with CIFAR-10 images

(c) Frost with CIFAR-10 images

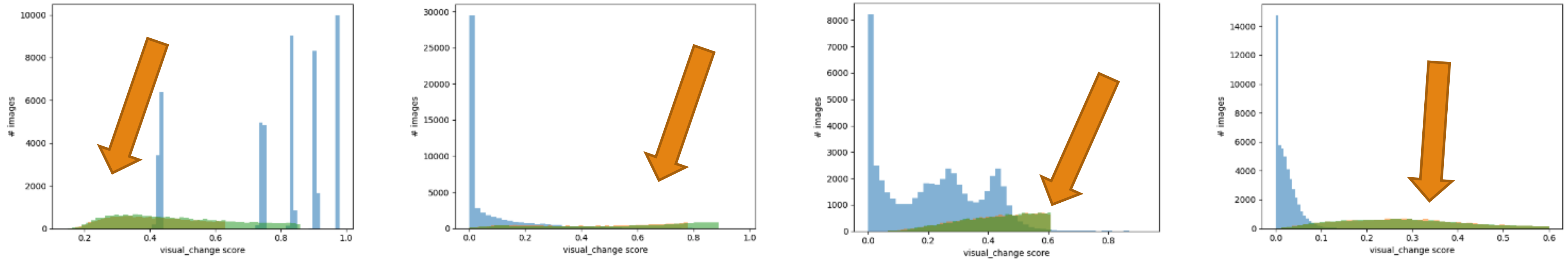
(d) JPEG Compression with CIFAR images

Blue: cifar-10-c benchmark images; Green: tests for prediction-preservation

Evaluating Our Ranges

Tests generated using our reliability requirements **VS** existing tests in benchmark dataset CIFAR-10-c

The human-tolerated range is not addressed.



(a) Contrast with CIFAR-10 images

(b) Brightness with CIFAR-10 images

(c) Frost with CIFAR-10 images

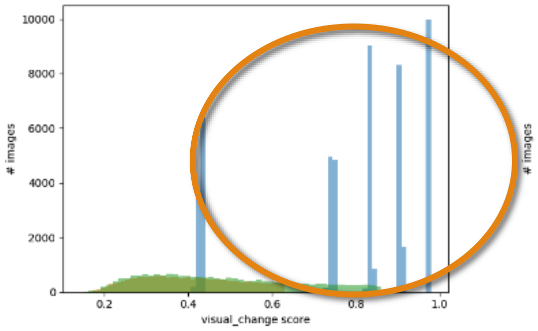
(d) JPEG Compression with CIFAR images

Blue: cifar-10-c benchmark images; Green: tests for prediction-preservation

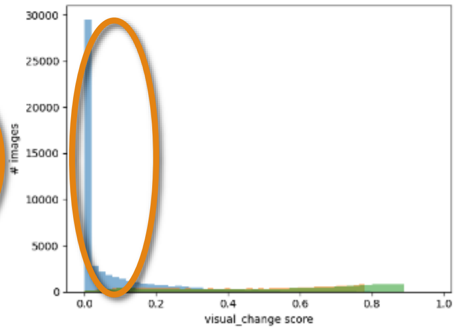
Evaluating Our Ranges

Tests generated using our reliability requirements **VS** existing tests in benchmark dataset CIFAR-10-c

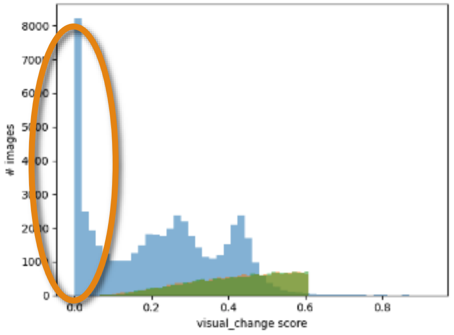
The testing results obtained this way may not be representative.



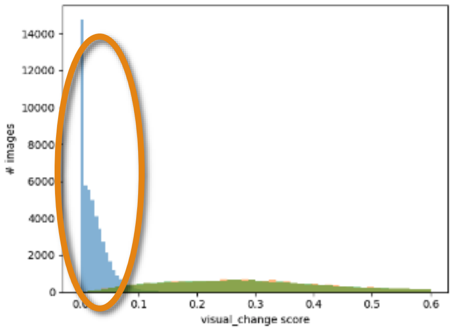
(a) Contrast with CIFAR-10 images



(b) Brightness with CIFAR-10 images



(c) Frost with CIFAR-10 images



(d) JPEG Compression with CIFAR images

Blue: cifar-10-c benchmark images; Green: tests for prediction-preservation

UNIVERSITY OF TORONTO

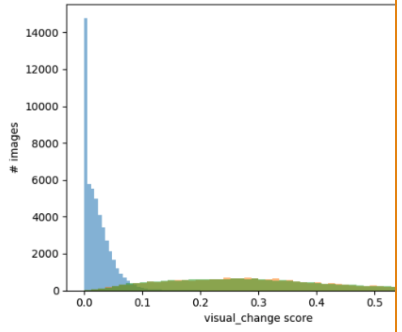
UNIVERSITY OF WATERLOO

Evaluate Usefulness of Our Requirements

How effective is our requirement checking method in identifying reliability gaps compared to existing approaches?

Testing with benchmark dataset **VS** testing our requirements

- Transformation: JPEG compression
- Generated transformed images (tests) within human tolerated range
- Tested models on the CIFAR-10-c leaderboard



CIFAR-10-c leaderboard model name	Rank on CIFAR-10-c	Rank of satisfying our *correctness preservation*	Rank of satisfying our *prediction preservation*
RLAT	1	5	1
RLATAugMixNoJSD	2	2	7
			2
			4
			3
			5
[illegible]	[illegible]	[illegible]	6

CIFAR-10-c: RLATAugMixNoJSD is the second most reliable against JPEG compression.

Our requirements: Even though RLATAugMixNoJSD has good accuracy on transformed images, its output is not consistent.

CIFAR-10-c ranking is only based on accuracy

Research Questions

1. **Evaluate our ranges**

How well do the existing reliability evaluation methods cover the human-tolerated range of changes?

Not addressed by existing benchmark

2. **Evaluate usefulness of our requirements**

How effective is our requirement checking method in identifying reliability gaps compared to existing approaches?

We can detect gaps missed by existing benchmark

It is important to check MVC reliability against our requirements.

Research Questions

1. **Evaluate our ranges**

How well do the existing reliability evaluation methods cover the human-tolerated range of changes?

Not addressed by existing benchmark

2. **Evaluate usefulness of our requirements**

How effective is our requirement checking method in identifying reliability gaps compared to existing approaches?

We can detect gaps missed by existing benchmark

Threats to validity

- [Construct] Human performance is hard for MVC to match
- [Internal] Testing with uniformly distributed transformation parameter values
- [External] Limited data considered due to budget consideration

Conclusion

Reliability of Machine Vision Components (MVC): `` **if a human can see it, so should the MVC**''

- An MVC should be reliably unaffected by image transformations, at least within the range of changes that does not affect humans
- 2 classes of reliability requirements: **correctness-preservation** and **prediction-preservation**, and a method to instantiate and check them
- Our framework revealed new reliability gaps not previously detected in state-of-the-art image classification models

Conclusion

Reliability of Machine Vision Components (MVC): `` **if a human can see it, so should the MVC**''

- An MVC should be reliably unaffected by image transformations, at least within the range of changes that does not affect humans
- 2 classes of reliability requirements: **correctness-preservation** and **prediction-preservation**, and a method to instantiate and check them
- Our framework revealed new reliability gaps not previously detected in state-of-the-art image classification models

Limitation:

- Only image classification
- Obtaining human data is expensive
- Simple testing method

Future Work:

- Extend reliability requirements to other type of MVCs
- Develop methods to reduce the cost of human performance
- Extend reliability checking method with reliability diagnosis

Thank you!