



IEEE Standard for Robustness Testing and Evaluation of Artificial Intelligence (AI)-based Image Recognition Service

IEEE Computer Society

Developed by the
Artificial Intelligence Standards Committee

IEEE Std 3129™-2023

STANDARDS

IEEE Standard for Robustness Testing and Evaluation of Artificial Intelligence (AI)-based Image Recognition Service

Developed by the

Artificial Intelligence Standards Committee
of the
IEEE Computer Society

Approved 15 February 2023

IEEE SA Standards Board

Abstract: Test specifications with a set of indicators for common corruption and adversarial attacks, which can be used to evaluate the robustness of artificial intelligence-based image recognition services are provided in this standard. Robustness attack threats and establishes an assessment framework to evaluate the robustness of artificial intelligence-based image recognition service under various settings are also specified in this standard.

Keywords: adversarial attacks, artificial Intelligence-based services, assessment framework, common corruption, IEEE 3129™, robustness

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2023 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 2 June 2023. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-9752-7 STD26190
Print: ISBN 978-1-5044-9753-4 STDPD26190

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile and Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

¹Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

²Available at: <https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html>.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are

³Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴Available at: <https://standards.ieee.org/standard/index.html>.

⁵Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this Standard was completed, the Robustness of Artificial Intelligence based Service Working Group had the following membership:

Qing An, Chair

<i>Organization Represented</i>	<i>Name of Representative</i>
Alibaba China Co. Ltd.	Juntao Peng
Alipay (China) Technology Co., Ltd.	Xiaoyuan Bai
Beijing RealAI Technology Co., Ltd.	Ying Guo
Chongqing Changan Automobile Co., Ltd.....	Yonggang Luo
Cloudwalk Technology	Jun Li
Huazhong University of Science and Technology	Kun He
Institute of Computing Technology, Chinese Academy of Sciences	Qingming Huang
Institute of Information Engineering, Chinese Academy of Sciences.....	Lingzhong Meng
Microsoft Corporation.	Praveen Palanisamy
Tsinghua Shenzhen International Graduate School	Guiguang Ding
Xi'an Jiaotong University.....	Chenhao Lin
Zhejiang University	Shouling Ji

The working group gratefully acknowledges the contributions of the following participants:

Hui Chen	Jinyan Ma	Hui Xue
Yuefeng Chen	Xiaofeng Mao	Yichen Yang
Juan Deng	Chao Shen	Zhiyong Yang
Yinpeng Dong	Hang Su	Zhen Yu
Yuan He	Yifeng Xiong	Xudong Zhang
Xiaodan Li	Qianqian Xu	Xuhong Zhang
Chang Liu		Hongru Zhu

The following members of the entity Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

OxSenses Corporation
1stCycle Corporation
Alibaba China Co. Ltd.
Ansteel Group Corporation Ltd
China Telecommunications
Corporation
Chongqing Changan Automobile
Co., Ltd.

Cloudwalk Technology
Huawei Technologies Co., Ltd
Huazhong University of Science
and Technology
Institute of Biomedical
Engineering, Chinese Academy
of Medical Sciences and Peking
Union Medical College

Institute of Software,
Chinese Academy of Sciences
Lenovo Group Limited
Shenzhen University
Tsinghua Shenzhen International
Graduate School
Xi'an Jiaotong University
YITU Technology

When the IEEE SA Standards Board approved this standard on 15 February 2023, it had the following membership:

David J. Law, *Chair*
Vacant Position, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Sara R. Biyabani
Ted Burse
Doug Edwards
Ramy Ahmed Fathy
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy

John D. Kulick
Joseph S. Levy
Howard Li
Johnny Daozhuang Lin
Gui Lin
Xiaohui Liu
Kevin W. Lu
Daleep C. Mohla
Andrew Myles

Paul Nikolich
Annette D. Reilly
Robby Robson
Lei Wang
F. Keith Waters
Karl Weber
Philip B. Winston
Don Wright

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 3129™-2023, IEEE Standard for Robustness Testing and Evaluation of Artificial Intelligence (AI)-based Image Recognition Service.

Artificial intelligence (AI) technology and image recognition services are continuously being developed and are widely used in all walks of life. AI-based service will not only encounter common corruptions such as image compression and environmental changes but will also face various adversarial sample attacks. All of these threats will affect the robustness of the service and reduce the accuracy of its results and may be used by unauthorized personnel, endangering cyberspace security, causing economic losses and even having adverse impacts on social stability.

There are several ad hoc solutions that attempt to solve these problems, but the source of the problem and the goal toward solving the problem must be understood. AI-based image recognition services in this standard refers to services that are based on AI and are provided to individuals or organizations through application programming interfaces (APIs). The API input normally includes unstructured image data, and the output may include the result image classification or object detection. “Robustness” in this standard refers to the ability of the image recognition service provided by AI to maintain its accuracy when the input image is subject to external interference such as harsh environmental conditions or adversarial attacks.

Service providers not only need to test the robustness of their algorithms and models through white-box methods in the algorithm development and training stages, but also need to conduct black-box testing of AI services through APIs that are provided to the market in order to give corresponding reference of evaluation of the use of the service. A standardized robustness black-box evaluation methodology for AI-based image recognition services (such as face recognition and object detection) does not currently exist. Therefore, it is necessary for this standard to provide guidance on testing and evaluation methods for the above-mentioned situations in the field of AI-based image recognition services.

Contents

1. Overview	11
1.1 Scope	11
1.2 Purpose	11
1.3 Word usage	11
2. Normative references	12
3. Definitions, acronyms, and abbreviations	12
3.1 Definitions	12
3.2 Acronyms and abbreviations	12
4. Robustness evaluation framework for AI-based image recognition services	12
4.1 Background	12
4.2 Image corruptions	13
4.3 Adversarial attacks	14
4.4 Evaluation practice	15
4.5 Roles	15
4.6 Evaluation process	16
4.7 Evaluation metrics	17
5. Test cases	23
5.1 Test case for <i>Accuracy</i> test	23
5.2 Test case for common corruptions	23
5.3 Test case for style transfer	23
5.4 Test case for adversarial perturbations	24
Annex A (informative) Example common corruptions and adversarial attacks	25
Annex B (informative) Example evaluation	29
Annex C (informative) Bibliography	30

IEEE Standard for Robustness Testing and Evaluation of Artificial Intelligence (AI)-based Image Recognition Service

1. Overview

1.1 Scope

This standard provides test specifications with a set of indicators for interference and adversarial attacks that can be used to evaluate the robustness of artificial intelligence (AI)-based image recognition services. This standard specifies robustness requirements and establishes an assessment framework to evaluate the robustness of AI-based image recognition services under various settings.

1.2 Purpose

The purpose of this standard is to guide individuals and organizations who provide, develop, or use AI-based image recognition services in testing and evaluating these services and in improving the robustness of these services. It is also applicable to guide third-party evaluation laboratories to test and evaluate the robustness of the service by applying standards-based testing to score individual algorithmic implementations.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{6,7}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

⁶The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

⁷The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

There are no normative references in this standard.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.⁸

adversary: An entity whose aim is to prevent the participants of the protocol from achieving their designed goal. For example, it could try to affect input privacy, result correctness, or result delivery.

common corruptions: Change of image color, texture, shape, resolution, and signal-to-noise ratio, etc. Human vision systems are robust to most image corruptions, even to abstract changes in structure and style, while AI-based image recognition services might be confused by some image corruptions.

3.2 Acronyms and abbreviations

AI	artificial intelligence
API	application programming interface
BIM	Basic Iteration Method
DNN	deep neural network
FGSM	Fast Gradient Sign Method
HSJA	HopSkipJumpAttack
IoU	intersection over union
mIoU	mean intersection over union
NGD	Natural Gradient Descent
ODS	Output Diversity Sampling
P-RGF	Prior-Guided Random Gradient Free
PGD	Projected Gradient Descent

4. Robustness evaluation framework for AI-based image recognition services

4.1 Background

This standard focuses on evaluating the robustness of image recognition services based on AI. The service is usually deployed in the cloud environment and accessible via application programming interface (API) calls. Multiple applications of image recognition services exist including face recognition, plant recognition, etc. [Figure 1](#) shows an image recognition service, which is the evaluation target of this standard.

⁸*IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

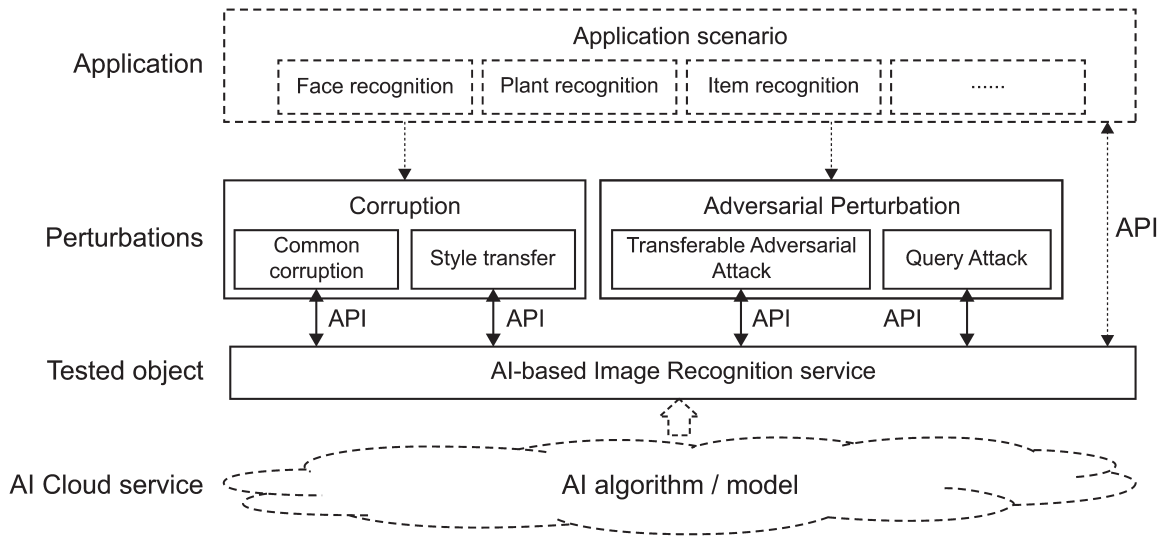


Figure 1—Evaluation scope for an image recognition service based on AI

When accessed via API calls, the target service shall demonstrate the ability to maintain its performance against various corruptions and adversarial attacks (Hendrycks and Dietterich [B14]).⁹ When a corrupted image is input to the target service, the target service may generate a false prediction result. In adversarial attacks, maliciously crafted images can also cause the service to produce incorrect prediction results, target results, or even random results. Therefore, in this standard, the robustness evaluation for the AI-based image recognition service shall encompass the evaluations of the robustness against corruptions and adversarial perturbations, as shown in Figure 1.

4.2 Image corruptions

4.2.1 Common corruptions

Image common corruptions include the following (Hendrycks and Dietterich [B14]):

- Noise* is typically defined as a random variation in brightness or color information. Common image noise includes Gaussian noise, Poisson noise, and impulse noise.
- Blur* refers to the loss of image details due to motion, defocus, and other reasons in the image acquisition process, including defocus blur, frosted glass blur, motion blur, and zoom blur.
- Weather* corruptions refer to the addition of adverse weather conditions to the original image. Common adverse weather conditions include snow, frost, fog, and brightness.
- Digital changes* are the changes in color, shape, and resolution of an image after digital processing. Common digital changes include contrast, geometric transformations, pixelation, and compression loss.
- Mask* means that part of the image is randomly occluded.

Common corruptions are described in detail in Annex A. All the corrupted images are generated by the corresponding corruption algorithms, as disturbance factors in the real world are difficult to control.

⁹The numbers in brackets correspond to those of the bibliography in Annex C.

4.2.2 Style transfer

When developing image-generation algorithms, it becomes easier to make style transfers to benign images. Style transfer refers to the computer vision technique that recomposes the content of an image in certain style, such as sketch, oil painting, Picasso, Mondrian, etc. The style transfer of an image can be achieved using deep learning algorithms. An example of style transfer can be found in “Neutral Style Transfer” [B25].

4.3 Adversarial attacks

An adversarial sample that is indiscernible from the original clean image to the human eye will lead the service to make wrong predictions (Goodfellow, Shlens, and Szegedy [B11]). The difference between the adversarial sample and the original clean image is generally measured using L_p norm.

Adversarial attacks that are widely recognized and prove effective on image recognition services include the Transferable Adversarial Examples Attack, explained in 4.3.1, and Query Attack, explained in 4.3.2. Details of specific adversarial attacks are described in Annex A.

4.3.1 Transferable Adversarial Attack

The transferability of adversarial examples refers to that adversarial examples crafted on an accessible service (or model) are also able to fool other inaccessible models with unknown architectures or parameters. A number of transferable adversarial attacks have been proposed to enhance the transferability of adversarial examples so as to improve the attack success rates in the totally black-box setting, as follows:

- a) Transferable Adversarial Attacks can be classified by target of the attacks, namely the following:
 - 1) *Decision Layer Attack*: The probability is predicted by the direct attack substitution model through one-step attack or multi-step iterative attack, optimizing adversarial samples.
 - 2) *Feature Layer Attack*: Different from the decision layer attack, feature layer attack mainly attacks the middle layer features of the alternative model so as to achieve the purpose of attacking the target service.
- b) Transferable Adversarial Attacks can also be classified by attack generation method, as follows:
 - 1) *Gradient Optimization Attack*: In this attack, a substitute model is obtained that has a similar decision boundary with the target model, the according loss function of the substitute model is set, and the gradient of the loss function is calculated against the input space for optimized adversarial perturbations. In practice, to approximate to the gradient of the target model, multiple gradient calculation methods are used, such as fast gradient sign method or momentum-based gradient calculation method.
 - 2) *Generative Attack*: Different from the previous gradient based anti-migration attacks, the alternative model is attacked by optimizing the generation model, and then the generation model is used to generate anti disturbance to attack the target service.
- c) There are strategies to improve the transferability of the attacks in Items a) and b) as follows:
 - 1) *Input transformation*: This strategy adopts various input transformations to further improve the transferability of adversarial examples. Specifically, such attacks create a batch of various patterns based on an input image and use the average gradient to optimize the adversarial examples.
 - 2) *Model ensemble*: This strategy uses multiple models simultaneously to improve the attack transferability. Often the predictions, logits, or losses of multiple models are fused. If an adversarial example can mislead multiple models simultaneously, it is likely to mislead another one as well.

4.3.2 Query Attack

Query Attack is one kind of black-box adversarial attack, which allows probing target models with queries. Compared with other kinds of black-box adversarial attacks, which can be grouped as zero-knowledge attacks, Query Attack can access the target model to obtain important information such as local gradient. Query Attack can be divided into two groups, including Score-Based Attack and Decision-Based Attack. Score-Based Attack can obtain not only the predictive labels but also the probability of the target labels. Decision-Based Attack can only obtain the predictive labels, such as boundary attack.

Query attacks can be classified by the information provided by the target service for input images, namely the following::

- a) *Score-based attacks*: The target service not only returns the predicted category but also feeds back the predicted probability and the output logits that could be utilized, which is convenient for the adversaries to generate adversarial examples. The researchers usually leverage the zero-order, first-order, or second-order information of the victim model to estimate the gradient with respect to the input sample. Then the white-box attacks could be naturally performed for efficient black-box attacks.
- b) *Decision-based attacks*: These are much more relevant in real-world machine learning applications where confidence scores or logits are rarely accessible, compared to score-based attacks. At the same time, decision-based attacks have the potential to be much more robust against standard defenses like gradient masking, intrinsic stochasticity, or robust training than attacks from other categories. Finally, compared to transfer-based attacks, they need much less information about the model (neither architecture nor training data) and are much simpler to apply. However, decision-based attacks are much more challenging due to the minimum information requirement for the attacks.

4.4 Evaluation practice

The robustness evaluation of an AI-base image recognition service can be carried out as self-evaluation, or third-party evaluation. In the former, the providers of the image recognition perform the evaluation on their own services, while in the latter, a third-party test laboratory performs the evaluation, generally upon the request and under the permission from the service provider.

4.5 Roles

Different roles are involved in the evaluation, namely, Test Requester, Tester Performing the Test, and Service Provider that provides the target service to be evaluated. These are explained as follows:

- *Test Requester*: A person or an entity that requests the test. A test requester can be either a user of an image recognition service or the provider of the service. The user of the service requests the test service in order to make informed decisions when selecting and using an image recognition service. The request is initiated when the service is already offered to the market. The service provider usually requests the service for verifying or improving the service, if any. Such a test generally occurs before the service is released to the market and can also occur after market as regular checks.
- *Tester*: A person or a testing laboratory performing the robustness test. A tester can be a user of the service who is equipped with necessary expertise, or the service provider, or a third-party qualified and independent tester.
- *Service Provider*: An entity that provides the image recognition service.

4.6 Evaluation process

Before the actual test, the Test Requester and Tester normally reach an agreement on the forthcoming testing, which is out of the scope of this standard.

The evaluation shall start with a Plan, followed by Corruption Robustness Evaluation, Adversarial Robustness Evaluation, and Report Generation, as shown in [Figure 2](#).

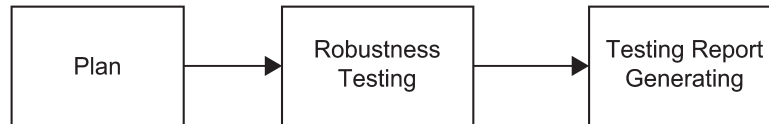


Figure 2—Evaluation Process

During the planning phase, the tester and the requester shall determine the information needed for the evaluation, including the following:

- The target service being evaluated comprising the type of target service, and the information needed for accessing the target service (e.g., API calls to use the service being evaluated, accounts and credentials)
- Image corruptions and adversarial attacks against which that the service will be evaluated ([Annex A](#) lists common corruptions and adversarial attacks that are well recognized and proved as effective on the image recognition services)
- The image source used for the evaluation
- Parameters related to the type of service under evaluation [for example, in object detection, the confidence threshold and the intersection of the union threshold for the predicted bounding box need to be determined, and in semantic segmentation, the mean intersection over union (mIoU) needs to be determined]
- Parameters related to specific corruptions or adversarial attacks (for example, the severity level of a corruption added to a test image, or the Lp bound used in an adversarial attack)

In Corruption Robustness Evaluation, the image corruptions shall be carried out by the Tester according to [5.2](#) and [5.3](#). In Adversarial Robustness Evaluation, the adversarial attacks shall be carried out according to [5.4](#). [Annex B](#) shows an example evaluation.

The evaluation report shall be comprised of at least the following:

- The target service being evaluated
- The tester information
- The test description including, for each test, the image source used, the attack or corruption performed, and the steps to carry out the test
- The metrics used
- The setting of the parameters related to the type of service
- The setting of the parameters specific to each corruption or attack performed by the tester
- The results

Other information may be deemed necessary according to the agreement between the test requester and the tester.

The evaluation report and other information necessary for the test requester to repeat the evaluation, should they decide to, are submitted by the tester.

4.7 Evaluation metrics

The following evaluation metrics should be used in the evaluation:

- *Accuracy*: The capability of the target service under evaluation to produce correct predictions when no corruptions or attacks are introduced
- *Robustness_Corr*: The average capability of the target service under evaluation to produce correct predictions under a corruption
- *Robustness_Adv*: The average capability of the target service under evaluation to produce correct predictions under an adversarial attack
- *WorstCase_Robustness_Corr*: The average capability of the target service under evaluation to produce correct predictions under all corruptions being tested against
- *WorstCase_Robustness_Adv*: The average capability of the target service under evaluation to produce correct predictions under all adversarial attacks being tested against

How the evaluation metrics are calculated are explained in the following subclauses, and the test pipeline only considers a single corruption per image in the robustness test. This is to get an understanding of how the target service behaves under certain corruption.

4.7.1 Accuracy evaluation

Accuracy evaluation is illustrated in Figure 3.

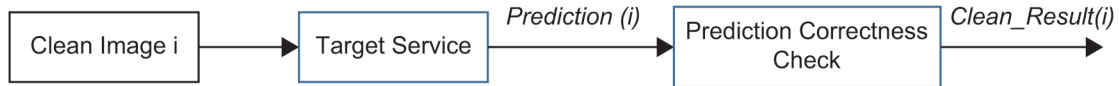


Figure 3—Accuracy Evaluation

In the test for *Accuracy*, a large number, M , of clean images shall be respectively input to the target service being evaluated. For each clean image i , the target service produces a prediction result, denoted as $prediction(i)$. When the prediction result $prediction(i)$ is correct, then $Clean_result(i)$, the indicator of prediction correctness for image i , has value 1, otherwise, it has a value of 0. *Accuracy* should be then calculated as in Equation (1).

$$Accuracy = \frac{\sum_{i=1}^M Clean_Result(i)}{M} \quad (1)$$

Equation (1) denotes the capability of the service to produce correct predictions when no corruptions or attacks are introduced.

How to determine whether the prediction from the target service is correct is presented in 4.7.6.

4.7.2 Robustness against corruptions

The target service shall be tested against all the corruptions that are decided during the test planning phase. To test the capability of the target service being evaluated against a specific corruption j , the M images shall be respectively input to the corruption algorithm j to generate the corruption test samples. $Corr_sample(i, j)$ in Figure 4 is the corrupted sample image obtained from image i with corruption j added, and $Prediction_Corr(i, j)$ is the prediction result for $Corr_sample(i, j)$.

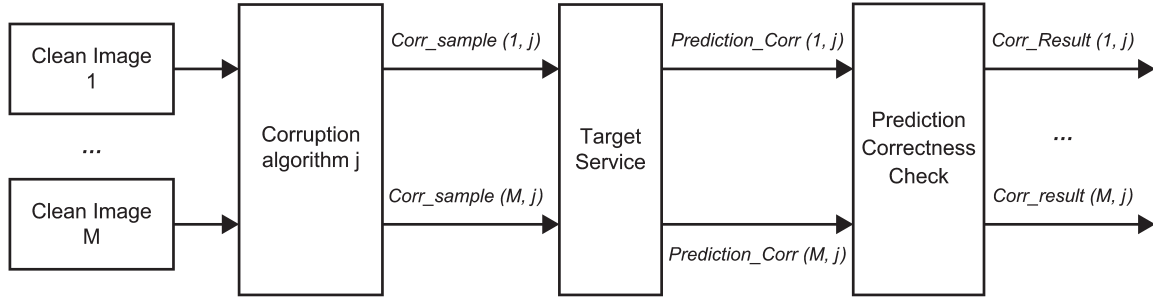


Figure 4—Evaluation of Robustness against Corruption j

If the target service produces the correct prediction for $Corr_sample(i, j)$, then $Corr_Result(i, j)$, the prediction correction indicator, has value 1, otherwise, it has value 0.

$Robustness_Corr(j)$ should be calculated as in Equation (2), and it denotes the capability of the service under evaluation to produce correct predictions under the corruption j .

$$Robustness_Corr(j) = \frac{\sum_{i=1}^M Corr_Result(i, j)}{M} \quad (2)$$

$Average_Robustness_Corr$, which denotes the average capability of the service under evaluation to resist a corruption, should then be calculated as shown in Equation (3). N is the number of corruptions to be tested against.

$$Average_Robustness_Corr = \frac{\sum_{j=1}^N Robustness_Corr(j)}{N} \quad (3)$$

4.7.3 Robustness against adversarial attacks

The test against adversarial attacks shall be performed similarly, as shown in Figure 5.

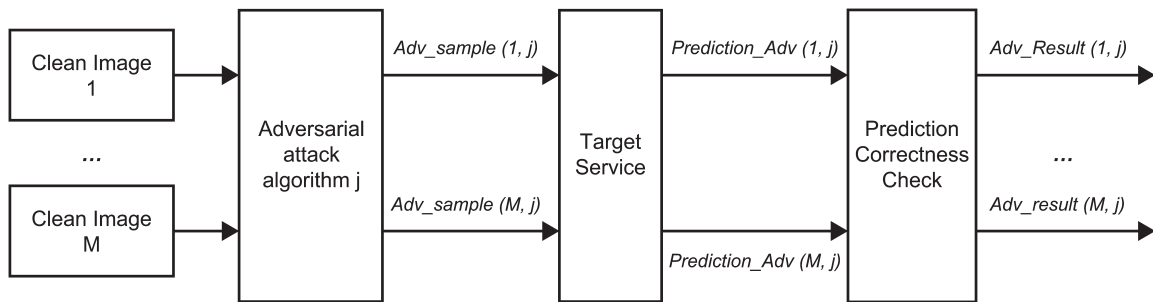


Figure 5—Evaluation of Robustness against Adversarial Attack j

$Robustness_Adv(j)$ is the capability of the service under evaluation to produce correct predictions under the adversarial attack j and should be calculated as in Equation (4).

$$Robustness_Adv(j) = \frac{\sum_{i=1}^M Adv_Result(i, j)}{M} \quad (4)$$

$Average_Robustness_Adv$, denotes the average capability of the service under evaluation to resist an adversarial attack, should be calculated as in Equation (5). K is the number of corruptions to be tested against.

$$Average_Robustness_Adv = \frac{\sum_{j=1}^K Robustness_Adv(j)}{K} \quad (5)$$

4.7.4 Worst-case robustness against corruptions

It may be useful to test the worst-case robustness of the target service against all corruptions (see Figure 6).

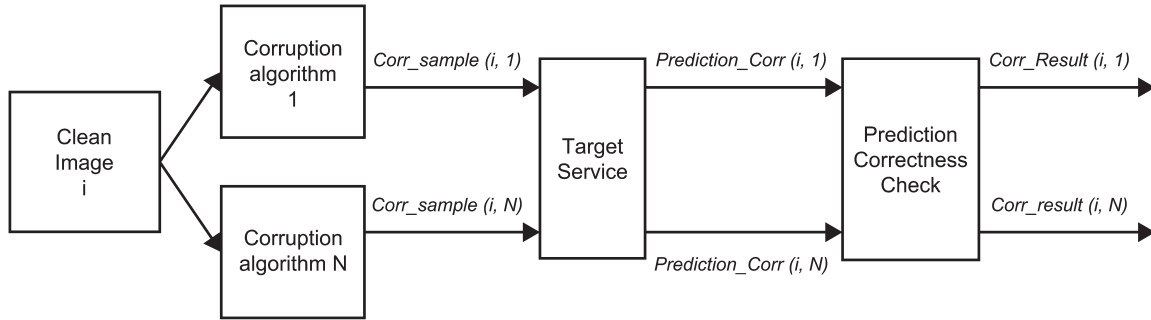


Figure 6—Worst-Case Robustness against Corruptions

For a clean image i , each corruption is respectively added, resulting in N corrupted images. $Corr_sample(i, j)$ is the corrupted image of image i with corruption j added. These corrupted images are fed into the target service respectively. If the target service produces the correct prediction for $Corr_sample(i, j)$, then $Corr_result(i, j)$ has value 1, otherwise, it has value 0. When the target service is able to produce correct predictions for all $Corr_sample(i, j)$, meaning $Corr_result(i, j) = 1$ for all $j = 1, ..N$, then $\prod_{j=1}^N Corr_Result(i, j) = 1$.

The worst-case robustness against corruptions should be calculated in Equation (6).

$$WorstCase_Robustness_Corr = \frac{\sum_{i=1}^M \prod_{j=1}^N Corr_Result(i, j)}{M} \quad (6)$$

$WorstCase_Robustness_Corr$ denotes the percentage of corruption test samples that are correctly recognized by the target service under all the corruptions being tested against.

4.7.5 Worst-case robustness against adversarial attacks

The tests for the worst-case robustness against adversarial attacks is performed similarly, as shown in Figure 7.

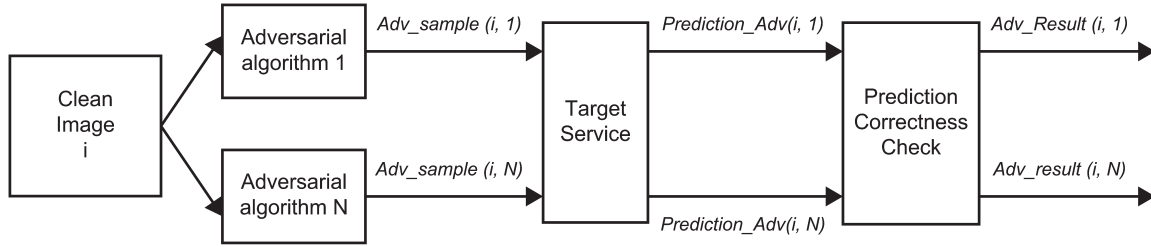


Figure 7—Worst-Case Robustness against Adversarial Attacks

$WorstCase_Robustness_Adv$ should be calculated in Equation (7), and it denotes the percentage of adversarial test samples that are correctly recognized by the target service under all the adversarial attacks being tested against.

$$WorstCase_Robustness_Adv = \frac{\sum_{i=1}^M \prod_{j=1}^N Adv_Result(i, j)}{M} \quad (7)$$

4.7.6 Metrics calculation in operation

4.7.6.1 Label sets

In this standard, before the test, the tester shall obtain from the service provider the knowledge on the label sets used by the target service. In preparing and labeling the clean test images, the tester shall use the same labeling sets as those of the training images for the target service.

4.7.6.2 Prediction correctness for services generating predictions on labels

In performing the tests and calculating the metrics, how to determine whether the predictions generated by the target service depends on whether the prediction is single or multiple.

For a target service that generates a single prediction for a given test image, the target service should be considered to have generated the correct prediction, if the prediction matches the label for the test image. Figure 8 shows an example case for a single prediction.

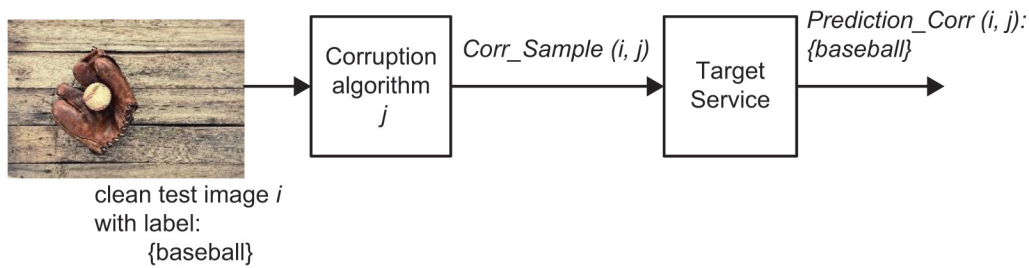


Figure 8—An example case for single prediction

A clean test image i with the label $\{baseball\}$ is used. After applying corruption j , the image i with corruption j added is input to the target service. The target service produces result $prediction_Corr(i, j)$, and $prediction_Corr(i, j)$ is also $\{baseball\}$. Therefore, the prediction is correct, since the prediction matches the label for the clean image.

For a target service that generates multiple predictions with confidences, in preparing and labeling the test images, a given clean test image may have multiple labels, denoted in this standard as $l^1 \dots l^s$, where $l^i \in L^i$ and $L^i, i=1 \dots s$ are the label sets. Figure 9 shows an example case where there are two label sets, L^1 and L^2 , for the training images for the target service, and a clean test image is labeled according the two label sets and has two labels, namely, $l^1 = \{ball\ games\} \in L^1$ and $l^2 = \{baseball\} \in L^2$.

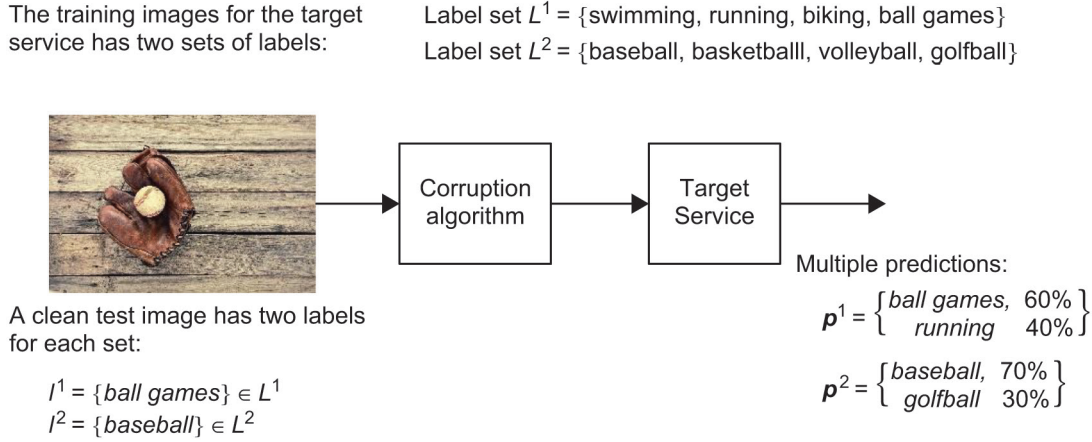


Figure 9—An example case for multiple label sets and multiple labels for a given image

After a certain corruption or adversarial attack is applied to the clean test image, the target service usually generates multiple predictions with each $p^i = \{p_1^i, \dots, p_t^i\}$, $i=1 \dots s$ where each $p_j^i \in p^i$ has varied confidence, and p^i is the multi-label prediction associated with Label set L^i . In the example in Figure 9, the clean image with two labels is added with certain corruption, and then input to the target service that generates two predictions p^1 and p^2 . In p^1 , the target service generates two results, *ball games* with confidence of 60%, and *running* with confidence of 40%. In p^2 , the target service generates two results, *baseball* with confidence of 70%, and *golfball* with confidence of 40%.

The approach to determine whether the predictions are correct shall be negotiated between the tester and the test requester. Approaches in the following subclauses should be used.

4.7.6.2.1 Correctness of Prediction p^i

There are multiple ways to determine the correctness of $p^i = \{p_1^i, \dots, p_t^i\}$, which is the prediction associated with Label Set L^i . Since each result $p_j^i \in p^i$ has a varied confidence associated, it is desirable to select the ones with strong confidence. The following alternatives can be selected.

- *Alternative 1:* Select p_j^i with the most confidence, and if p_j^i equals to l^i which is the label associated with the label Set L^i , then p^i should be considered correct. In the example shown in Figure 9, in p^1 , $\{ball\ games\}$ with confidence of 60% is selected as it has the largest confidence among the two, and p^1 is considered correct because the selected result, which is $\{ball\ games\}$, matches $l^1 = \{ball\ games\}$ of the clean test image. Likewise, in p^2 , $\{baseball\}$ with confidence of 70% is selected, p^2 is considered correct as the selected $\{baseball\}$ matches l^2 .

- *Alternative 2:* Select K p_j^i whose confidence are among the top- K confidences, and if the selected K p_j^i overlap l^i , then p^i should be considered correct. If we let $K = 2$, then in p^1 , both ball games and running are selected, and p^1 is considered correct because it overlaps l^1 . Likewise, p^2 is considered correct because it overlaps l^2 .
- *Alternative 3:* Set a threshold and select p_j^i whose confidence exceeds σ . If the selected p_j^i overlap l^i , then p^i should be considered correct. If we set threshold equals to 65%, then in p^2 {baseball} with confidence of 70% is selected, and p^2 is considered correct. But since in p^1 , neither result is selected as neither exceeds the threshold of 65%, p^1 is considered incorrect.

Which alternative to use and the parameter setting for K or σ shall be agreed with between the tester and the test requester.

4.7.6.2.2 Correctness of Prediction for an image

For an image with certain corruption or adversarial perturbation, the target service generates multiple p^i , $i = 1 \dots s$. To determine whether the target service has generated the correct prediction for the image, the correctness of each p^i should be taken into account. The following alternatives should be employed:

- *Alternative 1:* If at least Q p^i are correct where $1 \ll Q \ll s$, then the target service should be considered to have generated correct prediction for the image.
- *Alternative 2:* If p^i is correct for each $i = 1 \dots s$, then the target service should be considered to have generated correct prediction for the image.

Which alternative to use and the parameter setting for Q shall be agreed with between the tester and the test requester.

4.7.6.3 Prediction correctness for service generating prediction on bounding boxes

4.7.6.3.1 Object Detection Service

Subclause 4.7.6.1 and 4.7.6.2 describe the cases where the predictions generated by the target service are single or multiple labels. For object detection service, the predictions generated by the service may also include bounding boxes. The approaches in 4.7.6.2 generally apply with the following changes:

- A given clean test image may have multiple ground truth bounding boxes, $l^1 \dots l^s$, where l^i is the ground truth bounding box associated with object i to be detected.
- After applying certain corruption or adversarial perturbation to the clean test image, the target object detection service usually generates multiple predictions with each $p^i = \{p_1^i, \dots, p_t^i\}$, $i = 1 \dots s$ where p^i comprises the bounding boxes associated with object i , and each bounding box $p_j^i \in p^i$ has varied intersection over union (IoU) with respect to the truth bounding box l^i .
- The approaches to determine whether p^i is correct is the same to those in 4.7.6.2, except that IoU should be used instead of confidence.

4.7.6.3.2 Semantic segmentation

The semantic segmentation service additionally classifies each pixel into a class from a set of classes. In preparing and labeling a test image, the image has one label for each pixel, denoted at this standard as $l^1 \cdots l^s$, where $l^i \in \mathbf{L}^i$ and \mathbf{L}^i is a class set, and s is the number of pixels.

After applying certain corruption or adversarial attack to the clean test image, the semantic segmentation service usually multiple predictions with each $\mathbf{p}^i = \{p_1^i, \cdots p_t^i\}$, $i = 1 \cdots s$ where \mathbf{p}^i is the each $p_j^i \in \mathbf{p}^i$ has varied confidence.

In determining the correctness of \mathbf{p}^i , Alternative 1 in 4.7.6.2 should be used.

To determine whether the segmentation is correct for the test image, most of Y interested target classes should be selected, and if the mIoU of the Y interested target classes exceeds \gg , then the target service should be considered to have generated correct segmentation for the image. The parameter setting for \gg shall be agreed with between the tester and the test requester.

5. Test cases

5.1 Test case for Accuracy test

To gain statistical significance, the number of clean images, M , used in the test should be no smaller than 10000. Clean images can be obtained from public data sets (IMAGENET [B17]), from web crawling, from the image repository maintained by testers, or from test requesters. It is preferable the clean images are provided with labels, otherwise, image labeling is needed. These clean images are fed into the service under evaluation, and *Accuracy* should be calculate as explained in 4.7.1.

5.2 Test case for common corruptions

In testing the performance of target service against common corruptions, the M clean images in 5.1 shall be used. For each of the corruptions to be tested against, how to test and calculate *Average_Robustness_Corr* and *WorstCase_Robustness_Corr* should be performed as specified in 4.7.2 and 4.7.4. This standard does not intend to enumerate all the corruptions nor intend to suggest test all the corruptions enumerated. Which corruptions are to be tested against shall be decided by the tester requester and the tester. Corruption algorithms that apply corruptions to a clean image can generally be found in open source projects, such as GitHub (Busalev, et al. [B4], Hendrycks and Dietterich [B14]).

The severity of the corruptions has great impact on the image quality as well as classification *Accuracy*. Therefore, robustness on different corruptions should be evaluated under different severities. The severities to be used shall be negotiated between the tester and test requester.

5.3 Test case for style transfer

In testing the performance of the target service against style transfer, the M clean images in 5.1 shall be used. Common style transfer includes sketch, oil painting, Picasso, and Mondrian. Which styles to test against shall be determined by the tester and test requester. How to test and calculate the evaluation metrics for style transfer is similar to those of common corruptions. Different style algorithms which transfer a clean test image can be found in open source project (Huang and Belongie [B15]).

5.4 Test case for adversarial perturbations

In testing the performance of the target service against adversarial perturbation, the M clean images in 5.1 shall be used. What adversarial attacks will be tested against shall be agreed with between the tester and the test requester. How to test and calculate the evaluation metrics *Average_Robustness_Corr* and *WorstCase_Robustness_Corr* should follow 4.7.3 and 4.7.6. Adversarial perturbation algorithms that apply adversarial perturbation to a clean test image can be found in is available from open source project and open toolbox (Busalev, et al. [B4], Hendrycks and Dietterich [B14], etc.).

L_p bound should be pre-determined before the tests against adversarial attacks. which measures the perturbation added to the original image when crafting an adversarial sample.

All the query-based attacks are measured by the attack success rate and the query numbers, which also reflect the robustness of the target model. A maximum query number Q_m shall be set up. During Q_m number of queries, once the target model is successfully misclassified, the query process will end and output the number of current queries Q . Finally, the final attack success rate and the average query number shall be calculated, which measures the attack performance of the query-based attack.

Annex A

(informative)

Example common corruptions and adversarial attacks

A.1 Examples of common corruptions

A.1.1 Noise

Noise is typically defined as a random variation in brightness or color information. Common image noise includes Gaussian noise, Poisson noise, and impulse noise.

- a) Gaussian noise is statistical noise where a probability density function equals the normal distribution. A noisy image has pixels that are made up of the sum of their original pixel values plus a random Gaussian noise value.
- b) Poisson noise is electronic noise caused by the discrete nature of light itself. Like Gaussian noise, a noisy image has pixels that are made up of the sum of their original pixel values plus a random Poisson noise value.
- c) Impulse noise is a color analog of salt-and-pepper noise and can be caused by bit errors. Impulse noise is always independent and uncorrelated to the image pixels and is randomly distributed over the image. Hence unlike Gaussian noise, for an impulse noise corrupted image, a number of image pixels will be noisy and the rest of the pixels will be noise-free. There are different types of impulse noise, namely, salt and pepper type of noise and random valued impulse noise.

A.1.2 Blur

Blur refers to the loss of image details due to motion, defocus, and other reasons in the process of image acquisition, including defocus blur, frosted glass blur, motion blur, zoom blur, and Gaussian blur.

- a) Defocus blur occurs when an image is out of focus. It can be implemented by filtering, where the kernel is a Gaussian blurred disk.
- b) Frosted glass blur appears with “frosted glass” windows or panels. It can be implemented by swapping adjacent pixels randomly.
- c) Motion blur appears when a camera is moving quickly, including translation and rotation. It can be implemented by filtering, where the kernel is the average of the translation/rotation kernel and the identity kernel.
- d) Zoom blur occurs when a camera moves toward an object rapidly. It can be implemented by mixing a scaled image and the original image.
- e) Gaussian blur, as known as Gaussian smoothing, occurs when high-frequency information is lost. It can be implemented by filtering the original image with a Gaussian kernel.

A.1.3 Weather

Weather corruptions refer to adding some adverse weather conditions to the original image. The common adverse weather conditions include snow, frost, fog, and brightness.

- a) Snow is a visually obstructive form of precipitation.
- b) Frost forms when lenses or windows are coated with ice crystals.
- c) Fog shrouds objects and is rendered with the diamond-square algorithm.
- d) Brightness varies with daylight intensity.

A.1.4 Digital changes

Digital changes are the changes in color, shape, and resolution of an image after digital processing. Common digital changes include brightness, contrast, geometric transformations, pixelation, and compression loss.

- a) Brightness varies with daylight intensity, which can be modified by increasing or decreasing the H channel value of the image in HSV format.
- b) Contrast can be high or low depending on lighting conditions and the photographed object's color. The contrast of an image can be modified by interpolating the original image and the mean value.
- c) Geometric transformations include similarity transformation and shear mapping, and affine transformation with six degrees of freedom can be obtained by superimposing the above two types of operations. Geometric transformation of an image can be obtained by a pixel-by-pixel affine transformation, and pixels outside the field of view are usually filled with a constant.
- d) Pixelation occurs when up-sampling a low-resolution image. It can be obtained by down-sampling the original image, and then up-sample the low-resolution one into the original shape.
- e) Compression loss occurs when compressing images with a lossy image compression format like JPEG. It will increase image pixelation and introduces artifacts.

A.1.5 Mask

Mask refers to a part of the image that is randomly occluded.

A.2 Example adversarial attacks

A.2.1 Transferable adversarial attacks

The following transferable adversarial attacks should be tested:

- *Fast Gradient Sign Method (FGSM)*: uses the gradient direction of the loss function with respect to the input image to generate a fixed amount of perturbation (Goodfellow, Shlens, and Szegedy [B11]).
- *Basic Iteration Method (BIM)*: runs multiple iterations of FGSM with a small perturbation (Kurakin, Goodfellow, and Bengio [B18]).
- *Projected Gradient Descent (PGD)*: is a noisy version of BIM with larger step size and projection operations (Madry, et al. [B22]).

- *Momentum Iterative Fast Gradient Sign Method (MI-FGSM)*: boosts the adversarial attack with momentum (Dong, et al. [B10]).
- *Spatial Attack* [B11]: generates adversarial examples based on spatial transformation instead of direct manipulation of pixel values (Xia et al. [B31]).
- *Decoupling Direction Norm Attack*: optimizes the cross-entropy loss, and instead of penalizing the norm in each iteration, projects the perturbation onto a L_2 -sphere centered at the original image (Rony et al. [B27]).
- *Dispersion Reduction Attack*: builds the dispersion reduction as a strong baseline attack to evaluate model robustness against black box attacks, which generates adversarial examples using simple and readily-available image classification models (Lu et al. [B21]).
- *CWAttack*: create a set of attacks that can be used to construct an upper bound on the robustness of neural networks (Carlini and Wagner [B5]).
- *Elastic-net Attacks*: formulate the process of attacking deep neural networks (DNNs) via adversarial examples as an elastic-net regularized optimization problem. Elastic-net attacks to DNNs feature L_1 -oriented adversarial examples and include the state-of-the-art L_2 attack as a special case (Chen, Sharma, et al. [B7]).
- *Fast Adaptive Boundary Attacks*: concentrates on the internal layers of DNN representations to produce a new class of adversarial images (Sabour, Cao, Faghri, and Fleet [B28]).
- *Sparse L_1 Descent Attack*: improves the efficiency of the PGD attacks by employing finer control over the sparsity of an update step (Tramèr and Boneh [B30]).
- *Fast Feature Attack*: concentrates on the internal layers of DNN representations in generating adversarial images. The Euclidian distance between the internal DNN presentations of an adversarial image and the clean image is as small as possible (Sabour, Cao, Faghri, and Fleet [B28]).

A.2.2 Query attacks

The following query attacks should be tested.

- *Zeroth order optimization (ZOO)*: performs pixel-level gradient estimation first and then performs white-box C&W attack with the estimated gradients which exhibits high success rate but suffers from intensive computation and huge queries due to element-wise gradient estimation (Chen, Zhang, et al. [B8]).
- *Boundary Attack*: starts from a large adversarial perturbation and performs random walks on the decision boundary while keeping adversarial (Brendel, Rauber and Bethge [B3]).
- *HopSkipJumpAttack (HSJA)*: boosts BoundaryAttack by estimating the gradient direction via binary information at the decision boundary, which is based on a Monte Carlo estimate (Chen, Jordan, and Wainright [B6]).
- *Query Efficient Boundary-based Attack*: enhances HSJA for better gradient estimation using the perturbation sampled from various subspaces, including spatial, frequency, and intrinsic components (Li et al. [B19]).
- *qFool*: assumes that the curvature of the boundary is small around adversarial examples and adopts several perturbation vectors for efficient gradient estimation (Liu, Moosavi-Dezfooli, and Frossard [B20]).
- *GeoDA*: approximates the local decision boundary by a hyperplane and searches the closest point to the benign sample on the hyperplane as the adversary (Rahmaati, Moosavi-Dezfooli, Frossard and Dai [B26]).

- *Surfree*: iteratively constructs a circle on the decision boundary and adopts binary search to find the intersection of the constructed circle and decision boundary as the adversary without any gradient estimation (Maho, Furon, and Le Merrer [B23]).
- *Bandit Attack*: generates the black-box adversarial example attacks by introducing gradient priors. Two to four times fewer queries are needed when using this method (Illayas, Engstrom, and Madry [B16]).
- *Greedy Local Search*: adds perturbation to a randomly selected single pixel or a small set of them. Greedy local-search is used in selecting a small set of pixels to perturb. This improves the effectiveness of this attack (Narodytska and Kasiviswanathan [B24]).
- *Discrete Cosine Transformation Attack*: proposes to find a low-dimensional subspace that contains a high density of adversarial examples, in order to improve query inefficiency. The method utilizes the discrete cosine transform, which decomposes a signal into cosine wave components (Guo, Frank, and Weinberger [B12]).
- *Natural Gradient Descent (NGD) Based Attack*: proposes a zeroth-order natural gradient descent method to design the adversarial attacks. The method incorporates the zeroth order gradient estimation technique catering to the black-box attack scenario and the second-order natural gradient descent to achieve higher query efficiency (Zhao, Chen, Wang, and Lin [B32]).
- *Output Diversity Sampling (ODS) Method*: proposes to perturb an input away from the original image as measured directly by distances in the output space. First, a direction is randomly specified in the output space. Next, gradient-based optimization is performed to generate a perturbation in the input space that yields a large change in the specified direction (Toshiro, Song, and Ermon [B29]).
- *Prior-Guided Random Gradient Free (P-RGF) Method*: proposes a P-RGF method in generating adversarial perturbations. The P-RGF method uses a transfer-based prior and the query information simultaneously (Cheng, et al. [B9]).
- *SignHunter Attacks*: focuses on estimating just the sign of the gradient by reformulating the problem as minimizing the Hamming distance to the gradient design (Al-Dujaili and O'Reilly [B1]).
- *SimBA Attacks*: are efficient black-box attacks that exploit the confidence scores to create the adversarial perturbations (Guo, Gardner, You, Wilson, and Weinberger [B13]).
- *Square Attack*: is a score-based attack based on a randomized search scheme that selects localized square-shaped updates at random positions so that at each iteration the perturbation is situated approximately at the boundary of the feasible set (Andriushchenko, Croce, Flammarion, and Hein [B2]).

Annex B

(informative)

Example evaluation

Table B.1 gives an example of a target image classification service, the test image used, the corruptions and adversarial attacks to be tested against.

Table B.1—Example of target image classification service

No.	Tested target	Metrics	Prediction Correctness	Clean Image Data set	Corruptions	Adversarial Attack
1	Image classification service	<i>Accuracy</i> <i>Robustness_Corr</i> <i>Robustness_Adv</i> <i>WorstCase_Robustness_Corr</i> <i>WorstCase_Robustness_Adv</i>	4.7.6.1	ImageNet	Noise Gaussian noise, Poisson noise, impulse noise. Blur defocus blur, frosted glass blur, motion blur, zoom blur. Weather snow, frost, fog, brightness. Digital changes contrast, geometric transformations, pixelate, compression loss. Mask image occlusion. Style Transfer sketch oil painting Picasso Mondrian	Transferable Adversarial Attack SpatialAttack DDNAttack DRAttack CWAttack EADAttack Fast_adaptive_boundary FGSM BIM PGD MIFGSM SparseL1 DescentAttack FastFeatureAttack Query Attack BanditAttack LocalSearch DCTAttack NES NGD ODS RGF P-RGF SignHunter SimBA Square: Square Attack SinglePixelAttack

Annex C

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

- [B1] Al-Dujaili, A. and U.-M. O'Reilly, "There are No Bit Parts for Sign Bits in Black-Box Attacks."¹⁰
- [B2] Andriushchenko, M., F. Croce, N. Flammarion, and M. Hein, "Square Attack: A Query-Efficient Black-Box Adversarial Attack via Random Search," European Conference on Computer Vision, pp. 484–501, 2020.¹¹
- [B3] Brendel, W., J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models."¹²
- [B4] Busalev, A., A. Parinov, V. I. Iglovikov, E. Khvedchenya, and M. Druzhinin, "Albumentations."¹³
- [B5] Carlini, N. and D. Wagner, "Towards evaluating the robustness of neural networks," IEEE Symposium on Security and Privacy (SP), pp. 39–57, 2017.^{14,15}
- [B6] Chen, J., M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A Query-Efficient Decision-Based Attack," IEEE Symposium on Security and Privacy (SP), pp. 1277–1294, 2020.
- [B7] Chen, P.-Y., Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples," AAAI Conference on Artificial Intelligence, 2018.¹⁶
- [B8] Chen, P.-Y., H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models," Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 15–26, 2017.¹⁷
- [B9] Cheng, S., Y. Dong, T. Pang, H. Su, and J. Zhu, "Improving Black-box Adversarial Attacks with a Transfer-based Prior."¹⁸
- [B10] Dong, Y., F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting Adversarial Attacks with Momentum," IEEE Conference on Computer Vision and Pattern Recognition, 2018.
- [B11] Goodfellow, I. J., J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," International Conference on Learning Representations, 2015.¹⁹
- [B12] Guo, C., J. S. Frank, and K. Q. Weinberger, "Low frequency adversarial perturbation."²⁰

¹⁰Available at: <https://arxiv.org/abs/1902.06894>

¹¹Available at: <https://arxiv.org/abs/1912.00049>.

¹²Available at: arXiv preprint arXiv:1712.04248.

¹³Available at: <https://github.com/albumentations-team/albumentations>.

¹⁴The IEEE standards or products referred to in this annex are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

¹⁵IEEE publications are available from The Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (<https://standards.ieee.org/>).

¹⁶Available at: <https://arxiv.org/abs/1709.04114>.

¹⁷Available at: <https://arxiv.org/abs/1708.03999>.

¹⁸Available at: <https://arxiv.org/abs/1906.06919>.

¹⁹Available at: <https://arxiv.org/abs/1412.6572>.

²⁰Available at: <https://arxiv.org/abs/1809.08758>.

- [B13] Guo, C., J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, “Simple Black-box Adversarial Attacks.”²¹
- [B14] Hendrycks, D. and T. Dietterich, “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations,” 2019.²²
- [B15] Huang, X. and S. Belongie, “AdaIN-style.”²³
- [B16] Ilayas, A., L. Engstrom, and A. Madry, “Prior convictions: Black-box adversarial attacks with bandits and priors.”²⁴
- [B17] IMAGENET.²⁵
- [B18] Kurakin, A., I. Goodfellow, and S. Bengio, Adversarial examples in the physical world. Artificial Intelligence Safety and Security, 2018, pp. 99–112.²⁶
- [B19] Li, H., X. Xu, X. Zhang, S. Yang, and B. Li, “QEBA: Query-Efficient Boundary-Based Blackbox Attack,” Conference on Computer Vision and Pattern Recognition(CVPR), 2020.
- [B20] Liu, Y., S.-M. Moosavi-Dezfooli, and P. Frossard, “A geometry-inspired decision-based attack.” *arXiv preprint arXiv:1903.10826*.²⁷
- [B21] Lu, Y., Y. Jia, J. Wang, B. Li, W. Chai, L. Carin, and S. Velipasalar, “Enhancing Cross-task Black-Box Transferability of Adversarial Examples with Dispersion Reduction,” IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2020.
- [B22] Madry, A., L. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards Deep Learning Models Resistant to Adversarial Attacks.”²⁸
- [B23] Maho, T., T. Furon, and E. Le Merrer, “SurFree: A Fast Surrogate-Free Black-Box Attack,” IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2021.
- [B24] Narodytska, N and S. P. Kasiviswanathan, “Simple Black-Box Adversarial Perturbations for Deep Networks.”²⁹
- [B25] Neural Style Transfer, 2022.³⁰
- [B26] Rahmati, A., S.-M. Moosavi-Dezfooli, P. Frossard, and H. Dai, “GeoDA: A geometric framework for black-box adversarial attacks,” IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
- [B27] Rony, J., L. G. Hafemann, L. S. Oliveria, I. Ben Ayed, R. Sabourin, and E. Granger, “Decoupling Direction and Norm for Efficient Gradient-Based L2 Adversarial Attacks and Defenses.”³¹

²¹ Available at: <https://arxiv.org/abs/1905.07121>.

²² <https://github.com/hendrycks/robustness>.

²³ Available at: <https://github.com/xunhuang1995/AdaIN-style>.

²⁴ Available at: arXiv preprint arXiv:1807.07978.

²⁵ Available at: <https://www.image-net.org/index.php>.

²⁶ Available at: <https://arxiv.org/abs/1607.02533>.

²⁷ Available at: arXiv preprint arXiv:1903.10826.

²⁸ Available at: <https://arxiv.org/abs/1706.06083>.

²⁹ Available at: <https://arxiv.org/abs/1612.06299>.

³⁰ Available at https://www.tensorflow.org/tutorials/generative/style_transfer.

³¹ Available at: <https://arxiv.org/abs/1811.09600>.

- [B28] Sabour, S., Y. Cao, F. Faghri, and D. J. Fleet, “Adversarial manipulation of deep representations.”³²
- [B29] Tashiro, Y., Y. Song, and S. Ermon, “Diversity can be Transferred: Output Diversification for White- and Black-box Attacks.”³³
- [B30] Tramèr, F. and D. Boneh, “Adversarial training and robustness for multiple perturbations,” International Conference on Neural Information Processing Systems, pp. 5866–5876, 2019.³⁴
- [B31] Xia, C., J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, “Spatially transformed adversarial examples.”³⁵
- [B32] Zhao, P., P.-Y. Chen, S. Wang, and X. Lin, “Towards Query-Efficient Black-Box Adversary with Zeroth-Order Natural Gradient Descent.”³⁶

³² Available at: arXiv preprint arXiv:1511.05122.

³³ Available at: <https://arxiv.org/abs/2003.06878>.






³⁴ Available at: <https://arxiv.org/abs/1904.13000>.

³⁵ Available at: <https://arxiv.org/abs/1801.02612>.

³⁶ Available at: <https://arxiv.org/abs/2002.07891>.

RAISING THE WORLD'S STANDARDS

Connect with us on:

-  **Twitter:** twitter.com/ieeesa
-  **Facebook:** facebook.com/ieeesa
-  **LinkedIn:** linkedin.com/groups/1791118
-  **Beyond Standards blog:** beyondstandards.ieee.org
-  **YouTube:** youtube.com/ieeesa

standards.ieee.org
Phone: +1 732 981 0060