FEDERAL UNIVERSITY OF ALAGOAS
COMPUTING INSTITUTE
GRADUATE PROGRAM IN INFORMATICS

Filipe Falcão Batista dos Santos

ASSESSING THE RELIABILITY OF MACHINE LEARNING CLOUD SERVICES THROUGH FAULT INJECTION

Maceió - AL
2022

FILIPE FALCÃO BATISTA DOS SANTOS

ASSESSING THE RELIABILITY OF MACHINE LEARNING CLOUD SERVICES THROUGH FAULT INJECTION

Dissertation presented to the Graduate Program in Informatics of the Federal University of Alagoas as a requirement for obtaining the Master's degree in Informatics.

Advisor: Baldoino Fonseca dos Santos Neto, Ph.D.
Co-Advisor: Márcio de Medeiros Ribeiro, Ph.D.

Maceió - AL
2022

Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

S237a Santos, Filipe Falcão Batista dos.
Assessing the realiability of machine learning cloud services through fault injetion / Filipe Falcão Batista dos Santos. – 2022.
45 f. : il.

Orientador: Baldoino Fonseca dos Santos Neto.
Co-orientador: Márcio de Medeiros Ribeiro.
Dissertação (mestrado em informática) - Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2022.
Texto em inglês.

Bibliografia: f. 42-45.

1. Computação em nuvem. 2. Injeção de falhas (Engenharia de *software*. 3. Aprendizagem de máquina. I. Título.

CDU: 004.81:159.953.5

I would like to dedicate this work to my wife, Renata, for her love and affection, for always being at my side, and for tirelessly supporting me during the rough months of writing this dissertation.

To my parents, Marli and Josimar; to my siblings, Josimar Jr. and Juliana; to my brother-in-law, Weslley; and to my niece, Letícia, a big Dog Patrol fan; for their love, support, guidance, and dedication, relentlessly helping me to achieve my goals and dreams.

To my advisor, Baldoino Fonseca; to my co-advisor Márcio Ribeiro; and to all my research colleagues; for their guidance and support in my research.

To all my friends and colleagues for their friendship, support, and good moments.

To my canine friends, Flash and Mel, for being the best companions a human being could ever ask for.

ACKNOWLEDGEMENTS

I would like to thank my wife and my parents for their love, dedication, and support at all times. To Prof. DSc. Baldoino Fonseca and Prof. DSc. Márcio Ribeiro, for the time invested in me and in my work. To Prof. DSc. Rohit Gheyi and Prof. DSc. Marcelo Costa for their willingness and interest in contributing to the improvement of this dissertation. And to all my research colleagues for their invaluable contribution to the elaboration of this work.

RESUMO

Construir sistemas de Aprendizado de Máquina (AM) pode ser complexo, e as tecnologias de AM são conhecidas por terem uma curva de aprendizado acentuada. Essas dificuldades levaram à popularização dos serviços de AM. Embora muitos estudos recentes tenham abordado a vulnerabilidade dos modelos de AM para ataques direcionados, pouca atenção foi direcionada ao efeito de falhas de dados típicas na confiabilidade dos serviços de AM. Tais falhas podem ter origem nas aplicações que dependem de serviços de AM, causadas por falhas de hardware ou conexão, bugs ou comportamento indefinido. Consequentemente, essas falhas podem ser refletidas nos dados produzidos por tais aplicações e enviados aos serviços de AM. Buscando avaliar a confiabilidade desses serviços e com foco no domínio da Visão Computacional, este trabalho apresenta um estudo empírico sobre a injeção de falhas comuns de dados nos dados enviados aos serviços de visão computacional. Os resultados de 11 serviços comerciais indicam que falhas de dados podem afetar significativamente a confiabilidade dos serviços, conforme evidenciado por taxas de classificação incorreta que variam de 14% a até 63%. Por outro lado, as falhas de dados não parecem ter o mesmo impacto na equidade dos serviços, embora algumas configurações de falhas levem a impactos significativos na equidade.

Palavras-chaves: Serviços em Nuvem. Confiabilidade. Injeção de Falhas. Aprendizado de Máquina.

ABSTRACT

Building Machine Learning (ML) systems can be tricky, and ML technologies are known to have a steep learning curve. Those difficulties led to the popularization of ML services. Although many recent studies have addressed the vulnerability of ML models to target attacks, not enough attention has been directed to the effect of typical data faults on the reliability of ML services. Such faults may originate in the applications that rely on ML services, caused by hardware or connection failures, bugs, or undefined behaviour. Consequently, those faults can be reflected in the data produced by such applications and sent to the ML services. Seeking to evaluate the reliability of these services and focusing on the Computer Vision (CV) domain, this work presents an empirical study on the injection of common data faults into the data sent to CV services. The results from 11 commercial CV services indicate that data faults may significantly affect the reliability of the CV services, as evidenced by misclassification rates ranging from 14% to up to 63%. On the other hand, data faults do not seem to have the same impact on the fairness of CV services, even though some fault configurations lead to significant fairness impacts.

Keywords: Cloud Services. Reliability. Fault Injection. Machine Learning.

LIST OF FIGURES

Figure 1 – Image faults applied to a sample image 21
Figure 2 – The most impactful image faults 28
Figure 3 – The top-1 misclassification rates of ZBLUR, SNOW, SNOISE, CONT, and GNOISE faults 30
Figure 4 – The true positive rate (TPR) of the privileged (male - M) and unprivileged (female - F) groups for both clean and faulty images 31
Figure 5 – The average misclassification rates of the computer vision services when data faults are set at their initial parameter levels 34
Figure 6 – The average misclassification rates of the object detection and text detection services 35

LIST OF TABLES

Table 1 – List of selected computer vision services 18
Table 2 – The top-1 misclassification rate of the computer vision services when subjected to the data faults 27
Table 3 – The top-5 faults with the highest difference in the true positive rate (TPR) for the attribute sex 32

CONTENTS

1 INTRODUCTION 11
1.1 Context and Problem 11
1.2 Objectives 12
1.3 Contributions 13
1.4 Thesis Structure 13

2 BACKGROUND 14
2.1 Black-box Computer Vision Services 14
2.2 Faults in Computer Vision Systems 15
2.3 Fairness 15

3 STUDY DESIGN 17
3.1 Selection of Computer Vision Services 18
3.2 Fault Model and Evaluation 19
3.3 Data Faults 20
3.3.1 Blurring 21
3.3.2 Brightness (BRI) 22
3.3.3 Chromatic Aberration (CHR) 22
3.3.4 Climate Conditions 22
3.3.5 Contrast (CONT) 22
3.3.6 Grayscale (GRAY) 23
3.3.7 Noise 23
3.3.8 Pixelation (PIX) 23

4 EXPERIMENTS 24
4.1 Datasets 24
4.2 Fault Injection Campaigns 25
4.3 Assessing Effects of Data Faults (RQ1 & RQ2) 26
4.3.1 General-purpose Services 26
4.3.2 Domain-specific Services 27
4.3.3 Differences Across Cloud Providers 28
4.3.4 Effects of Data Faults 29
4.4 Assessing Effects on Fairness (RQ3) 31

5 DISCUSSION 34
5.1 Effect of Data Faults at Initial Parameters 34
5.2 The Role of k 35

6 IMPLICATIONS 37
6.1 To Developers 37
6.2 To Cloud Providers 38

7 THREATS TO VALIDITY 39
7.1 Construct and Internal Validity 39
7.2 Conclusion and External Validity 39

8 RELATED WORK 40
8.1 Black-box Adversarial Attacks 40
8.2 Attacking Machine Learning Services 40
8.3 Faults in Machine Learning Systems 41
8.4 Fairness 41
8.5 Our Contributions 41

9 CONCLUSION 43

Bibliography 44

1 INTRODUCTION

In this Chapter, we summarize our research by first presenting the context and defining the studied problem, leading to the objectives and contributions.

1.1 CONTEXT AND PROBLEM

The growing amount of public and private data generated from different data sources, like websites, social networks, mobile devices, and news providers, has increased the interest in technologies capable of extracting useful knowledge from large, usually unstructured, data collections. Furthermore, the popularization of wireless sensors, mobile devices, and wearable devices makes it possible to collect data in increasingly specific contexts (RIBEIRO; GROLINGER; CAPRETZ, 2015). Machine Learning (ML) techniques have been successfully employed to extract useful knowledge from such data by both academia and the software industry. Time-consuming tasks, like image classification (GUO et al., 2017), object detection (WANG; SHEN; SHAO, 2017), product recommendation (KUMAR; THAKUR, 2018), and forecasting based on past data (TEALAB, 2018), can be performed by machine learning systems automatically in a matter of seconds.

However, building ML systems can be tricky since massive training data and expensive computational resources are often required. Moreover, the widely used machine learning frameworks, like *Tensorflow*,[1] *PyTorch*,[2] and *scikit-learn*,[3] have a steep learning curve and present several challenges when deployed in a production environment (JAHANGIROVA et al., 2019). To make ML systems more accessible and affordable, many cloud providers (*e.g.*, Amazon Web Services, Google Cloud, Microsoft Azure) offer machine learning tools as services. Application developers can perform ML tasks through those services by simply sending their data to a cloud provider over APIs (Application Programming Interfaces), reducing the effort required to perform ML tasks (RIBEIRO; GROLINGER; CAPRETZ, 2015; SHOKRI et al., 2017).

Recent studies (HESAMIFARD et al., 2018; HUNT et al., 2018; KUMAR et al., 2020; LI et al., 2017; PAPERNOT et al., 2017; SHOKRI et al., 2017; LI et al., 2019; WANG; GONG, 2018) analyzed different aspects of machine learning systems, like privacy (HESAMIFARD et al., 2018; HUNT et al., 2018), scalability (LI et al., 2017), and security (KUMAR et al., 2020; PAPERNOT et al., 2017; SHOKRI et al., 2017; LI et al., 2019; WANG; GONG, 2018). Regarding security, several works have addressed the vulnerability of machine learning models, especially deep learning (GOODFELLOW et al., 2016) models, to adversarial examples (PAPERNOT et al., 2016; GOODFELLOW; SHLENS; SZEGEDY, 2014) and inference attacks (SHOKRI et

[1] <https://www.tensorflow.org>
[2] <https://pytorch.org>
[3] <https://scikit-learn.org>

al., 2017; WANG; GONG, 2018; TRUEX et al., 2019). While adversarial examples consist of malicious, carefully-constructed input data that causes machine learning models to yield incorrect results (PAPERNOT et al., 2016; GOODFELLOW; SHLENS; SZEGEDY, 2014; LI et al., 2019; NARODYTSKA; KASIVISWANATHAN, 2016), inference attacks aim at causing the leak of the data or parameters used to train ML models (SHOKRI et al., 2017; TRUEX et al., 2019; WANG; GONG, 2018).

Although the research on ML models security has indicated that these models are commonly vulnerable to malicious adversaries, little research has addressed the effect of typical *data faults* (*i.e.*, faults whose impact can be checked in the input data of a system) (NURMINEN et al., 2019) on the reliability (*i.e.*, the ability of a system to properly function under stated conditions for a period of time (BIROLINI, 2017)) of machine learning services. Such faults may originate in the applications that rely on ML services, being caused by hardware or connection failures, bugs, and undefined behavior (NURMINEN et al., 2019; CECCARELLI; SECCI, 2022; JHA et al., 2019; JHA et al., 2018). Consequently, those faults can be reflected on the data produced by such applications and sent to the machine learning services (*e.g.*, low image resolution, noise, and missing chunks of data).

Understanding how reliable ML services are when exposed to typical data faults may help application developers and cloud providers. While application developers can be better aware of what to expect from ML services in faulty scenarios (*e.g.*, defective sensors, connection problems), the cloud providers can improve their models to be more reliable when exposed to specific faults. Moreover, the importance of a reliability assessment is even more significant when critical applications, such as medical image analysis (LITJENS et al., 2017) or violence detection (RAMZAN et al., 2019), use ML services.

1.2 OBJECTIVES

Seeking to evaluate the reliability of machine learning services, this work presents an empirical study on the injection of common data faults into the input data passed to a set of commercial ML services, specifically computer vision (CV) services. Performing the injection of faults and then analyzing a system's behaviour is a recognized practice to gather the desired understanding of the reliability of the target system (ZIADE et al., 2004).

To achieve the objective aforementioned, we conducted fault injection campaigns with 14 different image faults representing common issues on 11 computer vision services, including general-purpose (*e.g.*, object detection) and domain-specific (*e.g.*, violence detection), from three widely used cloud providers: Amazon Web Services, Google Cloud, and Microsoft Azure. Moreover, we analyzed the impacts of data faults on the reliability of CV services both in terms of misclassifications and fairness.

1.3 CONTRIBUTIONS

We summarize the main contributions of our study below:

- The injected data faults are able to consistently impact the reliability of CV services, as reflected by the obtained misclassification rates of up to 63%;
- Domain-specific services appear to be more reliable than general-purpose services, as evidenced by consistently lower misclassification rates. Moreover, services with similar capabilities but from different providers may differ significantly in their reliability;
- While the minimum obtained misclassification rate for a fault is 20%, the Zoom Blur, Rain/snow, Salt and Pepper Noise, Contrast, and Gaussian Noise faults lead to a higher impact on reliability, even at initial severity levels;
- Common data faults do not systematically affect the fairness of the studied services, even though some fault configurations can lead to a significant impact. To the best of our knowledge, this is the first work to assess the effect of data faults on the fairness of CV services.

1.4 THESIS STRUCTURE

The remainder of this thesis is structured as follows:

- **Chapter 2** introduces the base concepts required to proceed with the thesis.
- **Chapter 3** introduces our research questions, details the computer vision services used in our study, and presents the fault model and the data faults investigated.
- **Chapter 4** defines our experiments, including the datasets used and the workflow of the fault injection campaigns. Then, we present the experiments' results.
- **Chapter 5** further discusses the results of our experiments.
- **Chapter 6** discusses the implications of our study.
- **Chapter 7** details the limitations and threats to the validity of this study.
- **Chapter 8** discusses the related work on black-box adversarial attacks, attacking machine learning services, faults in machine learning systems, and fairness. This Chapter also compares our work with the mentioned related work in terms of contributions and limitations.
- Finally, **Chapter 9** concludes this thesis with our contributions and future work.

2 BACKGROUND

This chapter contextualizes our work, presenting an overview of black-box computer vision services and faults in computer vision systems, and introduces the concept of fairness.

2.1 BLACK-BOX COMPUTER VISION SERVICES

Due to the high cost involved in training machine learning models with massive training data, which includes the necessity of expensive hardware, and the steep learning curve of ML technologies (RIBEIRO; GROLINGER; CAPRETZ, 2015), many companies are contributing to the trend initiated in recent years of migrating ML applications to the cloud. Indeed, Kaggle's *State of Data Science and Machine Learning 2022* survey[1] shows strong growth in 2022 for all major cloud computing providers. Moreover, the 2021 edition[2] of the same survey shows that less than 20% of the respondents do not use cloud services. Since the providers own both massive amounts of data and computing resources, they provide services that perform complex ML tasks using pre-trained models for a fee. Fees vary across services in a provider and across providers (*e.g.*, 0.001 USD per image for Amazon Rekognition image services),[3] even though all providers have a free tier that also varies across services and providers, but generally allow thousands of free queries.

Provided services belong to different machine learning domains, such as natural language processing (NLP), computer vision, and fraud detection. From the computer vision domain, the focus of our work, commonly provided services are face detection and recognition, label detection, text detection, and inappropriate or offensive content detection. Since pre-trained models power such services, and the cloud providers provide little to no details on the internals of the models, the services are essentially black-box. Hence, developers relying on those services do not have knowledge of the training data, the model architecture, data augmentation steps, or the parameters used for training. The only data developers can access are the input passed to the service and the outputted predictions.

Alternatively, cloud providers also provide Automated Machine Learning (AutoML) services. With AutoML services, developers can leverage the pre-trained models from black-box services to build custom models that perform a particular task (*e.g.*, distinguishing healthy and infected leaves). To accomplish that, developers must provide a few hundred labelled images which will be used to re-train the black-box model, resulting in the custom one. Although developers now have control over a portion of the training data when using AutoML services, they still do not know the majority of the internals of the resulting model.

[1] <https://www.kaggle.com/kaggle-survey-2022>
[2] <https://www.kaggle.com/kaggle-survey-2021>
[3] <https://aws.amazon.com/pt/rekognition/pricing>

2.2 FAULTS IN COMPUTER VISION SYSTEMS

Data-intensive systems, like machine learning applications, deeply rely on the quality of their data and may be subjected to faults caused by hardware or connection failures, bugs, or undefined behavior (CECCARELLI; SECCI, 2022; NURMINEN et al., 2019; JHA et al., 2019). Those faults can then be reflected in the data produced by those systems and passed as input to the machine learning systems powering them. For instance, a noisy image might be the result of a faulty RGB camera. Such fault would then be propagated to the ML model as the faulty input image is forwarded to the model to return a prediction. If the model returns a prediction different from the one that would have been returned if the input image was not faulty, then the fault present in the image successfully affected the model's decision.

To assess the tolerance of ML-powered systems to faults, different studies (JHA et al., 2018; JHA et al., 2019; CECCARELLI; SECCI, 2022) in the field of autonomous vehicles guided by computer vision models have proposed the process of injecting faults into the input data of ML models to simulate the occurrence of real-world faults. Similarly, other works (MA et al., 2018) on the field of mutation testing have proposed injecting faults both at the input data of ML models but also on the models themselves (*i.e.*, removing a layer from a deep learning model). However, in the case of mutation testing, the goal is to test the quality of the test suite instead of the fault tolerance of the model.

For computer vision systems, commonly investigated faults are related to RGB cameras, system bugs, or climate conditions. Previously investigated faults include different formats of noise and blur, pixelated images due to bugs or network issues, or climate conditions that outdoor cameras may easily be exposed to. Moreover, the severity of such faults can, in most cases, be configured (*e.g.*, the number of pixels in an image affected by noise) and the fault tolerance at different levels of severity may lead to relevant insights.

2.3 FAIRNESS

The growth of machine learning in recent years has led to many decisions now being taken by ML models trained over large sets of domain-specific data. Among those models are the ones that perform critical decisions, such as defining if an individual will get a loan (WANG et al., 2020) or be a target of predictive policing (ALIKHADEMI et al., 2021). Although one might think that, since machine learning systems learn to uncover patterns from examples, they would not include human bias in their decision-making process, many cases (BUOLAMWINI; GEBRU, 2018; BAROCAS; HARDT; NARAYANAN, 2019) have shown that fairness issues (*e.g.*, related race, gender, disabilities) may exist in accurate ML systems due to pre-existing biases in the training data.

Assessing the fairness of machine learning systems before releasing them is a growing concern, as doing so would reduce the harm caused by unfair decisions (BAROCAS; HARDT;

NARAYANAN, 2019). To perform such an assessment, many different criteria of fairness have been proposed (VERMA; RUBIN, 2018). Those criteria are based on the concept of *privileged* and *unprivileged* groups of a *sensitive feature* (*i.e.*, a feature that may be a target for discrimination, such as gender, race, or marital status). Although many criteria of fairness exist, some criteria may yield a positive result for bias while others may not since some of the existing criteria are mathematically incompatible (VERMA; RUBIN, 2018).

An example of the application of a range of different fairness criteria can be observed in the work of Verma and Rubin (VERMA; RUBIN, 2018). The authors presented an evaluation of the fairness of a logistic regression classifier of credit scores. The classifier was trained with data containing sensitive gender-related features from the German Credit Dataset (DUA; GRAFF, 2017). While some criteria (*e.g.*, equal opportunity (HARDT; PRICE; SREBRO, 2016)) deem the classifier as fair, other criteria (*e.g.*, equalized odds (HARDT; PRICE; SREBRO, 2016)) deem the classifier as unfair for not satisfying its criteria of fairness. The study concludes that there are indications of gender-related bias toward single males and that different fairness criteria yield different fairness assessments.

3 STUDY DESIGN

This chapter describes the proposed study design to assess the reliability of computer vision cloud services through the injection of common data faults. To perform our study, three research questions are defined:

RQ1. *To what extent are computer vision services able to tolerate the effects of common data faults?*

Studies have shown that computer vision models can change their behaviour and present misclassifications of the input data when exposed to adversarial attacks in both white (PAPERNOT et al., 2016; GOODFELLOW; SHLENS; SZEGEDY, 2014; YUAN et al., 2019; ZHANG; LI, 2019; MOOSAVI-DEZFOOLI; FAWZI; FROSSARD, 2016) and black-box (NARODYTSKA; KASIVISWANATHAN, 2016; PAPERNOT et al., 2017; ILYAS et al., 2018; MICHAELIS et al., 2020; HENDRYCKS; DIETTERICH, 2019) settings. However, little research has addressed the effect of typical data faults on the reliability of computer vision services. Understanding which CV services can better tolerate the effects of data faults may help users that intend to use CV services. For instance, developers building data-intensive CV applications may consider the fault tolerance of computer vision services as a factor before making long-term commitments. Hence, **RQ1** aims to investigate if and to what extent CV services can tolerate data faults.

RQ2. *Are there data faults, or groups of faults, that have a greater effect on computer vision services?*

Different data faults might impact computer vision models in different ways. Understanding which data faults may have a more significant impact on computer vision services may help the developers of such services to prioritize efforts when targeting data faults. Thus, **RQ2** aims at investigating if there is a data fault or collection of data faults that are more harmful to the CV services of cloud providers.

RQ3. *Do common data faults affect the fairness of computer vision services?*

Making sure that computer vision models are not only highly accurate but also fair is a growing concern, as doing so will reduce the harm caused by unfair decisions (BAROCAS; HARDT; NARAYANAN, 2019). Analyzing whether data faults can also affect the fairness of CV models exposed as cloud services may help application developers and services providers understand the effects that common data faults can cause in such an essential aspect of modern computer vision applications.

3.1 SELECTION OF COMPUTER VISION SERVICES

We select services from three major cloud providers for this study. The criteria we use to choose these services are:

1. The services must be commercially available and widely used—this ensures that the services are relevant, actively maintained, and validated by the industry;
2. The services must be ready-made—this ensures that the services are not highly configurable and are simple to use;
3. The services must perform computer vision tasks—this ensures that the services perform tasks from the studied domain.

After applying the criteria, the following set of computer vision services from the following cloud providers was selected: (i) Amazon Rekognition from **Amazon Web Services** (AWS);[1] (ii) Vision AI from **Google Cloud**;[2] and (iii) Vision Services from **Microsoft Azure**.[3] Table 1 presents the list of selected services, their providers, category, and considered labels. The list of considered labels for each service refers to the labels included in our analysis and predictions that do not match these labels are discarded. We can see that different services across providers belong to the same category as they perform the same or very similar tasks. For instance, all providers provide an object detection service.

Table 1 – List of selected computer vision services

Provider	Name	Category	Considered Labels
Amazon Web Services (AWS)	Celebrity Recognition	Celebrity Detection	All
	Label Detection	Object Detection	All
	Moderation Labels Detection	Violence Detection	Violence, Visually Disturbing
	Text Detection	Text Detection	–
	Face Detection	Face Detection	–
Google Cloud	Explicit Content Detection	Violence Detection	Violence
	Label Detection	Object Detection	All
	Text Detection	Text Detection	–
	Face Detection	Face Detection	–
Microsoft Azure	Tag Detection	Object Detection	All
	Face Detection	Face Detection	–

Source: Produced by the authors

Moreover, all providers offer a service to detect improper content (*e.g.*, violence, nudity, substances use) with varying levels of granularity. For instance, while some services simply

[1] <http://aws.amazon.com/machine-learning>
[2] <https://cloud.google.com/products/ai>
[3] <https://azure.microsoft.com/services/cognitive-services>

detect adult, racy, or violent content (*e.g.*, Google Cloud, Microsoft Azure), others provide second-level categories, like graphic nudity, swimwear/underwear, and weapons violence (*e.g.*, AWS). In this context, we consider in our study the violence detection services from AWS and Google Cloud. Since Azure only supports the detection of adult content and gory images (*i.e.*, graphic violence), to standardize our analysis, we do not include Azure's violence detection service in our study. Moreover, due to ethical concerns regarding nudity detection datasets, we will also not include in our study the adult content detection services made available by the three providers.

All providers also provide a text detection service. Although Azure provides such a service, due to a higher implementation complexity[4] than other services, we do not consider this service in our study. Moreover, only AWS provides a publicly accessible celebrity detection service. Finally, the three selected providers also offer a face detection service. Since the study of fairness in computer vision has been commonly linked to face detection and recognition tasks (BUOLAMWINI; GEBRU, 2018; BAROCAS; HARDT; NARAYANAN, 2019), we use the face detection services to answer **RQ3**. Moreover, we use all the remaining services in the scope of **RQ1** and **RQ2** analyses.

Unlike all other selected services, Google Cloud's explicit content detection service returns all supported labels associated with a likelihood value instead of returning only the detected labels with their associated confidence value. The following likelihood values can be returned from the service: `UNKNOWN`, `VERY_UNLIKELY`, `UNLIKELY`, `POSSIBLE`, `LIKELY`, `VERY_LIKELY`. In the next section, we explain how we applied the required treatment for this particularity.

3.2 FAULT MODEL AND EVALUATION

This study assumes that the fault injection experiments have only black-box access to the CV services. That means the only information accessible by the fault injector are the input instances and the predictions returned by the CV services for each instance.

To evaluate the effect of data faults on the accuracy of the studied services (**RQ1** and **RQ2**), we make use of the *top-k misclassification rate* (NARODYTSKA; KASIVISWANATHAN, 2016). It refers to the percentage of k-misclassified instances in a set of predictions. A k-misclassified instance means that a machine learning model ranked the true label of that instance below at least k other labels. In our experiments, we calculate the rate of misclassified instances in a fault injection experiment using $k = 1$, (*i.e.*, a simple misclassification of the highest ranked label). Although the misclassification of the highest ranked label is widely used in the context of adversarial examples (LI et al., 2019; HENDRYCKS; DIETTERICH, 2019; NARODYTSKA; KASIVISWANATHAN, 2016), it can be harsh since many applications relying

[4] At the date of writing, text detection in Microsoft Azure is split into two APIs, requiring synchronization to be manually implemented, thus no matching the second criteria.

on CV models commonly use the first few predicted labels. Thus, we also present additional analyses varying the value of *k*.

Due to the likelihood particularity described in the previous section and based on the definition of top-*k* misclassification rate, a change in the prediction of the Violence label from `LIKELY` to `VERY_LIKELY`, or `VERY_UNLIKELY` to `UNLIKELY` would yield a misclassification. To handle such a scenario, we group the "unlikely" labels in a `NEGATIVE` label and the "likely" labels in a `POSITIVE` label.

Regarding the effects of data faults on the fairness of CV services (**RQ3**), a well-known fairness criterion is analyzed: equal opportunity (HARDT; PRICE; SREBRO, 2016). Its definition states that the privileged and unprivileged groups of a sensitive feature (*e.g.*, race, sex, religion) should have equal true positive rates (*TPR*). To assess how similar the *TPRs* are for the privileged and unprivileged groups, we will analyze their difference

$$TPR(privileged) - TPR(unprivileged) \tag{3.1}$$

where

$$TPR = TP/(TP + FN) \tag{3.2}$$

and where (i) *TP* refers to the number of true positives—a case when the predicted and actual labels are both in the positive class; and (ii) *FN* refers to the number of false negatives—a case when the predicted and actual labels are in different classes.

3.3 DATA FAULTS

In this section, we discuss the data faults investigated in our study. A total of 14 image faults (CARLSON et al., 2018; CECCARELLI; SECCI, 2022; HENDRYCKS; DIETTERICH, 2019) that can result from camera failures, network issues, or application bugs are used to target the computer vision services. All these faults have been previously investigated and validated by other works in the computer vision field (CARLSON et al., 2018; NURMINEN et al., 2019; CECCARELLI; SECCI, 2022; HENDRYCKS; DIETTERICH, 2019; MICHAELIS et al., 2020). In the following paragraphs, we present the definition of the image faults addressed in this work. The faults were implemented using well-known data and image processing libraries, like *numpy*,[5] *skimage*,[6] *Pillow*,[7] and an image corruption library, *imagecorruptions*.[8] Figure 1 provides examples of the effects of such faults.

[5] <https://numpy.org>
[6] <https://scikit-image.org>
[7] <https://python-pillow.org>
[8] <https://github.com/bethgelab/imagecorruptions>

Figure 1 – Image faults applied to a sample image

(a) Original (b) GBLUR (c) MBLUR (d) ZBLUR

(e) BRI (f) CHR (g) COND (h) FOG

(i) FROST (j) SNOW (k) CONT (l) GRAY

(m) GNOISE (n) SNOISE (o) PIX

Source: Produced by the authors

Notes: Image faults applied to a randomly selected image (2a) from the COCO Object Detection Dataset (LIN et al., 2014). Images 2b-2o present the effects of the studied faults

3.3.1 Blurring

Blurred images occur when the image captured by the lens of a camera is out of focus (CARLSON et al., 2018; CHEONG et al., 2015). We blurring in an image by applying a Gaussian filter (*i.e.*, **Gaussian blur (GBLUR)** (CARLSON et al., 2018)). Moreover, two other types of blurring faults are considered in our study, **motion blur (MBLUR)** and **zoom blur (ZBLUR)**. While the former appears when a camera moves quickly, the latter occurs when a camera moves toward an object quickly (HENDRYCKS; DIETTERICH, 2019). To assess the impact of the blurring faults, we use the parameters from Michaelis *et al.* (MICHAELIS et al., 2020), mapped to 5 levels of severity *s*.

3.3.2 Brightness (BRI)

This fault represents the brightness alteration on an image by a certain factor *f*. If $f > 1$, the brightness of the image increases; similarly, the brightness decreases if $f < 1$. Changes in the brightness of an image can happen with the malfunction of the lens of a digital camera (CECCARELLI; SECCI, 2022) or because of daylight intensity (HENDRYCKS; DIETTERICH, 2019). We follow Ceccarelli and Secci (CECCARELLI; SECCI, 2022) on the choice of the f parameter and use the following values in our experiments: $\{0.3, 0.6, 1.5, 3, 4.5\}$.

3.3.3 Chromatic Aberration (CHR)

In this fault, an image presents colour distortions and a blurred appearance on the edges that separate subjects in the image (CARLSON et al., 2018; CECCARELLI; SECCI, 2022) due to the convergence of different light wavelengths that passes through the optical lens (KANG, 2007). We simulate such a fault by scaling the blue and green channels of an RGB image by a factor *f*. In our experiments, we use $f \in \{1, 2\}$ as in Ceccarelli and Secci (CECCARELLI; SECCI, 2022).

3.3.4 Climate Conditions

Adverse climate conditions are well-known challenges to computer vision techniques (MICHAELIS et al., 2020; NURMINEN et al., 2019; CECCARELLI; SECCI, 2022). Since applications that rely on CV services may be positioned outdoors (*e.g.*, violence detection based on external monitoring systems), poor climate conditions can play a major role in the reliability of these applications. Thus, we include in our study five common climate conditions, namely, **condensation (COND)**, **fog (FOG)**, **frost (FROST)** (*i.e.*, lenses or windows coated with ice crystals), and falling **rain/snow (SNOW)**. We simulate those conditions by using image masks (COND and FROST) (MICHAELIS et al., 2020; CECCARELLI; SECCI, 2022) or image processing techniques (FOG and SNOW) (MICHAELIS et al., 2020). Furthermore, for the FOG and SNOW faults we follow Michaelis *et al.* (MICHAELIS et al., 2020) and use 5 levels of severity *s*. As they rely on image masks, COND and FROST do not depend on any parameter.

3.3.5 Contrast (CONT)

The contrast of an image can easily vary between low or high depending on lighting conditions and the photographed object's color (HENDRYCKS; DIETTERICH, 2019). To emulate this fault, we change the contrast of an image by a severity factor $s \in \{0.4, .3, .2, .1, .05\}$, following Michaelis *et al.* (MICHAELIS et al., 2020).

3.3.6 Grayscale (GRAY)

In this fault, a chromatically wrong (*i.e.*, grayscaled) image is the result of malfunctions that may happen in the Bayer filter of a digital camera (CECCARELLI; SECCI, 2022). We apply a grayscale transformation to the input images to simulate this fault scenario. Like COND and FROST, this fault does not depend on any parameter.

3.3.7 Noise

The occurrence of noise in images is common and different factors may be responsible for its introduction (BONCELET, 2009; VERMA; ALI, 2013; HENDRYCKS; DIETTERICH, 2019). The most frequently occurring noise is additive **Gaussian noise (GNOISE)** (BONCELET, 2009). We simulate the occurrence of this fault by adding a random, Gaussian distributed noise value to every pixel of an image. Another type of image noise, the **salt and pepper noise (SNOISE)** is caused by sharp and sudden changes in the image signal (VERMA; ALI, 2013). This noise results in the appearance of sprinkled black and white dots on an image (BONCELET, 2009), and we simulate its occurrence by randomly changing a percentage p of the pixels of an image either to black or white. Similar to other faults, we use the parameters from Michaelis *et al.* (MICHAELIS et al., 2020), mapped to 5 levels of severity s, in our experiments.

3.3.8 Pixelation (PIX)

Pixelation could occur as an application defect by upsampling a low-resolution image (HENDRYCKS; DIETTERICH, 2019). Once again, we follow Michaelis *et al.* (MICHAELIS et al., 2020) and make use of 5 levels of severity s.

4 EXPERIMENTS

To answer our research questions, we initially conduct two experiments. The first addresses the effect of the common data faults on CV services (**RQ1** and **RQ2**), while the second addresses the effect of such faults on the fairness of those services (**RQ3**). More in detail, each experiment is a set of fault injection campaigns we conduct to assess the reliability of a target CV service when exposed to a particular fault. Each subject service is provided input data from the same domain they were designed to operate on.

4.1 DATASETS

Computer vision services trained to perform specific tasks would only be fairly assessed if they receive data that matches their domain. In our experiments, the selected services from Table 1 are grouped into five categories based on the task they perform, namely: *Celebrity Detection*, *Object Detection*, *Violence Detection*, *Text Detection*, and *Face Detection*.

For each category, we select a dataset according to the following criteria: (i) the datasets must be publicly available — this ensures that our work is easily accessible and reproducible; (ii) the datasets must have been previously used in CV research — this ensures the validity of the dataset; and (iii) all instances in the datasets must be labelled — this ensures that we can accurately compare the predicted label for an image with its ground truth. After applying the criteria, we selected the following datasets to match our service categories:

- **CelebA** (LIU et al., 2015) (Celebrity Detection): a large-scale dataset with more than 200,000 celebrity images both in-the-wild, and aligned and cropped, each with 40 attribute annotations.
- **COCO Object Detection Dataset** (LIN et al., 2014) (Object Detection): a large-scale object detection, segmentation, and captioning dataset, with 80 different object categories. The selected portion of the dataset is the 2017 validation set,[1] which is composed of 5,000 instances.
- **Human Rights UNderstanding Dataset (HRUN)** (KALLIATAKIS et al., 2017) (Violence Detection): a dataset with 400 images of four types of human rights violations: child labour, child soldiers, police violence, and refugees. Only the child soldiers' images displaying weapons and police violence images are selected for our experiments since violence detection services do not cover the other types of human rights violations.

[1] <http://images.cocodataset.org/zips/val2017.zip>

- **KAIST Scene Text Database** (LEE et al., 2010) (Text Detection): a scene text dataset comprised of 3,000 images captured outdoors and indoors. The dataset includes scenes in Korean, English, and mixed text. Only images containing English text are selected for our experiments due to the limited support for other languages from the text detection services.
- **UTKFace** (ZHANG; SONG; QI, 2017) (Face Detection): a large-scale face dataset with over 20,000 images and long age span (0 to 116). The images are annotated for attributes like age, gender, and ethnicity.

Given the overhead and cost of making API calls to the computer vision services, we do not employ the complete datasets in our experiments. Instead, we randomly sample 100 instances from each of the selected datasets.

4.2 FAULT INJECTION CAMPAIGNS

To perform our study, fault injection campaigns are conducted on the selected computer vision services. The workflow of these campaigns is as follows:

1. For every CV service, we select its matching dataset as specified on Table 1 and Section 4.1;
2. For every instance in the matching dataset, we perform the associated CV task using a service, saving its predictions. This process results in the *clean* predictions (*i.e.*, the predictions not influenced by any data fault) from the service;
3. Then, for each data fault, we inject it on all instances in the dataset using the set of parameters defined in Section 3.3. This process results in the *faulty* datasets (*i.e.*, the datasets under the effects of the data faults);
4. For every instance in the faulty dataset, we employ the CV service a second time to perform the same CV task, saving its predictions. This results in the *faulty* predictions (*i.e.*, the predictions under the effects of the data faults);
5. Finally, for each workflow configuration evaluated (*i.e.*, the combinations of CV service, data fault, and predefined parameters), we compute the proposed metrics (Section 3.2) using the clean and faulty prediction logs.

To perform this workflow, we implemented a tool, *mlaas-fi*[2]. The tool runs the fault injection workflow described in a configuration file, accepting any dataset of JPG/JPEG images as input and allowing the user to choose from a set of implemented faults and ML services

[2] <https://github.com/filipefalcaos/mlaas-fi>

from three major providers. Moreover, any user should be able to easily extend the tool to add support for new ML services or faults.

4.3 ASSESSING EFFECTS OF DATA FAULTS (RQ1 & RQ2)

To answer the **RQ1** and **RQ2** research questions, we execute the fault injection campaigns as described in the previous section, using the datasets from Section 4.1 and the nine CV services, presented in Table 1, from the categories Celebrity Detection, Object Detection, Violence Detection, and Text Detection. For each service, a total of 55 fault configurations are used. Since each dataset has 100 images, we execute 5,600 API calls per service (5,500 for faulty and 100 for clean images). Thus, to complete all fault injection campaigns necessary to answer the first two research questions, we performed a total of 50,400 API calls.

Table 2 presents the results that support **RQ1** and **RQ2**. While the first column describes the studied faults, the remaining columns describe the *top-1 misclassification rate* for a CV service when subjected to a fault. To calculate this value, we divide all the misclassified faulty predictions by the total faulty predictions while varying the fault parameters described in Section 3.3. For instance, if a dataset has a hundred images, 500 faulty predictions are performed for the Gaussian blur fault (GBLUR) since it has an intensity parameter with five possible values. Moreover, while the last line of the table represents the average misclassification rate for a service, the last column represents the average misclassification rate for a fault when injected onto multiple services. Finally, cells in green highlight the faults with the least effect over a service, while cells in red highlight the faults with the most effect.

We observe that all selected services are affected by the studied data faults, with a minimum average misclassification rate of 14% observed on Google Cloud's violence detection service and a maximum of 63% observed on the text detection service from the same provider. As our results highlight, 40 injected faults (36%) resulted in a misclassification rate above 50%. In fact, the GRAY and SNOISE faults in Google's violence detection experiment are the only faults with the highest effect on a service to result in a misclassification rate below 50%. Moreover, although the average misclassification rates of the faults are mainly below the 50% mark, these rates can considerably impact sensitive tasks (*e.g.*, violence detection). These results indicate that common faults can consistently affect the selected services and thus should be considered when relying on CV services.

4.3.1 General-purpose Services

The object detection services of multiple providers presented some of the highest misclassification rates observed in our experiments, with an average ranging from 39% on Microsoft Azure to 57% on AWS. Similarly, the two text detection services analyzed also present

Table 2 – The top-1 misclassification rate of the computer vision services when subjected to the data faults

Fault	AWS				Google Cloud			Azure	Average
	Celebrity	Object	Violence	Text	Object	Violence	Text	Object	
GBLUR	0.51	0.52	0.49	0.34	0.38	0.14	0.65	0.26	0.41
MBLUR	0.39	0.51	0.44	0.39	0.47	0.13	0.7	0.28	0.41
ZBLUR	0.31	0.71	0.67	0.81	0.73	0.15	0.85	0.61	0.6
BRI	0.12	0.42	0.23	0.21	0.32	0.12	0.55	0.22	0.27
CHR	0.02	0.55	0.56	0.34	0.53	0.14	0.56	0.5	0.4
COND	0.02	0.42	0.22	0.21	0.42	0.13	0.58	0.28	0.29
FOG	0.15	0.58	0.51	0.25	0.51	0.15	0.58	0.43	0.4
FROST	0.06	0.59	0.47	0.24	0.46	0.13	0.63	0.41	0.37
SNOW	0.25	0.7	0.68	0.43	0.63	0.15	0.75	0.52	0.51
CONT	0.22	0.69	0.65	0.31	0.63	0.14	0.64	0.49	0.47
GRAY	0.02	0.42	0.27	0.18	0.42	0.17	0.45	0.43	0.29
GNOISE	0.25	0.61	0.34	0.32	0.67	0.15	0.63	0.42	0.42
SNOISE	0.21	0.65	0.41	0.38	0.72	0.17	0.66	0.49	0.46
PIX	0.07	0.6	0.24	0.25	0.29	0.11	0.58	0.14	0.28
Average	0.19	0.57	0.44	0.33	0.51	0.14	0.63	0.39	

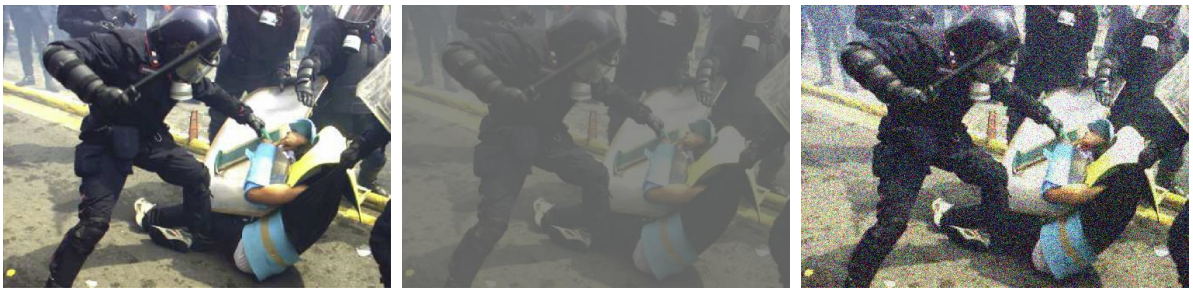
Source: Produced by the authors

high misclassification rates: 33% on AWS and 63% on Google Cloud. That is possibly related to the number of labels returned by the general-purpose services: an object detection service can perhaps return dozens of labels, and a fault that causes the top-1 predicted label to be moved down to third would already result in a misclassification (see Chapter 5).

4.3.2 Domain-specific Services

On the other hand, our experiments indicate a tendency towards a higher degree of reliability from domain-specific services when subjected to common data faults. The domain-specific services (*i.e.*, celebrity detection and violence detection) on all three cloud providers are more reliable than the general-purpose services. In fact, the two most reliable services in our experiments are domain-specific: Google Cloud's violence detection and AWS's celebrity detection. These two services present an average misclassification rate ranging from 14% to 19%, and only a single fault going over the 50% mark. Moreover, except for AWS's violence detection service, which presents the fourth highest misclassification rate, the remaining domain-specific services of a provider present lower misclassification rates than the general-purpose services of the same provider.

Figure 2 – The most impactful image faults



(a) Original (b) CONT (c) GNOISE

(d) SNOISE (e) ZBLUR

Source: Produced by the authors

Notes: The most impactful image faults applied to a randomly selected image of violence (3a) from the Human Rights UNderstanding Dataset (HRUN) (KALLIATAKIS et al., 2017). All faults are injected with the severity parameter s set to 3

4.3.3 Differences Across Cloud Providers

Moreover, within the same category of CV service, we can see significant differences from one provider to another. Although the object detection service has similar misclassification rates for AWS and Google Cloud, our results indicate that for most of the remaining services, there is a considerable difference between these two providers. Google Cloud's misclassification rate for the violence detection service is close to $^1/_3$ of AWS's. On the other hand, the misclassification rate of Google Cloud's text detection service almost doubles its AWS counterpart. Finally, Azure's object detection service presents significantly better misclassification rates than AWS and Google Cloud.

Summary for RQ1. *Results indicate that common data faults consistently affect the reliability of the CV services, with average misclassification rates ranging from 14% up to 63%. Domain-specific services, like celebrity detection, appear to be more reliable when exposed to those faults. Moreover, services with the same capabilities but different providers may differ significantly in their reliability, as evidenced by the text detection and violence detection services.*

4.3.4 Effects of Data Faults

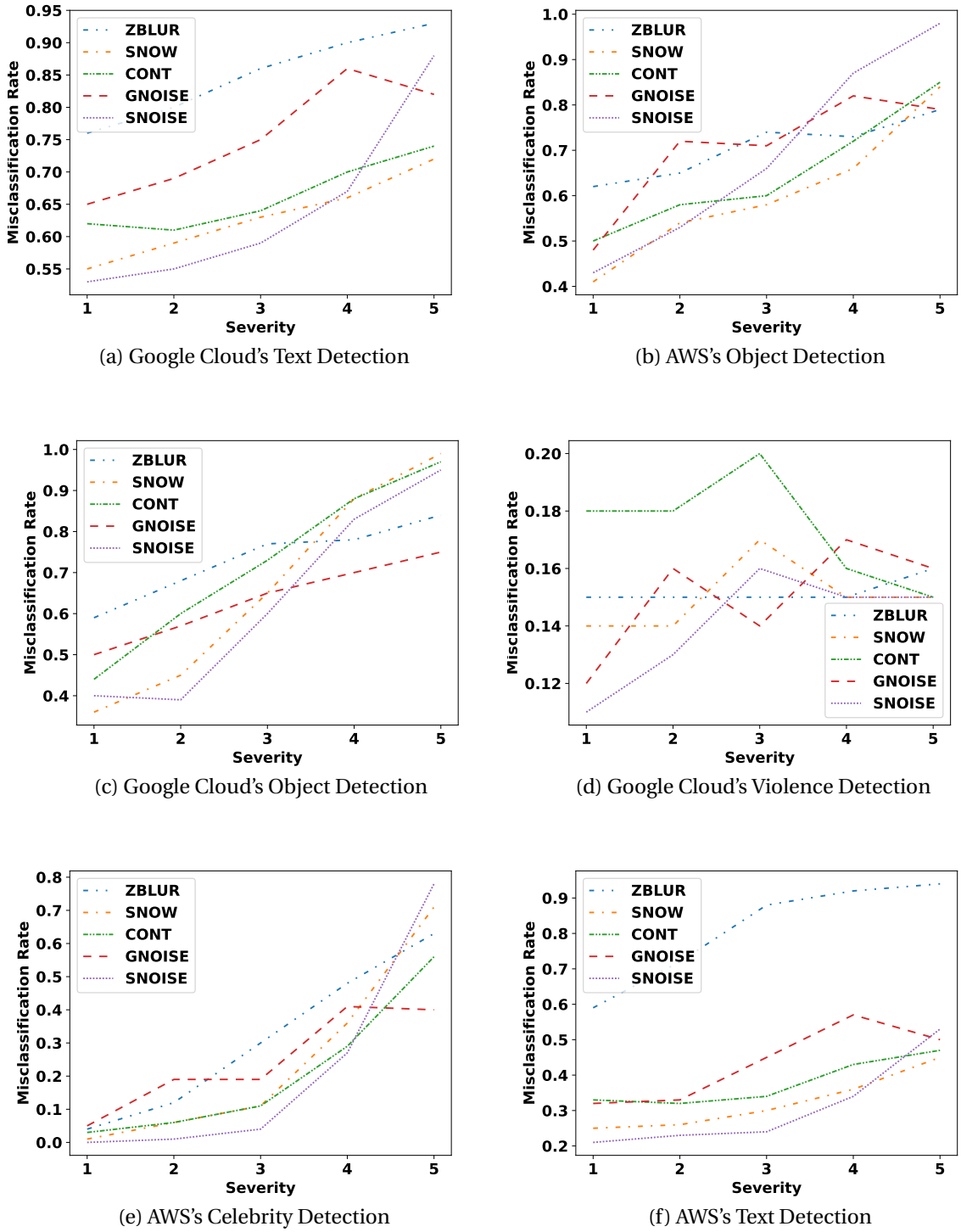
Regarding the injected data faults, we can notice that all faults are able to affect the computer vision services to some degree. The BRI, PIX, COND, and GRAY faults had, on average, the lowest effect across all experiments, being the only faults to show an average misclassification rate in the 20% range. The low impact of the BRI fault is in line with results reported in previous work (LI et al., 2019), where significant changes in the image brightness are required to achieve a high misclassification rate. However, in two of our experiments, we observed 42% and 55% misclassification rates for that fault. That indicates that although the effect of this fault tends to be smaller, it can also lead to considerable impacts. It is worth noting that these higher misclassification rates are found in experiments where all faults reached rates higher than 40%. On the other hand, the COND fault has been previously linked to high effects on model accuracy in autonomous driving applications (CECCARELLI; SECCI, 2022). That may indicate that this fault can impact other domain-specific applications more.

Moreover, the ZBLUR, SNOW, SNOISE, CONT, and GNOISE faults are the ones that resulted in the most significant effect in our experiments, as shown by their average misclassification rates of at least 42%. As depicted in Figure 2, these faults can significantly modify images. To better understand the effects of these faults, we present in Figure 3 the misclassification rates when varying the parameters of the most impactful faults identified above. To include contrast in our analysis, we analyze both the three services with the highest and lowest average misclassification rates.

We notice a clear upward trend in the misclassification rate on all faults as the severity parameter increases in the services analyzed. The notable exception is Google Cloud's violence detection service, where no clear trend can be observed across faults. Moreover, we can see that the ZBLUR, SNOW, SNOISE, CONT, and GNOISE faults resulted in misclassification rates of at least 40% in the initial severity level, where perturbations to the image are less perceptible. For instance, for SNOISE, $s = 1$ is mapped to a perturbation of only 3% of the image pixels. We can also notice that, as the severity increases, misclassification rates quickly reach elevated levels. That is evidenced by the linear and exponential behaviour dominant on the plots in Figure 3.

Furthermore, we can see that the most impactful fault varies according to the service analyzed. While ZBLUR affects Google Cloud's text detection service the most, the object detection services are mostly affected by the SNOISE and SNOW faults. However, the SNOISE fault has constantly resulted in high misclassification rates obtained being the highest in the three AWS services, especially when $s = 5$. Finally, at high perturbation levels, even the more reliable services (*i.e.*, the services with the lowest average misclassification rates) were consistently affected by the studied faults.

Figure 3 – The top-1 misclassification rates of ZBLUR, SNOW, SNOISE, CONT, and GNOISE faults



(a) Google Cloud's Text Detection

(b) AWS's Object Detection

(c) Google Cloud's Object Detection

(d) Google Cloud's Violence Detection

(e) AWS's Celebrity Detection

(f) AWS's Text Detection

Source: Produced by the authors

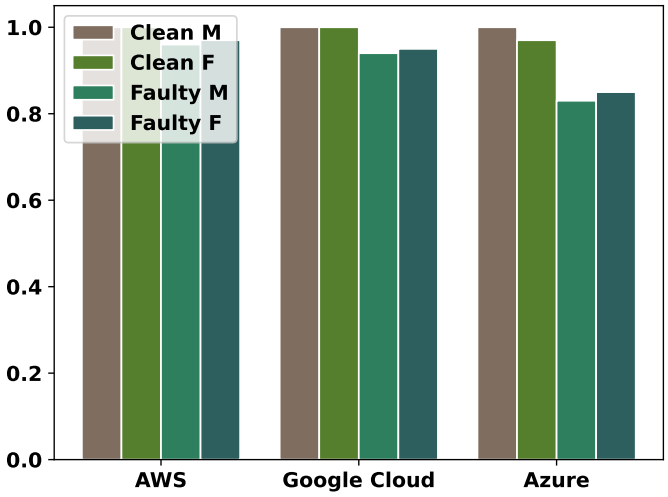
Notes: The top-1 misclassification rates when varying their severity *s* parameter. The figures refer to the services with the top-3 highest and lowest average misclassification rates

Summary for RQ2. *The BRI, PIX, COND, and GRAY faults resulted in the least impact in our experiments; ZBLUR, SNOW, SNOISE, CONT, and GNOISE resulted in the most impact. Some services can experience high misclassification rates even at initial severity levels, where perturbations are less perceptible. Moreover, misclassification rates can quickly reach elevated levels as severity increases. Finally, the most impactful faults vary across services.*

4.4 ASSESSING EFFECTS ON FAIRNESS (RQ3)

To answer **RQ3**, we first define the sensitive attribute of our fairness assessment. In this analysis, we use the sampled UTKFace dataset and the annotated *sex* as our sensitive attribute. We choose this attribute due to its availability on the dataset and the history of CV models that present gender bias (BUOLAMWINI; GEBRU, 2018; BAROCAS; HARDT; NARAYANAN, 2019). Also, based on such history, in our analysis, we refer to the male group as the privileged one and the female group as the unprivileged one. We perform API calls to the face detection services using the clean images from the sampled UTKFace dataset and calculate the true positive rates (TPRs) of the privileged and unprivileged groups, which compose the equal opportunity criteria defined in Section 3.2. Finally, we execute three fault injection campaigns, targeting the face detection services and calculating the TPRs for each fault configuration.

Figure 4 – The true positive rate (TPR) of the privileged (male - M) and unprivileged (female - F) groups for both clean and faulty images



Source: Produced by the authors

Figure 4 presents the results of our fairness analysis. For each provider, the first two bars represent the TPR of the privileged (male - M) and unprivileged (female - F) groups for the clean images. The two remaining bars, represent the average TPR when considering all

Table 3 – The top-5 faults with the highest difference in the true positive rate (TPR) for the attribute sex

AWS	**Google Cloud**	**Microsoft Azure**
-0.22 (CONT, $s = 5$)	-0.18 (SNOW, $s = 4$)	-0.34 (GNOISE, s = 5)
-0.18 (SNOISE, $s = 5$)	-0.1 (CONT, $s = 4$)	-0.25 (FOG, s = 4)
-0.14 (GNOISE, $s = 5$)	-0.1 (SNOW, $s = 3$)	-0.23 (CONT, s = 3)
0.08 (FOG, $s = 5$)	0.08 (FOG, $s = 4$)	-0.21 (SNOW, s = 3)
-0.06 (SNOW, $s = 4$)	0.06 (SNOW, $s = 1$)	-0.19 (SNOISE, s = 5)

Source: Produced by the authors

fault configurations. We observe that for the clean images, the face detection service from both AWS and Google Cloud returned equal TPRs of 100%, indicating fairness with respect to the equal opportunity criteria. On the other hand, Azure's face detection service present a small difference of 3% in the TPRs, possibly indicating the presence of a small bias towards the privileged group. We also note that, when exposed to faults, the three face detection services presented a reduction on the TPRs of both groups. Moreover, the injection of faults resulted in a small, 1-2% difference of TPRs for the face detection services. On all three cases, the female group presented the better TPR, which means for Azure, the direction of the difference changed groups due to the injected faults. Nevertheless, the TPR differences, both for the clean and faulty images, are small and in the 1-3% range.

Although the fairness of the services does not seen to be affected by the studied faults when analyzing the average TPRs, we notice in our experiments that some fault configurations are able to cause an isolated impact on fairness. Table 3 presents the top-5 fault configurations with the highest difference in the TPRs of privileged and unprivileged groups for each face detection service. Each column of the table describes the TPR difference for a fault configuration (*e.g.*, SNOISE with $s = 5$) in a service.

We observe that a few particular fault configurations are able to cause an impact on fairness, as measured by the TPR difference. The CONT, SNOW, and FOG faults appeared on the top-5 for the three face detection services. When injected with the parameter s between 3 and 5, the CONT fault resulted on a TPR difference of -10% (Google Cloud) up to -34% (Azure). Similarly, the SNOW fault resulted on a difference of 6% (Google Cloud), when $s = 1$, up to -21% (Azure), when $s = 3$. The FOG fault resulted on the same TPR difference of 8% for AWS and Google Cloud but at different values of the s parameter: 4 and 5, respectively. However, the FOG fault resulted in a far higher TPR difference of -0.25% when $s = 4$ on Azure.

We also observe that the GNOISE and SNOISE faults appeared on the top-5 highest TPR differences for AWS and Azure. For AWS, we observe a difference of -14% and -18% for GNOISE and SNOISE when $s = 5$, respectively. For Azure, we see a similar TPR difference of -19%, but a far higher difference for GNOISE: -34% when $s = 5$. In fact, we can notice that

the TPR differences for Azure's face detection service are significantly higher than the ones observed for AWS and Google Cloud. Moreover, although the results in Table 3 show that fairness can be affected in certain scenarios, it is worth noticing that these scenarios mostly occur at higher perturbation levels, when $s > 3$. In fact, that condition accounts for 73% of the top-5 highest TPR differences.

Summary for RQ3. *The fairness, in terms of the equal opportunity criteria for the sex attribute, of the face detection services is not systematically affected by data faults. However, under certain fault configurations, significant impacts on fairness may be observed, specially at higher fault parameter levels. Moreover, different face detection services may experience higher or lower levels of fairness impact.*

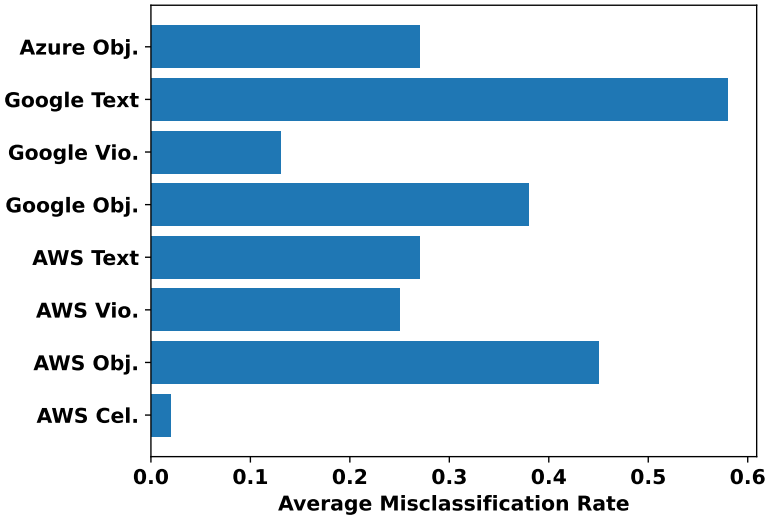
5 DISCUSSION

In this chapter, we further discuss our experiments' results. More in detail, we will discuss (i) the effect of the studied data faults at their initial parameter levels; and (ii) the role of the k parameter in our analyses.

5.1 EFFECT OF DATA FAULTS AT INITIAL PARAMETERS

Figure 3 indicates that even at initial parameter levels, some faults can result in high misclassification rates. Aiming to better understand the impact of the studied faults at initial parameter levels, we perform a complementary analysis on the misclassification rates of the services studied for **RQ1** and **RQ2**. Figure 5 presents the average misclassification rates of the CV services when data faults are set at their initial parameter levels (*i.e.*, $f = 1$ for BRI and CHR, and $s = 1$ for the remaining faults).

Figure 5 – The average misclassification rates of the computer vision services when data faults are set at their initial parameter levels



Source: Produced by the authors

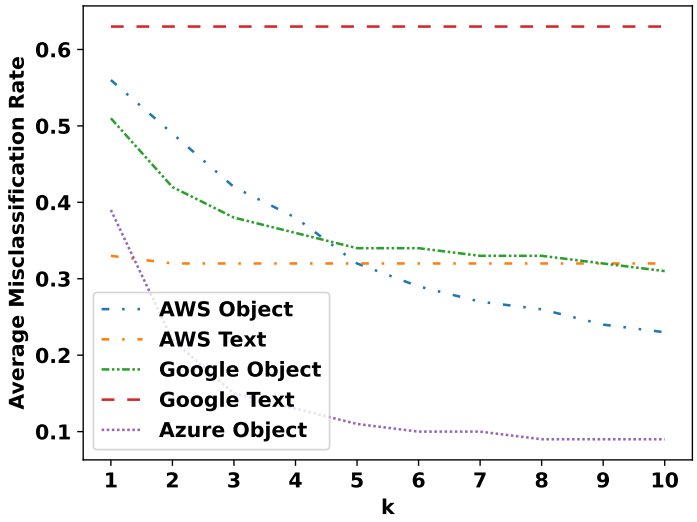
We observe that three services, AWS's object detection and Google's object and text detection services, reached misclassification rates above 30% at initial parameter levels. Google's text detection service, the most extreme case, reached a 58% misclassification rate. On the other hand, the domain-specific services are significantly less affected at initial parameter levels. For instance, the three domain-specific services presented the three lowest misclassification rates at initial parameter levels. It is worth noticing though that all services are affected to some degree at initial parameter levels. In fact, 7 out of 8 services reached

misclassification rates above 10%. These results indicate that, even when perturbations are small, data faults can affect the reliability of the studied CV services.

5.2 THE ROLE OF *K*

Many works in the adversarial examples field have relied on the top-1 misclassification to present their results (LI et al., 2019; HENDRYCKS; DIETTERICH, 2019; NARODYTSKA; KASIVISWANATHAN, 2016). During our experiments, we encountered many cases where an injected fault caused a reduction in the confidence level of the top prediction, moving it one place down and thus resulting in misclassification. Although this shows that the fault indeed caused an impact, this impact would be less severe if the top few labels were used by the application relying on the affected service. That is the case of services that naturally return many predicted labels, like the object and text detection services.

Figure 6 – The average misclassification rates of the object detection and text detection services



Source: Produced by the authors

Notes: Misclassification rates obtained when k varies in the $[1, 10]$ interval

Aiming to analyze the role of k when assessing the reliability of a CV service, we compute the average misclassification rate for a service, as presented in the last row of Table 2, for the selected object detection and text detection services. In this analysis, we follow the same methodology used to answer **RQ1**, but now we vary k in the $[1, 10]$ interval. Figure 6 presents the results of that analysis. We observe that, for the object detection services, the average misclassification rate decreases as the value of k increases. We also notice that, even at $k = 10$, the object detection services from AWS and Google Cloud were still consistently affected by the faults, as evidenced by misclassification rates higher than 20%. Moreover,

Azure's object detection service presented the largest reduction in misclassification rate, reaching 9% at $k = 10$.

On the other hand, the misclassification rates for the text detection services were not affected by the value of k. By analyzing the predictions from these two services, we notice that this occurs because the injected faults caused text to be incorrectly detected (*i.e.*, a character was recognized wrongly) or no longer detected at all, instead of pushing predictions down. Unlike object detection services, the predictions of text detection services are not predefined. Therefore, increasing the value of k does not affect these scenarios. These results on the role of the k parameter suggest that for services that output multiple predefined labels per prediction (*e.g.*, object detection), relying on more than only the top predicted label might increase reliability.

6 IMPLICATIONS

We present the implications of our study, both in terms of the developers (*i.e.*, the clients of the cloud services) and the cloud providers, as follows.

6.1 TO DEVELOPERS

The obtained results indicate that the studied faults consistently impact the reliability of computer vision cloud services from three major providers. Therefore, developers considering using these services to power computer vision applications should take into account the reliability of such services when exposed to faults. However, it may be difficult for regular developers to assess how likely it is for some of the studied faults to occur (*e.g.*, Gaussian blur (GBLUR), Chromatic Aberration (CHR)) in the environment their CV applications will be deployed on. On the other hand, some of the studied faults can be easily mapped to different deployment environments and, thus, should be considered by developers when choosing which CV service to use. For instance, applications receiving image or video input from outdoor cameras will likely be exposed to the climate condition faults studied and variations in Brightness (BRI). Similarly, applications tracking moving targets may be exposed to Motion Blur (MBLUR) or Zoom Blur (ZBLUR).

Moreover, our results indicate that significant differences in reliability exist across services performing the same computer vision task (*e.g.*, object detection, violence detection) but from different providers. Such a finding indicates that the reliability of services can be important in deciding which provider would power a CV application. Furthermore, the obtained levels of reliability in our experiments indicate that general-purpose services (*i.e.*, object detection, text detection) are less reliable than domain-specific services (*i.e.*, celebrity detection, violence detection). Such a result suggests that developers planning on building CV applications that rely on general-purpose services should be more cautious about addressing the effects of data faults on the reliability of their applications.

Additionally, some data faults (*i.e.*, ZBLUR, SNOW, SNOISE, CONT, and GNOISE) presented a higher effect on the reliability of the studied CV services, consistently affecting even the more reliable services overall. Conversely, other faults (*i.e.*, BRI, PIX, COND, and GRAY) resulted in little effect on the reliability of such services. Therefore, developers may decide to give more priority to reliability risks if they know high-effect faults are more likely to occur on their applications. Finally, the fairness of the computer vision services does not seem to be affected by data faults in our experiments. However, this result should not be generalized, as it was obtained by evaluating a single sensitive attribute, sex, in face detection services from three providers.

6.2 TO CLOUD PROVIDERS

Since the obtained results indicate that the studied faults consistently impact the reliability of the CV services, the providers of such services may use our results to assess and possibly improve the reliability of their services. Remarkably, the differences in reliability across providers may indicate that some providers could be less impacted by data faults than how they currently are. Additionally, as our results suggest that the services can be significantly impacted even at low perturbation levels, providers may want to improve their services to be more reliable when facing small perturbations. Furthermore, our results on the faults that pose the most and the most negligible impact on the studied services may provide the development teams from the cloud providers with a direction on how to improve the overall reliability of their services.

Finally, the indication that fairness does not seem to be affected by data faults in our experiments should not be generalized, as the underlying analyses considered only a single sensitive attribute. In this context, both the developers and the cloud providers would undoubtedly benefit from more fine-grained analyses performed by the providers of the effect of data faults on fairness, potentially improving the confidence in the fairness of the CV services when exposed to data faults.

7 THREATS TO VALIDITY

We discuss the threats to the study validity (WOHLIN et al., 2012), with the respective actions to mitigate them, as follows.

7.1 CONSTRUCT AND INTERNAL VALIDITY

The data faults employed in our study and their parameters may not fully represent the faults that systems that rely on computer vision services can experience. To mitigate this threat, we selected faults that can occur in real-life computer vision applications, like autonomous driving (CECCARELLI; SECCI, 2022; CARLSON et al., 2018), and have been used in related studies (HENDRYCKS; DIETTERICH, 2019; MICHAELIS et al., 2020; CECCARELLI; SECCI, 2022; CARLSON et al., 2018).

Regarding the fairness analyses, we employed only a single sensitive feature, sex. Thus, the results of our fairness analyses could differ for other sensitive features, like race. Moreover, only a single fairness criterion, equal opportunity, was employed in our analyses. Therefore, as shown in the work of Verma and Rubin (VERMA; RUBIN, 2018), other fairness criteria could have deemed a cloud service as unfair rather than fair.

Furthermore, we did not employ a large number of images per dataset in our experiments due to the overhead and cost related to using CV services. However, we sampled a limited number of instances from carefully selected datasets previously used in computer vision research (see Section 4.1).

7.2 CONCLUSION AND EXTERNAL VALIDITY

We carefully performed our descriptive and statistical analyses. All authors validated every analysis in this paper, and both the code and the data from our study are publicly available at <https://github.com/filipefalcaos/mlaas-fi>. Regarding the generality of our findings, we only selected CV services for our fault injection experiments. Although we have analyzed a wide range of computer vision services, our results might not hold for different computer vision tasks (*e.g.*, describing images with human-readable language (WANG; ZHANG; YU, 2020)). They also might not hold for different input types for CV tasks (*i.e.*, videos instead of images) or other ML domains (*e.g.*, natural language processing, time series analysis).

8 RELATED WORK

Early works mainly studied machine learning applications' security and robustness in a white-box setting (PAPERNOT et al., 2016; GOODFELLOW; SHLENS; SZEGEDY, 2014; YUAN et al., 2019; ZHANG; LI, 2019; MOOSAVI-DEZFOOLI; FAWZI; FROSSARD, 2016). However, the number of studies working on a black-box setting, which includes ML services, has been increasing, given the difficulty of performing a white-box attack in the real world.

8.1 BLACK-BOX ADVERSARIAL ATTACKS

In Narodytska and Kasiviswanathan (NARODYTSKA; KASIVISWANATHAN, 2016), the authors propose a black-box elementary adversarial attack based on the idea of greedy local search. Their results show that even simple attacks in a black-box setting can fool neural networks. In Papernot *et al.* (PAPERNOT et al., 2017), the authors introduced a strategy of training an adversary to replace a target Deep Neural Network (DNN), using inputs crafted by the adversary and labelled by the target DNN. They tested their approach against baseline models and models trained by the authors on AWS and Google Cloud, yielding high misclassification rates. The work of Ilyas *et al.* (ILYAS et al., 2018) proposes new approaches to overcome the challenge of limited queries and information on black-box attacks. The authors show the effectiveness of their attacks by targeting Google Cloud's Vision API. The works of Michaelis *et al.* (MICHAELIS et al., 2020), and Hendrycks and Dietterich (HENDRYCKS; DIETTERICH, 2019) show that common image perturbations, like the ones investigated in our work, can affect the performance of computer vision models.

8.2 ATTACKING MACHINE LEARNING SERVICES

Other studies focus specifically on attacking machine learning services with adversarial examples (LI et al., 2019; GOODMAN, 2020; TRAMÈR et al., 2016; WANG; GONG, 2018). Tramer *et al.* (TRAMÈR et al., 2016) presented a model extraction attack that attempts to steal the parameters of ML models deployed as services. Similarly, Wang and Gong (WANG; GONG, 2018) proposed an approach to stealing hyperparameters of models accessible as services. They empirically demonstrated the ability of their attack to steal the target hyperparameters with a small margin of error. Li *et al.* (LI et al., 2019) assessed the robustness of three types of cloud-based detectors from five different providers with four black-box attack methods. The proposed attacks leverage semantic segmentation, and the results of their experiments show high attack success rates. Goodman (GOODMAN, 2020) proposed a novel attack method to achieve a high bypass rate with a limited number of queries. The author also presents the

results of an initial empirical study of the effectiveness of the proposed approach, which achieved over 90% of success rate.

8.3 FAULTS IN MACHINE LEARNING SYSTEMS

Some previous studies focus on the role of data faults in machine learning systems (CECCARELLI; SECCI, 2022; JHA et al., 2018; JHA et al., 2019; NURMINEN et al., 2019; MA et al., 2018). Ceccarelli and Secci (CECCARELLI; SECCI, 2022) assessed the effect of RGB camera failures on the behaviour of AI/ML applications for autonomous driving. Their results show that camera failures have a relevant impact on object detectors, and even faults that cause small perturbations may alter the system's decisions. Similarly, other works have proposed fault injection frameworks for ML systems. Jha *et al.* (JHA et al., 2018; JHA et al., 2019) presents fault injection frameworks to assess the reliability of autonomous vehicles. Among the faults considered in their studies are noise models and weather conditions used in our work. In Nurminen *et al.* (NURMINEN et al., 2019), the authors proposed a framework for injecting data faults to test machine learning systems. *DeepMutation* (MA et al., 2018) also applies a similar method by generating artificial faults, both in the data and in the model of a deep learning system, then changing the original system (*i.e.*, applying a mutation).

8.4 FAIRNESS

Issues regarding the fairness of machine learning systems have been gaining more importance over the last few years (BAROCAS; HARDT; NARAYANAN, 2019). The work of Buolamwini and Gebru (BUOLAMWINI; GEBRU, 2018) shows that commercial gender classification tools presented a significant disparity in error rates of darker-skinned females and lighter-skinned males. De Vries *et al.* (VRIES et al., 2019) found a considerable disparity between countries in the accuracy of object detection models. Images from lower-income countries experienced higher error rates because objects may look very different in different countries. Concerning adversarial examples and fairness, Delobelle *et al.* (DELOBELLE et al., 2021) and Zhang and Sang (ZHANG; SANG, 2020) have proposed the use of adversarial examples to improve the fairness of machine learning models. Xu *et al.* (XU et al., 2021) show in their work that adversarial training, a common defence against adversarial examples, can introduce severe disparity between different data groups. The authors then proposed a new framework to perform adversarial training while avoiding unfairness.

8.5 OUR CONTRIBUTIONS

Our study differs from prior work by considering the effects of common data faults in CV services that may arise from applications that depend on these services instead of proposing target attacks. Also, different from most studies that cover CV services, we

perform our study on a more extensive set of services, covering both general-purpose services (*i.e.*, object detection and text detection) as well as domain-specific services (*i.e.*, celebrity detection, violence detection). We also compare our results across services and cloud providers. Moreover, we introduce a fairness analysis in face recognition services concerning data faults. To the best of our knowledge, this is the first work to assess the effect of data faults on the fairness of CV services.

9 CONCLUSION

This work investigated the reliability of commercial computer vision services by injecting common data faults into their input data. We extensively analyzed 11 computer vision services provided by three major cloud providers and also implemented 14 image faults that could arise from camera failures, network issues, or application bugs. First, we analyzed to what extent CV services can tolerate common data faults. Then, we evaluated the effect of specific faults, or groups of faults, in CV services. Finally, we introduced an extensive analysis of the impact of data faults on the fairness of computer vision services.

Our results indicate that not only do the injected faults consistently impact the reliability of CV services, but services with similar capabilities from different providers can present different degrees of reliability. Results also indicate that general-purpose services, like object detection and text detection, appear less reliable than domain-specific services, as highlighted by their lower levels of reliability obtained in our experiments. Moreover, we found that some faults tend to have a higher impact on the reliability of CV services. On the other hand, results indicate that the fairness of those services is not consistently impacted.

In future work, we plan to work on mechanisms to mitigate the effects of data faults on ML services. Developers relying on such services would likely benefit from having easily accessible tools to reduce the reliability risks presented in this paper. Hopefully, our work will help both application developers relying on CV services to make more informed decisions and cloud providers designing more robust services.

BIBLIOGRAPHY

ALIKHADEMI, K. et al. A review of predictive policing from the perspective of fairness. **Artificial Intelligence and Law**, Springer, p. 1–17, 2021.

BAROCAS, S.; HARDT, M.; NARAYANAN, A. **Fairness and Machine Learning**. [S.l.]: fairmlbook.org, 2019. <http://www.fairmlbook.org>.

BIROLINI, A. Reliability engineering. **IEEE Software**, Springer, v. 34, 2017.

BONCELET, C. Chapter 7 - image noise models. In: BOVIK, A. (Ed.). **The Essential Guide to Image Processing**. Boston: Academic Press, 2009. p. 143–167. ISBN 978-0-12-374457-9. Disponível em: <https://www.sciencedirect.com/science/article/pii/B978012374457900007X>.

BUOLAMWINI, J.; GEBRU, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In: PMLR. **Conference on fairness, accountability and transparency**. [S.l.], 2018. p. 77–91.

CARLSON, A. et al. Modeling camera effects to improve visual learning from synthetic data. In: **Proceedings of The European Conference on Computer Vision (ECCV) Workshops**. [S.l.: s.n.], 2018. p. 0–0.

CECCARELLI, A.; SECCI, F. Rgb cameras failures and their effects in autonomous driving applications. **IEEE Transactions on Dependable and Secure Computing**, IEEE, 2022.

CHEONG, H. et al. Fast image restoration for spatially varying defocus blur of imaging sensor. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 15, n. 1, p. 880–898, 2015.

DELOBELLE, P. et al. Ethical adversaries: Towards mitigating unfairness with adversarial machine learning. **ACM SIGKDD Explorations Newsletter**, ACM New York, NY, USA, v. 23, n. 1, p. 32–41, 2021.

DUA, D.; GRAFF, C. **UCI Machine Learning Repository**. 2017. Disponível em: <http://archive.ics.uci.edu/ml>.

GOODFELLOW, I. et al. **Deep learning**. [S.l.]: MIT press Cambridge, 2016. v. 1.

GOODFELLOW, I. J.; SHLENS, J.; SZEGEDY, C. Explaining and harnessing adversarial examples. **arXiv preprint arXiv:1412.6572**, 2014.

GOODMAN, D. Transferability of adversarial examples to attack cloud-based image classifier service. **arXiv preprint arXiv:2001.03460**, 2020.

GUO, T. et al. Simple convolutional neural network on image classification. In: IEEE. **2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)**. [S.l.], 2017. p. 721–724.

HARDT, M.; PRICE, E.; SREBRO, N. Equality of opportunity in supervised learning. **Advances in neural information processing systems**, v. 29, 2016.

HENDRYCKS, D.; DIETTERICH, T. Benchmarking neural network robustness to common corruptions and perturbations. **arXiv preprint arXiv:1903.12261**, 2019.

HESAMIFARD, E. et al. Privacy-preserving machine learning as a service. **Proceedings on Privacy Enhancing Technologies**, Sciendo, v. 2018, n. 3, p. 123–142, 2018.

HUNT, T. et al. Chiron: Privacy-preserving machine learning as a service. **arXiv preprint arXiv:1803.05961**, 2018.

ILYAS, A. et al. Black-box adversarial attacks with limited queries and information. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2018. p. 2137–2146.

JAHANGIROVA, G. et al. Taxonomy of real faults in deep learning systems. **arXiv preprint arXiv:1910.11015**, 2019.

JHA, S. et al. Avfi: Fault injection for autonomous vehicles. In: IEEE. **2018 48th annual ieee/ifip international conference on dependable systems and networks workshops (dsn-w)**. [S.l.], 2018. p. 55–56.

JHA, S. et al. Kayotee: A fault injection-based system to assess the safety and reliability of autonomous vehicles to faults and errors. **arXiv preprint arXiv:1907.01024**, 2019.

KALLIATAKIS, G. et al. Detection of human rights violations in images: Can convolutional neural networks help? **arXiv preprint arXiv:1703.04103**, 2017.

KANG, S. B. Automatic removal of chromatic aberration from a single image. In: **2007 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2007. p. 1–8.

KUMAR, P.; THAKUR, R. S. Recommendation system techniques and related issues: a survey. **International Journal of Information Technology**, Springer, v. 10, n. 4, p. 495–501, 2018.

KUMAR, R. S. S. et al. Adversarial machine learning-industry perspectives. In: IEEE. **2020 IEEE Security and Privacy Workshops (SPW)**. [S.l.], 2020. p. 69–75.

LEE, S. et al. Scene text extraction with edge constraint and text collinearity. In: IEEE. **2010 20th international conference on pattern recognition**. [S.l.], 2010. p. 3983–3986.

LI, L. E. et al. Scaling machine learning as a service. In: **International Conference on Predictive Applications and APIs**. [S.l.: s.n.], 2017. p. 14–29.

LI, X. et al. Adversarial examples versus cloud-based detectors: A black-box empirical study. **IEEE Transactions on Dependable and Secure Computing**, IEEE, 2019.

LIN, T.-Y. et al. Microsoft coco: Common objects in context. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 740–755.

LITJENS, G. et al. A survey on deep learning in medical image analysis. **Medical Image Analysis**, v. 42, p. 60 – 88, 2017. ISSN 1361-8415. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1361841517301135>.

LIU, Z. et al. Deep learning face attributes in the wild. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 3730–3738.

MA, L. et al. Deepmutation: Mutation testing of deep learning systems. In: IEEE. **2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)**. [S.l.], 2018. p. 100–111.

MICHAELIS, C. et al. **Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming**. 2020.

MOOSAVI-DEZFOOLI, S.-M.; FAWZI, A.; FROSSARD, P. Deepfool: a simple and accurate method to fool deep neural networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 2574–2582.

NARODYTSKA, N.; KASIVISWANATHAN, S. P. Simple black-box adversarial perturbations for deep networks. **arXiv preprint arXiv:1612.06299**, 2016.

NURMINEN, J. K. et al. Software framework for data fault injection to test machine learning systems. In: IEEE. **2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)**. [S.l.], 2019. p. 294–299.

PAPERNOT, N. et al. Practical black-box attacks against machine learning. In: **Proceedings of the 2017 ACM on Asia conference on computer and communications security**. [S.l.: s.n.], 2017. p. 506–519.

PAPERNOT, N. et al. The limitations of deep learning in adversarial settings. In: IEEE. **2016 IEEE European symposium on security and privacy (EuroS&P)**. [S.l.], 2016. p. 372–387.

RAMZAN, M. et al. A review on state-of-the-art violence detection techniques. **IEEE Access**, IEEE, v. 7, p. 107560–107575, 2019.

RIBEIRO, M.; GROLINGER, K.; CAPRETZ, M. A. Mlaas: Machine learning as a service. In: IEEE. **2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)**. [S.l.], 2015. p. 896–902.

SHOKRI, R. et al. Membership inference attacks against machine learning models. In: IEEE. **2017 IEEE Symposium on Security and Privacy (SP)**. [S.l.], 2017. p. 3–18.

TEALAB, A. Time series forecasting using artificial neural networks methodologies: A systematic review. **Future Computing and Informatics Journal**, Elsevier, v. 3, n. 2, p. 334–340, 2018.

TRAMÈR, F. et al. Stealing machine learning models via prediction {APIs}. In: **25th USENIX security symposium (USENIX Security 16)**. [S.l.: s.n.], 2016. p. 601–618.

TRUEX, S. et al. Demystifying membership inference attacks in machine learning as a service. **IEEE Transactions on Services Computing**, IEEE, 2019.

VERMA, R.; ALI, J. A comparative study of various types of image noise and efficient noise removal techniques. **International Journal of advanced research in computer science and software engineering**, v. 3, n. 10, 2013.

VERMA, S.; RUBIN, J. Fairness definitions explained. In: IEEE. **2018 ieee/acm international workshop on software fairness (fairware)**. [S.l.], 2018. p. 1–7.

VRIES, T. D. et al. Does object recognition work for everyone? In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2019. p. 52–59.

WANG, B.; GONG, N. Z. Stealing hyperparameters in machine learning. In: IEEE. **2018 IEEE symposium on security and privacy (SP)**. [S.l.], 2018. p. 36–52.

WANG, H.; ZHANG, Y.; YU, X. An overview of image caption generation methods. **Computational intelligence and neuroscience**, Hindawi, v. 2020, 2020.

WANG, W.; SHEN, J.; SHAO, L. Video salient object detection via fully convolutional networks. **IEEE Transactions on Image Processing**, IEEE, v. 27, n. 1, p. 38–49, 2017.

WANG, Y. et al. A comparative assessment of credit risk model based on machine learning——a case study of bank loan data. **Procedia Computer Science**, Elsevier, v. 174, p. 141–149, 2020.

WOHLIN, C. et al. **Experimentation in software engineering**. [S.l.]: Springer Science & Business Media, 2012.

XU, H. et al. To be robust or to be fair: Towards fairness in adversarial training. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2021. p. 11492–11501.

YUAN, X. et al. Adversarial examples: Attacks and defenses for deep learning. **IEEE transactions on neural networks and learning systems**, IEEE, v. 30, n. 9, p. 2805–2824, 2019.

ZHANG, J.; LI, C. Adversarial examples: Opportunities and challenges. **IEEE transactions on neural networks and learning systems**, IEEE, v. 31, n. 7, p. 2578–2593, 2019.

ZHANG, Y.; SANG, J. Towards accuracy-fairness paradox: Adversarial example-based data augmentation for visual debiasing. In: **Proceedings of the 28th ACM International Conference on Multimedia**. [S.l.: s.n.], 2020. p. 4346–4354.

ZHANG, Z.; SONG, Y.; QI, H. Age progression/regression by conditional adversarial autoencoder. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 5810–5818.

ZIADE, H. et al. A survey on fault injection techniques. **Int. Arab J. Inf. Technol.**, v. 1, n. 2, p. 171–186, 2004.