

SAEED BAKHSI GERMI

# Deep Neural Classifiers in Safety-Critical Applications

Safety concerns and mitigation methods



SAEED BAKHSHI GERMI

Deep Neural Classifiers in  
Safety-Critical Applications  
Safety concerns and mitigation methods

ACADEMIC DISSERTATION

To be presented, with the permission of  
the Faculty of Information Technology and Communication Sciences  
of Tampere University,  
for public discussion in the auditorium TB104  
of the Hervanta Campus, Korkeakoulunkatu 1, Tampere,  
on May 17th, at 12 o'clock.

# ACADEMIC DISSERTATION

Tampere University, Faculty of Information Technology and Communication Sciences  
Finland

*Responsible supervisor and Custos*          Professor  
Esa Rahtu  
Tampere University  
Finland

*Pre-examiners*                                  Professor  
Heikki Kälviäinen  
LUT University  
Finland

Doctor  
Mohammad Aref  
Cargotec HIAB  
Finland

*Opponents*                                      Professor  
Heikki Kälviäinen  
LUT University  
Finland

Professor  
Antti Honkela  
University of Helsinki  
Finland

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Copyright ©2024 author

Cover design: Roihu Inc.

ISBN 978-952-03-3404-8 (print)

ISBN 978-952-03-3405-5 (pdf)

ISSN 2489-9860 (print)

ISSN 2490-0028 (pdf)

<http://urn.fi/URN:ISBN:978-952-03-3405-5>



Carbon dioxide emissions from printing Tampere University dissertations have been compensated.

PunaMusta Oy – Yliopistopaino  
Joensuu 2024

# PREFACE

The culmination of a four-year odyssey has brought me to the edge of the precipice, staring into the unknown abyss of the future. The obstacles I faced along the way were no mere bumps in the road - they were towering mountains that tested the very limits of my resilience and strength. Nevertheless, these trials and tribulations have forged me into the person I am today - a fearless explorer, a passionate researcher, and a fierce advocate for the power of knowledge. I owe an immeasurable debt of gratitude to Dr. Heikki Huttunen and Prof. Esa Rahtu for entrusting me with the opportunity to embark on this transformative journey of self-discovery.

Navigating a new life in a foreign land and grappling with personal challenges while far away from loved ones is no easy feat. Through it all, I have been blessed with my family's unwavering support and love, who have been my guiding light through the darkest times. Alongside them, my colleagues, especially Bishwo Adhikari, Francesco Lomio, and Xingyang Ni, have been my steadfast companions in this quest for knowledge, making the research experience more enjoyable and rewarding than I could have ever imagined.

Through my research, I have been privileged to connect with brilliant minds in academia and industry, forging bonds that will last a lifetime. I would like to express my profound appreciation to Antti Siren, who has facilitated meetings and provided the bridge I needed to connect with such individuals.

As I stand on the cusp of this new chapter in my life, I am reminded of the words of Ursula K. Le Guin, who famously said, "It is good to have an end to journey towards, but it is the journey that matters in the end." To all those who dare to embark on their journey of self-discovery, I urge you to embrace the challenges and savor the journey itself - for it is the journey that will define you.

Saeed Bakhshi Geremi  
2024-02-19  
Tampere, Finland



# ABSTRACT

Deep learning has demonstrated tremendous potential in solving complex computational tasks such as human re-identification, optical character recognition, and object detection. Despite achieving high performance on various synthetic and real-life datasets, the absence of functional safety standards in this field hinders the development of practical solutions for safety-critical applications. Therefore, this dissertation emphasizes the safety aspect of deep learning to dissect the problem and propose potential solutions.

The first objective of this study is to investigate classification as the fundamental component of most deep learning algorithms from a safety perspective. The aim is to identify and categorize faults and their underlying causes in a typical visual classification system. The research systematically categorizes faults from three key phases: training, evaluation, and inference. Subsequently, eight distinct safety concerns were defined, and the existing mitigation methods for each fault were discussed to evaluate their effectiveness and limitations. Furthermore, potential solutions were presented directed toward the limitations. This list could be used alongside other resources to build a safety case for utilizing deep learning methods in safety-critical applications.

The second objective delves deeper into the training phase and explores the faults related to the training dataset, aiming to enhance the existing mitigation methods with safety in mind. Improved algorithms are introduced to mitigate label noise, detect outlier data, and bridge the domain gap. These problems have been analyzed from various perspectives to find practical approaches to address them. The proposed methods utilize low-cost extra resources to improve overall performance. The trade-off between cost and performance was a significant focus point in these studies. The proposed methods were compared to state-of-the-art alternatives with the help of public benchmarks to evaluate their performance.

The AI tools used in my thesis and the purpose of their use have been described below:

### **OpenAI ChatGPT (GPT-3.5)**

#### **Purpose of use and the part in which it was used:**

ChatGPT was used primarily to find and correct any grammatical mistakes, inconsistencies, or incoherent text over the entire thesis. I wrote the initial text and processed it by ChatGPT on a sub-chapter level. The prompt was “check for grammatical mistakes and enhance the text to prevent inconsistencies and improve coherency without changing the overall writing style”. Afterward, I manually checked the results, removed any artifacts, and reverted unnecessary changes that didn’t suit my writing style.

Moreover, ChatGPT has generated the description part for tools and datasets used in this thesis (e.g. ResNet structure or Clothing 1M dataset) based on the information given by their respective authors in the original webpage. The prompt was “generate a description for this tool/database for my doctoral thesis based on the provided information”. Similarly, I double-checked the results to make sure the information was correct, and the text matched my writing style.

Finally, the "Preface" part of the manuscript heavily relied on using ChatGPT. My original text was processed multiple times by ChatGPT to get a sophisticated, dramatic entry to the thesis. The prompt was “make the written text more sophisticated and dramatic while keeping it to the same length”.

I am aware that I am totally responsible for the entire content of the thesis, including the parts generated by AI, and accept the responsibility for any violations of the ethical standards of publications.



# CONTENTS

1	Introduction . . . . .	13
1.1	Objectives and Scope of the Thesis . . . . .	14
1.2	Research Questions . . . . .	14
1.3	Summary of the Publications . . . . .	15
2	Background . . . . .	19
2.1	Visual Deep Neural Classifier . . . . .	19
2.2	Safety-Critical Systems & Functional Safety . . . . .	20
2.3	Classification Models used in Thesis . . . . .	22
2.4	Datasets used in Thesis . . . . .	22
2.5	Evaluation Metrics used in Thesis . . . . .	25
3	Safety Concerns in Visual Deep Neural Classifiers . . . . .	29
3.1	Safety & Deep Learning. . . . .	29
3.2	Training Stage . . . . .	32
3.2.1	Safety Concern 1: Incomplete Dataset . . . . .	32
3.2.2	Safety Concern 2: Inadequate Dataset . . . . .	33
3.2.3	Safety Concern 3: Insufficient/Noisy Dataset . . . . .	34
3.2.4	Safety Concern 4: Ill-Matched Architecture . . . . .	34
3.3	Evaluation Stage . . . . .	35
3.3.1	Safety Concern 5: Imperfect Metrics/Benchmarks. . . . .	35
3.3.2	Safety Concern 6: Black-Box Behavior . . . . .	36
3.4	Inference Stage. . . . .	37
3.4.1	Safety Concern 7: Defective Hardware. . . . .	37
3.4.2	Safety Concern 8: Harsh Environment. . . . .	38
3.5	Discussion . . . . .	38

4	Mitigation Methods for Data-Related Safety Concerns. . . . .	41
4.1	Selective Visual Classification . . . . .	41
4.2	Data-Recalibration in Visual Classification . . . . .	46
4.3	Domain Adaptation in Visual Classification . . . . .	51
4.4	Discussion . . . . .	54
5	Conclusions . . . . .	55
	References . . . . .	59
	Publication I . . . . .	75
	Publication II . . . . .	83
	Publication III . . . . .	93
	Publication IV . . . . .	107

# ABBREVIATIONS

AI	Artificial Intelligence
AUROC	Area under the Receiver Operating Characteristic curve
CCN	Class-Conditional Noise
CNN	Convolutional Neural Network
DL	Deep Learning
e.g.	for example, from latin <i>exempli grantia</i>
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
GDA	Gradual Domain Adaptation
GNN	Graph Neural Network
GPU	Graphics Processing Unit
IDN	Instance-Dependent Noise
ISO	International Organization for Standardization
MC	Monte Carlo
ML	Machine Learning
PNN	Probabilistic Neural Network
ResNet	Residual Neural Network
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic

SOTIF	Safety of the Intended Functionality
SR	Softmax Response
SVM	Support Vector Machine
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
TPU	Tensor Processing Unit

## ORIGINAL PUBLICATIONS

- Publication I      Saeed Bakhshi Germi, Esa Rahtu, and Heikki Huttunen. “Selective Probabilistic Classifier Based on Hypothesis Testing”. In: *European Workshop on Visual Information Processing (EUVIP)*. Prais, France: IEEE, 2021. DOI: 10.1109/EUVIP50544.2021.9483967.
- Publication II     Saeed Bakhshi Germi and Esa Rahtu. “A Practical Overview of Safety Concerns and Mitigation Methods for Visual Deep Learning Algorithms”. In: *AAAI’s Workshop on Artificial Intelligence Safety (SafeAI)*. Virtual: CEUR, 2022.
- Publication III    Saeed Bakhshi Germi and Esa Rahtu. “Enhanced Data-Recalibration: Utilizing Validation Data to Mitigate Instance-Dependent Noise in Classification”. In: *International Conference on Image Analysis and Processing (ICIAP)*. Lecce, Italy: Springer, 2022. DOI: 10.1007/978-3-031-06427-2\_52.
- Publication IV     Saeed Bakhshi Germi and Esa Rahtu. “IFMix: Utilizing Intermediate Filtered Images for Domain Adaptation in Classification”. In: *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*. Lisbon, Portugal: SciTePress, 2023. DOI: 10.5220/0011713600003417.

### *Author's contribution*

The current author's contribution to each publication is as follows.

- Publication I      The author was responsible for proposing and implementing of the idea, and writing the manuscript as the corresponding author. Co-authors supervised the work and provided feedback on how to progress.
- Publication II     The author was responsible for proposing the idea and writing the manuscript as the corresponding author. The co-author supervised the work and provided feedback to refine the idea.
- Publication III    The author was responsible for proposing and implementing of the idea, and writing the manuscript as the corresponding author. The co-author supervised the work and provided feedback on how to progress.
- Publication IV     The author was responsible for proposing and implementing of the idea, and writing the manuscript as the corresponding author. The co-author supervised the work and provided feedback on how to progress.

# 1 INTRODUCTION

The extraordinary achievements of Artificial Intelligence (AI) and Machine Learning (ML) in a variety of real-world applications have drawn interest from safety-critical fields such as heavy machinery [20], logistics [114], and healthcare [107]. Deep Learning (DL), an advanced subset of ML, notably excels in tackling complex, non-linear problems that involve multi-dimensional inputs and extensive optimization needs [94]. These challenges extend from object detection in autonomous vehicles [83] to semantic segmentation in advanced medical imaging [125].

Deep neural networks are trained to identify patterns from numerous training samples, subsequently applying the learned logic to unseen samples to predict outcomes [94]. Each component of this intricate mechanism is vulnerable to faults [134], which must be identified and addressed for safety-critical applications. However, this process is impeded by the demand for comprehensive functional safety standards.

Traditional standardization methods are designed toward deterministic systems and struggle with deep learning algorithms due to their distinctive traits. For instance, conventional software development follows the 'V model,' a systematic process from requirements to maintenance, with predictable outcomes [85]. However, DL algorithms deviate from this pattern due to their data-driven nature. Their outcomes are reliant on the quality and quantity of data, and they may exhibit unprogrammed behaviors. Consequently, traditional rules prove insufficient, demanding novel development, testing, and validation approaches for DL algorithms.

Thus, this dissertation aims to investigate the faults within deep neural networks designed for visual input classification to determine how to implement such networks within the confines of safety-critical applications.

## 1.1 Objectives and Scope of the Thesis

Classification, a fundamental component of most high-level decision-making AI systems, is susceptible to various faults in the implementation cycle [134]. Faults can occur at different stages, including structured training data collection, appropriate hyperparameter selection, performance measurement metric definition, and fair comparison conduct. This research aims to provide a comprehensive list of faults within a visual deep neural classification system.

Additionally, this study aims to propose practical strategies for enhancing existing mitigation methods. While prior works have offered generic lists of safety concerns and mitigation methods [120, 96, 118], uncertainty and safety wrappers [53, 31], or targeted specific faults [134, 106], most haven't adequately considered the practicality or completeness of their solutions. This study aims to address these shortcomings by proposing practical and effective mitigation strategies built on realistic assumptions.

## 1.2 Research Questions

The research questions of this dissertation arise from two main viewpoints. The first perspective examines the underlying causes of faults in a visual deep neural classification system. Since faults can manifest at any stage of the algorithm development process, it is critical to systematically identify faults, understand their underlying causes, and design effective mitigation strategies to ensure safety.

The second perspective investigates the practical implementation of mitigation methods to address data-related faults, such as label noise, outlier data, and domain shift in a visual deep neural classification system. Current methods often rely on unrealistic assumptions, posing challenges to their practical implementation in safety-critical applications. Given the vital role of training and testing in the classifier's overall performance and robustness, finding appropriate data-related mitigation methods is crucial in deploying classifiers in safety-critical applications.

This work attempts to answer the following research questions:

- *Research Question 1:*  
Which faults can lead to the failure of visual deep neural classification systems, and how can they be systematically categorized?



- *Research Question 2:*  
What are the existing mitigation methods for addressing safety concerns in visual deep neural classification systems, and how effective are they?
- *Research Question 3:*  
How can the existing mitigation methods for data-related faults in visual deep neural classification systems be enhanced to ensure practical implementation?

### 1.3 Summary of the Publications

The overall outcome of this thesis can be seen in Figure 1.1. The results of individual publications are summarized as follows:

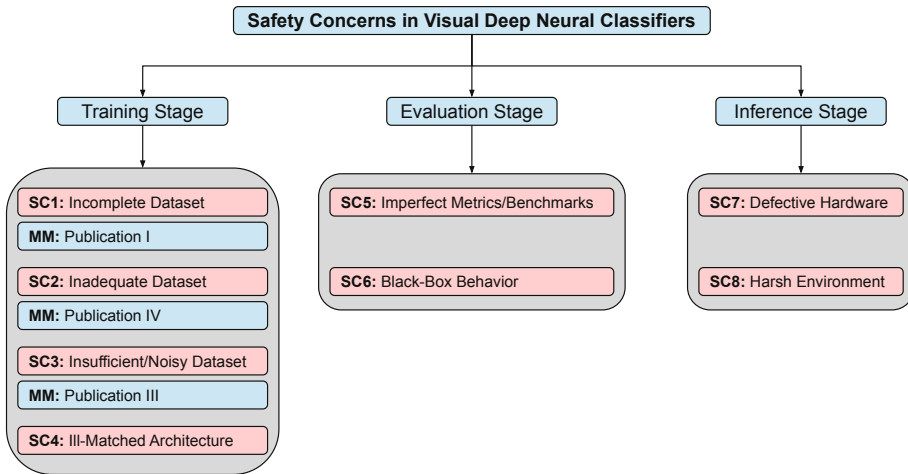
Publication I      This paper proposes a reject option based on hypothesis testing with probabilistic neural networks, aiming to implement a selective classifier that mitigates out-of-distribution data during the inference stage. The proposed method relies on an estimated distribution of outcomes to detect out-of-distribution data based on each outcome’s statistical significance. Various experiments were conducted to evaluate the proposed method using a well-known network configuration (ResNet), benchmark datasets (COCO and CIFAR), and various synthetic disturbances. The performance was compared against the traditional Softmax Response method. The results demonstrated an average improvement of 0.15 AUROC value over all test cases. Moreover, the proposed method offers a broader range of trade-off options between FPR and TPR.

Publication II     This paper delves into the safety concerns of visual deep learning algorithms and their existing mitigation methods. The research identifies a gap between the current functional safety standards and the state-of-the-art methods, hindering the validation/verification process of potential deep-learning solutions. To overcome this, the paper investigates the underlying causes of faults in visual deep learning algorithms and provides a practical and comprehensive list of safety concerns to help build a safety case for

these algorithms. Additionally, the paper highlights the limitations of current mitigation methods, underscoring the need for further research in this area. The findings offer valuable insights for researchers and practitioners working on the safety of visual deep learning algorithms.

Publication III This paper proposes an iterative data-recalibration method based on validation data to mitigate noisy labels in classification tasks. The proposed method relies on a small, clean validation dataset to update the training dataset labels in each iteration based on the network's performance. Several experiments were conducted to evaluate the proposed method using well-known network configurations (ResNet and VGG), benchmark datasets (CIFAR, Animal10N, Food-101N, Clothing1M), and different noise models (instance dependent and independent). The performance was compared against state-of-the-art algorithms based on accuracy. The results showed an average of 1 - 1.5% increase in accuracy in most of the experiments.

Publication IV This paper proposes an iterative intermediate domain generation algorithm for domain adaptation in classification tasks. The proposed method relies on a variable ratio for mixing low and high-passed images from the source and target domains to create multiple intermediate domains and train parallel networks to leverage different perspectives for better performance. Several experiments were conducted to evaluate the proposed method using a well-known network configuration (ResNet) and benchmark datasets (Office-31, Office-Home, and VisDa-2017). The performance was compared against state-of-the-art algorithms based on accuracy, and the results showed an average of 0.2 - 1.7% increase in accuracy over alternative state-of-the-art algorithms.



**Figure 1.1** The relation between publications. The overall list is proposed in publication II, with other publications dealing with specific mitigation methods.

## Structure of the Thesis

The dissertation is systematically structured to offer a comprehensive understanding of safety concerns in visual deep learning algorithms. Chapter 2 introduces the field, including the history of machine learning and deep learning, the existing gaps in functional safety standards, and the tools and datasets used for experimentation. Chapter 3 analyzes the faults in visual deep learning algorithms and the effectiveness of existing mitigation methods. The chapter summarizes the author’s work on developing a practical safety concern list. Chapter 4 presents the author’s contributions to the field, proposing and implementing various mitigation methods for data-related safety concerns, and presents experimental results based on benchmark datasets. Finally, Chapter 5 concludes the thesis with discussions on the current state of the field and future research directions, emphasizing the author’s contributions to advancing the field towards safer, more reliable deep learning algorithms.



## 2 BACKGROUND

This chapter provides a general overview of the key concepts and terminologies related to deep neural classification and functional safety to contextualize the research work conducted in this thesis. Additionally, this chapter lists the network architectures, benchmark datasets, and evaluation metrics utilized in this thesis.

### 2.1 Visual Deep Neural Classifier

Visual deep neural classifier can be explained as an image classification algorithm based on deep neural network. In computer vision, image classification involves assigning a label or category to an image based on its contents. This task requires analyzing the visual features of an input image and comparing them with the learned features of a pre-trained model to determine the most likely class label [101].

The landscape of image classification underwent a significant transformation with the advancement of deep learning techniques. One of the key advancements in this field was Convolutional Neural Networks (CNNs), which proved particularly effective for image classification tasks [9, 87].

Following the success of AlexNet [56], other deep learning architectures were introduced to further improve performance on image classification tasks. In 2016, He et al. introduced Deep Residual Networks (ResNets), which utilized residual connections to mitigate the vanishing gradient problem, enabling the training of much deeper networks [44]. Another noteworthy architecture is the Densely Connected Convolutional Networks (DenseNets) introduced by Huang et al., which improved information flow within the network by connecting each layer to every other layer in a feed-forward fashion [47].

In addition to supervised [71] classification methods, unsupervised [17] and semi-supervised [21] methods have also been explored for situations where labeled data is scarce or unavailable. Clustering methods like k-means and hierarchical clustering

can group visually similar instances without labels [70], while generative models such as Variational AutoEncoders and Generative Adversarial Networks learn to generate new instances that resemble the training data [130, 6].

The field of image classification is an active area of research, with new methods and applications being developed constantly. Open-source libraries such as PyTorch [78] and TensorFlow [1] offer a variety of pre-trained models that can be fine-tuned on specific datasets or used as feature extractors. The open sharing of research findings and resources has significantly contributed to advancements in visual classification tasks.

## 2.2 Safety-Critical Systems & Functional Safety

In safety engineering, a failure is defined as "the inability of a system to perform its required functions within specified criteria", while a fault refers to "a flaw in a component". A system is considered safety-critical when failures could result in outcomes such as loss of life, serious harm to people, significant damage or destruction of property, equipment, or environment. In these systems, faults could lead to failures which significantly increases the safety risk for people and environment [104].

Such risks are typically mitigated through safety engineering practices and tools. Functional safety ensures that all systems operate correctly in response to their inputs, even when things go wrong. Given the increasingly complex nature of modern machinery and its broad applications, such safety considerations are indispensable [35].

The history of functional safety dates back to the Machine Directive EN 2006/42/EC, which laid the foundation for functional safety in 2006. It was born out of the pressing need to provide guidance on the safety of machinery, ensuring their intended safe usage and minimizing associated risks throughout their life-cycle [27]. However, as technology evolved, the directive needed to be complemented by more detailed guidelines.

The Safety of Machinery ISO 13849 standard, published in 2006, provided a robust framework for assessing machinery safety by delving deeper into the safety requirements for the control systems used in machinery. This standard aimed to reduce the risk of accidents caused by machinery by providing explicit guidance on the design and implementation of the control systems [89].

As the use of electronics and software in vehicles increased, the need for a specific standard for functional safety in the automotive industry became apparent. In response to this need, the International Organization for Standardization published the ISO 26262 standard in 2011. This standard provides guidance on the functional safety of electrical and electronic systems in road vehicles. It extends its coverage over the system's entire life-cycle, providing a comprehensive framework to manage associated risks [85]. Other industries soon followed suit, with standards like ISO 25119 for agriculture and forestry machinery [110] and ISO 19014 for earth-moving machinery [28].

In 2019, the ISO/PAS 21448 standard (Safety of the Intended Functionality or SOTIF) was introduced, further expanding the horizon of safety standards. Recognizing that risks could stem from the system's intended functionality and unforeseen interactions between systems, SOTIF offers guidance on identifying and managing such risks [86].

All these aforementioned standards were based on traditional definition for software, in which the faults of any software component are deterministic by nature and could be reproduced by following specific steps. However, the emergence of AI tools, specifically complicated algorithms of DL, resulted in a new methodology for software development where the faults were not deterministic. New standards such as UL4600 adopt a risk-based approach to safety, acknowledging that eliminating all risks is implausible, but managing them to an acceptable level is paramount [31]. They emphasize testing and validation, providing direction on these activities and integrating their requirements with existing standards like ISO 26262.

This work will try to breach the gap between the existing standards and the practical approach to ensure safety of deep learning algorithms. This is done by following the risk-based approach, and applying it to components of a DL algorithm to find sources of faults. It is worth noting that the ISO/IEC JTC 1/SC 42 committee is currently developing safety standards related to artificial intelligence, such as ISO/IEC TR 5469 [12], indicating the growing integration of AI systems in our lives. However, at the time of doing the research the mentioned standard was not published officially.

## 2.3 Classification Models used in Thesis

To achieve the objectives of this thesis, a range of deep neural network architectures were employed to process and analyze the visual data under consideration. These architectures were selected based on state-of-the-art publications, allowing for easier comparison with alternative methods. This section describes the various architectures used in this thesis.

### ResNet

Residual Network, or ResNet, is a deep convolutional neural network introduced in 2015 [44]. The main idea behind ResNet is the use of residual connections, which allow information from earlier layers to bypass some of the layers in between and be directly added to later layers. This helps alleviate the problem of vanishing gradients that often arises in very deep networks. ResNet has become a popular choice for many applications in computer vision due to its impressive performance and the availability of pre-trained models. Its architecture has been extended and adapted for various tasks, including image segmentation, face recognition, and video analysis.

### VGG

The VGG architecture is a deep convolutional neural network introduced in 2014 [100]. It is characterized by its use of tiny 3x3 convolutional filters and its deep architecture, with up to 19 layers in the network. VGG has achieved outstanding results on various image classification benchmarks, including the ImageNet Large Scale Visual Recognition Challenge [87]. Its architecture has also served as a starting point for other network designs. Although it is less widely used than ResNet, VGG remains an essential milestone in developing deep neural networks for computer vision.

## 2.4 Datasets used in Thesis

In order to achieve the objectives of this thesis, various classification datasets were employed for training and testing. These datasets were carefully selected to represent real-world challenges and were based on state-of-the-art publications to facilitate



comparison with alternative methods. This section describes the datasets used in this thesis.

## CIFAR

The CIFAR-10 and CIFAR-100 datasets contain color images of different objects. CIFAR-10 consists of 60,000 images divided into 10 mutually exclusive classes such as airplane, dog, and truck. On the other hand, CIFAR-100 consists of 60,000 images divided into 100 classes with hierarchical relationships where every category of CIFAR-10 is divided into 10 subcategories [55]. These datasets have become popular benchmarks for evaluating the performance of image classification algorithms. The relatively small and low-resolution images make them suitable for experimentation and proof-of-concept across various tasks.

## COCO

COCO is a large-scale dataset designed for object detection, segmentation, and captioning tasks [66]. It comprises over 330,000 images of complex everyday scenes, such as street scenes, urban environments, and indoor settings. The images are annotated with 80 different object categories, including people, animals, vehicles, and various household items. Additionally, COCO provides segmentation masks for many objects, allowing for more precise localization and segmentation of objects in images. The COCO dataset serves as a challenging benchmark for object detection and other computer vision tasks due to its large size, diverse object categories, and complex scenes. While initially intended for object detection tasks, the dataset can be adjusted for classification by manually cropping the images to extract each object based on the annotations, resulting in a complex classification dataset suitable for performance evaluation.

## Animal-10N

The Animal-10N dataset is valuable for evaluating computer vision models' robustness to noisy labels across various tasks, including image classification and object recognition [105]. This dataset comprises 55,000 images of ten animals, collected through online search engines and categorized into five pairs of visually similar an-

imals such as cat and lynx. The noise rate of the dataset, estimated through cross-validation and human inspection, is approximately 8%, reflecting the noise present in real-world datasets. This makes the Animal-10N dataset a challenging benchmark for evaluating the robustness of computer vision models.

## Food-101N

The Food-101N dataset is a large-scale dataset designed to address label noise with minimal human supervision [61]. It contains 310,000 images of food recipes belonging to 101 different classes, with an estimated noise rate of around 20%. Each image in the dataset has a verification label indicating whether the class label is correct or not. The verification labels are manually assigned to a subset of the images for training and validation. The Food-101N dataset facilitates learning image classification with label noise and label noise detection, making it a challenging benchmark for developing and evaluating robust image classifiers that can handle label noise. It is an extension of the Food-101 dataset [18] and provides a noisier and larger environment for model evaluation.

## Clothing1M

The Clothing1M dataset is a large-scale dataset of clothing images with noisy labels [121]. It comprises one million clothing images from 14 classes such as T-shirts, Shirts, and Knitwear. The data was collected from various online shopping websites, resulting in the inclusion of many mislabeled samples. To address this issue, the dataset also includes 50k, 14k, and 10k images with clean labels for training, validation, and testing, respectively. The dataset also includes the surrounding text provided by the sellers, which can be used as visual tags.

## Office-31

The Office-31 dataset serves as a popular benchmark in domain adaptation and transfer learning [88]. It contains 4,110 images across three domains: Amazon, DSLR, and Webcam, with 31 object categories. The Amazon domain comprises clean background images captured from online merchants, the DSLR domain consists of high-quality images captured by a DSLR camera, and the Webcam domain contains low-

resolution and noisy images captured by a webcam. The dataset has been widely used to evaluate the ability of algorithms to generalize across different domains, such as adapting a classifier trained on one domain to perform well on a different domain.

## Office-Home

The Office-Home dataset is created for evaluating domain adaptation algorithms for object recognition using deep learning methods [112]. It consists of 15,500 images of 65 object categories from four domains: Artistic, Clip Art, Product, and Real-World images. The images were collected using a web crawler that searched through different search engines and image directories. They were filtered to ensure that the desired object was present in each picture, and categories were filtered to ensure a minimum number of images per category. One of the distinctive features of this dataset is the diversity of images, including variations in color, lighting, viewpoint, and background. The dataset poses challenges due to domain shifts between domains and intra-class variations, making it suitable for evaluating the robustness of object recognition algorithms in various scenarios.

## VisDa-2017

The VisDa-2017 dataset is designed for unsupervised domain adaptation for image classification [80]. It includes 280,000 samples from 12 object categories, such as airplane, horse, and person. The dataset is divided into a training domain (source) with synthetic 2D renderings of 3D models generated from different angles and lighting conditions and a validation domain (target) with photo-realistic or real-image samples. The complexity and realism of the dataset make it a valuable benchmark for domain adaptation problems, as algorithms must adapt to different image domains without explicit guidance from labeled data.

## 2.5 Evaluation Metrics used in Thesis

Several metrics were used to evaluate the performance of the proposed methods, including accuracy, the area under the receiver operating characteristic curve, and the Z-test. Each metric was selected based on its relevance to the specific task and its ability to comprehensively evaluate the effectiveness of the proposed methods. These

metrics were also used to compare the proposed methods with baseline approaches and identify areas for further improvement. This section describes the various evaluation metrics used in this thesis. The definitions for these metrics are taken from ISO/IEC TS 4213 [49].

#### True Positive (TP)

The number of samples correctly classified as positive.

#### True Negative (TN)

The number of samples correctly classified as negative.

#### False Positive (FP)

The number of samples wrongly classified as positive, also known as *False alarm* or *Type I error*.

#### False Negative (FN)

The number of samples wrongly classified as negative, also known as *Miss* or *Type II error*.

#### Accuracy

The number of correctly classified samples divided by all classified samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

#### True Positive Rate (TPR)

The number of samples correctly classified as positive divided by all positive samples. Also known as *Sensitivity*, *Recall*, and *Hit rate*.

$$TPR = \frac{TP}{TP + FN}$$

## True Negative Rate (TNR)

The number of samples correctly classified as negative divided by all negative samples. Also known as *Specificity* and *Selectivity*.

$$TNR = \frac{TN}{TN + FP}$$

## False Positive Rate (FPR)

The number of samples incorrectly classified as positive divided by all negative samples. Also known as *Fall-out*.

$$FPR = \frac{FP}{TN + FP}$$

## False Negative Rate (FNR)

The number of samples incorrectly classified as negative divided by all positive samples. Also known as *Miss rate*.

$$FNR = \frac{FN}{TP + FN}$$

## Receiver Operating Characteristic (ROC) curve

A graphical method for displaying the true positive rate (TPR) vs. the false positive rate (FPR) across multiple thresholds.

## Area Under Receiver Operating Characteristic curve (AUROC)

The overall area under ROC curve.

## Two-sample Z-test

A statistical test used to compare the means of two independent samples to determine whether they have a significant difference [25].

$$Z = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Where  $\mu_1$  and  $\mu_2$  are the sample means,  $\sigma_1$  and  $\sigma_2$  are the standard deviations, and  $n_1$  and  $n_2$  are the sample sizes.

# 3 SAFETY CONCERNS IN VISUAL DEEP NEURAL CLASSIFIERS

Despite the undeniable potential of deep learning in tackling complex computational tasks, their heuristic and unexplainable nature hinders the verification and validation for safety-critical applications as traditional functional safety standards and reliability requirements do not account for the unique properties of deep learning algorithms. To address this issue, recent research has focused on developing safety concern lists as effective safety-case frameworks for practically applying these algorithms. However, many existing works in this area only consider safety criteria and do not address the implementation challenges of proposed mitigation methods.

This chapter provides an overview of the work on identifying the underlying causes of faults in visual deep learning algorithms. The aim is to generate a list of safety concerns and potential state-of-the-art mitigation techniques. The approach involves breaking down the process of building a deep learning algorithm into three phases: training, evaluation, and inference. After that, the relevant components in each phase are analyzed to determine their impact on the algorithm's overall performance. The focus on practical implementation distinguishes this work from previous research. This chapter corresponds to publication II.

## 3.1 Safety & Deep Learning

The question of whether deep learning algorithms can be used in safety-critical applications is important. To answer it, an understanding of how deep learning algorithms relate to safety must be established. In theory, deep learning algorithms can either replace or work in conjunction with traditional safety mechanisms. The three possible interactions between deep learning and safety can be defined as follows:

- Deep learning algorithms can entirely replace the traditional safety mechanisms and assume full responsibility for ensuring the safe operation of the system. For example, in the domain of forestry machines, a deep learning algorithm can autonomously analyze sensor data to detect potential hazards like falling branches, unstable terrain, or nearby obstacles. By leveraging a trained model and real-time data, the algorithm can make decisions to avoid accidents and maintain the safety of the machine and its operator.
- Deep learning algorithms can cooperate with traditional safety mechanisms to enhance their capabilities and improve overall system safety. For instance, in port cranes, deep learning algorithms can analyze video feeds from multiple cameras in combination with existing collision detection systems. By integrating the knowledge extracted from visual data with the crane's sensor-based collision avoidance system, the deep learning algorithm can provide additional information and insights, enabling more accurate and robust decision-making to prevent collisions and ensure the safety of personnel.
- Deep learning algorithms can be employed in safety-critical applications without directly influencing the safety mechanisms of the system. Instead, they serve other purposes such as optimizing operations or providing additional insights. For example, in mining machines, a deep learning algorithm can analyze sensor data to predict optimal maintenance schedules based on patterns indicative of wear and tear. By proactively scheduling maintenance tasks, the algorithm contributes to the overall efficiency and reliability of the machine, indirectly supporting safety by reducing the likelihood of unexpected failures.

A major drawback in utilizing deep learning algorithms is the heuristic and multi-dimensional nature which makes them difficult to explain and interpret. As current safety standards rely on the traditional definition of software, it is impossible to practically implement standard verification and validation methods for deep learning algorithms. Consequently, standards such as IEC 61508 [35] advise against using artificial intelligence (including deep learning algorithms) in systems with a safety integrity level of two or higher [19].

Researchers have proposed alternative approaches to address the challenges associated with safety in deep learning algorithms. One prominent approach involves the development of explainable safety concern lists, which aim to identify potential safety risks specific to deep learning algorithms and offer appropriate mitigation

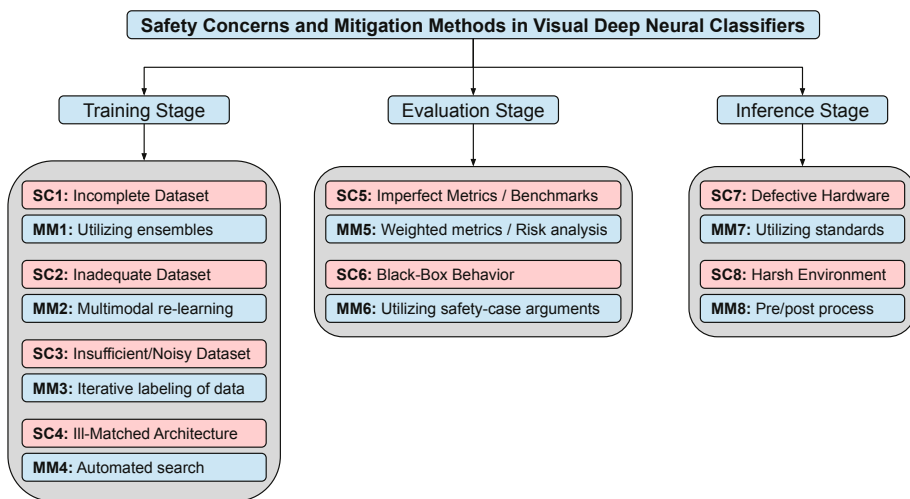


methods.

A notable example of such an approach is UL4600, which provides a safety-case solution tailored for implementing artificial intelligence in autonomous vehicles [31]. UL4600 focuses on creating a comprehensive list of arguments that address safety concerns associated with AI systems. This safety-case approach helps stakeholders in the autonomous vehicle industry assess and evaluate the safety implications of their systems.

However, it is essential to note that while UL4600 addresses the evaluation aspect of safety, it does not provide detailed practical guidance for the entire implementation cycle. The practical implementation of safety measures involves various stages, including system design, development, verification, and validation. These stages require comprehensive guidelines and standards to ensure AI's safe and reliable integration in autonomous vehicles.

Thus, this chapter provides a practical safety concern and mitigation method list for deep learning algorithms. The overall results are illustrated in Figure 3.1. Combining this with existing safety case arguments, proper monitoring tools, contingency protocols, and practical explanations would result in a safe implementation of deep learning algorithms.



**Figure 3.1** The complete list of safety concerns and mitigation methods in a visual deep neural classifier.

## 3.2 Training Stage

By minimizing the empirical risk of the training data, a deep learning algorithm estimates the relationship between input data and the desired output. Consequently, a suitable training dataset is crucial to achieving the desired outcome. Additionally, various model structures have distinct advantages and disadvantages. Selecting the appropriate architecture, configuring a suitable loss function and optimization algorithm, and identifying the ideal hyperparameters are all necessary for optimal performance.

Extracting useful information from visual data is a complex task, posing a challenge for humans and machines. Ideally, a perfect training dataset would be [134]:

- *Complete*: Contains samples from a defined input-output space.
- *Adequate*: Contains samples with identical distribution to the workspace.
- *Ample*: Contains enough samples for training the algorithm.
- *Clean*: Contains well-labeled and noise-free samples.

However, it is nearly impossible to perfectly fulfill these conditions in real-world scenarios. The violation of each condition will result in a unique form of fault in the deep learning algorithm. Additionally, the incorrect selection of model and hyperparameters might result in issues such as overfitting, the divergence of the model, and overconfidence, which can further contribute to faults.

### 3.2.1 Safety Concern 1: Incomplete Dataset

The inherent complexity of the real world implies that the defined space for a task is always much smaller than the unexplored broader space. Potential sources of faults, such as outlier classes (known unknowns) and adversarial attacks (unknown unknowns), pose a considerable risk to the algorithm. These factors can cause the algorithm to produce overconfident yet incorrect predictions [134].

Recent studies suggest utilizing out-of-distribution detectors to identify unseen samples and reject the algorithm's overconfident outcomes [22, 92]. While uncertainty metrics have the potential to measure reliability, they often result in a trade-off between accuracy and safety due to the conservative nature of uncertainty methods [95]. Alternatively, open-world recognition systems can expand the algorithm's

workspace by incorporating outlier samples encountered during the inference stage [77, 15]. However, learning constantly requires high computational power and might result in a complicated model that cannot operate efficiently on the original classes. As for the adversarial attacks, the model can be trained to defend against them by incorporating attack patterns into the training dataset [123, 128, 39]. Nevertheless, the attack algorithms keep evolving at the same rate.

A suitable mitigation method could involve utilizing an ensemble of models. By combining multiple models with different architectures and training datasets, an ensemble can improve the model's overall performance and provide a more robust safety guarantee. For example, the ensemble can consist of models trained on different subsets of the training dataset or models with different architectures or hyperparameters. The models in the ensemble can be weighted differently depending on their performance and safety metrics, and their combined output can be used to make the final prediction. Using an ensemble, the algorithm can identify and reject out-of-distribution samples and improve its accuracy and safety in safety-critical applications.

### 3.2.2 Safety Concern 2: Inadequate Dataset

The dynamic nature of the real world implies that the distribution of data samples is likely to vary between the training and inference stages over time. Various factors within and outside system's control, such as hardware equipment or weather conditions, could result in a distribution shift and affect the algorithm's performance [134].

Recent research has proposed using domain adaptation techniques to address the domain gap between the training and inference stages [33, 138]. These methods often require a small batch of data during inference to adapt the model to the new environment. State-of-the-art techniques can be fine-tuned to meet the needs of most safety-critical applications. Alternatively, an algorithm can integrate multiple sources of information to perform a single task (e.g., person identification using face, iris, voice, and fingerprint) [14, 42]. Multimodal methods, such as sensor fusion in autonomous vehicles (using LIDAR, GPS, and IMU for localization), have already been successfully implemented and are well-suited for safety-critical applications.

A suitable mitigation method could involve utilizing multiple data sources and sensors during the inference phase and iteratively re-training the model within stan-

standard time frames. For instance, in an autonomous driving scenario, LIDAR and camera data can be combined to enhance the detection and tracking of objects, while GPS and IMU can be used to improve the vehicle's localization accuracy. Using multiple data sources and sensors can help ensure the model adapts to the changing environment and maintains its performance over time.

### 3.2.3 Safety Concern 3: Insufficient/Noisy Dataset

The manual labeling of samples in a large dataset can be susceptible to errors and noisy labels due to various factors such as unacquainted workers, insufficient information, confusing patterns, and the massive scale of data. Despite the existence of iterative labeling methods [4], the initial cost of gathering and maintaining a clean dataset increases exponentially compared to its size. With the unavoidable noise in every training dataset, the algorithm will memorize the noise pattern, resulting in poor generalization [134].

Recent research has discovered that one way to mitigate noise in datasets is by combining various methods, such as robust loss and iterative relabeling [106, 26, 5]. Alternatively, data augmentation methods can create additional samples for the training dataset using different transformations and generative algorithms [115, 98, 82, 76]. Although synthesized data may not perfectly represent the real world, it can bridge the gap between the two domains. Additionally, semi-supervised and unsupervised training techniques can reduce dependency on a clean dataset [29, 93]. However, fully supervised methods usually yield higher accuracy.

A suitable mitigation method could generate an initial dataset using iterative labeling techniques and extend it with synthetic data generation methods. This would help bridge the gap between the real-world and synthesized data and increase the dataset's size, leading to improved model generalization.

### 3.2.4 Safety Concern 4: Ill-Matched Architecture

It can be time-consuming and expensive to manually compare various models and hyperparameters to determine the best fit for a particular task. Additionally, such a process necessitates the involvement of an industry expert to provide insights into the issue at hand. Selecting the wrong architecture can result in unforeseen issues or performance degradation caused by inherent weaknesses in specific scenarios.

Automated methods for hyperparameter optimization [127, 67, 48] and neural architecture search [119, 84] have been developed to alleviate the burden of manual work and eliminate the requirement for an expert. These methods use various search algorithms to locate the optimal model and hyperparameters within the relevant domain.

A suitable mitigation method could involve utilizing automated tools with proper validation benchmarks to select the model and hyperparameters for each new task.

### 3.3 Evaluation Stage

To ensure the algorithm’s reliability, the testing dataset should consist of samples from all identified situations, even infrequent ones. An appropriate dataset should also uphold similar characteristics to the training dataset. Furthermore, selecting appropriate performance metrics during testing has a crucial effect on the comparisons, requiring a prior understanding of the task. Moreover, existing verification/validation methods rely on the explainability and interpretability of the algorithm, thus making additional information about the algorithm’s function helpful.

While expert knowledge is expected in safety-critical applications, sweeping the entire working space during the tests is impossible. Furthermore, the black-box behavior of deep learning algorithms makes it challenging to explore the unknown workspace for potential safety concerns.

#### 3.3.1 Safety Concern 5: Imperfect Metrics/Benchmarks

While accuracy is the most frequently used performance metric in deep learning algorithms, it does not fully capture the algorithm’s reliability, uncertainty, and other attributes.

Recent studies suggest using weighted cost functions that correspond to different types of errors as performance metrics, which enables the algorithm to be assessed according to safety criteria [137, 38, 91]. Such novel evaluation metrics would make it easier to discern the performance and safety trade-offs. Designing a cost function requires specialized expertise since a poor decision could result in a non-converging algorithm. Therefore, solid mathematical arguments are needed to prove the algorithm’s convergence.

Conversely, a risk analysis can compile a comprehensive list of all hazardous sce-

narios relevant to a given task for inclusion in the testing dataset [129, 59]. Such datasets could serve as benchmarks for evaluating and verifying the performance of algorithms. While exploring the entire workspace is almost impossible, sharing suitable benchmarks could reduce the cost burden.

A suitable mitigation method could involve utilizing third-party experts to model more sophisticated and specialized metrics for individual tasks and perform risk analysis to ensure the quality of existing benchmark datasets.

### 3.3.2 Safety Concern 6: Black-Box Behavior

The massive number of parameters and non-linear functions used in deep learning algorithms create an uninterpretable system or a black box. Given the need for a clear correlation between the inputs and outputs of this system and the impossible challenge of testing the entire workspace, evaluating the reliability and safety of deep learning algorithms is a difficult task.

Recent studies suggest using representation learning to identify the connection between input data and output in an interpretable manner by revealing the feature selection process [132, 64], which helps to understand how the network perceives input data and which data components are more significant in determining the output. Alternatively, a map of pixel relevance demonstrates each pixel's significance in the output calculation [54, 13]. These heat maps can provide information on individual pixels or the interactions between different pixels. Analyzing these maps can reveal the impact of minor variations in input on the output and assist in identifying potential safety hazards. However, this information cannot be utilized to verify/validate the algorithm according to existing standards.

This issue is one of the most crucial topics in the current research community, with no adequate solution. A suitable mitigation method could involve using safety case arguments to simplify the process of verification/validation. Since safety case arguments involve constructing a logical argument based on available evidence to demonstrate that the system is safe rather than fully attempting to understand the algorithm's inner workings, they could explain the reliability and safety of the deep learning algorithm even in the presence of black-box behavior.

## 3.4 Inference Stage

Once the algorithm is trained and evaluated, it needs multiple hardware parts to interact with the real world. A visual deep learning algorithm requires a camera to capture input data, a communication channel to transmit the data, a processing unit to compute the results, and a power supply to sustain its operation. Different hardware setups may introduce unintended biases not present during the training or evaluation stages, resulting in unpredictable behavior.

Furthermore, hardware failures or changes in the environment can affect the system's performance, leading to unexpected outputs or failures. Therefore, the hardware components used during the inference stage must be verified and validated to ensure they function correctly under different operating conditions.

### 3.4.1 Safety Concern 7: Defective Hardware

Hardware faults can impact the algorithm's outcome in various ways. For instance, sensor faults can cause input data disturbances, leading to data corruption or distortion. Similarly, communication channel faults can result in data loss or corruption, while processing unit faults can cause incorrect calculations, algorithmic delays, or system freeze. Lastly, power supply faults can damage other hardware components or result in a complete system shutdown.

The mentioned hardware components have been in use for safety-critical applications for decades. Thus, functional safety standards such as ISO 26262 [85] and ISO/PAS 21448 [86] offer valuable recommendations for verifying and validating hardware components. Additionally, technical reports based on functional safety standards can aid in selecting or developing secure hardware components like a camera [30], communication channel [7], and operating system [103]. Furthermore, safety-critical applications have found other measures helpful, including redundant hardware, adequate noise shielding, and data fusion techniques [102, 24].

A suitable mitigation method for hardware faults could involve conducting a thorough risk analysis to identify possible hardware failures and their consequences. Then, risk mitigation strategies could be implemented, such as selecting hardware components that meet specific safety standards, implementing hardware redundancy, and monitoring the system for potential hardware failures. Finally, regular main-

tenance and testing of hardware components could prevent hardware faults from leading to catastrophic consequences.

### 3.4.2 Safety Concern 8: Harsh Environment

Once the hardware is chosen based on appropriate functional safety standards, it should operate without significant safety concerns. However, the mitigation method only partially eliminates disturbances or data corruption. Environmental factors, such as poor illumination, movement, and obscured objects, can affect input image quality without causing hardware failure. While some of these issues may be unrecognizable by a human annotator, the deep learning algorithm may encounter faults based on the type and severity of corruption. Additionally, less severe hardware failures may cause noise variations in the input data.

Recent studies suggest using image processing techniques, such as denoising [32, 40, 50], deblurring [133, 75, 3], and enhancement methods [81], to mitigate the effect of environmental disturbances on the input data. These techniques are proven to be effective when the corruption type is known. Otherwise, such functions would negatively affect the input data, like removing or fading edges. Thus, it is assumed that corruption of data is inevitable.

As these methods are closely tied with mitigation methods for training data, both problems can be handled by multiple data-related approaches to mitigate the effect of noise, corruption, and domain gap. The next chapter explains the work in mitigating data-related safety concerns.

## 3.5 Discussion

This chapter discussed various safety concerns for visual deep learning algorithms, including dataset bias, domain gap, adversarial attacks, overfitting, interpretability, defective hardware, and harsh environments. This work presented the underlying causes of each safety concern and state-of-the-art solutions to mitigate them, highlighting the limitations of existing mitigation methods and the need for further research in the field.

The chapter also emphasized the importance of considering the entire life cycle of the algorithm, including the training, validation, and inference stages, as well as the hardware and environmental factors that could affect the algorithm's performance



and safety. Furthermore, the chapter discussed the need for specialized expertise in designing and implementing suitable mitigation methods for these safety concerns.

Overall, the complexity and non-linearity of deep learning algorithms pose significant challenges to traditional broad-spectrum standardization methods. However, alternative solutions such as functional safety standards and safety case arguments, combined with data-related approaches, can help ensure visual deep learning algorithms' safe and reliable operation. As such, continued research and development in this field are crucial to address emerging safety concerns and enable the use of deep learning algorithms in safety-critical applications.



## 4 MITIGATION METHODS FOR DATA-RELATED SAFETY CONCERNS

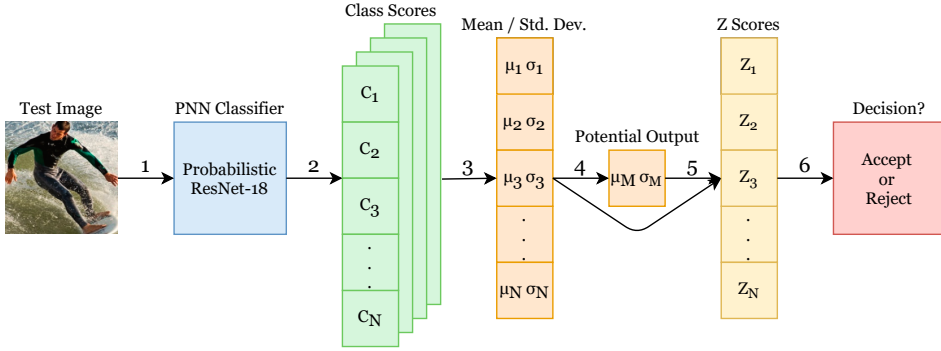
In recent years, data-related safety concerns in deep learning algorithms have received significant attention from the research community. While many academic works propose complex innovations for state-of-the-art solutions, these methods' practical implementation and real-world applicability are often overlooked. Moreover, the performance metrics used in academia may not fully address the requirements of safety-critical applications, resulting in a gap between research and practice.

This chapter aims to bridge this gap by presenting practical improvements to existing mitigation methods for data-related safety concerns. Specifically, the focus is on three areas: mitigating the effect of outliers and distorted data with probabilistic selective classifiers, mitigating the effect of label noise with enhanced data recalibration, and mitigating the domain gap in transfer learning with iterative intermediate domain generation. The proposed methods improve upon existing approaches by incorporating more realistic assumptions about the data and the demands of real-world applications. These works correspond to publications I, III, and IV, respectively.

Given the lack of functional safety standards in this field, evaluating the safety of the proposed methods is challenging. Therefore, the performance evaluation relies on the existing metrics from the literature.

### 4.1 Selective Visual Classification

Despite achieving high accuracy on public benchmark datasets, state-of-the-art classifiers may perform incorrectly when faced with circumstances outside the training set. As stated in the previous chapter, the training dataset is assumed to be complete, and the testing dataset is supposed to be clean. Thus, the algorithm does not expect an out-of-distribution or a distorted sample at the inference stage. However, enforcing these restrictions on datasets in real-world applications is high impossible. Thus,



**Figure 4.1** The structure of the proposed selective classifier. (1) Feed the test image to the probabilistic classifier  $k$  times. (2) Record the class scores generated in each inference. (3) Calculate the mean and standard deviation for each class. (4) Identify and designate the maximum mean value as potential output. (5) Conduct two-sample Z-tests between the potential output and all other classes, and record the resulting Z-scores. (6) Determine the outcome by comparing the Z-scores against a threshold value. Reproduced with permission from publication I (©2021 IEEE).

the classifier may produce erroneous results when presented with such samples.

Based on the direction of previous research, a better metric to measure a classifier’s ability to generalize and be robust to environmental changes is essential. Uncertainty is one such metric that measures the algorithm’s confidence in its decision [10]. Utilizing such a metric would allow a selective classifier to reject uncertain outcomes. Previous studies have used techniques such as modified activation functions [16, 65, 99], modified loss functions [126, 122], voting systems [58, 36], and combinations of different ideas [113] to achieve this goal.

Probabilistic Neural Network (PNN) is a type of neural network that uses stochastic weights to perform classification and pattern recognition tasks [72]. During training, the model learns to estimate the probability distribution function of each class. When presented with new input data, PNN estimates the probability of each class and assigns the input data to the class with the highest posterior probability, thereby reducing the probability of misclassification. Since each inference produces a slightly different output, a low standard deviation between multiple inferences indicates a higher level of network certainty, making it a valuable metric for measuring uncertainty.

This work proposes utilizing hypothesis testing on the output of a PNN to calculate the uncertainty of the algorithm. The proposed approach involves performing

a Z-test on the distribution of outputs generated by the PNN to determine the statistical significance of a given result and reject any insignificant outcomes. The main distinction between the proposed method and previous state-of-the-art approaches, such as ODIN [65], is that the proposed method does not restrict the use of different network architectures. Algorithm 1 summarizes the proposed method in steps, and Figure 4.1 shows its structure. The proposed method is compared to Softmax Response (SR), which can outperform other alternatives such as Monte Carlo (MC) dropout, according to Geifman and El-Yaniv [37].

---

**Algorithm 1:** Selective Probabilistic Classifier

---

**Require:** A trained probabilistic classifier, a threshold value for statistical significance  $T$

- 1: Run the test image through the classifier  $k$  times
- 2: Store mean ( $\mu$ ) and standard deviation ( $\sigma$ ) for all  $N$  classes
- 3: Find the potential output, the class with the highest mean value ( $c_M$ )
- 4: **for**  $i \in 1, 2, \dots, N; i \neq M$  **do**
- 5:   Compute the two-sample Z-test between  $c_M$  and  $c_i$
- 6:   Store the  $Z_i$  score
- 7: **end for**
- 8: **if**  $Z_i > T$  for  $i \in 1, 2, \dots, N; i \neq M$  **then**
- 9:    $C_M$  is chosen as the output
- 10: **else**
- 11:    $C_M$  is rejected as an output
- 12: **end if**

**Return** Data classes based on threshold value  $T$

---

The main contributions of this work are twofold. First, a technique is proposed based on statistical analysis and PNN to estimate the uncertainty of a classifier and build a reject option to mitigate out-of-distribution and distorted samples during inference time. Second, proposed method is evaluated by conducting extensive tests to simulate out-of-distribution and distorted samples and demonstrate the effectiveness of the proposed method over the baseline SR method. The proposed method is not restricted to any specific architecture and can be combined with similar methods.

The experiment was conducted using the CIFAR [55] and COCO [66] datasets. The CIFAR dataset was chosen for the feasibility study, where *Automobile* and *Truck* classes were excluded from the training set to simulate the out-of-distribution samples during the inference phase. The COCO dataset was chosen to evaluate the

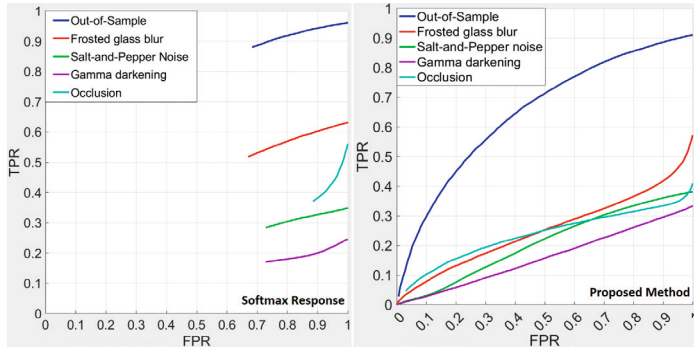


**Figure 4.2** Distortions on the sample image. (A) Original image. (B) Motion blur. (C) Frosted glass blur. (D) Gaussian blur. (E) Noise. (F) Gamma darkening. (G) Gamma lightening. (H) Occlusion. Reproduced with permission from publication I (©2021 IEEE).

performance on challenging datasets. Since COCO is designed as an object detection dataset, the instances of objects were manually extracted and categorized into four classes: Human, Vehicle (any four-wheeled vehicle), Animal (any four-legged animal), and Background (patches of images with no overlapping objects) where the *Animal* class was excluded from the training set to simulate the out-of-distribution samples during the inference phase.

The ResNet-18 [44] network with probabilistic weights was used to test the proposed method and evaluate its performance against the SR method. The comparison did not include other state-of-the-art methods due to reliance on a specific structure or limited test capacity with complex datasets. AUROC was used as a threshold-independent metric for a fair comparison, and both networks were trained from scratch with the same initial configuration. Several experiments were conducted to simulate the effect of out-of-distribution and distorted samples. Out-of-distribution samples were selected from the classes not present during the training, and distortions were added artificially based on Kamann’s work (as shown in Figure 4.2) [51].

The ROC curves for the COCO test indicate that the proposed method has a larger span over the TPR-vs-FPR trade-off (Figure 4.3). While the SR method can theoretically achieve a 0% FPR, it only happens if the algorithm is set to reject all outputs. Thus, AUROC is calculated using only the valid parts of the ROC curve, and the results are presented in Table 4.1. Judging by the results, the proposed method offers a better option for the trade-off between accuracy and uncertainty, resulting in a higher value for AUROC. Combined with the fact that the cost of



**Figure 4.3** ROC curves for the COCO dataset. The worst performance of each category was chosen to be presented. Increasing the threshold value decreases TPR and FPR. Reproduced with permission from publication I (©2021 IEEE).

false positives versus false negatives may vary in different safety-critical applications, having a broad range of trade-off is beneficial.

Dataset	Method	Out of Distribution	Blur			Noise		Gamma correction		Occlusion
			Motion	Frosted glass	Gaussian	Gaussian	S&P	Darkening	Lightening	
COCO	Proposed	0.65	0.34	0.25	0.38	0.22	0.21	0.16	0.17	0.23
	SR	0.29	0.20	0.18	0.22	0.14	0.09	0.04	0.05	0.06
CIFAR	Proposed	0.89	0.50	0.50	0.59	0.38	0.39	0.37	0.42	0.48
	SR	0.52	0.44	0.34	0.47	0.35	0.25	0.22	0.26	0.31

**Table 4.1** AUROC values for selective classifier tests. Reproduced with permission from publication I (©2021 IEEE).

## Conclusion

In conclusion, the proposed method incorporates a rejection option into probabilistic classifiers through Z-test analysis. The Z-test analyzes multiple runs' mean and standard deviation to identify uncertain results and estimate network certainty. Multiple experiments were conducted using a well-known network configuration (ResNet) and datasets (CIFAR and COCO), comparing the proposed method with the SR method using a threshold-independent metric. The proposed method improved the AUROC value by an average of 0.15 over all test cases while offering a broad range of trade-offs between accuracy and uncertainty.



**Figure 4.4** Samples of the same category in different datasets: (A) Number two in MNIST [60], (B) Deer in CIFAR-10 [55], (C) Caesar salad in Food-101N [61], and (D) Underwear in Clothing1M [121]. The images on the top row are more straightforward to label than those on the bottom. Reproduced with permission from publication III.

## 4.2 Data-Recalibration in Visual Classification

As stated in the previous chapter, the training dataset is assumed to be sufficiently large and clean for practical training of deep learning models. However, manual labeling of a large dataset can be challenging, with factors such as inexperienced annotators, domain-related expertise requirements, complex samples, and the massive volume of data contributing to the possibility of errors and noisy labels [34, 111]. This poses a challenge to deep learning algorithms, as they may memorize the noisy patterns, resulting in poor generalization and reduced performance during inference [131]. This crucial issue in various sectors, such as safety-critical applications [5] and medical imaging [97], lead researchers to develop mitigation methods for label noise [34, 8, 106].

While many recent studies assume that annotations are affected by Class-Conditional Noise (CCN) [69] and try to estimate [45] or mediate it by modifying the architecture [11], Chen has demonstrated that label noise in real-world datasets, such as Clothing1M [121], is instance-dependent [23]. Figure 4.4 provides a better understanding of this, showing that two samples of the same category have different labeling complexities, indicating that label noise varies per instance. Based on Chen’s findings, researchers have started formulating Instance-Dependent Noise (IDN) patterns to study effective mitigation methods.

One of the widely used mitigation methods for noisy labels is iterative data recal-



ibration. The process involves an iterative loop in which the network is trained to find the top-performing network, and the noisy labels are corrected based on confidence scores obtained from the top-performing network. This process continues until convergence, resulting in a model with improved accuracy and better generalization ability [63]. This work proposes an iterative data-recalibration method that utilizes a clean validation dataset to mitigate the effect of label noise in classification. The approach involves iteratively training a network with labels from the previous stage, evaluating its performance on a smaller clean validation dataset, and using the best-performing model to re-label the training dataset. The main distinction between the proposed method and previous state-of-the-art approaches is the involvement of validation data in selecting the best-performing network (also known as the Oracle network). Algorithm 2 summarizes the proposed method in detailed steps.

---

**Algorithm 2:** Enhanced Data-Recalibrator

---

**Require:** Initial classifier  $f^\circ$ , Threshold value  $\theta$ , Number of epochs  $T$ ,  
Training set  $\tilde{D}_{train}^\circ = \{(x_i, \tilde{y}_i^\circ)\}_{i=1}^n$ , Validation set  $D_{valid} = \{(x_i, y_i)\}_{i=1}^m$

- 1: **for**  $t \in 1, \dots, T$  **do**
- 2:   Train  $f^{t-1}$  on  $\tilde{D}_{train}^{t-1}$  to get  $f^t$
- 3:   Calculate the performance score  $S^t$  of  $f^t$  on  $D_{valid}$  and compare to previous scores
- 4:   Find best performing classifier  $f^B$
- 5:   **for**  $(x, \tilde{y}) \in \tilde{D}_{train}^{t-1}$  **do**
- 6:     Get the confidence scores  $(C_1, \dots, C_k)$  of  $f^B$  on  $x$
- 7:     Find the best confidence score  $C_M$  and the confidence score of the previous label  $C_N$
- 8:     Calculate  $P = |\log(C_M) - \log(C_N)|$
- 9:     **if**  $P \geq \theta$  **then**
- 10:       Set new label  $\tilde{y}^t = M$
- 11:     **else**
- 12:       Keep old label  $\tilde{y}^t = \tilde{y}^{t-1}$
- 13:     **end if**
- 14:   **end for**
- 15:   **if**  $\forall i \in [1, \dots, n], \tilde{y}_n^t = \tilde{y}_n^{t-1}$  **then**
- 16:     Decrease  $\theta$  by a small amount
- 17:   **end if**
- 18: **end for**

**return** Best trained network  $f^B$  and cleaned dataset  $\tilde{D}_{train}^T$

---

The main contributions of this work are twofold. First, an enhanced data-recalibration method is proposed based on using a clean validation dataset. The cost of collecting this dataset is negligible compared to the performance gain. Second, the proposed method is evaluated by conducting extensive tests on synthetic noise patterns and public benchmarks to demonstrate the effectiveness of the proposed method over the state-of-the-art alternatives. Multiple techniques were used in the implementation stage to improve the performance of the proposed method. First, a high initial learning rate was used to allow the network to achieve satisfactory accuracy without overfitting to noise [131]. Second, the confidence scores from multiple high-performing networks are averaged to prevent accidental bias toward a specific class due to randomness. Third, the selection pool for averaging networks is limited to recent iterations to reduce the effect of confirmation bias from unexpected occurrences of falsely confident models.

CIFAR [55] dataset was used for the initial tests, where synthetic noise was added to samples. To ensure comparability, the IDN patterns were based on descriptions provided by Zhang [135]. The formulas can be seen in Equation 4.1, where  $\aleph_{C_1, C_2}$  is the probability of changing the label of a sample from the most confident class  $C_1$  to the second-most confident class  $C_2$ , and  $f^*$  is an oracle classifier trained on clean samples. Additionally, the CCN patterns were based on descriptions provided by Patrini [79]. The formulas can be seen in Equation 4.2, where  $\beth_{C_1, C_2}$  is the probability of changing the label of a sample from one class to another,  $\mathcal{R}$  is the noise rate, and  $k$  is the total number of classes.

$$\begin{aligned}
\aleph_{C_1, C_2}^{\text{I}}(x) &= \frac{1}{2} - \frac{1}{2} \left[ f_{C_1}^*(x) - f_{C_2}^*(x) \right]^2 \\
\aleph_{C_1, C_2}^{\text{II}}(x) &= 1 - \left[ f_{C_1}^*(x) - f_{C_2}^*(x) \right]^3 \\
\aleph_{C_1, C_2}^{\text{III}}(x) &= 1 - \frac{1}{3} \left[ f_{C_1}^*(x) - f_{C_2}^*(x) \right]^3 - \frac{1}{3} \left[ f_{C_1}^*(x) - f_{C_2}^*(x) \right]^2 \\
&\quad - \frac{1}{3} \left[ f_{C_1}^*(x) - f_{C_2}^*(x) \right]
\end{aligned} \tag{4.1}$$

$$\begin{aligned} \mathcal{D}_{C_1, C_2}^{\text{Uniform}} &= \begin{cases} \frac{\mathcal{R}}{k-1} & C_1 \neq C_2 \\ 1 - \mathcal{R} & C_1 = C_2 \end{cases} \\ \mathcal{D}_{C_1, C_2}^{\text{Asymmetrical}} &= \begin{cases} \mathcal{R} & C_1 \neq C_2 \\ 1 - \mathcal{R} & C_1 = C_2 \end{cases} \end{aligned} \quad (4.2)$$

To evaluate the performance of the proposed method in realistic cases, Animal-10N [105], Food-101N [61], and Clothing1M [121] datasets were used. These datasets represent real-world samples with complex noise patterns. In each case, part of the dataset was manually re-labeled to create the clean validation set. ResNet-34, ResNet-50 [44], and VGG-19 [100] network configurations were used to test the proposed method and evaluate its performance against alternatives. In synthetic experiments, 10% of the clean training data was reserved as the validation set, while in benchmark experiments, random samples were re-labeled to create the validation set. The results are presented in Tables 4.2 and 4.3. The proposed approach outperforms the alternatives in most cases, as illustrated in the tables. Moreover, some alternative methods exhibited a high standard deviation rate in specific cases, suggesting potential instability in those approaches.

## Conclusion

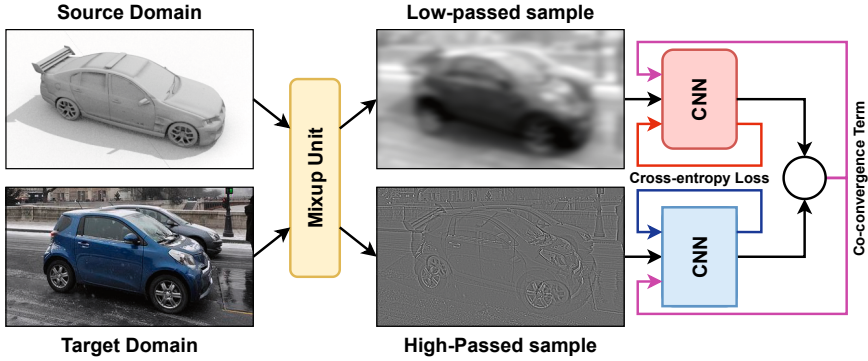
In conclusion, the proposed method enhances the data-recalibration process by incorporating a clean validation set to improve the quality of the trained network. Utilizing existing tools to prepare a small validation set is relatively cheap compared to the increased performance and stability it offers. Multiple experiments were conducted using well-known network configurations (ResNet and VGG), noise patterns (CCN and IDN), and datasets (CIFAR, Animal-10, Food-101N, and Clothing1M), comparing the proposed method with state-of-the-art alternatives. The proposed method has an average of 1.57% higher accuracy in synthetic tests. In benchmark tests, the average improvement is down to 0.06% since the proposed method is beaten by an alternative in Food-101N dataset. However, the proposed method has a slight edge on the alternative when considering both test cases (Animal-10N and Food-101N), in which the proposed method has a 0.14% improvement on average.

Dataset	Noise Info	SL[116]	LRT[136]	PLC[135]	Proposed	
CIFAR-10	$\mathcal{N}_{35\%}^I$	$79.76 \pm 0.7$	$80.98 \pm 0.8$	<u><math>82.80 \pm 0.3</math></u>	<b><math>83.60 \pm 0.3</math></b>	
	$\mathcal{N}_{70\%}^I$	$36.29 \pm 0.7$	$41.52 \pm 4.5$	<u><math>42.74 \pm 2.1</math></u>	<b><math>46.47 \pm 1.1</math></b>	
	$\mathcal{N}_{35\%}^{II}$	$77.92 \pm 0.9$	$80.74 \pm 0.3$	<u><math>81.54 \pm 0.5</math></u>	<b><math>83.41 \pm 0.3</math></b>	
	$\mathcal{N}_{70\%}^{II}$	$41.11 \pm 1.9$	$44.67 \pm 3.9$	<u><math>46.04 \pm 2.2</math></u>	<b><math>46.24 \pm 0.9</math></b>	
	$\mathcal{N}_{35\%}^{III}$	$78.81 \pm 0.3$	$81.08 \pm 0.4$	<u><math>81.50 \pm 0.5</math></u>	<b><math>83.16 \pm 0.3</math></b>	
	$\mathcal{N}_{70\%}^{III}$	$38.49 \pm 1.5$	$44.47 \pm 1.2$	<u><math>45.05 \pm 1.1</math></u>	<b><math>46.33 \pm 1.1</math></b>	
	$\mathcal{N}_{35\%}^I + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$	$77.79 \pm 0.5$	$75.97 \pm 0.3$	<u><math>79.04 \pm 0.5</math></u>	<b><math>80.94 \pm 0.2</math></b>	
	$\mathcal{N}_{35\%}^{II} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$	$77.14 \pm 0.7$	$76.96 \pm 0.5$	<u><math>78.31 \pm 0.4</math></u>	<b><math>79.93 \pm 0.5</math></b>	
	$\mathcal{N}_{35\%}^{III} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$	$75.08 \pm 0.5$	$75.94 \pm 0.6$	<u><math>80.08 \pm 0.4</math></u>	<b><math>81.07 \pm 0.2</math></b>	
	$\mathcal{N}_{35\%}^{II} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$	$75.43 \pm 0.4$	$77.03 \pm 0.6$	<u><math>77.63 \pm 0.3</math></u>	<b><math>79.90 \pm 0.5</math></b>	
	$\mathcal{N}_{35\%}^{III} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$	$76.22 \pm 0.1$	$75.66 \pm 0.6$	<u><math>80.06 \pm 0.5</math></u>	<b><math>80.54 \pm 0.3</math></b>	
	$\mathcal{N}_{35\%}^{III} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$	$76.09 \pm 0.1$	$77.19 \pm 0.7$	<u><math>77.54 \pm 0.7</math></u>	<b><math>79.54 \pm 0.5</math></b>	
	CIFAR-100	$\mathcal{N}_{35\%}^I$	$55.20 \pm 0.3$	$56.74 \pm 0.3$	<u><math>60.01 \pm 0.4</math></u>	<b><math>63.85 \pm 0.3</math></b>
		$\mathcal{N}_{70\%}^I$	$40.02 \pm 0.9$	$45.29 \pm 0.4$	<u><math>45.92 \pm 0.6</math></u>	<b><math>46.38 \pm 0.3</math></b>
$\mathcal{N}_{35\%}^{II}$		$56.10 \pm 0.7$	$57.25 \pm 0.7$	<u><math>63.68 \pm 0.3</math></u>	<b><math>63.91 \pm 0.3</math></b>	
$\mathcal{N}_{70\%}^{II}$		$38.45 \pm 0.6$	$43.71 \pm 0.5$	<u><math>45.03 \pm 0.5</math></u>	<b><math>46.63 \pm 0.2</math></b>	
$\mathcal{N}_{35\%}^{III}$		$56.04 \pm 0.7$	$56.57 \pm 0.3$	<u><math>63.68 \pm 0.3</math></u>	<b><math>63.92 \pm 0.4</math></b>	
$\mathcal{N}_{70\%}^{III}$		$39.94 \pm 0.8$	$44.41 \pm 0.2$	<u><math>44.45 \pm 0.6</math></u>	<b><math>46.22 \pm 0.2</math></b>	
$\mathcal{N}_{70\%}^{III}$		$39.94 \pm 0.8$	$44.41 \pm 0.2$	<u><math>44.45 \pm 0.6</math></u>	<b><math>46.22 \pm 0.2</math></b>	
$\mathcal{N}_{35\%}^I + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$		$51.34 \pm 0.6$	$45.66 \pm 1.6$	<u><math>60.09 \pm 0.2</math></u>	<b><math>61.46 \pm 0.4</math></b>	
$\mathcal{N}_{35\%}^{II} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$		$50.18 \pm 1.0$	$52.04 \pm 0.2$	<u><math>56.40 \pm 0.3</math></u>	<b><math>59.94 \pm 0.4</math></b>	
$\mathcal{N}_{35\%}^{III} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$		$50.58 \pm 0.3$	$43.86 \pm 1.3$	<u><math>60.01 \pm 0.6</math></u>	<b><math>61.16 \pm 0.3</math></b>	
$\mathcal{N}_{35\%}^{II} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$		$49.46 \pm 0.2$	$52.11 \pm 0.5$	<b><math>61.43 \pm 0.3</math></b>	<u><math>59.34 \pm 0.5</math></u>	
$\mathcal{N}_{35\%}^{III} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$		$50.18 \pm 0.5$	$42.79 \pm 1.8$	<u><math>60.14 \pm 1.0</math></u>	<b><math>61.82 \pm 0.3</math></b>	
$\mathcal{N}_{35\%}^{III} + \begin{cases} \text{Uniform} \\ \text{Asymmetrical} \end{cases}_{30\%}$		$48.15 \pm 0.9$	$50.31 \pm 0.4$	<u><math>54.56 \pm 1.1</math></u>	<b><math>59.76 \pm 0.5</math></b>	

**Table 4.2** Accuracy on the CIFAR dataset for different noise patterns and rates. The best accuracy is indicated in bold, and the second best is underlined. Reproduced with permission from publication III.

Dataset	Method	Accuracy	Dataset	Method	Accuracy	Dataset	Method	Accuracy
Animal-10N	SELFIE [105]	79.40	Food-101N	DeepSelf [43]	79.40	Clothing1M	DeepSelf [43]	74.45
	Co-Learning [108]	82.95		PLC [135]	83.40		CleanNet [61]	74.69
	PLC [135]	83.40		Proposed	86.34		DivideMix [62]	74.76
	Proposed	<b>84.47</b>		Co-Learning [108]	<b>87.57</b>		Proposed	<b>75.11</b>

**Table 4.3** Accuracy on the Animal-10N, Food-101N, and Clothing1M datasets. The best accuracy is indicated in bold, and the second best is underlined. Reproduced with permission from publication III.



**Figure 4.5** The structure of the proposed method. Samples from the same category are mixed utilizing low/high pass filters. The resulting data is formed into training datasets for two distinct neural networks. Training is done with categorical cross-entropy loss and a co-convergence term. Reproduced with permission from publication IV.

### 4.3 Domain Adaptation in Visual Classification

As mentioned in the previous chapter, acquiring a sizable and labeled dataset for training deep learning algorithms for practical applications is a significant challenge, especially considering safety regulations. To overcome this challenge, a common approach is transfer learning, where the algorithm is initially trained on a label-rich source dataset (such as synthesized or simulated data) and then fine-tuned using a much smaller target dataset (such as data collected from the real world) [138]. However, the algorithm's performance would be adversely affected if there is a significant discrepancy between the two domains.

Recent studies propose Gradual Domain Adaptation (GDA), which incorporates data from intermediary domains to address the discrepancy between source and target domains [57]. However, these intermediary domains do not naturally exist for most real-world applications. Therefore, scientists propose methods to create intermediary domains to overcome this challenge [90, 2, 74]. This work proposes an iterative domain adaptation algorithm that utilizes filtered images and a variable mixup technique to create intermediate domains artificially. The proposed approach involves iteratively merging the low-pass and high-pass filtered images from the source and target domains based on a dynamic ratio. The resulting domains are then used to train two distinct models in parallel, with the final output calculated based on the

average of these models. The main distinction between the proposed method and previous state-of-the-art approaches is the involvement of a labeled sub-dataset from the target domain to create intermediate domains. Additionally, the mixup technique results in two distinct domains, which train two models with different characteristics. Algorithm 3 summarizes the proposed method in detailed steps, while Figure 4.5 illustrates it.

---

**Algorithm 3:** IFMix: Intermediate Filtered Mixup

---

**Require:** Source dataset  $\mathcal{D}^s$ , Labeled Target subset  $\mathcal{D}_l^t$ , Number of epochs  $T$ , Batch size  $B$ , Warm-up period  $W$ , Mixup ratio  $H_o$ , Mixup increment rate  $\alpha$

- 1: **for**  $t \in 1, \dots, T$  **do**
- 2:   Select samples from same category in  $\mathcal{D}^s$  and  $\mathcal{D}_l^t$  and apply filters.
- 3:   Create intermediate domains:  

$$x_i^{lo} = (1 - H_i) \times \text{LowPass}(x_i^s) + H_i \times \text{LowPass}(x_j^t)$$

$$x_i^{hi} = (1 - H_i) \times \text{HighPass}(x_i^s) + H_i \times \text{HighPass}(x_j^t)$$
- 4:   **for**  $b \in 1, \dots, B$  **do**
- 5:     Update loss functions:  

$$\mathcal{L}_{cce}^{lo} = \frac{1}{B} \sum_i^B y_i^{lo} \times \log(p(y|x_i^{lo}))$$

$$\mathcal{L}_{cce}^{hi} = \frac{1}{B} \sum_i^B y_i^{hi} \times \log(q(y|x_i^{hi}))$$
- 6:     **if**  $i \geq W$  **then**
- 7:       Update co-convergence term:  

$$\mathcal{L}_{cct} = \frac{1}{B} \sum_i^B y_i \times \log\left(\frac{p(y|x_i^{lo}) + q(y|x_i^{hi})}{2}\right)$$
- 8:     **end if**
- 9:   **end for**
- 10:   Update the mixup ratio:  $H_{i+1} = H_i + \alpha \times t$
- 11: **end for**

**return** Two trained networks.

---

The main contributions of this work are twofold. First, an iterative intermediate domain creation technique was proposed based on utilizing filtered images and a labeled subset from the target domain. The cost of preparing the required data is negligible compared to the performance gain. Second, the proposed method was evaluated by conducting extensive tests on public benchmarks and demonstrate the effectiveness of the proposed method over the state-of-the-art alternatives. The experiment was conducted using the Office-31 [88], Office-Home [112], and VisDa-2017 [80] datasets to evaluate the performance of the proposed method against alternatives. The Office-31 and Office-Home datasets were chosen for the feasibility study.

The domains in Office-31 were named  $A$  for images taken from Amazon.com,  $D$  for photos taken with a DSLR camera, and  $W$  for photos taken with a webcam, while the domains in Office-Home were named  $A$  for arts and paintings,  $C$  for clip-art images,  $P$  for product images without a background, and  $R$  for real-world images taken with a camera. The VisDa-2017 dataset was chosen to evaluate the performance on challenging datasets, where the domains were named  $S$  for simulation and  $R$  for real-world. Moreover, ResNet-50 and ResNet-101 [44] network configurations were used to test the proposed method and evaluate its performance against alternatives. In each experiment, 5% of the target domain samples were utilized as the labeled target subsets, while the remaining 95% of samples were reserved as test data. Tables 4.4, 4.5, and 4.6 show the performance of the proposed method compared to alternatives on the Office-31, Office-Home, and VisDa-2017 datasets, respectively. The proposed method outperforms the alternatives in most cases, with the average improvement of 0.2%, 0.8%, and 1.7% in accuracy, respectively.

## Conclusion

In conclusion, the proposed method enhances the domain adaptation process by incorporating a labeled subset from the target domain and high/low pass filters to generate intermediate domains iteratively. Utilizing existing tools to prepare a small labeled subset from the target domain is relatively cheap compared to the increased performance and stability it offers. Multiple experiments were conducted using well-known network configurations (ResNet) and datasets (Office-31, Office-Home, and VisDa-2017), comparing the proposed method with state-of-the-art alternatives. The proposed method improved the accuracy by an average of 0.2 - 1.7% in different tests.

Method	$A \rightarrow D$	$A \rightarrow W$	$D \rightarrow A$	$D \rightarrow W$	$W \rightarrow A$	$W \rightarrow D$	Average
GSDA [46]	94.8	95.7	73.5	99.1	74.9	100	89.7
SRDC [109]	95.8	95.7	76.7	<u>99.2</u>	77.1	100	90.8
RSDA [41]	95.8	96.1	77.4	<b>99.3</b>	78.9	100	91.1
FixBi [74]	95	96.1	<b>78.7</b>	<b>99.3</b>	<u>79.4</u>	100	91.4
CoVi [73]	<b>98</b>	<b>97.6</b>	77.5	<b>99.3</b>	78.4	100	<u>91.8</u>
Proposed	<u>97.6</u>	<u>97.5</u>	<u>77.9</u>	<b>99.3</b>	<b>79.7</b>	100	<b>92</b>

**Table 4.4** Accuracy on the Office-31 dataset. The best accuracy is indicated in bold, and the second best is underlined. Reproduced with permission from publication IV.

Method	$A \rightarrow C$	$A \rightarrow P$	$A \rightarrow R$	$C \rightarrow A$	$C \rightarrow P$	$C \rightarrow R$	$P \rightarrow A$	$P \rightarrow C$	$P \rightarrow R$	$R \rightarrow A$	$R \rightarrow C$	$R \rightarrow P$	Average
MetaAlign [117]	59.3	76	80.2	65.7	74.7	75.1	65.7	56.5	81.6	74.1	61.1	85.2	71.3
FixBi [74]	58.1	<u>77.3</u>	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
CoVi [73]	58.5	78.1	80	68.1	80	77	66.4	60.2	82.1	76.6	<b>63.6</b>	86.5	73.1
CDTrans [124]	60.6	79.5	82.4	75.6	81	82.3	72.5	56.7	84.4	77	59.1	85.5	74.7
WinTR [68]	<u>65.3</u>	<b>84.1</b>	85	<u>76.8</u>	<b>84.5</b>	<u>84.4</u>	<u>73.4</u>	<u>60</u>	<u>85.7</u>	<u>77.2</u>	<u>63.1</u>	<u>86.8</u>	<u>77.2</u>
Proposed	<b>66.1</b>	<b>84</b>	<b>86.6</b>	<b>77.4</b>	<u>84.1</u>	<b>86.1</b>	<b>75.2</b>	<b>61.1</b>	<b>86.5</b>	<b>78.4</b>	62.8	<b>87.4</b>	<b>78</b>

**Table 4.5** Accuracy on the Office-Home dataset. The best accuracy is indicated in bold, and the second best is underlined. Reproduced with permission from publication IV.

Method	Plane	Bike	Bus	Car	Horse	Knife	Motor	Human	Plant	Skate	Train	Truck	Average
CAN [52]	97	87.2	82.5	74.3	97.8	<u>96.2</u>	90.8	80.7	96.6	96.3	87.5	59.9	87.2
FixBi [74]	96.1	87.8	90.5	90.3	96.8	95.3	92.8	<b>88.7</b>	97.2	<u>94.2</u>	90.9	25.7	87.2
CDTrans [124]	97.1	90.5	82.4	77.5	96.6	96.1	93.6	<u>88.6</u>	<u>97.9</u>	86.9	90.3	<b>62.8</b>	88.4
CoVi [73]	96.8	85.6	88.9	88.6	97.8	93.4	91.9	87.6	96	93.8	93.6	48.1	88.5
WinTR [68]	<b>98.7</b>	<u>91.2</u>	<b>93</b>	<u>91.9</u>	<u>98.1</u>	96.1	<b>94</b>	72.7	97	<b>95.5</b>	<b>95.3</b>	57.9	<u>90.1</u>
Proposed	<u>98.2</u>	<b>91.7</b>	<u>92.9</u>	<b>92.2</b>	<b>98.5</b>	<b>96.5</b>	<u>93.7</u>	88	<b>98</b>	<b>95.5</b>	<u>94.8</u>	<u>61.8</u>	<b>91.8</b>

**Table 4.6** Accuracy on the VisDa-2017 dataset. The best accuracy is indicated in bold, and the second best is underlined. Reproduced with permission from publication IV.

## 4.4 Discussion

This chapter explored three different mitigation methods for data-related safety concerns. First, the Selective Probabilistic Classifier Based on Hypothesis Testing was introduced, which utilized a hypothesis testing approach to reject uncertain samples based on their probability distribution. The proposed method demonstrated superior performance compared to traditional classifiers in handling out-of-distribution and distorted samples. Next, Enhanced Data-Recalibration was discussed, which utilized a clean validation set to identify and mitigate instance-dependent noise in classification tasks. The proposed method demonstrated promising results in improving the robustness and reliability of classification models in the presence of noisy labels. Finally, IFMix was introduced, which utilized a small labeled subset from the target domain and high/low pass filters to generate intermediate domains iteratively for domain adaptation. The proposed method demonstrated better performance compared to alternative methods.



## 5 CONCLUSIONS

Modern visual classifiers present a significant challenge to existing safety standards owing to their complexity and the conventional approaches used to define software in established standards. This dissertation has addressed two crucial safety aspects in visual classifiers: fault identification and mitigation. First, the faults in visual deep neural classifiers were systematically identified and categorized as safety concerns. Then, the efficacy and limitations of current mitigation methods were evaluated. After that, potential mitigation methods for each safety concern were proposed with a focus on data-related safety concerns, such as noisy labels, outlier data, and domain gaps. The following holds the detailed answers to the research questions posed in this study:

**Research Question 1:** Which faults could result in visual classification systems failing, and how can they be systematically categorized?

The faults in a visual deep neural classifier and their underlying causation was investigated. The research methodology involved systematically decomposing a visual deep neural classifier into three key phases: training, evaluation, and inference. Through a comprehensive analysis of each phase, the faults commonly associated with them were identified. Subsequently, eight distinct safety concerns were formulated based on the impact of each identified fault on the overall system.

**Research Question 2:** What are the existing mitigation methods for dealing with safety concerns in visual classification systems, and how effective are they?

An overview of existing mitigation methods for each identified safety concern was provided with a critical evaluation of their effectiveness and limitations. It is important to note that assigning a definitive safety level to each mitigation method is

not feasible due to the absence of appropriate standardization. However, a suitable combination of meticulously crafted safety-case questions and appropriate metrics can aid in building a compelling case for the system’s overall safety. Additionally, potential mitigation methods for each safety concern was presented, with the intend of mitigating the limitations of individual methods as much as possible.

**Research Question 3:** How can the existing mitigation methods for data-related faults be improved in visual classification systems to ensure practical implementation?

A selective classification method was proposed based on probabilistic neural networks and statistical significance tests to estimate the uncertainty of the classifier, which can serve as a metric for rejecting uncertain outcomes. To evaluate the efficacy of the proposed method, several experiments were conducted using a well-known network configuration (ResNet), benchmark datasets (COCO and CIFAR), and various synthetic disturbances. The performance of the proposed method was compared against the traditional Softmax Response method, using the Area Under the Receiver Operating Characteristic curve as a threshold-independent metric. The experimental results demonstrated that the proposed method outperformed the traditional method while offering a wider range of trade-off options between accuracy and uncertainty.

Furthermore, an iterative data-recalibration method was proposed based on utilizing a small clean validation dataset to iteratively cleanse the noisy labels from the training dataset for visual classifiers. To evaluate the effectiveness of the proposed method, several experiments were conducted using well-known network configurations (ResNet and VGG), benchmark datasets (CIFAR, Animal-10N, Food-101N, Clothing-1M), and different noise models (instance dependent and independent). The performance of the proposed method was compared against state-of-the-art algorithms based on accuracy. The experimental results indicated that the proposed method offers a robust solution and a significant improvement over existing alternatives at a negligible cost of manually cleaning a small validation dataset.

Finally, an iterative intermediate domain generation method was proposed based on utilizing a small clean subset from the target domain, low-/high-pass filters, and a dynamic mixing ratio to iteratively bridge the gap between the source and target domains in visual classifiers. To evaluate the effectiveness of the proposed

method, several experiments were conducted using a well-known network configuration (ResNet) and benchmark datasets (Office-31, Office-Home, and VisDa-2017). The performance of the proposed method was compared against state-of-the-art algorithms based on accuracy. The experimental results indicated that the proposed method offers a robust solution and a significant improvement over existing alternatives at a negligible cost of manually labeling a small portion of the target domain.

Throughout this work, the focus was on simplifying complicated and unnecessary assumptions to reflect the practical use cases in safety-critical applications. Furthermore, the trade-off between cost and safety was considered and negligible costs that favored higher performance were embraced in the proposed approaches wherever possible.

## Future Work

The utilization of AI in safety-critical applications is swiftly expanding, and the challenges of using deep neural classifiers in this domain have been explored in this thesis. These challenges have been under review for a significant duration, and notable headway has been made with every new research outcome. To further advance the current state of research, some potential areas for future investigation are proposed.

The use of AI in safety-critical applications is gaining prominence and is increasingly acknowledged in emerging standards for functional safety in industries such as the automotive sector. While practical solutions may take time to develop, the demand for autonomy and the growing importance of AI reinforces the need for continued research in this area. In this context, exploring the potential of safety-case arguments and safety concern lists for using AI in various fields may prove beneficial. Further research can also investigate the development of standardized approaches for assigning safety levels to mitigation methods and the identification of novel mitigation strategies that address the limitations of current techniques. Finally, research can focus on developing methods that consider AI systems' uncertainty and generalization capabilities in safety-critical applications to ensure their reliability and robustness.

While publication I proposed a successful selective classification method, the testing was limited to synthetic data, and the comparison was made with simple baseline methods. Future research should focus on evaluating the performance of the pro-

posed method with real-world data to identify a more sophisticated approach for enhancing the performance of visual classifiers. Furthermore, exploring the scalability of the proposed method for large datasets and complex network architectures should also be a focus for future research.

Revisiting the mathematical foundation of mitigating label noise in visual classifiers is imperative due to recent discoveries about the nature of label noise. Although Publication III has provided a comprehensive approach using synthetic and real-world data, further exploration is needed to expand and adapt the current mathematical framework. New insights and discoveries can be integrated to advance the field of label noise mitigation and provide more accurate and reliable visual classifiers.

## REFERENCES

- [1] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. DOI: 10.48550/arXiv.1603.04467.
- [2] Samira Abnar et al. *Gradual Domain Adaptation in the Wild: When Intermediate Distributions are Absent*. 2021. DOI: 10.48550/arXiv.2106.06080.
- [3] Abdullah Abuolaim et al. “NTIRE 2021 Challenge for Defocus Deblurring Using Dual-pixel Images: Methods and Results”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2021, pp. 578–587. DOI: 10.1109/CVPRW53098.2021.00070.
- [4] Bishwo Adhikari and Heikki Huttunen. “Iterative Bounding Box Annotation for Object Detection”. In: *International Conference on Pattern Recognition (ICPR)*. 2021, pp. 4040–4046. DOI: 10.1109/ICPR48806.2021.9412956.
- [5] Bishwo Adhikari et al. “Effect of Label Noise on Robustness of Deep Neural Network Object Detectors”. In: *International Workshop on Artificial Intelligence Safety Engineering (WAISE)*. 2021, pp. 239–250. DOI: 10.1007/978-3-030-83906-2\_19.
- [6] Alankrita Aggarwal, Mamta Mittal, and Gopi Battineni. “Generative Adversarial Network: An Overview of Theory and Applications”. In: *International Journal of Information Management Data Insights* 1.1 (2021). DOI: 10.1016/j.jjimei.2020.100004.
- [7] Jarmo Alanen, Marita Hietikko, and Timo Malm. *Safety of Digital Communications in Machines*. VTT Technical Research Centre of Finland, 2004.
- [8] Görkem Algan and Ilkay Ulusoy. “Image Classification with Deep Learning in the Presence of Noisy Labels: A Survey”. In: *Knowledge-Based Systems* 215 (2021). DOI: 10.1016/j.knosys.2021.106771.

- [9] Neena Aloysius and M. Geetha. “A Review on Deep Convolutional Neural Networks”. In: *International Conference on Communication and Signal Processing (ICCSP)*. 2017, pp. 588–592. DOI: 10.1109/ICCSP.2017.8286426.
- [10] Vincent Aravantinos and Peter Schlicht. “Making the Relationship Between Uncertainty Estimation and Safety Less Uncertain”. In: *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2020, pp. 1139–1144. DOI: 10.23919/DATE48585.2020.9116541.
- [11] Eric Arazo et al. *Unsupervised Label Noise Modeling and Loss Correction*. 2019. DOI: 10.48550/arXiv.1904.11238.
- [12] *Artificial Intelligence - Functional Safety and AI Systems*. en. Standard. International Organization for Standardization, 2024.
- [13] Sebastian Bach et al. “On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation”. In: *PLOS ONE* 10.7 (2015), pp. 1–46. DOI: 10.1371/journal.pone.0130140.
- [14] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. “Multi-modal Machine Learning: A Survey and Taxonomy”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (2019), pp. 423–443. DOI: 10.1109/TPAMI.2018.2798607.
- [15] Abhijit Bendale and Terrance Boulton. “Towards Open World Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1893–1902. DOI: 10.1109/CVPR.2015.7298799.
- [16] Abhijit Bendale and Terrance E. Boulton. “Towards Open Set Deep Networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1563–1572. DOI: 10.1109/CVPR.2016.173.
- [17] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2016.
- [18] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. “Food-101 - Mining Discriminative Components with Random Forests”. In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 446–461. DOI: 10.1007/978-3-319-10599-4\_29.

- [19] Jens Braband and Hendrik Schäbe. “On safety assessment of artificial intelligence”. In: *Dependability* 20.4 (2020), pp. 25–34. DOI: 10.21683/1729-2646-2020-20-4-25-34.
- [20] Henrique César de Lima Araújo et al. “Artificial Intelligence in Urban Forestry — A Systematic Review”. In: *Urban Forestry & Urban Greening* 66 (2021). DOI: 10.1016/j.ufug.2021.127410.
- [21] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 2010.
- [22] Jiefeng Chen et al. *Robust Out-of-Distribution Detection for Neural Networks*. 2021. DOI: 10.48550/arXiv.2003.09711.
- [23] Pengfei Chen et al. “Beyond Class-Conditional Assumption: A Primary Attempt to Combat Instance-Dependent Label Noise”. In: *AAAI Conference on Artificial Intelligence* 35.13 (2021), pp. 11442–11450. DOI: 10.1609/aaai.v35i13.17363.
- [24] O Ciftcioglu and E Turkcan. *Data Fusion and Sensor Management for Nuclear Power Plant Safety*. 1996.
- [25] Susan D. Cochran. *Two sample z-tests*. URL: <http://www.stat.ucla.edu/~cochran/stat10/winter/lectures/lect21.html>.
- [26] Filipe R. Cordeiro and Gustavo Carneiro. “A Survey on Deep Learning with Noisy Labels: How to Train Your Model when You Cannot Trust on the Annotations?” In: *SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. 2020, pp. 9–16. DOI: 10.1109/SIBGRAPI51738.2020.00010.
- [27] *Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006 on Machinery, and Amending Directive 95/16/EC*. en. Directive 2006/42/EC. European Union, 2006.
- [28] *Earth-Moving Machinery — Functional Safety*. en. Standard. International Organization for Standardization, 2008.
- [29] Jesper E. van Engelen and Holger H. Hoos. “A Survey on Semi-Supervised Learning”. In: *Machine Learning* 109.2 (2020), pp. 373–440. DOI: 10.1007/s10994-019-05855-6.
- [30] *Evaluating Functional Safety in Automotive Image Sensors*. en. Technical report. ON Semiconductor, 2018.

- [31] *Evaluation of Autonomous Products*. en. Standard. UL Standards & Engagement, 2022.
- [32] Linwei Fan et al. “Brief Review of Image Denoising Techniques”. In: *Visual Computing for Industry, Biomedicine, and Art 2.1* (2019). DOI: 10.1186/s42492-019-0016-7.
- [33] Abolfazl Farahani et al. “A Brief Review of Domain Adaptation”. In: *Advances in Data Science and Information Engineering*. 2021, pp. 877–894. DOI: 10.1007/978-3-030-71704-9\_65.
- [34] Benoit Frenay and Michel Verleysen. “Classification in the Presence of Label Noise: A Survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 25.5 (2014), pp. 845–869. DOI: 10.1109/TNNLS.2013.2292894.
- [35] *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*. en. Standard. International Electrotechnical Commission, 2010.
- [36] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *International Conference on Machine Learning (ICML)*. 2016, pp. 1050–1059. DOI: 10.5555/3045390.3045502.
- [37] Yonatan Geifman and Ran El-Yaniv. “Selective Classification for Deep Neural Networks”. In: *International Conference on Neural Information Processing Systems (NIPS)*. 2017, pp. 4885–4894. DOI: 10.5555/3295222.3295241.
- [38] Mohamad Gharib and Andrea Bondavalli. “On the Evaluation Measures for Machine Learning Algorithms for Safety-Critical Systems”. In: *European Dependable Computing Conference (EDCC)*. 2019, pp. 141–144. DOI: 10.1109/EDCC.2019.00035.
- [39] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2014. DOI: 10.48550/ARXIV.1412.6572.
- [40] Bhawna Goyal et al. “Image Denoising Review: From Classical to State-of-the-Art Approaches”. In: *Information Fusion* 55 (2020), pp. 220–244. DOI: 10.1016/j.inffus.2019.09.003.



- [41] Xiang Gu, Jian Sun, and Zongben Xu. “Spherical Space Domain Adaptation With Robust Pseudo-Label Loss”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 9098–9107. DOI: 10.1109/CVPR42600.2020.00912.
- [42] Wenzhong Guo, Jianwen Wang, and Shiping Wang. “Deep Multimodal Representation Learning: A Survey”. In: *IEEE Access* 7 (2019), pp. 63373–63394. DOI: 10.1109/ACCESS.2019.2916887.
- [43] Jiangfan Han, Ping Luo, and Xiaogang Wang. “Deep Self-Learning From Noisy Labels”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 5137–5146. DOI: 10.1109/ICCV.2019.00524.
- [44] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
- [45] Dan Hendrycks et al. “Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty”. In: *International Conference on Neural Information Processing Systems (NIPS)*. 2019, pp. 15663–15674. DOI: 10.5555/3454287.3455690.
- [46] Lanqing Hu et al. “Unsupervised Domain Adaptation With Hierarchical Gradient Synchronization”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4042–4051. DOI: 10.1109/CVPR42600.2020.00410.
- [47] Gao Huang et al. “Densely Connected Convolutional Networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.
- [48] Frank Hutter, Jörg Lücke, and Lars Schmidt-Thieme. “Beyond Manual Tuning of Hyperparameters”. In: *Künstliche Intelligenz* 29.4 (2015), pp. 329–337. DOI: 10.1007/s13218-015-0381-0.
- [49] *Information Technology — Artificial Intelligence — Assessment of Machine Learning Classification Performance*. en. Standard. International Organization for Standardization, 2022.

- [50] Rusul Sabah Jebur, Chen Soong Der, and Dalal Abdulmohsin Hammood. “A Review and Taxonomy of Image Denoising Techniques”. In: *International Conference on Interactive Digital Media (ICIDM)*. 2020, pp. 1–6. DOI: 10.1109/ICIDM51048.2020.9339674.
- [51] Clemens Kamann and Carsten Rother. “Benchmarking the Robustness of Semantic Segmentation Models with Respect to Common Corruptions”. In: *International Journal of Computer Vision* 129.2 (2021), pp. 462–483. DOI: 10.1007/s11263-020-01383-2.
- [52] Guoliang Kang et al. “Contrastive Adaptation Network for Unsupervised Domain Adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4888–4897. DOI: 10.1109/CVPR.2019.00503.
- [53] Michael Kläs and Lisa Jöckel. “A Framework for Building Uncertainty Wrappers for AI/ML-Based Data-Driven Components”. In: *International Workshop on Artificial Intelligence Safety Engineering (WAISE)*. 2020, pp. 315–327. DOI: 10.1007/978-3-030-55583-2\_23.
- [54] Gunnar König et al. “Relative Feature Importance”. In: *International Conference on Pattern Recognition (ICPR)*. 2021, pp. 9318–9325. DOI: 10.1109/ICPR48806.2021.9413090.
- [55] Alex Krizhevsky and Geoffrey Hinton. *Learning Multiple Layers of Features from Tiny Images*. 2009.
- [56] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90. DOI: 10.1145/3065386.
- [57] Ananya Kumar, Tengyu Ma, and Perc Liang. “Understanding Self-Training for Gradual Domain Adaptation”. In: *International Conference on Machine Learning (ICML)*. Vol. 119. 2020, pp. 5468–5479. DOI: 10.5555/3524938.3525445.
- [58] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. *Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles*. 2016. DOI: 10.48550/ARXIV.1612.01474.

- [59] John Lambert et al. “MSeg: A Composite Dataset for Multi-Domain Semantic Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2023), pp. 796–810. DOI: 10.1109/TPAMI.2022.3151200.
- [60] Y. Lecun et al. “Gradient-Based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.
- [61] Kuang-Huei Lee et al. “CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 5447–5456. DOI: 10.1109/CVPR.2018.00571.
- [62] Junnan Li, Richard Socher, and Steven C. H. Hoi. *DivideMix: Learning with Noisy Labels as Semi-Supervised Learning*. 2020. DOI: 10.48550/ARXIV.2002.07394.
- [63] Junnan Li et al. “Learning to Learn From Noisy Labeled Data”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5046–5054. DOI: 10.1109/CVPR.2019.00519.
- [64] Yingming Li, Ming Yang, and Zhongfei Zhang. “A Survey of Multi-View Representation Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 31.10 (2019), pp. 1863–1883. DOI: 10.1109/TKDE.2018.2872063.
- [65] Shiyu Liang, Yixuan Li, and R. Srikant. *Enhancing The Reliability of Out-of-Distribution Image Detection in Neural Networks*. 2017. DOI: 10.48550/ARXIV.1706.02690.
- [66] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1\_48.
- [67] Gang Luo. “A Review of Automatic Selection Methods for Machine Learning Algorithms and Hyper-Parameter Values”. In: *Network Modeling Analysis in Health Informatics and Bioinformatics* 5.1 (2016). DOI: 10.1007/s13721-016-0125-6.
- [68] Wenxuan Ma et al. *Exploiting Both Domain-Specific and Invariant Knowledge via a Win-win Transformer for Unsupervised Domain Adaptation*. 2021. DOI: 10.48550/arXiv.2111.12941.

- [69] Xingjun Ma et al. “Normalized Loss Functions for Deep Learning with Noisy Labels”. In: *International Conference on Machine Learning (ICML)*. Vol. 119. 2020, pp. 6543–6553. DOI: 10.5555/3524938.3525545.
- [70] T. Soni Madhulatha. *An Overview on Clustering Methods*. 2012. DOI: 10.48550/arXiv.1205.1117.
- [71] Tom M. Mitchell. *Machine Learning*. MacGraw-Hill, 1997.
- [72] Behshad Mohebbi et al. “Probabilistic Neural Networks”. In: *Handbook of Probabilistic Models*. 2020, pp. 347–367. DOI: 10.1016/B978-0-12-816514-0.00014-X.
- [73] Jaemin Na et al. “Contrastive Vicinal Space for Unsupervised Domain Adaptation”. In: *European Conference on Computer Vision (ECCV)*. 2022, pp. 92–110. DOI: 10.1007/978-3-031-19830-4\_6.
- [74] Jaemin Na et al. “FixBi: Bridging Domain Spaces for Unsupervised Domain Adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 1094–1103. DOI: 10.1109/CVPR46437.2021.00115.
- [75] Seungjun Nah et al. “NTIRE 2021 Challenge on Image Deblurring”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2021, pp. 149–165. DOI: 10.1109/CVPRW53098.2021.00025.
- [76] Sergey I Nikolenko. *Synthetic Data for Deep Learning*. Springer, 2021.
- [77] Jitendra Parmar et al. “Open-World Machine Learning: Applications, Challenges, and Opportunities”. In: *ACM Computing Surveys* 55.10 (2023), pp. 1–37. DOI: 10.1145/3561381.
- [78] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *International Conference on Neural Information Processing Systems (NIPS)*. 2019, pp. 8026–8037. DOI: 10.5555/3454287.3455008.
- [79] Giorgio Patrini et al. “Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2233–2241. DOI: 10.1109/CVPR.2017.240.
- [80] Xingchao Peng et al. *VisDA: The Visual Domain Adaptation Challenge*. 2017. DOI: 10.48550/arXiv.1710.06924.

- [81] Ridho Putra, Tito Purboyo, and Anggunmeka Prasasti. “A Review of Image Enhancement Methods”. In: *International Journal of Applied Engineering Research* 12.23 (2017), pp. 13596–13603. DOI: 10.1109/JPROC.2020.3004555.
- [82] Trivellore E. Raghunathan. “Synthetic Data”. In: *Annual Review of Statistics and Its Application* 8.1 (2021), pp. 129–140. DOI: 10.1146/annurev-statistics-040720-031848.
- [83] Ratheesh Ravindran, Michael J. Santora, and Mohsin M. Jamali. “Multi-Object Detection and Tracking, Based on DNN, for Autonomous Vehicles: A Review”. In: *IEEE Sensors Journal* 21.5 (2021), pp. 5668–5677. DOI: 10.1109/JSEN.2020.3041615.
- [84] Pengzhen Ren et al. “A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions”. In: *ACM Computing Surveys* 54.4 (2021), pp. 1–34. DOI: 10.1145/3447582.
- [85] *Road Vehicles — Functional Safety*. en. Standard. International Organization for Standardization, 2018.
- [86] *Road vehicles — Safety of the Intended Functionality*. en. Standard. International Organization for Standardization, 2022.
- [87] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.
- [88] Kate Saenko et al. “Adapting Visual Category Models to New Domains”. In: *European Conference on Computer Vision (ECCV)*. 2010, pp. 213–226. DOI: 10.1007/978-3-642-15561-1\_16.
- [89] *Safety of Machinery — Safety-Related Parts of Control Systems*. en. Standard. International Organization for Standardization, 2006.
- [90] Shogo Sagawa and Hideitsu Hino. *Gradual Domain Adaptation via Normalizing Flows*. 2022. DOI: 10.48550/arXiv.2206.11492.
- [91] Tara Salman et al. “Safety Score as an Evaluation Metric for Machine Learning Models of Security Applications”. In: *IEEE Networking Letters* 2.4 (2020), pp. 207–211. DOI: 10.1109/LNET.2020.3016583.

- [92] Chandramouli Shama Sastry and Sageev Oore. “Detecting Out-of-Distribution Examples with Gram Matrices”. In: *International Conference on Machine Learning (ICML)*. 2020, pp. 8491–8501. DOI: 10.5555/3524938.3525725.
- [93] Lars Schmarje et al. “A Survey on Semi-, Self- and Unsupervised Learning for Image Classification”. In: *IEEE Access* 9 (2021), pp. 82146–82168. DOI: 10.1109/ACCESS.2021.3084358.
- [94] Hannes Schulz and Sven Behnke. “Deep Learning”. In: *KI - Künstliche Intelligenz* 26.4 (2012), pp. 357–363. DOI: 10.1007/s13218-012-0198-z.
- [95] Adrian Schwaiger et al. “Is Uncertainty Quantification in Deep Learning Sufficient for Out-of-Distribution Detection?” In: *Workshop on Artificial Intelligence Safety*. 2020.
- [96] Gesina Schwalbe et al. “Structuring the Safety Argumentation for Deep Neural Network Based Perception in Automotive Applications”. In: *International Workshop on Artificial Intelligence Safety Engineering (WAISE)*. 2020, pp. 383–394. DOI: 10.1007/978-3-030-55583-2\_29.
- [97] Jialin Shi and Ji Wu. *Distilling Effective Supervision for Robust Medical Image Segmentation with Noisy Labels*. 2021. DOI: 10.48550/ARXIV.2106.11099.
- [98] Connor Shorten and Taghi M Khoshgoftaar. “A Survey on Image Data Augmentation for Deep Learning”. In: *Journal of Big Data* 6.1 (2019), pp. 1–48. DOI: 10.1186/s40537-019-0197-0.
- [99] Lei Shu, Hu Xu, and Bing Liu. “DOC: Deep Open Classification of Text Documents”. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2017, pp. 2911–2916. DOI: 10.18653/v1/D17-1314.
- [100] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. DOI: 10.48550/ARXIV.1409.1556.
- [101] Ankita Singh and Pawan Singh. “Image Classification: A Survey”. In: *Journal of Informatics Electrical and Electronics Engineering (JIEEE)* 1.2 (2020), pp. 1–9. DOI: 10.54060/JIEEE/001.02.002.
- [102] J. R. Sklaroff. “Redundancy Management Technique for Space Shuttle Computers”. In: *IBM Journal of Research and Development* 20.1 (1976), pp. 20–28. DOI: 10.1147/rd.201.0020.

- [103] Juraj Slačka and Miroslav Halás. “Safety Critical RTOS for Space Satellites”. In: *International Conference on Process Control (PC)*. 2015, pp. 250–254. DOI: 10.1109/PC.2015.7169971.
- [104] Ian Sommerville. *Software Engineering*. Addison-Wesley, 2010.
- [105] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. “SELFIE: Refurbishing Unclean Samples for Robust Deep Learning”. In: *International Conference on Machine Learning (ICML)*. 2019, pp. 5907–5915.
- [106] Hwanjun Song et al. “Learning From Noisy Labels With Deep Neural Networks: A Survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 34.11 (2023), pp. 8135–8153. DOI: 10.1109/tnnls.2022.3152527.
- [107] Kenji Suzuki. “Overview of Deep Learning in Medical Imaging”. In: *Radiological Physics and Technology* 10.3 (2017), pp. 257–273. DOI: 10.1007/s12194-017-0406-5.
- [108] Cheng Tan et al. “Co-Learning: Learning from Noisy Labels with Self-Supervision”. In: *ACM International Conference on Multimedia*. 2021, pp. 1405–1413. DOI: 10.1145/3474085.3475622.
- [109] Hui Tang, Ke Chen, and Kui Jia. “Unsupervised Domain Adaptation via Structurally Regularized Deep Clustering”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 8722–8732. DOI: 10.1109/CVPR42600.2020.00875.
- [110] *Tractors and Machinery for Agriculture and Forestry — Safety-Related Parts of Control Systems*. en. Standard. International Organization for Standardization, 2010.
- [111] Andreas Veit et al. “Learning From Noisy Large-Scale Datasets With Minimal Supervision”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6575–6583. DOI: 10.1109/CVPR.2017.696.
- [112] Hemanth Venkateswara et al. “Deep Hashing Network for Unsupervised Domain Adaptation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5385–5394. DOI: 10.1109/CVPR.2017.572.
- [113] Apoorv Vyas et al. “Out-of-Distribution Detection Using an Ensemble of Self Supervised Leave-out Classifiers”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 560–574. DOI: 10.1007/978-3-030-01237-3\_34.

- [114] Fei-Yue Wang. “Toward a Revolution in Transportation Operations: AI for Complex Systems”. In: *IEEE Intelligent Systems* 23.6 (2008), pp. 8–13. DOI: 10.1109/MIS.2008.112.
- [115] Xiang Wang, Kai Wang, and Shiguo Lian. “A Survey on Face Data Augmentation for the Training of Deep Neural Networks”. In: *Neural Computing and Applications* 32.19 (2020), pp. 15503–15531. DOI: 10.1007/s00521-020-04748-3.
- [116] Yisen Wang et al. “Symmetric Cross Entropy for Robust Learning With Noisy Labels”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 322–330. DOI: 10.1109/ICCV.2019.00041.
- [117] Guoqiang Wei et al. “MetaAlign: Coordinating Domain Alignment and Classification for Unsupervised Domain Adaptation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 16638–16648. DOI: 10.1109/CVPR46437.2021.01637.
- [118] Oliver Willers et al. “Safety Concerns and Mitigation Approaches Regarding the Use of Deep Learning in Safety-Critical Perception Tasks”. In: *International Workshop on Artificial Intelligence Safety Engineering (WAISE)*. 2020, pp. 336–350. DOI: 10.1007/978-3-030-55583-2\_29.
- [119] Martin Wistuba, Ambrish Rawat, and Tejaswini Pedapati. *A Survey on Neural Architecture Search*. 2019. DOI: 10.48550/arXiv.1905.01392.
- [120] Ernest Wozniak et al. “A Safety Case Pattern for Systems with Machine Learning Components”. In: *International Workshop on Artificial Intelligence Safety Engineering (WAISE)*. 2020, pp. 370–382. DOI: 10.1007/978-3-030-55583-2\_28.
- [121] Tong Xiao et al. “Learning From Massive Noisy Labeled Data for Image Classification”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2691–2699. DOI: 10.1109/CVPR.2015.7298885.
- [122] Guibiao Xu, Bao-Gang Hu, and Jose C. Principe. “Robust C-Loss Kernel Classifiers”. In: *IEEE Transactions on Neural Networks and Learning Systems* 29.3 (2018), pp. 510–522. DOI: 10.1109/TNNLS.2016.2637351.



- [123] Han Xu et al. “Adversarial Attacks and Defenses in Images, Graphs and Text: A Review”. In: *International Journal of Automation and Computing* 17.2 (2020), pp. 151–178. DOI: 10.1007/s11633-019-1211-x.
- [124] Tongkun Xu et al. *CDTrans: Cross-Domain Transformer for Unsupervised Domain Adaptation*. 2021. DOI: 110.48550/arXiv.2109.06165.
- [125] Ruixin Yang and Yingyan Yu. “Artificial Convolutional Neural Network in Object Detection and Semantic Segmentation for Medical Imaging Analysis”. In: *Frontiers in Oncology* 11 (2021). DOI: 10.3389/fonc.2021.638182.
- [126] Qing Yu and Kiyoharu Aizawa. “Unsupervised Out-of-Distribution Detection by Maximum Classifier Discrepancy”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9517–9525. DOI: 10.1109/ICCV.2019.00961.
- [127] Tong Yu and Hong Zhu. *Hyper-Parameter Optimization: A Review of Algorithms and Applications*. 2020. DOI: 10.48550/arXiv.2003.05689.
- [128] Xiaoyong Yuan et al. “Adversarial Examples: Attacks and Defenses for Deep Learning”. In: *IEEE Transactions on Neural Networks and Learning Systems* 30.9 (2019), pp. 2805–2824. DOI: 10.1109/TNNLS.2018.2886017.
- [129] Oliver Zendel et al. “WildDash - Creating Hazard-Aware Benchmarks”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 407–421. DOI: 10.1007/978-3-030-01231-1\_25.
- [130] Junhai Zhai et al. “Autoencoder and Its Various Variants”. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2018, pp. 415–419. DOI: 10.1109/SMC.2018.00080.
- [131] Chiyuan Zhang et al. “Understanding Deep Learning (Still) Requires Rethinking Generalization”. In: *Communications of the ACM* 64.3 (2021), pp. 107–115. DOI: 10.1145/3446776.
- [132] Daokun Zhang et al. “Network Representation Learning: A Survey”. In: *IEEE Transactions on Big Data* 6.1 (2020), pp. 3–28. DOI: 10.1109/TBDAT A.2018.2850013.
- [133] Kaihao Zhang et al. “Deep Image Deblurring: A Survey”. In: *International Journal of Computer Vision* 130.9 (2022), pp. 2103–2130. DOI: 10.1007/s11263-022-01633-5.

- [134] Xu-Yao Zhang, Cheng-Lin Liu, and Ching Y. Suen. “Towards Robust Pattern Recognition: A Review”. In: *Proceedings of the IEEE* 108.6 (2020), pp. 894–922. DOI: 10.1109/JPROC.2020.2989782.
- [135] Yikai Zhang et al. *Learning with Feature-Dependent Label Noise: A Progressive Approach*. 2021. DOI: 10.48550/ARXIV.2103.07756.
- [136] Songzhu Zheng et al. “Error-Bounded Correction of Noisy Labels”. In: *International Conference on Machine Learning (ICML)*. 2020, pp. 11447–11457. DOI: 10.5555/3524938.3525999.
- [137] Jianlong Zhou et al. “Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics”. In: *Electronics* 10.5 (2021). DOI: 10.3390/electronics10050593.
- [138] Fuzhen Zhuang et al. “A Comprehensive Survey on Transfer Learning”. In: *Proceedings of the IEEE* 109.1 (2021), pp. 43–76. DOI: 10.1109/JPROC.2020.3004555.

## PUBLICATIONS



# PUBLICATION

|

## **Selective Probabilistic Classifier Based on Hypothesis Testing**

Saeed Bakhshi Gerami, Esa Rahtu, and Heikki Huttunen

In: *European Workshop on Visual Information Processing (EUVIP)*. Prais, France: IEEE,  
2021

DOI: 10.1109/EUVIP50544.2021.9483967

©2021 IEEE. Reprinted with the permission of the copyright holders and authors.



# SELECTIVE PROBABILISTIC CLASSIFIER BASED ON HYPOTHESIS TESTING

Saeed Bakhshi Germi, Esa Rahtu

Heikki Huttunen

Tampere University  
Tampere, Finland

Visy Oy  
Tampere, Finland

## ABSTRACT

In this paper, we propose a simple yet effective method to deal with the violation of the Closed-World Assumption for a classifier. Previous works tend to apply a threshold either on the classification scores or the loss function to reject the inputs that violate the assumption. However, these methods cannot achieve the low False Positive Ratio (FPR) required in safety applications. The proposed method is a rejection option based on hypothesis testing with probabilistic networks. With probabilistic networks, it is possible to estimate the distribution of outcomes instead of a single output. By utilizing Z-test over the mean and standard deviation for each class, the proposed method can estimate the statistical significance of the network certainty and reject uncertain outputs. The proposed method was experimented on with different configurations of the COCO and CIFAR datasets. The performance of the proposed method is compared with the Softmax Response, which is a known top-performing method. It is shown that the proposed method can achieve a broader range of operation and cover a lower FPR than the alternative.

**Index Terms**— Selective Classifier, Probabilistic Neural Network, Statistical Analysis, Uncertainty Estimation

## 1. INTRODUCTION

Artificial Intelligence (AI) is becoming a vital part of many real-life applications such as healthcare, logistics, surveillance, and industry. Classification is a common concept in the AI field, and it can be considered one of the building blocks for higher-level reasoning and decision-making systems. With the increasing demand for robust and reliable algorithms, especially in safety-critical systems [1], the research community has been trying to define the robustness [2], evaluation metrics [3], and solutions to satisfy the requirements of a robust classifier [4].

State-of-the-art classifiers have achieved high accuracy numbers when dealing with simple datasets such as MNIST [5] or challenging ones like ImageNet [6]. However, several open questions remain on how the classifier should behave in the circumstances not covered in the training set, for example, when unseen classes appear (out-of-distribution samples) or when inputs are distorted in a way not seen in the training set.

In such cases, a classifier might generate faulty results. So it becomes clear that accuracy is not enough for measuring the performance of classifiers, and the generalization to new environments and robustness to environmental changes should also be considered.

In their review, Zhang *et al.* argue that unexpected faulty result in a pattern recognition algorithm can happen due to the violation of either of the following assumptions[7]: (1) Closed-World Assumption where the data is assumed to have a fixed number of classes, all covered in the training set, (2) Independent and Identically Distributed Assumption where the classes in the data are assumed to be independent of each other and have the same distribution, and (3) Clean and Big Data Assumption where the data is assumed to be well-labeled and large enough for training the network properly. While fulfilling these assumptions is more accessible in a controlled environment, real-world applications rarely cover them completely.

This paper deals with the violation of the Closed-World Assumption. While a straightforward way of dealing with this issue is introducing a *trash* class in the training set to cover all out-of-distribution samples, the complex distribution of them makes it impossible to train an effective classifier in most cases. Moreover, different distortions might make a sample not easy to classify, even for a human. While there is ongoing research for adversarial attacks, the phenomenon is not that common in the everyday use of AI algorithms. In a typical case, distortions usually are from these categories: blur, noise, occlusion, and digital alteration of the image.

Recent works try to solve this issue by formulating it to reliable rejection of the predictions when the network is uncertain. The rejection option, also known as selective classification, is a central concept in different classification applications when dealing with uncertainty (e.g., optical character recognition). Previous works either rely on using a specific type of activation function in the classifier, such as OpenMax [8], temperature scaling for SoftMax [9], and Sigmoid [10], modifying the loss function such as discrepancy loss [11], using more resources such as an ensemble of multiple classifiers [12] and Monte-Carlo dropout [13]. Moreover, some also suggest a combination of different ideas [14].

The proposed method is a rejection option based on hypothesis testing with probabilistic networks. By utilizing a

Z-test over the distribution of outcomes from a probabilistic network, it is possible to estimate the statistical significance of a given output and reject insignificant results. The main difference between the proposed method and previous state-of-the-art methods such as ODIN [9] is the non-restricted use of different architectures. The proposed method can be applied to any architecture and improve the performance when dealing with violation of the Closed-World Assumption by not limiting the network to a specific loss function or activation function.

In their work, Geifman and El-Yaniv show that Softmax Response (SR) is a simple yet top-performing method in selective classifiers [15] that outperforms Monte Carlo (MC) dropout. However, this paper shows that if utilized correctly, the probabilistic network can easily outperform the SR method, making it a viable choice.

The main contributions of this paper are as follows:

- Proposing a simple yet effective method (rejection based on the statistical significance of probabilistic network output) to deal with the violation of the Closed-World Assumption in classifiers. This method can be utilized in any modern network architecture by changing the structure into a probabilistic model, which is possible with the help of existing tools.
- Testing the proposed method on state-of-the-art architecture (ResNet) with a diverse set of distortions (blur, noise, gamma correction, and occlusion) to show the effectiveness of the proposed method over the baseline SR method.

The rest of this paper is structured as follows. The details of the proposed method are presented in Section 2. Then Section 3 deals with the experiments and their results. Finally, Section 4 concludes the work and suggests potential research directions for the future.

## 2. METHODS

### 2.1. Proposed method

The proposed method requires a fully trained probabilistic classifier to work. Due to the nature of the probabilistic classifier, each inference of it will result in a slightly different class score. To utilize this fact, first, the test image is passed through the network  $n$  times to get the mean and standard deviation values for each class. After that, the maximum mean value between classes is chosen as the potential output. Next, two-sample Z-tests [16] are deployed between the potential output and all other classes to find the statistical significance between their difference. Finally, if the Z-scores indicate a significant difference, then the potential output is chosen to be correct. Algorithm 1 summarizes these steps and Figure 1 shows the structure of the proposed method.

---

### Algorithm 1: Selective Probabilistic Classifier

---

**Require:** A trained probabilistic classifier.

- 1: run the image through the classifier  $n$  times
- 2: find mean ( $\mu$ ) and std. dev. ( $\sigma$ ) for all  $N$  classes
- 3: find the class with the highest mean value ( $c_M$ )
- 4: **for**  $i \in 1, 2, \dots, N; i \neq M$  **do**
- 5:   run the two-sample Z-test between  $c_M$  and  $c_i$
- 6:   store the  $Z_i$  score
- 7: **end for**
- 8: **if**  $Z_i > z$  for  $i \in 1, 2, \dots, C; i \neq M$  **then**
- 9:   set output to be  $C_M$
- 10: **else**
- 11:   set output to be Reject
- 12: **end if**

**return** output value for the image

---

#### 2.1.1. Probabilistic Neural Network

A probabilistic neural network (PNN) classifier [17] uses a stochastic weighting system. The classifier can allocate a class to an input sample by utilizing the posterior probability, which means each run of the network will result in a slightly different output. The amount of difference between several runs is the key to network certainty. A low standard deviation between several runs indicates a higher level of certainty for the network, making standard deviation a suitable metric for selective classification. The convolution layers for such a network are constructed based on Flipout [18]. The code can be found in the Tensorflow probability directory [19].

#### 2.1.2. Two-Sample Z-test

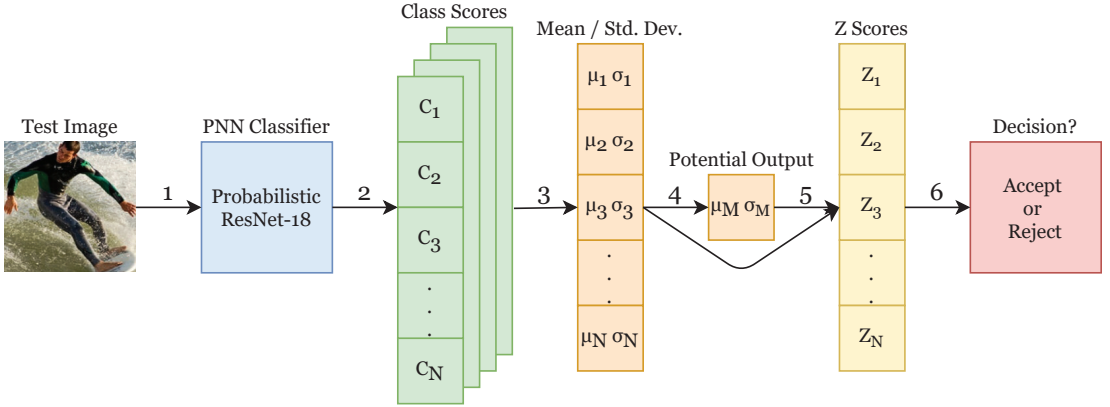
A Z-test [20] refers to any statistical test that can approximate the distribution of the hypothesis by a normal distribution. The two-sample Z-test can be used to test whether two samples are similar to each other or not. The formula is as follows:

$$Z = \frac{\mu_1 - \mu_2 - \Delta}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Where  $\mu_1$  and  $\mu_2$  are the mean values for two samples,  $\Delta$  is the hypothesized difference between the means (0 if testing for equality),  $\sigma_1$  and  $\sigma_2$  are the standard deviations, and  $n_1$  and  $n_2$  are the sample sizes (which are equal in this paper).

By setting the null hypothesis as  $H_0 : \mu_1 = \mu_2$ , the alternative hypothesis as  $H_a : \mu_1 \neq \mu_2$ , and  $\Delta$  to zero, the two-sample Z-test will result in a score that indicates the likelihood of two samples being different from each other. A higher score means more likelihood for the samples to be different. This score can be compared to critical values to get the percentage for the likelihood of a significant difference





**Fig. 1.** The structure of the proposed method. (1) Pass the test image through the probabilistic classifier. (2) Repeat it  $n$  times and store the class scores for each inference. (3) Calculate the mean and standard deviation for each class. (4) Find the maximum mean value and label it as potential output. (5) Run two-sample Z-tests between the potential output and all other classes, then store the Z-scores. (6) Compare Z-scores with the threshold value to decide the acceptance or rejection of the potential output.

between samples. These values can be found in any Z-Score table, such as [21].

## 2.2. Softmax Response

The SR method applies a threshold directly to the output of the Softmax layer from a deep neural network (DNN) and rejects any output below the threshold. This method was chosen as the baseline for comparison. While the method is simple, it is a known top-performer [15].

## 3. EXPERIMENTS AND RESULTS

The proposed method was experimented on with the well-known ResNet-18 network configuration [22]. The goal is to show the performance of it in case of violating the Closed-World Assumption. A comparison with the SR was made to evaluate the performance. This comparison was based on the area under the Receiver Operating Characteristic curve (ROC), which is threshold-independent. Both networks are trained from scratch with the same initial configuration to have a fair comparison. Other state-of-the-art methods were not included in the comparison as they either require a specific structure for the model, limiting the use case, or were only tested on more simple datasets such as MNIST.

Multiple experiments were conducted to represent various violations of Closed-World Assumption in real-world applications. In these experiments, the classifiers are trained with a limited number of classes and presented with both in-distribution and out-of-distribution samples. Further experi-

ments also distort the test samples to see the effect of each distortion on the performance. The chosen distortions were based on [23]. Before discussing the results, the dataset and distortions are explained in detail.

### 3.1. Dataset and Distortions

**COCO** — COCO [24] was chosen as the first dataset. It is a complex dataset where the objects have various sizes, qualities, and overlaps. Since the COCO is originally an object detection dataset, all instances were extracted from it manually based on the bounding boxes provided in the dataset. The data was separated into four classes: Human, Vehicle (containing 4-wheeled vehicles), Animal (containing 4-legged animals), and Background (patches of images with no overlapping objects). 260k images were used for training, excluding the animal class, and 40k images were used as test samples. The reason behind using a commonly known object detection dataset for classification is to have a more realistic dataset where an external source does not filter the samples.

**CIFAR** — CIFAR [25] was chosen as the second dataset. It is a more straightforward dataset where objects are classified into ten categories. The dataset is small yet sufficiently complex, which makes it an ideal case for testing algorithms. 40k images were used for training, excluding the automobile and truck classes, and 10k images were used as test samples.

**Blur** — Three different blurring algorithms were used to see their effect on the performance: Motion blur, Frosted glass blur, and Gaussian blur. The effect of each algorithm can be seen in Figure 2(B-D). Each algorithm will simulate a situ-



**Fig. 2.** Distortions on the image. (A) Original image. (B) Motion blur. (C) Frosted glass blur. (D) Gaussian blur. (E) Noise. (F) Gamma darkening. (G) Gamma lightening. (H) Occlusion.

ation where the object is not sharp (e.g., the camera is not focused, the object is moving, a semi-transparent object is between the camera and the object)

**Noise** — Two different noises were added to test samples to see their effect on the performance: Gaussian noise and Salt-and-pepper noise. The effect of a sample noise can be seen in Figure 2(E). It will simulate a situation where the input is noisy due to internal or external sources.

**Gamma Correction** — The gamma correction technique was applied to each test sample to see the illumination effect on the performance. The effect of darkening and lightening can be seen in Figure 2(F-G). It will simulate a situation where the amount of light in the environment changes due to environmental factors.

**Occlusion** — A black patch was added to test samples to see the effect of occlusion on the performance. The effect of occlusion can be seen in Figure 2(H). It will simulate a situation where the object is partially visible.

### 3.2. Results

After conducting the tests, ROC curves were used to examine the effectiveness of each algorithm. These curves can be seen in Figure 3-4. In general, each point in the ROC curve corresponds to a specific threshold value for the rejection option. If this threshold is set to 0, the algorithm will not reject any input, resulting in a 100% FPR. The more extreme threshold values will result in lower FPR and True Positive Ratio (TPR) until, at some point, the algorithm rejects all inputs (0% FPR and TPR). The SR method hits this value when the threshold

is set to 1. As the output of Softmax cannot be larger than 1, any output will be rejected. However, since a DNN typically generates high scores for the output, this threshold ends up preventing the SR algorithm from reaching lower FPR rates. On the other hand, the proposed method does not rely on the limit of Softmax output, as it compares the significance of each class to the others. Such a limit will cause a significant gap in AUROC scores, as seen in Table 1.

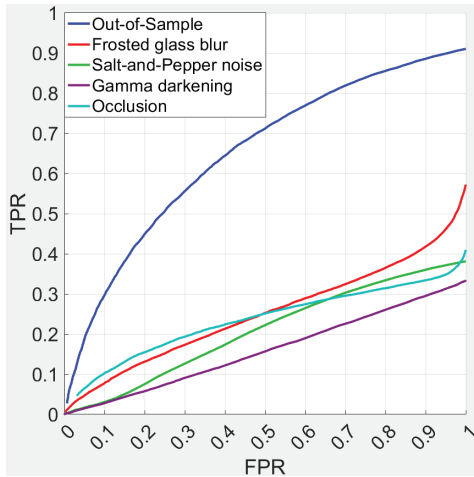
Judging by the ROC curves, both algorithms start roughly on the same point. This means that both algorithms function similarly when it comes to classification. However, the SR method has the mentioned drawback, which is visible in the curves.

The comparison must be threshold-independent for it to be fair. Thus, the area under the ROC curve (AUROC) was used as a comparison method. The area calculation must consider the limitations of both algorithms. While the SR algorithm can reach 0% FPR, it only happens when the threshold is at one (1) or higher, which means the output is not valid. Thus, only the area under the valid parts of the ROC curve was used in calculating the AUROC values. These values can be found in Table 1.

While every distortion reduces the performance, gamma correction has the most significant effect, and blurring has an almost negligible effect on the proposed method. It can be justified by how a classifier works, as changing the intensity of the image makes it harder to separate the objects from the Background class. That being said, the proposed algorithm still outperforms the SR method by a notable margin.

Dataset	Method	Out of Distribution	Blur			Noise		Gamma correction		Occlusion
			Motion	Frosted glass	Gaussian	Gaussian	S&P	Darkening	Lightening	
COCO	Proposed SR	<b>0.65</b>	<b>0.34</b>	<b>0.25</b>	<b>0.38</b>	<b>0.22</b>	<b>0.21</b>	<b>0.16</b>	<b>0.17</b>	<b>0.23</b>
		0.29	0.20	0.18	0.22	0.14	0.09	0.04	0.05	0.06
CIFAR	Proposed SR	<b>0.89</b>	<b>0.50</b>	<b>0.50</b>	<b>0.59</b>	<b>0.38</b>	<b>0.39</b>	<b>0.37</b>	<b>0.42</b>	<b>0.48</b>
		0.52	0.44	0.34	0.47	0.35	0.25	0.22	0.26	0.31

**Table 1.** AUROC values of the tests. The values are calculated by taking the area under the ROC where the algorithm could produce a valid response.

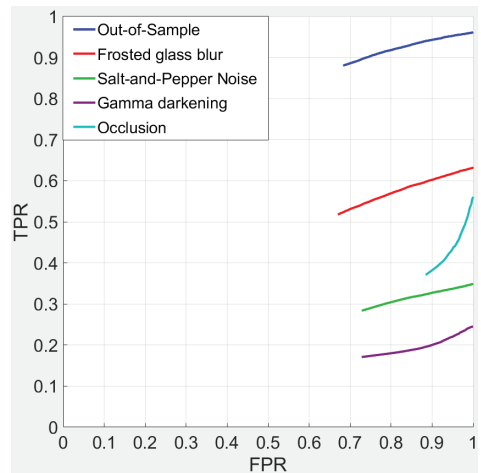


**Fig. 3.** ROC curves for proposed method in COCO test. The worst performance of each category was chosen to present the tolerance of the algorithm to extreme distortions.

#### 4. CONCLUSION

In this paper, we propose a rejection option for probabilistic classifiers based on Z-test analysis. This method will address the violation of the Closed-World Assumption. By utilizing a probabilistic classifier, each run results in a slightly different class score. A Z-test analyses the mean and standard deviation values for multiple runs to estimate network certainty and filter out uncertain results.

We designed several experiments based on a well-known network configuration (ResNet-18) and datasets (COCO and CIFAR). A comparison with the SR method was made based on AUROC as a threshold-independent metric. The proposed method was shown to have better performance than the SR method by a notable margin while maintaining robustness in the presence of distortions. This makes the proposed method more suitable in safety applications.



**Fig. 4.** ROC curves for SR method in COCO test. The worst performance of each category was chosen to present the tolerance of the algorithm to extreme distortions.

In the future, we will consider expanding the method by merging it with existing tools such as ODIN and covering more complex systems such as object detection.

#### Acknowledgment

This research is done as part of a Ph.D. study co-funded by Tampere University and Forum for Intelligent Machines ry (FIMA).

#### 5. REFERENCES

- [1] Vincent Aravantinos and Peter Schlicht, "Making the relationship between uncertainty estimation and safety less uncertain," *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1139–1144, 2020.

- [2] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard, "The robustness of deep networks: A geometrical perspective," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 50–62, 2017.
- [3] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," *IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017.
- [4] Guibiao Xu, Bao-Gang Hu, and Jose C. Principe, "Robust c-loss kernel classifiers," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3, pp. 510–522, 2018.
- [5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.
- [7] Xu-Yao Zhang, Cheng-Lin Liu, and Ching Y. Suen, "Towards robust pattern recognition: A review," *Proceedings of the IEEE*, vol. 108, no. 6, pp. 894–922, 2020.
- [8] Abhijit Bendale and Terrance E. Boult, "Towards open set deep networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1563–1572, 2016.
- [9] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *ArXiv*, vol. 1706.02690, 2017.
- [10] Lei Shu, Hu Xu, and Bing Liu, "DOC: Deep open classification of text documents," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2911–2916, 2017.
- [11] Qing Yu and Kiyoharu Aizawa, "Unsupervised out-of-distribution detection by maximum classifier discrepancy," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9518–9526, 2019.
- [12] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *ArXiv*, vol. 1612.01474, 2016.
- [13] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1050–1059, 2016.
- [14] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke, "Out-of-distribution detection using an ensemble of self supervised leave-out classifiers," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 550–564, 2018.
- [15] Yonatan Geifman and Ran El-Yaniv, "Selective classification for deep neural networks," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4885–4894, 2017.
- [16] Online source, "Two-sample z-test for comparing two means," [www.cliffsnotes.com/study-guides/statistics/univariate-inferential-tests/two-sample-z-test-for-comparing-two-means](http://www.cliffsnotes.com/study-guides/statistics/univariate-inferential-tests/two-sample-z-test-for-comparing-two-means).
- [17] Behshad Mohebbi, Amirhessam Tahmassebi, Anke Meyer-Baese, and Amir H. Gandomi, "Probabilistic neural networks: a brief overview of theory, implementation, and application," *Handbook of Probabilistic Models*, pp. 347–367, 2020.
- [18] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse, "Flipout: Efficient pseudo-independent weight perturbations on mini-batches," *ArXiv*, vol. 1803.04386, 2018.
- [19] Online source, "Tensorflow probability," <https://www.tensorflow.org/probability>.
- [20] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, John Wiley and Sons, 2003.
- [21] Online source, "Z-score table," [www.z-table.com](http://www.z-table.com).
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [23] Christoph Kamann and Carsten Rother, "Benchmarking the robustness of semantic segmentation models with respect to common corruptions," *International Journal of Computer Vision*, pp. 1–22, 2020.
- [24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft coco: Common objects in context," *Computer Vision - ECCV*, pp. 740–755, 2014.
- [25] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.

# PUBLICATION

II

## **A Practical Overview of Safety Concerns and Mitigation Methods for Visual Deep Learning Algorithms**

Saeed Bakhshi Gerami and Esa Rahtu

In: *AAAI's Workshop on Artificial Intelligence Safety (SafeAI)*. Virtual: CEUR, 2022

Reprinted with the permission of the copyright holders and authors.



# A Practical Overview of Safety Concerns and Mitigation Methods for Visual Deep Learning Algorithms

Saeed Bakhshi Germi, Esa Rahtu

Tampere University  
Korkeakoulunkatu 7, 33720 Tampere, Finland  
saeed.bakhshigermi@tuni.fi, esa.rahtu@tuni.fi

## Abstract

This paper proposes a practical list of safety concerns and mitigation methods for visual deep learning algorithms. The growing success of deep learning algorithms in solving non-linear and complex problems has recently attracted the attention of safety-critical applications. While the state-of-the-art methods achieve high performance in synthetic and real-case scenarios, it is impossible to verify/validate their reliability based on currently available safety standards. Recent works try to solve the issue by providing a list of safety concerns and mitigation methods in generic machine learning algorithms from the standards' perspective. However, these solutions are either vague, and non-practical when dealing with deep learning methods in real-case scenarios, or they are shallow and fail to address all potential safety concerns. This paper provides an in-depth look at the underlying cause of faults in a visual deep learning algorithm to find a practical and complete safety concern list with potential state-of-the-art mitigation strategies.

## 1 Introduction

Deep learning is a powerful tool that solves mathematically challenging tasks with high dimensional inputs and multi-variable optimization requirements such as human re-identification, optical character recognition, and object detection. The learning process involves using heuristic and numerical methods, which are often hard to explain or interpret as the dimension grows (black-box behavior).

While state-of-the-art deep learning algorithms achieve high performance in various synthetic and real-life cases, there is no guarantee for the reliability requirements that safety-critical applications typically demand since available safety standards do not provide a suitable verification/validation method for deep learning models.

Recent works found another way of dealing with the problem. By explaining the potential safety concerns of a deep learning algorithm, it is possible to provide suitable mitigation methods around them. While the overall strategy sounds effective, most works fail to provide a practical list of safety concerns and mitigation methods. These lists are typically vague, impractical to implement, shallow, and incomplete.

This paper focuses on the underlying cause of faults in a visual deep learning algorithm to provide a list of safety concerns and potential state-of-the-art mitigation methods. The main contributions of this paper are:

- Providing a practical, complete, and categorical list of possible faults with their underlying cause for different visual deep learning algorithm components.
- Providing potential state-of-the-art mitigation methods to deal with the faults.

The rest of this paper is structured as follows. Section 2 covers related works. Next, Section 3 explains safety concerns related to a visual deep learning algorithm and provides existing mitigation methods to deal with them. Finally, Section 4 concludes the work.

## 2 Related Works

A visual deep learning algorithm is prone to different types of faults. Recent papers focus on either solving specific faults or providing an overview of all system-related safety concerns. Here we discuss some of the most important contemporary works:

Zhang's review of recent papers explains how violation of critical assumptions in the training stage would lead to faults and a non-robust system (Zhang, Liu, and Suen 2020). This review also categorically covers existing mitigation methods and discusses each technique's effectiveness. Song focuses on learning with noisy labels and discusses major strategies to overcome the challenges of this topic (Song et al. 2021). While these works and similar titles provide potential mitigation methods for specific faults, they do not offer a complete list of all safety concerns.

Kläs suggests using uncertainty wrappers on deep learning components to ensure the outcome is dependable (Kläs and Jöckel 2020). However, these wrappers rely on specific metrics that require prior knowledge of data, which is considered impractical in the deep learning field.

Wozniak, Schwalbe, and Willers suggest different approaches to providing a safety concern list and mitigation methods for developing a deep learning algorithm (Wozniak et al. 2020; Schwalbe et al. 2020; Willers et al. 2020). The proposed strategies contain various goals related to the dataset, model, and training/inference stage. However, some goals are vague and non-practical, with no explanation on

how to achieve them or what to do if the goal is not achievable. Moreover, the list is not complete in either work.

Houben provides an extensive list of practical methods to improve the safety of a deep learning algorithm (Houben et al. 2021). The work covers the current state-of-the-art methods to deal with specific problems. However, the provided safety concern list is neither complete nor adequately categorized.

Other similar works, such as (Heyn et al. 2021), also suffer from the same issues. The flaws of recent works can be listed as one or more of the following:

- Not covering the underlying causes of faults, which might lead to poor choice of mitigation methods.
- Providing non-practical and vague mitigation methods, which are not suitable for implementation.
- Overestimating the practical capabilities of mitigation methods in dealing with faults and not providing backup plans in case of failure.

### 3 Safety Concerns (SC) and Mitigation Methods (MM)

The development of a visual deep learning algorithm has three major stages: (1) **training**, (2) **evaluation**, and (3) **inference**. This section presents the list of possible faults within each stage.

#### 3.1 Faults in the Training Stage

Visual data is one of the significant sources of information for deep learning algorithms. Extracting useful information from visual data is a complex task that makes it prone to faults.

A deep learning algorithm approximates the relationship between the input data and the objects in the real world by reducing the empirical risk on training data. Thus, having a proper training dataset is essential to reach the desired quality in the algorithm. A training dataset should be:

- *Complete*: contain samples from the defined output space for the task.
- *Adequate*: contain samples with identical distribution to real-world.
- *Ample*: contain a sufficient amount of samples for convergence of the algorithm.
- *Clean*: contain well-labeled samples.

Moreover, different model structures come with specific sets of benefits and weaknesses. Choosing the correct model, setting up a suitable loss function and optimization algorithm, and finding the perfect hyperparameters are essential to achieve the best performance.

**SC 1 – Incomplete Dataset:** Due to the natural complexity of the real world, there is always a much larger open space than the defined output space for the task. Even with defined boundaries for the output space, known unknowns (e.g., outlier classes) and unknown unknowns (e.g., adversarial attacks) pose a significant issue for the algorithm by producing over-confident wrong predictions.

**SC 2 – Inadequate Dataset:** Due to the ever-changing nature of real-world conditions, the collected data for training will not have identical distribution with the real-world environment in the inference stage. Even a slight mismatch in the distribution can cause a significant drop in performance and result in poor generalization.

**SC 3 – Insufficient/Noisy Dataset:** The cost of manually labeling a dataset increases exponentially with its size. While having a small clean validation dataset is feasible, larger datasets tend to have noisy labels. A deep learning algorithm can memorize this noise, leading to poor generalization and low performance.

**SC 4 – Ill-Matched Architecture:** Manually comparing different models and hyperparameters to find the best match for the task is time-consuming and costly. Moreover, it requires an expert in the field to provide an insight into the problem. An ill-matched architecture could result in unforeseen faults due to inherent weakness against specific situations that might exist.

**MM 1 – Learning with Unseen Data:** Modern deep learning tools could be utilized to force the boundaries of the training dataset even further. Out-of-distribution detectors can be used in the algorithm to detect unseen samples in the inference stage and reject the over-confident results of the algorithm. These methods introduce uncertainty metrics to determine whether the algorithm should be trusted or not (Chen et al. 2020; Sastry and Oore 2020; Bakhshi Germi, Rahtu, and Huttunen 2021).

Also, open-world recognition systems can be used to extend the output space of the algorithm as it encounters outlier samples in the inference stage. These methods continue to learn new classes during the inference stage to reduce the chance of over-confident wrong predictions (Parmar, Chouhan, and Rathore 2021; Bendale and Boulton 2015).

Moreover, the model could be trained to defend against adversarial attacks by including such patterns in the training dataset (Xu et al. 2020; Yuan et al. 2019).

**Discussing MM 1:** Out-of-distribution detectors typically result in lower accuracy, open-world recognition systems are slow and demanding, and adversarial attacks keep evolving and changing every day. The mentioned methods all have their limitation. A suitable backup plan would involve utilizing several models with various mitigation methods to create an ensemble to vote for the final result.

**MM 2 – Learning with Unequally Distributed Data:** Modern deep learning tools could be utilized to reduce the distribution mismatch between the training and inference domain. Transfer learning and domain adaptation can be used to fine-tune the algorithm online during the inference stage. These methods help the model to adapt to new environments quickly and achieve better generalization by using a small batch of data in the inference stage (Farahani et al. 2020; Zhuang et al. 2020).

On the other hand, the algorithm can achieve higher performance by utilizing multiple sources of information for a single task (e.g., person identification with face, iris,





Figure 1: Samples of the same category in MNIST (Top) (Lecun et al. 1998) and CIFAR-10 (Bottom) (Krizhevsky 2009) datasets (Taken from (Chen et al. 2021)). From left to right, the difficulty of classifying is increasing for both manual and automatic label assignment, thus resulting in the increased chance of noisy labels.

voice, and fingerprint). Multimodal learning methods incorporate supplementary and complementary data from multiple modalities to the performance of a single task (Baltrušaitis, Ahuja, and Morency 2018; Guo, Wang, and Wang 2019).

**Discussing MM 2:** Transfer learning and domain adaptation methods typically rely on having a decent starting point (trained network) and quality samples from the inference stage to fine-tune the model successfully. While the requirements are hard to achieve, it is not impossible. Moreover, multimodal methods have already been used with sensor fusion in autonomous vehicles (LIDAR, GPS, IMU, and so on), making them a strong candidate for use in deep learning systems. A suitable backup plan would involve storing the input data during the inference stage to re-evaluate and re-calibrate the algorithm by replacing parts of the older and non-useful training dataset in an iterative cycle.

**MM 3 – Learning with Noisy Labels and Small Dataset:** Modern deep learning tools could be utilized to reduce the effect of label noise or eliminate the need for a large labeled dataset. Robust loss, sample selection, relabeling, and weighted training are all potential solutions to deal with noisy labels in the training dataset (Song et al. 2021; Cordeiro and Carneiro 2020; Adhikari et al. 2021). A combination of multiple methods usually leads to better results.

On the other hand, data augmentation methods can be used to create additional samples for the training dataset. These methods typically involve rotating, scaling, shifting, and flipping data (Wang, Wang, and Lian 2020; Shorten and Khoshgoftaar 2019). More advanced synthesizing techniques can lead to the creation of entire datasets (Raghunathan 2021; Nikolenko 2019). Additionally, existing public datasets can be utilized to extend the samples at a lower cost.

Moreover, the cost and time for manually labeling datasets can be drastically reduced by using iterative labeling methods (Adhikari and Huttunen 2021).

Finally, semi-supervised and unsupervised training techniques can be used to decrease the dependency on a clean training dataset (Van Engelen and Hoos 2020; Schmarje et al. 2021).

**Discussing MM 3:** Recent works prove that the label noise is instance-dependent, as shown in Figure 1. This discovery means most state-of-the-art methods in dealing with label noise need revision on how to mitigate the effects of label noise. Recent works happen to focus on this topic and provide effective solutions. While these solutions do not have mathematical proof, they perform decently on public benchmarks.

Meanwhile, the research around synthesized data indicates that it may not represent the real world in every situation due to the limitations of simulation environments and lack of involved experts in the process. Moreover, the existing public datasets might not suit the specific task or have other inconsistencies, such as low-quality images and noisy labels.

A suitable backup plan would involve developing a more realistic simulation environment while including the physical knowledge about the task in the training process.

**MM 4 – Automated Architecture Selection:** Modern deep learning tools could be utilized to select the optimum model and hyperparameter for a given task. Automated hyperparameter optimization (Yu and Zhu 2020; Luo 2016; Hutter, Lücke, and Schmidt-Thieme 2015) and neural architecture search (Wistuba, Rawat, and Pedapati 2019; Ren et al. 2021) methods can reduce manual labor while eliminating the need for an expert. These methods rely on different search algorithms to find the best model and hyperparameters within the working domain.

**Discussing MM 4:** Relying on search algorithms requires high computational power and proper comparison tools. While it will cost money and time to do it, the solution is not impossible or impractical in most safety-critical applications.

### 3.2 Faults in the Evaluation Stage

Evaluation of a trained deep learning algorithm requires prior knowledge about the task. A testing dataset should include samples from all scenarios, no matter how rare, to ensure the safety of the algorithm. Also, proper performance metrics should be selected during the tests to obtain comparable outputs.



Figure 2: Effects of camera faults on the input image (Taken from (TND6233-D)): (A) Faulty clocking system, (B) Faulty pipeline, and (C) Faulty row addressing logic.

Moreover, formal verification/validation methods depend on having an interpretable algorithm, which contrasts deep learning.

**SC 5 – Incompatible Metrics and Benchmarks:** The most common performance metric in deep learning algorithms is accuracy. However, other metrics might hold more value in safety-critical applications as the importance of false-positive and false-negative grow exponentially in this field. Moreover, gathering a proper dataset to use as a benchmark has similar challenges to the training dataset.

**MM 5 – Using Safety-Aware Metrics and Hazard-Aware Benchmarks:** By including a weighted cost for each type of fault in the performance metric, the algorithm can be evaluated according to safety requirements (Zhou et al. 2021; Gharib and Bondavalli 2019; Salman et al. 2020). These new evaluation metrics would make the trade-off between performance and safety more visible.

On the other hand, a list of all hazardous scenarios can be prepared for every task for inclusion in the testing dataset by performing a risk analysis on the task (Zendel et al. 2018; Lambert et al. 2020). Such datasets could be treated as benchmarks for comparing different algorithms or validating their performance.

**Discussing MM 5:** While formulating a new cost function requires expert knowledge, it is within the scope of expectations in a safety-critical application. Various combination of weighted metrics can be utilized and compared to find the most suitable one for the task. However, a bad decision could result in a non-converging algorithm, thus there is a necessity for mathematical proof about the convergence of the algorithm.

Moreover, the competitive nature of industry typically prevents them from sharing any suitable benchmarks or cost functions publicly, which means each company has to spend time and resources on developing their own system. A suitable backup plan would involve third-party associations funded by multiple companies to handle the problem for the benefit of all members.

**SC 6 – Black-Box Behavior:** The large volume of parameters and non-linear functions in deep learning algorithms result in an uninterpretable system. With no clear relation between the input and output of this black-box system and the impossible task of testing the entire input domain, it is hard to verify/validate deep learning algorithms based on safety standards.

**MM 6 – Opening the Black-Box:** Representation learning enables the deep learning algorithm to discover the relation between input data and output in a presentable way by showing the process of feature selection (Zhang et al. 2018; Li, Yang, and Zhang 2018). Understanding this process helps to gain an insight into how the network interprets input data, and which parts of data play a more significant role in deciding the outcome.

Another way to gain such insight is to present a map of pixel relevance for the algorithm. These heat maps illustrate the importance of each pixel when calculating the output (König et al. 2021; Bach et al. 2015). Such information can be about isolated pixels or the interconnection of different pixels. Studying these maps could show the effects of slight changes in input on the output and help find potential hazardous cases.

**Discussing MM 6:** This specific problem could be one of the most important ones with the least proper solutions as of yet. While it is possible to gain some insight into the operation of deep learning algorithms, the information cannot be used in any form to verify/validate the algorithm based on traditional standards. A suitable backup plan would involve using safety case arguments and other similar approaches to bypass the need for verification/validation for now.

### 3.3 Faults in the Inference Stage

In a typical case, a similar sensor used for collecting offline data provides the online data for the implemented algorithm. On top of it, other hardware components are required for the algorithm to work correctly. These components can be summarized as:



Figure 3: Effects of environmental factors on the input image (Taken from (Bakhshi Gerami, Rahtu, and Huttunen 2021)): (A) Original image, (B) Movement of camera/object (Motion blur), (C) Raindrop on the lens (Frosted-glass blur), (D) Out-of-focus object (Gaussian blur), (E) Low illumination (Gaussian noise), (F,G) Improper balance of light and darkness (Low/High brightness), and (H) Obscured object (Occlusion).

- A camera to capture the input image.
- A communication channel to transfer the captured image.
- A processing unit to host the deep learning algorithm.
- A power supply to keep the system running.

**SC 7 – Defective Hardware:** The first concern in deep learning algorithms is providing the necessary hardware mentioned above. Hardware faults can have a wide range of effects on the algorithm based on the faulty component, an example being the results of a faulty camera on the captured image, as shown in Figure 2. An implementation of the algorithm might run into problems based on the defective hardware component:

- Camera faults that might result in various disturbances in the input image, such as pixel corruption or image distortion.
- Communication channel faults that might result in data corruption or data loss.
- Processing unit faults that might result in wrong calculations, lagging, or freezing of the algorithm.
- Power supply faults might result in breaking other hardware components or total system shutdown.

**MM 7.1 – Following Functional Safety Standards:** The mentioned hardware components are not unique to deep learning algorithms and have been used for decades in safety-critical applications. As a result, the current functional safety standards such as ISO 26262 (ISO 26262)

and ISO/PAS 21448 (ISO/PAS 21448) provide practical guidelines for verifying and validating hardware components. Also, technical reports based on functional safety standards can help develop or choose safe hardware components such as a camera (TND6233-D), communication channel (Alanen, Hietikko, and Malm 2004), and operating system (Slačka and Halás 2015).

Moreover, other precautions such as using redundant hardware, proper noise shielding, and data fusion techniques have already proved helpful in safety-critical applications (Sklaroff 1976; Ciftcioglu and Turkcan 1996).

**Discussing MM 7.1:** Assuming the hardware is chosen based on the proper functional safety standards, it should operate without significant safety concerns. However, this mitigation method does not guarantee the complete removal of any disturbance or corruption of data. Environmental factors such as lousy illumination, movement, and obscured objects can affect input image quality without causing a hardware failure, as seen in Figure 3. While some of these problems might not be recognizable by a human annotator, the deep learning algorithm could run into faults based on the type and severity of corruption. Moreover, less severe levels of hardware failure might cause noise variations on the input data. A suitable backup plan would involve utilizing another mitigation approach described as follows.

**MM 7.2 – Using Image Processing Techniques:** Since the exact relation between the input image and the output of the deep learning algorithm is not known, it is recom-

mended to have clean input data to reduce the change of unwanted outcomes. The current state-of-the-art image processing techniques such as denoising (Fan et al. 2019; Goyal et al. 2020; Jebur, Der, and Hammood 2020), deblurring (Sada and Goyani 2018; Nah et al. 2021; Abuolaim, Timofte, and Brown 2021), and enhancement (Putra, Purboyo, and Prasasti 2017) methods can improve the quality of the input images and remove most of the disturbances not covered by the previous mitigation method. Most image processing techniques have solid mathematical foundations and passed extensive testing cycles to prove their effectiveness, making them easy to validate and verify for safety-critical applications.

**Discussing MM 7.2:** Image processing techniques are only valid when it's known that the image is corrupted. Otherwise, such functions can negatively affect a clean image during the operation (e.g., removing/fading edges, brightening the image without necessity, etc.). Applying a filter without knowing the type of corruption is almost as dangerous as not utilizing any technique. So, it is safe to assume that some form of corruption is inevitable. A suitable backup plan would involve using the rejection option as described before to reduce the amount of overconfident wrong outputs.

## 4 Conclusion

The research around using deep learning algorithms in safety-critical applications is growing rapidly, with the current state-of-the-art answers partially fulfilling the requirements of old standards. However, the nature of the problem demands to move away from the traditional broad-spectrum method of standardization as it is not suitable for deep learning algorithms. There is a high demand for task-specific standards to be developed. Until such standards are developed, the research community focuses on alternative approaches and empirical analysis to provide practical solutions on specific cases.

This paper provides a practical list of safety concerns for a visual deep learning algorithm by explaining the underlying cause of faults and providing current state-of-the-art solutions to mitigate them. By presenting the limitations of existing mitigation methods, the need for further study is expressed. We hope this paper offers an insight to those who want to utilize deep learning algorithms in their applications or those who want to develop proper standard or safety case arguments for such systems.

## Acknowledgments

This research is done as part of a Ph.D. study co-funded by Tampere University and Forum for Intelligent Machines ry (FIMA).

## References

Abuolaim, A.; Timofte, R.; and Brown, M. S. 2021. NTIRE 2021 Challenge for Defocus Deblurring Using Dual-pixel Images: Methods and Results. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 578–587.

Adhikari, B.; and Huttunen, H. 2021. Iterative Bounding Box Annotation for Object Detection. In *25th International Conference on Pattern Recognition (ICPR)*, 4040–4046.

Adhikari, B.; Peltomäki, J.; Bakhshi Germi, S.; Rahtu, E.; and Huttunen, H. 2021. Effect of Label Noise on Robustness of Deep Neural Network Object Detectors. In *Computer Safety, Reliability, and Security. SAFECOMP Workshops*, 239–250.

Alanen, J.; Hietikko, M.; and Malm, T. 2004. *Safety of Digital Communications in Machines*. VTT Technical Research Centre of Finland. ISBN 951-38-6502-9.

Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7): 1–46.

Bakhshi Germi, S.; Rahtu, E.; and Huttunen, H. 2021. Selective Probabilistic Classifier Based on Hypothesis Testing. In *9th European Workshop on Visual Information Processing (EUVIP)*.

Baltrušaitis, T.; Ahuja, C.; and Morency, L.-P. 2018. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2): 423–443.

Bendale, A.; and Boulton, T. 2015. Towards Open World Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1893–1902.

Chen, J.; Li, Y.; Wu, X.; Liang, Y.; and Jha, S. 2020. Robust Out-of-distribution Detection for Neural Networks. arXiv:2003.09711.

Chen, P.; Ye, J.; Chen, G.; Zhao, J.; and Heng, P.-A. 2021. Beyond Class-Conditional Assumption: A Primary Attempt to Combat Instance-Dependent Label Noise. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11442–11450.

Ciftcioglu, O.; and Turkcan, E. 1996. Data fusion and sensor management for nuclear power plant safety.

Cordeiro, F. R.; and Carneiro, G. 2020. A Survey on Deep Learning with Noisy Labels: How to train your model when you cannot trust on the annotations? arXiv:2012.03061.

Fan, L.; Zhang, F.; Fan, H.; and Zhang, C. 2019. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2(1): 1–12.

Farahani, A.; Voghoei, S.; Rasheed, K.; and Arabnia, H. R. 2020. A Brief Review of Domain Adaptation. arXiv:2010.03978.

Gharib, M.; and Bondavalli, A. 2019. On the evaluation measures for machine learning algorithms for safety-critical systems. In *15th European Dependable Computing Conference (EDCC)*, 141–144.

Goyal, B.; Dogra, A.; Agrawal, S.; Sohi, B.; and Sharma, A. 2020. Image denoising review: From classical to state-of-the-art approaches. *Information Fusion*, 55: 220–244.

Guo, W.; Wang, J.; and Wang, S. 2019. Deep Multimodal Representation Learning: A Survey. *IEEE Access*, 7: 63373–63394.

- Heyn, H.-M.; Knauss, E.; Muhammad, A. P.; Eriksson, O.; Linder, J.; Subbiah, P.; Pradhan, S. K.; and Tungal, S. 2021. Requirement Engineering Challenges for AI-intense Systems Development. arXiv:2103.10270.
- Houben, S.; Abrecht, S.; Akila, M.; Bär, A.; Brockherde, F.; Feifel, P.; Fingscheidt, T.; Gannamaneni, S. S.; Ghobadi, S. E.; Hammam, A.; Haselhoff, A.; Hauser, F.; Heinze-mann, C.; Hoffmann, M.; Kapoor, N.; Kappel, F.; Klingner, M.; Kronenberger, J.; Küppers, F.; Löhdefink, J.; Mlynarski, M.; Mock, M.; Mualla, F.; Pavlitskaya, S.; Poretschkin, M.; Pohl, A.; Ravi-Kumar, V.; Rosenzweig, J.; Rottmann, M.; Rüping, S.; Sämann, T.; Schneider, J. D.; Schulz, E.; Schwalbe, G.; Sicking, J.; Srivastava, T.; Varghese, S.; Weber, M.; Wirkert, S.; Wirtz, T.; and Woehrl, M. 2021. In-spect, Understand, Overcome: A Survey of Practical Methods for AI Safety. arXiv:2104.14235.
- Hutter, F.; Lücke, J.; and Schmidt-Thieme, L. 2015. Beyond manual tuning of hyperparameters. *KI-Künstliche Intelligenz*, 29(4): 329–337.
- ISO 26262. 2018. Road vehicles – Functional safety. Standard, International Organization for Standardization.
- ISO/PAS 21448. 2019. Road vehicles — Safety of the intended functionality. Standard, International Organization for Standardization.
- Jebur, R. S.; Der, C. S.; and Hammood, D. A. 2020. A Review and Taxonomy of Image Denoising Techniques. In *6th International Conference on Interactive Digital Media (ICIDM)*.
- Klås, M.; and Jöckel, L. 2020. A Framework for Building Uncertainty Wrappers for AI/ML-Based Data-Driven Components. In *Computer Safety, Reliability, and Security. SAFECOMP Workshops*, 315–327.
- Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report.
- König, G.; Molnar, C.; Bischl, B.; and Grosse-Wentrup, M. 2021. Relative Feature Importance. In *25th International Conference on Pattern Recognition (ICPR)*, 9318–9325.
- Lambert, J.; Liu, Z.; Sener, O.; Hays, J.; and Koltun, V. 2020. MSeg: A composite dataset for multi-domain semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2879–2888.
- Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.
- Li, Y.; Yang, M.; and Zhang, Z. 2018. A Survey of Multi-View Representation Learning. *IEEE Transactions on Knowledge and Data Engineering*, 31(10): 1863–1883.
- Luo, G. 2016. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1): 1–16.
- Nah, S.; Son, S.; Lee, S.; Timofte, R.; and Lee, K. M. 2021. NTIRE 2021 Challenge on Image Deblurring. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 149–165.
- Nikolenko, S. I. 2019. Synthetic Data for Deep Learning. arXiv:1909.11512.
- Parmar, J.; Chouhan, S. S.; and Rathore, S. S. 2021. Open-world Machine Learning: Applications, Challenges, and Opportunities. arXiv:2105.13448.
- Putra, R.; Purboyo, T.; and Prasasti, A. 2017. A Review of Image Enhancement Methods. *International Journal of Applied Engineering Research*, 12: 13596–13603.
- Raghunathan, T. E. 2021. Synthetic Data. *Annual Review of Statistics and Its Application*, 8(1): 129–140.
- Ren, P.; Xiao, Y.; Chang, X.; Huang, P.-y.; Li, Z.; Chen, X.; and Wang, X. 2021. A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions. *ACM Computing Surveys*, 54(4): 1–34.
- Sada, M. M.; and Goyani, M. M. 2018. Image Deblurring Techniques—A Detail Review. *International Journal of Scientific Research in Science, Engineering and Technology*, 4: 176–188.
- Salman, T.; Ghubaish, A.; Unal, D.; and Jain, R. 2020. Safety Score as an Evaluation Metric for Machine Learning Models of Security Applications. *IEEE Networking Letters*, 2(4): 207–211.
- Sastry, C. S.; and Oore, S. 2020. Detecting Out-of-Distribution Examples with Gram Matrices. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, 8491–8501.
- Schmarje, L.; Santarossa, M.; Schröder, S.-M.; and Koch, R. 2021. A Survey on Semi-, Self- and Unsupervised Learning for Image Classification. *IEEE Access*, 9: 82146–82168.
- Schwalbe, G.; Knie, B.; Sämann, T.; Dobberphul, T.; Gauerhof, L.; Raafatnia, S.; and Rocco, V. 2020. Structuring the Safety Argumentation for Deep Neural Network Based Perception in Automotive Applications. In *Computer Safety, Reliability, and Security. SAFECOMP Workshops*, 383–394.
- Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1): 1–48.
- Sklaroff, J. R. 1976. Redundancy Management Technique for Space Shuttle Computers. *IBM Journal of Research and Development*, 20(1): 20–28.
- Slačka, J.; and Halás, M. 2015. Safety critical RTOS for space satellites. In *20th International Conference on Process Control (PC)*, 250–254.
- Song, H.; Kim, M.; Park, D.; Shin, Y.; and Lee, J.-G. 2021. Learning from Noisy Labels with Deep Neural Networks: A Survey. arXiv:2007.08199.
- TND6233-D. 2018. Evaluating Functional Safety in Automotive Image Sensors. White paper, ON Semiconductor.
- Van Engelen, J. E.; and Hoos, H. H. 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2): 373–440.
- Wang, X.; Wang, K.; and Lian, S. 2020. A survey on face data augmentation for the training of deep neural networks. *Neural computing and applications*, 1–29.
- Willers, O.; Sudholt, S.; Raafatnia, S.; and Abrecht, S. 2020. Safety Concerns and Mitigation Approaches Regarding the

Use of Deep Learning in Safety-Critical Perception Tasks. In *Computer Safety, Reliability, and Security. SAFECOMP Workshops*, 336–350.

Wistuba, M.; Rawat, A.; and Pedapati, T. 2019. A Survey on Neural Architecture Search. arXiv:1905.01392.

Wozniak, E.; Cărlan, C.; Acar-Celik, E.; and Putzer, H. J. 2020. A Safety Case Pattern for Systems with Machine Learning Components. In *Computer Safety, Reliability, and Security. SAFECOMP Workshops*, 370–382.

Xu, H.; Ma, Y.; Liu, H.-C.; Deb, D.; Liu, H.; Tang, J.-L.; and Jain, A. K. 2020. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2): 151–178.

Yu, T.; and Zhu, H. 2020. Hyper-Parameter Optimization: A Review of Algorithms and Applications. arXiv:2003.05689.

Yuan, X.; He, P.; Zhu, Q.; and Li, X. 2019. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9): 2805–2824.

Zendel, O.; Honauer, K.; Murschitz, M.; Steininger, D.; and Domínguez, G. F. 2018. WildDash - Creating Hazard-Aware Benchmarks. In *Computer Vision – ECCV*, 407–421.

Zhang, D.; Yin, J.; Zhu, X.; and Zhang, C. 2018. Network Representation Learning: A Survey. *IEEE Transactions on Big Data*, 6(1): 3–28.

Zhang, X.-Y.; Liu, C.-L.; and Suen, C. Y. 2020. Towards Robust Pattern Recognition: A Review. *Proceedings of the IEEE*, 108(6): 894–922.

Zhou, J.; Gandomi, A. H.; Chen, F.; and Holzinger, A. 2021. Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics. *Electronics*, 10(5).

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2020. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1): 43–76.

# PUBLICATION

## III

### **Enhanced Data-Recalibration: Utilizing Validation Data to Mitigate Instance-Dependent Noise in Classification**

Saeed Bakhshi Germi and Esa Rahtu

In: *International Conference on Image Analysis and Processing (ICIAP)*. Lecce, Italy:  
Springer, 2022

DOI: 10.1007/978-3-031-06427-2\_52



Reproduced with permission from Springer Nature.







# Enhanced Data-Recalibration: Utilizing Validation Data to Mitigate Instance-Dependent Noise in Classification

Saeed Bakhshi Geremi<sup>(\*)</sup>  and Esa Rahtu 

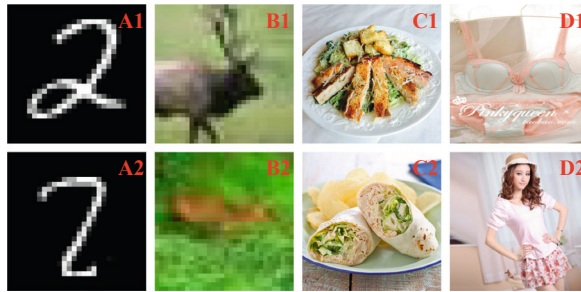
Tampere University, Tampere, Finland  
{saeed.bakhshigermi, esa.rahtu}@tuni.fi

**Abstract.** This paper proposes a practical approach to deal with instance-dependent noise in classification. Supervised learning with noisy labels is one of the major research topics in the deep learning community. While old works typically assume class conditional and instance-independent noise, recent works provide theoretical and empirical proof to show that the noise in real-world cases is instance-dependent. Current state-of-the-art methods for dealing with instance-dependent noise focus on data-recalibrating strategies to iteratively correct labels while training the network. While some methods provide theoretical analysis to prove that each iteration results in a cleaner dataset and a better-performing network, the limiting assumptions and dependency on knowledge about noise for hyperparameter tuning often contrast their claims. The proposed method in this paper is a two-stage data-recalibration algorithm that utilizes validation data to correct noisy labels and refine the model iteratively. The algorithm works by training the network on the latest cleansed training Set to obtain better performance on a small, clean validation set while using the best performing model to cleanse the training set for the next iteration. The intuition behind the method is that a network with decent performance on the clean validation set can be utilized as an oracle network to generate less noisy labels for the training set. While there is no theoretical guarantee attached, the method's effectiveness is demonstrated with extensive experiments on synthetic and real-world benchmark datasets. The empirical evaluation suggests that the proposed method has a better performance compared to the current state-of-the-art works. The implementation is available at <https://github.com/Sbakhshigermi/EDR>.

**Keywords:** Label noise · Classification · Data-recalibration

## 1 Introduction

Inexperienced workers, insufficient information about samples, confusing patterns, tiresome nature of the work, and other factors make the manual labeling



**Fig. 1.** Multiple samples of the same category in different datasets: (A) Number two in MNIST [15], (B) Deer in CIFAR-10 [14], (C) Caesar salad in Food-101N [16], and (D) Underwear in Clothing1M [34]. The images on the top row are more straightforward to label than the images on the bottom row.

of samples in a large dataset prone to errors and noisy labels [10, 29]. Unfortunately, deep learning algorithms have the potential to memorize these noisy labels, which leads to poor generalization and lower performance on clean test datasets [37]. Due to the importance of the topic in different sectors, such as safety-critical applications [2] and medical imaging [23], researchers have been developing methods to mitigate such label noise [3, 10, 27].

Most recent works assume the labels to be affected by a class-conditional noise (CCN) where the noise is instance-independent [20]. This type of noise can be estimated [13] or mitigated by adding extra loss terms in the model [4]. However, Chen utilized visual examples and mathematical analysis to prove that the label noise in a real-world dataset (Clothing1M [34]) is actually instance-dependent. To better understand why this is the case, take a look at Fig. 1. As seen in this figure, two samples of the same category have different complexity of labeling, which suggests that the label noise is instance-dependent.

With the previous assumption of CCN proven wrong, a new mathematical foundation for mitigation methods had to be developed. Therefore, researchers started defining variations of instance-dependent noise (IDN) patterns to represent synthetic noise and propose mitigation approaches based on them. One of the effective strategies used in the state-of-the-art methods is the iterative data-recalibration [18]. These methods use the predictions of a network trained over noisy samples to select and correct samples iteratively.

While recent works on IDN provided theoretical analysis to prove the convergence of their models to an oracle Bayes classifier [5, 38], the limiting assumptions in their theories cannot be met in practical implementation, as shown by their empirical findings. Due to these limitations, this paper will focus on empirical experiments on synthetic and real-world datasets to showcase the effectiveness of the proposed method.

This paper proposes an enhanced data-recalibration algorithm that corrects labels affected by instance-dependent noise by utilizing validation set. On each iteration, the proposed method trains a model with the cleansed data from the

last iteration to achieve higher performance on a small, clean validation set. Then, the best-performing model is chosen to correct labels in the training set based on the model’s confidence for the next iteration. The intuition is that better performance on the clean validation set means a better prediction of training labels than the previous iteration.

The main difference between the proposed method and previous works is utilizing a clean validation set to influence the training stage to help the network approach an oracle model that can predict ground truth labels. Small, clean validation sets can be easily obtained with computer-assisted tools [1]. While previous works often use the validation set as a selector of the final model for accuracy reports, none utilize it any further to the best of the authors’ knowledge. The main contributions of this paper are:

- Proposing a practical data-recalibration algorithm that utilizes easy-to-gather clean validation set to enhance the performance over the existing state-of-the-art methods.
- Providing empirical evaluation with extensive experiments on both synthetic and real-world datasets to show the effectiveness of the proposed method.

The rest of the paper is structured as follows. Section 2 covers the related works. Next, Sect. 3 explains the proposed method in detail. After that, Sect. 4 deals with the experiments and the empirical evaluation to show the effectiveness of the proposed method. Finally, Sect. 5 concludes the work.

## 2 Related Works

Menon provided one of the major theoretical frameworks for IDN in binary problems. This framework provided the basis to construct a loss function with specific criteria to mitigate IDN. While the work was necessary at the time, the method is not extensible to deep neural networks [21]. Chen provided mathematical proof that the label noise in a large real-world dataset called Clothing1M [34] follows the IDN pattern. They proposed a method of generating IDN patterns by averaging the predictions of an oracle classifier over the training session to find complex samples and flip their labels. The mitigation method provided also relies on averaging the predictions of a network, with the intuition that the network can find a soft representation of labels that are closer to ground truth over time. While this work provided essential information about IDN, the mitigation method is cost-heavy with low performance compared to other works [7].

Zhang defined a new family of noise called poly-margin diminishing (PMD). This new noise family follows the same intuition that data points near the decision boundary are more challenging to classify, thus more prone to noise. Based on the previously stated reasons for label noise, this definition seems realistic. To mitigate this family of noise, they proposed an iterative correction method that corrects the labels based on the network confidence over the training set in each iteration. While the work provided theories to prove the effectiveness, their

hyperparameter settings and assumption violation in implementing the method contradict their idea [38].

Several state-of-the-art methods managed to reach high performance on real-world benchmarks. Tan combined a supervised and an unsupervised network and co-teach them with the help of an encoder to maximize the agreement between the networks in latent space [28]. Wu utilized the spatial topology of data in the latent space of the network iteratively to collect clean labels and refine the network further [32]. Zhu focused on the second-order approach to estimate covariance terms for IDN with peer loss function [19] and defined a new loss function to change the problem to CCN [40]. Xia eliminated the need for anchor points in estimating the noise transition matrix [33]. Han described a two-stage algorithm where the trained network is used to select multiple class prototypes to represent the characteristics of the data better and correct the noisy labels [11]. Lee focused on reducing human supervision by introducing a method that required a small clean training set to extract the information about label noise [16]. Li divides the training data into labeled clean and unlabeled noisy samples to utilize semi-supervised learning techniques by training two networks and correcting more labels over each iteration [17]. Other methods such as PEN-CIL [36], ILFC [5], CORES<sup>2</sup> [8], Meta-Weight-Net [24], estimation of transition matrix [35], and JoCoR [31] are also noteworthy.

### 3 Proposed Method

In this section, we present the details of our proposed method. The proposed method alternates between training the network to find the best performance on the clean validation set and correcting the noisy labels based on confidence scores from the top-performing network. Before the proposed algorithm starts the process, we prepare a deep neural network by training it for a few epochs with a high learning rate, which allows the network to reach a reasonable confidence level without overfitting to noise [37].

#### 3.1 Preliminaries

Let  $\mathcal{X}$  be the feature space,  $\mathcal{L}$  be the label space,  $(x, y), (x, \tilde{y}) \in \mathcal{X} \times \mathcal{L}$  be a clean and a noisy sample respectively,  $D = \{(x_i, y_i)\}_{i=1}^n$  be a dataset,  $f^t(x) = (C_1, \dots, C_k)$  be a classifier at the  $t$ -th iteration of the algorithm, where  $C_i$  is the confidence score of the network for the  $i$ -th class (output of softmax layer in this paper), and  $k$  is the total number of classes. Finally, let  $S^t$  be the performance of the classifier over clean validation set at  $t$ -th iteration of the algorithm.

#### 3.2 Iterative Label Correction Method

The overall algorithm is summarized in Algorithm 1. In practice, we use an average of confidence scores from several top-performing networks. Since there is no

---

**Algorithm 1:** Enhanced Data-Recalibration

---

**Require:** Initial training set  $\tilde{D}_{train}^0 = \{(x_i, \tilde{y}_i^0)\}_{i=1}^n$ , Initial classifier  $f^0$ , threshold value  $\theta$ , Number of epochs  $T$ , Validation set  $D_{valid} = \{(x_i, y_i)\}_{i=1}^m$

- 1: **for**  $t \in 1, \dots, T$  **do**
- 2:     Train  $f^{t-1}$  on  $\tilde{D}_{train}^{t-1}$  to get  $f^t$  and get the performance score  $S^t$
- 3:     Compare  $S^t$  to previous scores  $\{S^i\}_{i=1}^{t-1}$  to find best performing classifier  $f^B$
- 4:     **for**  $(x, \tilde{y}) \in \tilde{D}_{train}^{t-1}$  **do**
- 5:         Get the confidence scores  $(C_1, \dots, C_k)$  of  $f^B$  on  $x$
- 6:         Find the best confidence score  $C_M$  and the noisy confidence score  $C_N$
- 7:         Calculate  $Gap = |\log(C_M) - \log(C_N)|$
- 8:         **if**  $Gap \geq \theta$  **then**
- 9:             Set new label  $\tilde{y}^t = M$
- 10:         **else**
- 11:             Keep old label  $\tilde{y}^t = \tilde{y}^{t-1}$
- 12:         **end if**
- 13:     **end for**
- 14:     **if**  $\forall i \in [1, \dots, n], \tilde{y}_n^t = \tilde{y}_n^{t-1}$  **then**
- 15:         Decrease  $\theta$  by a small amount
- 16:     **end if**
- 17: **end for**

**return** Best trained network  $f^B$

---

guarantee of improving the network on every iteration, there might be a random instance where the trained network arbitrarily achieves a high performance score. Averaging multiple confidence scores mitigates the effect of these random encounters as they do not introduce a bias towards any class. Moreover, the top-performing networks are selected from a range of recently trained networks to ensure that the network is not stuck in a loop. In the following subsections, we will describe what happens in the t-th iteration of the algorithm:

### 3.3 Stage One

In this stage, the algorithm starts training the network for one epoch with the labels acquired from the previous iteration. In other terms, the network from the previous iteration  $f^{t-1}$  is trained on the training set with labels generated in the previous iteration  $\tilde{D}_{train}^{t-1} = \{(x_i, \tilde{y}_i^{t-1})\}_{i=1}^n$  to obtain the new network  $f^t$ . Then, the performance of the network is evaluated to obtain the top-performing network for the next stage. It is done by evaluating the new network  $f^t$  on the clean validation set  $D_{valid} = \{(x_i, y_i)\}_{i=1}^m$  to get its performance score  $S^t$ . Then, this performance score  $S^t$  is compared to all previous scores  $\{S^i\}_{i=1}^{t-1}$  to find the best-performing network  $\{f^B \mid \forall i \leq t : S^B \geq S^i\}$ .

### 3.4 Stage Two

In this stage, the algorithm starts collecting the confidence scores of the chosen network on the training set. It is done by predicting the confidence scores

$(C_1, \dots, C_k)$  of the best-performing network  $f^B$  for each sample in training set from the previous iteration  $(x, \tilde{y}) \in \tilde{D}_{train}^{t-1}$ . Then, the confidence scores are evaluated to decide the labels for the next iteration. For each sample in the dataset  $(x, \tilde{y}) \in \tilde{D}_{train}^{t-1}$ , the highest confidence score  $\{C_M \mid \forall i \leq k : C_M \geq C_i\}$  and the confidence score for the noisy label  $C_{N=\tilde{y}}$  are considered. If the difference of logarithms between them is greater than a threshold  $|\log(C_M) - \log(C_N)| \geq \theta$ , then the sample is selected for correction. The intuition behind the process is that a noticeable gap between the prediction of the best-performing network and the current label suggests the label is noisy. After that, the labels for the next iteration are generated. It is done by swapping the label of the selected samples to the prediction of the best-performing network  $\tilde{y}_{sel}^t = M$  while keeping the labels of other samples the same as before  $\tilde{y}_{rest}^t = \tilde{y}^{t-1}$ . Finally, the threshold value is evaluated and reduced if the algorithm cannot select samples anymore. By initializing a high threshold value and lowering it in small steps, the best-performing network gains more trust from the algorithm gradually, which prevents confirmation bias to some degree.

## 4 Experiments and Evaluation

### 4.1 Synthetic Datasets

For proof of concept, the public datasets CIFAR-10 and CIFAR-100 [14] are chosen for synthetic experiments. Both datasets contain 50,000 training and 10,000 testing samples over ten categories. In the case of CIFAR-100, each category is further divided into ten subclasses. As argued by the previous works [5, 7, 38], a realistic noise does not uniformly affect all data space points. The most common solution among previous works to generate reliable IDN is to find challenging samples and then flip their label from the most confident category to the second most confident category. A challenging sample is typically located at the edges of the decision boundary and results in a low network confidence score. Such samples can be found by training an oracle network and selecting the low confidence samples [5] or averaging the network’s confidence over the training period and selecting the confusing samples [7]. To generate reliable and comparable IDN, we follow the definition for the PMD noise family [38].

Let  $\aleph_{C_1, C_2}(x) = \mathbb{P}[\tilde{y} = C_2 \mid y = C_1, x]$  be the probability of corrupting the label of a sample from the most confident class  $C_1$  to the second-most confident class  $C_2$ , and  $f^*(x)$  be an oracle classifier trained on clean samples. The three types of IDN used in our experiments are defined as in Eq. 1.

$$\begin{aligned}
 \aleph_{C_1, C_2}^I(x) &= \frac{1}{2} - \frac{1}{2} [f_{C_1}^*(x) - f_{C_2}^*(x)]^2 \\
 \aleph_{C_1, C_2}^{II}(x) &= 1 - [f_{C_1}^*(x) - f_{C_2}^*(x)]^3 \\
 \aleph_{C_1, C_2}^{III}(x) &= 1 - \frac{1}{3} [f_{C_1}^*(x) - f_{C_2}^*(x)]^3 \\
 &\quad - \frac{1}{3} [f_{C_1}^*(x) - f_{C_2}^*(x)]^2 - \frac{1}{3} [f_{C_1}^*(x) - f_{C_2}^*(x)]
 \end{aligned} \tag{1}$$

For the sake of completion, we also include the most common CCN noise types in our experiments: uniform and asymmetrical [22]. Let  $\supset_{C_1, C_2} = \mathbb{P}[\hat{y} = C_2 \mid y = C_1]$  be the probability of corrupting the label of a sample from class  $C_1$  to class  $C_2$ ,  $\mathcal{R}$  be the noise rate and  $k$  be the total number of classes. The two types of CCN used in our experiments are defined as in Eq. 2.

$$\supset_{C_1, C_2}^{\text{Uniform}} = \begin{cases} \frac{\mathcal{R}}{k-1} & C_1 \neq C_2 \\ 1 - \mathcal{R} & C_1 = C_2 \end{cases} \quad (2)$$

$$\supset_{C_1, C_2}^{\text{Asymmetrical}} = \begin{cases} \mathcal{R} & C_1 \neq C_2 \\ 1 - \mathcal{R} & C_1 = C_2 \end{cases}$$

The ResNet-34 [12] is used for synthetic experiments. All models are trained from scratch for 180 epochs with a batch size of 128 images. Stochastic gradient descent is used as the optimizer with a momentum value equal to  $9 \times 10^{-1}$  and a weight decay rate of  $5 \times 10^{-4}$ . The learning rate is initialized as  $1 \times 10^{-2}$  and gets divided by 2 after 40 and 80 epochs. Standard data augmentations are applied: random horizontal flip,  $32 \times 32$  random crop after padding 4 pixels, and standard normalizing with mean = (0.4914, 0.4822, 0.4465), std = (0.2023, 0.1994, 0.2010). In each experiment, 10% of the clean training data is reserved as the validation set. Each experiment is repeated 5 times to report the mean and standard deviation for final accuracy. The initial value for  $\theta$  in Algorithm 1 is set to  $7 \times 10^{-1}$  with a decrement step of  $1 \times 10^{-1}$ . The algorithm averages 5 top-performing networks from the last 30 epochs on each iteration.

**Table 1.** Final accuracy on the CIFAR datasets for different IDN patterns and rates.

Dataset	Noise info	SL [30]	LRT [39]	PLC [38]	Ours
CIFAR-10	$\mathcal{N}_{35\%}^{\text{I}}$	$79.76 \times 0.7$	$80.98 \times 0.8$	$82.80 \times 0.3$	<b><math>83.60 \times 0.3</math></b>
	$\mathcal{N}_{70\%}^{\text{I}}$	$36.29 \times 0.7$	$41.52 \times 4.5$	$42.74 \times 2.1$	<b><math>46.47 \times 1.1</math></b>
	$\mathcal{N}_{35\%}^{\text{II}}$	$77.92 \times 0.9$	$80.74 \times 0.3$	$81.54 \times 0.5$	<b><math>83.41 \times 0.3</math></b>
	$\mathcal{N}_{70\%}^{\text{II}}$	$41.11 \times 1.9$	$44.67 \times 3.9$	$46.04 \times 2.2$	<b><math>46.24 \times 0.9</math></b>
	$\mathcal{N}_{35\%}^{\text{III}}$	$78.81 \times 0.3$	$81.08 \times 0.4$	$81.50 \times 0.5$	<b><math>83.16 \times 0.3</math></b>
	$\mathcal{N}_{70\%}^{\text{III}}$	$38.49 \times 1.5$	$44.47 \times 1.2$	$45.05 \times 1.1$	<b><math>46.33 \times 1.1</math></b>
CIFAR-100	$\mathcal{N}_{35\%}^{\text{I}}$	$55.20 \times 0.3$	$56.74 \times 0.3$	$60.01 \times 0.4$	<b><math>63.85 \times 0.3</math></b>
	$\mathcal{N}_{70\%}^{\text{I}}$	$40.02 \times 0.9$	$45.29 \times 0.4$	$45.92 \times 0.6$	<b><math>46.38 \times 0.3</math></b>
	$\mathcal{N}_{35\%}^{\text{II}}$	$56.10 \times 0.7$	$57.25 \times 0.7$	$63.68 \times 0.3$	<b><math>63.91 \times 0.3</math></b>
	$\mathcal{N}_{70\%}^{\text{II}}$	$38.45 \times 0.6$	$43.71 \times 0.5$	$45.03 \times 0.5$	<b><math>46.63 \times 0.2</math></b>
	$\mathcal{N}_{35\%}^{\text{III}}$	$56.04 \times 0.7$	$56.57 \times 0.3$	$63.68 \times 0.3$	<b><math>63.92 \times 0.4</math></b>
	$\mathcal{N}_{70\%}^{\text{III}}$	$39.94 \times 0.8$	$44.41 \times 0.2$	$44.45 \times 0.6$	<b><math>46.22 \times 0.2</math></b>

Table 1 holds the results of testing the proposed method on synthetic data affected by three different IDN patterns with 35% and 70% noise rates. The

performance of baseline methods is obtained from [38]. As shown in this table, our method outperforms the alternatives in all cases. Judging by the numbers, some alternative approaches have a high standard deviation rate, indicating possible instability of that method.

**Table 2.** Final accuracy on the CIFAR datasets for different combinations of noise.

Dataset	Noise info	SL [30]	LRT [39]	PLC [38]	Ours
CIFAR-10	$\mathcal{N}_{35\%}^I + \mathcal{U}_{30\%}^{\text{Uniform}}$	$77.79 \times 0.5$	$75.97 \times 0.3$	$79.04 \times 0.5$	<b><math>80.94 \times 0.2</math></b>
	$\mathcal{N}_{35\%}^I + \mathcal{A}_{30\%}^{\text{Asymmetrical}}$	$77.14 \times 0.7$	$76.96 \times 0.5$	$78.31 \times 0.4$	<b><math>79.93 \times 0.5</math></b>
	$\mathcal{N}_{35\%}^{II} + \mathcal{U}_{30\%}^{\text{Uniform}}$	$75.08 \times 0.5$	$75.94 \times 0.6$	$80.08 \times 0.4$	<b><math>81.07 \times 0.2</math></b>
	$\mathcal{N}_{35\%}^{II} + \mathcal{A}_{30\%}^{\text{Asymmetrical}}$	$75.43 \times 0.4$	$77.03 \times 0.6$	$77.63 \times 0.3$	<b><math>79.90 \times 0.5</math></b>
	$\mathcal{N}_{35\%}^{III} + \mathcal{U}_{30\%}^{\text{Uniform}}$	$76.22 \times 0.1$	$75.66 \times 0.6$	$80.06 \times 0.5$	<b><math>80.54 \times 0.3</math></b>
	$\mathcal{N}_{35\%}^{III} + \mathcal{A}_{30\%}^{\text{Asymmetrical}}$	$76.09 \times 0.1$	$77.19 \times 0.7$	$77.54 \times 0.7$	<b><math>79.54 \times 0.5</math></b>
CIFAR-100	$\mathcal{N}_{35\%}^I + \mathcal{U}_{30\%}^{\text{Uniform}}$	$51.34 \times 0.6$	$45.66 \times 1.6$	$60.09 \times 0.2$	<b><math>61.46 \times 0.4</math></b>
	$\mathcal{N}_{35\%}^I + \mathcal{A}_{30\%}^{\text{Asymmetrical}}$	$50.18 \times 1.0$	$52.04 \times 0.2$	$56.40 \times 0.3$	<b><math>59.94 \times 0.4</math></b>
	$\mathcal{N}_{35\%}^{II} + \mathcal{U}_{30\%}^{\text{Uniform}}$	$50.58 \times 0.3$	$43.86 \times 1.3$	$60.01 \times 0.6$	<b><math>61.16 \times 0.3</math></b>
	$\mathcal{N}_{35\%}^{II} + \mathcal{A}_{30\%}^{\text{Asymmetrical}}$	$49.46 \times 0.2$	$52.11 \times 0.5$	<b><math>61.43 \times 0.3</math></b>	$59.34 \times 0.5$
	$\mathcal{N}_{35\%}^{III} + \mathcal{U}_{30\%}^{\text{Uniform}}$	$50.18 \times 0.5$	$42.79 \times 1.8$	$60.14 \times 1.0$	<b><math>61.82 \times 0.3</math></b>
	$\mathcal{N}_{35\%}^{III} + \mathcal{A}_{30\%}^{\text{Asymmetrical}}$	$48.15 \times 0.9$	$50.31 \times 0.4$	$54.56 \times 1.1$	<b><math>59.76 \times 0.5</math></b>

Table 2 holds the results of testing the proposed method on synthetic data simultaneously affected by IDN and CCN patterns. The final noise rate is typically lower than the sum of two individual noise rates due to overlaps in selected samples. As shown in this table, our method still outperforms the alternatives in almost all cases.

## 4.2 Real-World Datasets

To evaluate the performance of the proposed method on real-world cases, three commonly used datasets were chosen for testing:

**ANIMAL-10N** [26] – This dataset contains 50,000 training and 5,000 testing samples over ten categories. According to the creators of the dataset, the estimated noise rate is about 8%. Following the authors’ work, we chose VGG-19 [25] with a batch normalization for this experiment. The model is trained from scratch for 180 epochs with a batch size of 128 images. Stochastic gradient descent is used as the optimizer with a weight decay rate of  $1 \times 10^{-3}$ . The learning rate is initialized as  $1 \times 10^{-1}$  and gets divided by 5 after 50 and 75 epochs. Standard data augmentations are applied: random horizontal flip and standard normalizing with mean = (0.485, 0.456, 0.406), std = (0.229, 0.224, 0.225). 10% of the training data is manually labeled with the help of [1] and reserved as the validation set. The initial value for  $\theta$  in Algorithm 1 is set to  $7 \times 10^{-1}$  with a



decrement step of  $1 \times 10^{-1}$ . The algorithm averages 10 top-performing networks from the last 30 epochs on each iteration. Table 3 holds the results of testing the proposed method on the ANIMAL-10N dataset. The performance of baseline methods is obtained from their respective papers. As seen in this table, the proposed method outperforms the alternatives.

**Table 3.** Final accuracy on the Animal-10N and Food-101N datasets.

Dataset	Method	Accuracy	Dataset	Method	Accuracy
Animal-10N	SELFIE [26]	79.40	Food-101N	DeepSelf [11]	79.40
	Co-learning [28]	82.95		PLC [38]	83.40
	PLC [38]	83.40		Ours	86.34
	<b>Ours</b>	<b>84.47</b>		<b>Co-learning [28]</b>	<b>87.57</b>

**Food-101N** [16] – This dataset contains 310,000 training samples and utilizes the 25,000 testing samples provided by the Food-101 dataset [6] over 101 categories. According to the creators of the dataset, the estimated noise rate is about 10%. Following the authors’ work, we chose ResNet-50 with pre-trained weights on ImageNet [9] for this experiment. The model is fine-tuned for 30 epochs with a batch size of 32 images. Stochastic gradient descent is used as the optimizer with a weight decay rate of  $1 \times 10^{-3}$ . The learning rate is initialized as  $5 \times 10^{-3}$  and gets divided by 10 after 10 and 20 epochs. Standard data augmentations are applied: random horizontal flip,  $224 \times 224$  random crop, and standard normalizing with mean = (0.485, 0.456, 0.406), std = (0.229, 0.224, 0.225). 14% of the labels are verified by the creators of the dataset to be used as the validation set. The initial value for  $\theta$  in Algorithm 1 is set to  $9 \times 10^{-1}$  with a decrement step of  $1 \times 10^{-1}$ . The algorithm averages 4 top-performing networks from the last 8 epochs on each iteration. Table 3 holds the results of testing the proposed method on the Food-101N dataset. The performance of baseline methods is obtained from their respective papers. This table shows that the proposed method outperforms most of the alternatives but gets beaten by Co-Learning [28].

**Clothing1M** [16,34] – This dataset contains 1,000,000 samples over 14 categories, out of which 50,000 training, 14,000 validation, and 10,000 testing samples are verified by the creators of the dataset. Following the previous works [16,17,32], the clean training data is discarded. We chose ResNet-50 with pre-trained weights on ImageNet for this experiment. The model is fine-tuned for 20 epochs with a batch size of 32 images. Stochastic gradient descent is used as the optimizer with a momentum value equal to  $9 \times 10^{-1}$  and a weight decay rate of  $5 \times 10^{-4}$ . The learning rate is initialized as  $1 \times 10^{-3}$  and gets divided by 10 after 5 and 10 epochs. Standard data augmentations are applied: random horizontal flip,  $224 \times 224$  random crop, and standard normalizing with mean = (0.485, 0.456, 0.406), std = (0.229, 0.224, 0.225). The verified validation data is used

as the validation set. The initial value for  $\theta$  in Algorithm 1 is set to  $3 \times 10^{-1}$  with a decrement step of  $1 \times 10^{-1}$ . The algorithm averages 4 top-performing networks from the last 8 epochs on each iteration. Table 4 holds the results of testing the proposed method on the Clothing1M dataset. The performance of baseline methods is obtained from their respective papers. As seen in this table, the proposed method outperforms the alternatives.

**Table 4.** Final accuracy on the Clothing1M dataset.

Method	Accuracy
CAL [40]	74.17
Reweight [33]	74.18
DeepSelf [11]	74.45
CleanNet [16]	74.69
DivideMix [17]	74.76
<b>Ours</b>	<b>75.11</b>

## 5 Conclusion

This paper proposes a practical iterative label correction method that utilizes clean validation sets to achieve better performance when dealing with instance-dependent noise. The effectiveness of the proposed method is shown with empirical experiments on both synthetic and real-world benchmark datasets. The proposed method outperformed the current state-of-the-art methods in these experiments. The findings suggest that the proposed method’s intuition might be correct, and utilizing a clean validation set in iterative label correction methods is helpful.

## References

1. Adhikari, B., Huttunen, H.: Iterative bounding box annotation for object detection. In: 25th International Conference on Pattern Recognition (ICPR), pp. 4040–4046 (2021). <https://doi.org/10.1109/ICPR48806.2021.9412956>
2. Adhikari, B., Peltomäki, J., Germi, S.B., Rahtu, E., Huttunen, H.: Effect of label noise on robustness of deep neural network object detectors. In: Habli, I., Sujan, M., Gerasimou, S., Schoitsch, E., Bitsch, F. (eds.) SAFECOMP 2021. LNCS, vol. 12853, pp. 239–250. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-83906-2\\_19](https://doi.org/10.1007/978-3-030-83906-2_19)
3. Algan, G., Ulusoy, I.: Image classification with deep learning in the presence of noisy labels: a survey. *Knowl.-Based Syst.* **215** (2021). <https://doi.org/10.1016/j.knosys.2021.106771>
4. Arazo, E., Ortego, D., Albert, P., O’Connor, N., McGuinness, K.: Unsupervised label noise modeling and loss correction. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 312–321 (2019)

5. Berthon, A., Han, B., Niu, G., Liu, T., Sugiyama, M.: Confidence scores make instance-dependent label-noise learning possible. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 825–836 (2021)
6. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 - mining discriminative components with random forests. In: Computer Vision - ECCV, pp. 446–461. Proceedings of Machine Learning Research (2014)
7. Chen, P., Ye, J., Chen, G., Zhao, J., Heng, P.A.: Beyond class-conditional assumption: a primary attempt to combat instance-dependent label noise. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 13, pp. 11442–11450 (2021)
8. Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., Liu, Y.: Learning with instance-dependent label noise: a sample sieve approach (2021)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
10. Frenay, B., Verleysen, M.: Classification in the presence of label noise: a survey. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(5), 845–869 (2014). <https://doi.org/10.1109/TNNLS.2013.2292894>
11. Han, J., Luo, P., Wang, X.: Deep self-learning from noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
13. Hendrycks, D., Lee, K., Mazeika, M.: Using pre-training can improve model robustness and uncertainty. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 2712–2721 (2019)
14. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)
15. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998). <https://doi.org/10.1109/5.726791>
16. Lee, K.H., He, X., Zhang, L., Yang, L.: CleanNet: transfer learning for scalable image classifier training with label noise. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
17. Li, J., Socher, R., Hoi, S.C.H.: Dividemix: learning with noisy labels as semi-supervised learning (2020)
18. Li, J., Wong, Y., Zhao, Q., Kankanhalli, M.S.: Learning to learn from noisy labeled data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
19. Liu, Y., Guo, H.: Peer loss functions: learning from noisy labels without knowing noise rates. In: Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 6226–6236 (2020)
20. Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., Bailey, J.: Normalized loss functions for deep learning with noisy labels. In: Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 6543–6553 (2020)
21. Menon, A.K., van Rooyen, B., Natarajan, N.: Learning from binary labels with instance-dependent noise. *Mach. Learn.* 1561–1595 (2018). <https://doi.org/10.1007/s10994-018-5715-3>

22. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: a loss correction approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
23. Shi, J., Wu, J.: Distilling effective supervision for robust medical image segmentation with noisy labels (2021)
24. Shu, J., et al.: Meta-weight-net: learning an explicit mapping for sample weighting (2019)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
26. Song, H., Kim, M., Lee, J.G.: SELFIE: refurbishing unclean samples for robust deep learning. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 5907–5915 (2019)
27. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: a survey (2021)
28. Tan, C., Xia, J., Wu, L., Li, S.Z.: Co-learning: learning from noisy labels with self-supervision. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 1405–1413 (2019)
29. Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.: Learning from noisy large-scale datasets with minimal supervision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6575–6583 (2017). <https://doi.org/10.1109/CVPR.2017.696>
30. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J.: Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
31. Wei, H., Feng, L., Chen, X., An, B.: Combating noisy labels by agreement: a joint training method with co-regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
32. Wu, P., Zheng, S., Goswami, M., Metaxas, D., Chen, C.: A topological filter for learning with label noise (2020)
33. Xia, X., et al.: Are anchor points really indispensable in label-noise learning? In: Advances in Neural Information Processing Systems, vol. 32 (2021)
34. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
35. Yang, S., et al.: Estimating instance-dependent label-noise transition matrix using DNNs (2021)
36. Yi, K., Wu, J.: Probabilistic end-to-end noise correction for learning with noisy labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
37. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* **64**(3), 107–115 (2021). <https://doi.org/10.1145/3446776>
38. Zhang, Y., Zheng, S., Wu, P., Goswami, M., Chen, C.: Learning with feature-dependent label noise: a progressive approach (2021)
39. Zheng, S., et al.: Error-bounded correction of noisy labels. In: Proceedings of the 37th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 119, pp. 11447–11457 (2020)
40. Zhu, Z., Liu, T., Liu, Y.: A second-order approach to learning with instance-dependent label noise. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10113–10123 (2021)

# PUBLICATION

## IV

### **IFMix: Utilizing Intermediate Filtered Images for Domain Adaptation in Classification**

Saeed Bakhshi Germi and Esa Rahtu

In: *International Joint Conference on Computer Vision, Imaging and Computer Graphics  
Theory and Applications (VISAPP)*. Lisbon, Portugal: SciTePress, 2023

DOI: 10.5220/0011713600003417

Reprinted with the permission of the copyright holders and authors.



# IFMix: Utilizing Intermediate Filtered Images for Domain Adaptation in Classification

Saeed Bakhshi Germi<sup>a</sup> and Esa Rahtu<sup>b</sup>

*Computer Vision Group, Tampere University, Tampere, Finland*

**Keywords:** Domain Adaptation, Filtered Images, Classification, Mixup Technique.

**Abstract:** This paper proposes an iterative intermediate domain generation method using low- and high-pass filters. Domain shift is one of the prime reasons for the poor generalization of trained models in most real-life applications. In a typical case, the target domain differs from the source domain due to either controllable factors (e.g., different sensors) or uncontrollable factors (e.g., weather conditions). Domain adaptation methods bridge this gap by training a domain-invariant network. However, a significant gap between the source and the target domains would still result in bad performance. Gradual domain adaptation methods utilize intermediate domains that gradually shift from the source to the target domain to counter the effect of the significant gap. Still, the assumption of having sufficiently large intermediate domains at hand for any given task is hard to fulfill in real-life scenarios. The proposed method utilizes low- and high-pass filters to create two distinct representations of a single sample. After that, the filtered samples from two domains are mixed with a dynamic ratio to create intermediate domains, which are used to train two separate models in parallel. The final output is obtained by averaging out both models. The method's effectiveness is demonstrated with extensive experiments on public benchmark datasets: Office-31, Office-Home, and VisDa-2017. The empirical evaluation suggests that the proposed method performs better than the current state-of-the-art works.

## 1 INTRODUCTION

With the increasing popularity of deep learning algorithms in the heavy machine industry and the inclusion of artificial intelligence in new regulations (e.g., EU AI Act) and safety standards (e.g., ISO/IEC JTC 1/SC 42 Committee), the practical issues of utilizing such algorithms in safety-critical applications have become more apparent. One of the challenges for any practical application of a deep learning algorithm is collecting and labeling a large dataset for training the algorithm while considering the safety criteria for the application (Bakhshi Germi and Rahtu, 2022b). A standard method to deal with this issue is utilizing transfer learning (Zhuang et al., 2021), where the model is trained with a label-rich source dataset (e.g., synthesized or simulated data) and fine-tuned on a much smaller target dataset (e.g., data collected from the real world). However, a significant gap between these two domains would result in poor performance.

Gradual domain adaptation (GDA) deals with the gap problem by adding data from intermediate domains that interpolate between the source and the tar-

get domains (Kumar et al., 2020). The intermediate domains are assumed to be available with sufficient data for the training process. The accuracy of GDA methods is highly dependent on the distance between the source and the target domains. Moreover, GDA methods are usually unsupervised and do not require labels from intermediate or target domains. While unsupervised methods attract more attention in the research community, using a small labeled subset from the target domain is more realistic in real-world applications. Various annotation tools (Adhikari and Huttunen, 2021) and denoising techniques (Bakhshi Germi and Rahtu, 2022a) could be utilized to help with gathering the required labeled subset. Meanwhile, intermediate domains do not naturally exist for most real-world applications. Thus, this paper focuses on generating intermediate domains based on a large labeled source dataset and a small labeled target dataset.

This paper proposes IFMix, a domain adaptation algorithm that utilizes a filtered-image-based mixup technique to create intermediate domains iteratively. A new domain is created by merging the low-pass or high-pass filtered images from both domains with a dynamic ratio. The images are chosen from the same

<sup>a</sup> <https://orcid.org/0000-0003-3048-220X>

<sup>b</sup> <https://orcid.org/0000-0001-8767-0864>

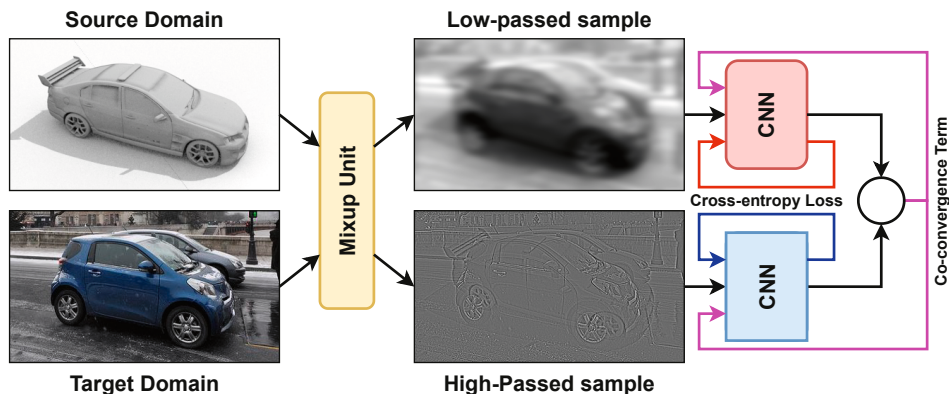


Figure 1: The overall structure of the proposed method. Two samples of the same category are chosen from two domains to be mixed. The mixup unit utilizes low-pass and high-pass filters to mix images with different ratios. The resulting images are used as training samples for two separate models. Each model is trained with a categorical cross-entropy loss. A co-convergence term is utilized to ensure the convergence of both models towards the same point.

category in both domains to keep the labels intact. After that, the proposed method utilizes the intermediate domains to train two separate models in parallel. Both models' average output is considered the proposed method's final output. The intuition behind the proposed method is that a supervised method that relies on a small amount of data from the target domain would be practical and realistic, the iterative domain creation would compensate for the lack of data in real-world applications, and the two models develop different perspectives based on their respective filters.

The main difference between the proposed method and previous works is utilizing a small labeled target dataset to create intermediate domains, resulting in accurate labels instead of pseudo-labels. Also, using the low- and high-pass filters would result in two distinct representations of the same sample, creating substantially different intermediate domains for training two different models. Moreover, the iterative and gradual nature of the algorithm ensures that the model is not overwhelmed by new information while the gap between the two domains is breached. The effectiveness of the proposed method is shown by comparing the performance with previous state-of-the-art methods in standard public benchmarks such as Office-31 (Saenko et al., 2010), Office-Home (Venkateswara et al., 2017), and VisDa-2017 (Peng et al., 2017). The main contributions of this paper are summarized as follows:

- Proposing an iterative intermediate domain creation technique based on filtered images to bridge the gap between the source and the target domains.

- Providing a practical domain adaptation algorithm based on the proposed intermediate domains.
- Providing empirical evaluation with extensive experiments on three standard benchmarks to show the effectiveness of the proposed method.

The rest of the paper is structured as follows. Section 2 covers the related works. Next, Section 3 explains the proposed method in detail. After that, Section 4 deals with the experiments and the empirical evaluation to show the effectiveness of the proposed method. Finally, Section 5 concludes the work.

## 2 RELATED WORKS

### 2.1 Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) methods utilize domain-invariant representation to generalize a model from a rich-labeled source domain to an unlabeled target domain (Wilson and Cook, 2020). The process can be done by either optimizing distribution discrepancy metrics (e.g., maximum mean discrepancy) (Li et al., 2021a; Peng et al., 2019) or utilizing adversarial training (Li et al., 2021b; Liu et al., 2019; Wang et al., 2019). On top of that, utilizing pseudo-labeling ideas from semi-supervised learning methods improves the performance of UDA algorithms (Chen et al., 2020; Liang et al., 2020; Liang et al., 2021; Liu et al., 2021; Zhang et al., 2021b). Moreover, the natural advantage of transformers in extracting transferable representations was studied further for application in domain adaptation (Ma et al.,





Figure 2: The creation of multiple intermediate domains by the proposed method. The shown samples are not filtered to understand better how the method works. Samples progress from the source domain (left) to the target domain (right) with each iteration based on the value of  $H$ .

2021; Xu et al., 2021; Yang et al., 2021a). Unsupervised methods have been the research focus for a while in academic applications. However, utilizing a small labeled dataset could result in a performance surge without significantly increasing the overall cost of gathering data.

## 2.2 Gradual Domain Adaptation

Gradual domain adaptation methods utilize intermediate domains to improve the performance of basic domain adaptation techniques (Choi et al., 2020; Cui et al., 2020; Dai et al., 2021; Hsu et al., 2020). GDA methods utilize generative models (e.g., generative adversarial networks) to create an intermediate domain by mixing the source and the target data at an arbitrary ratio (Sagawa and Hino, 2022). By doing so, the model can learn common features shared between two domains. In the original work, Kumar assumed that the intermediate domains gradually shift from the source to the target domain, and their sequence is known prior to learning (Kumar et al., 2020). However, the method is effective even if the sequence of these domains is unknown (Chen and Chao, 2021; Zhang et al., 2021a) or when no intermediate domain is available (Abnar et al., 2021; Na et al., 2021b). The main difference between current state-of-the-art GDA algorithms is their technique for creating intermediate domains.

## 2.3 Mixup Technique

Mixup techniques are a family of data augmentation methods based on mixing two or more data points. Mixup and its variants have proven helpful in supervised and semi-supervised learning (Berthelot et al., 2019; Yun et al., 2019; Zhang et al., 2017). Some recent domain adaptation methods tried utilizing this technique to create a continuous latent space across domains (Wu et al., 2020; Xu et al., 2020), obtain pseudo labels for intermediate domains (Na et al., 2021b; Yan et al., 2020; Yang et al., 2021b), or generate more positive/negative samples (Kalantidis et al., 2020; Zhang et al., 2022; Zhu et al., 2021).

This paper utilizes the intermediate domains from GDA, a mixup technique based on low- and high-

pass filters, and a small labeled subset from the target domain to achieve high performance in real-world scenarios. The assumptions in this paper are tailored around practical use cases of domain adaptation where a large labeled source domain and a small labeled target domain are available. While similar works exist in this field, the proposed method outperforms the existing state-of-the-art, as shown in Section 4.

## 3 PROPOSED METHOD

This section presents the details of the proposed method, as shown in Figure 1. Let  $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^n$  be the labeled dataset from the source domain,  $\mathcal{D}^t = \{(x_j^t)\}_{j=1}^m$  be the unlabeled dataset from the target domain, and  $\mathcal{D}_l^t = \{(x_k^t, y_k^t)\}_{k=1}^p$  be the labeled subset from the target domain. The task is transferring knowledge from  $\mathcal{D}^s$  to  $\mathcal{D}^t$  when there is a large distribution gap between them.

### 3.1 Iterative Filtered Mixup

The proposed method selects random samples with the same category label from  $\mathcal{D}^s$  and  $\mathcal{D}_l^t$ , applies low- and high-pass filters on them, and mixes them to create new samples as follows:

$$\begin{aligned} x_i^{lo} &= (1 - H) \times LoPass(x_i^s) + H \times LoPass(x_j^t) \\ x_i^{hi} &= (1 - H) \times HiPass(x_i^s) + H \times HiPass(x_j^t) \end{aligned} \quad (1)$$

Where  $(0 \leq H \leq 1)$  denotes a dynamic ratio for the mixing step,  $LoPass$  and  $HiPass$  denote the low-pass and high-pass filter functions, respectively. These filters could be implemented using the Gaussian filter function in the Multidimensional Image Processing package (scipy.ndimage). Moreover, the labels  $y_i^{lo}$  and  $y_i^{hi}$  for generated samples would be the same as the original label  $y_i$  due to choosing samples from the same category. Finally, the mixing ratio  $H$  is updated based on the number of epochs as follows:

$$H_{i+1} = H_i + \alpha \times t \quad (2)$$

Where  $\alpha$  is a positive constant and  $t$  is the current number of epochs. Two labeled datasets,  $\mathcal{D}_H^{lo}$  and  $\mathcal{D}_H^{hi}$ , are created with each iteration. These intermediate datasets fill the gap between the source and the target domains, as shown in Figure 2. Note that the figure shows unfiltered samples for a more straightforward interpretation of how the algorithm works.

### 3.2 Training and Loss Functions

In the next step, two models are trained on  $\mathcal{D}_H^{lo}$  and  $\mathcal{D}_H^{hi}$  using the categorical cross-entropy loss function:

$$\begin{aligned} \mathcal{L}_{cce}^{lo} &= \frac{1}{B} \sum_i y_i^{lo} \times \log \left( p \left( y | x_i^{lo} \right) \right) \\ \mathcal{L}_{cce}^{hi} &= \frac{1}{B} \sum_i y_i^{hi} \times \log \left( q \left( y | x_i^{hi} \right) \right) \end{aligned} \quad (3)$$

Where  $p(y|x_i^{lo})$  and  $q(y|x_i^{hi})$  denote the predicted class for each network on their respective input, and  $B$  is the batch size. The models are trained separately for a few epochs (warm-up period) to ensure they gain different perspectives without the influence of the other model.

### 3.3 Output and Co-Convergence Term

With each model training to recognize different characteristics of a given sample, their average output is used to determine the final output of the algorithm. Since the models should converge towards the same goal, a co-convergence term is added to the overall loss after the warm-up period. This term ensures that each model can influence the other model slightly to reach a similar conclusion on their output.

$$\mathcal{L}_{cct} = \frac{1}{B} \sum_i y_i \times \log \left( \frac{p(y|x_i^{lo}) + q(y|x_i^{hi})}{2} \right) \quad (4)$$

### 3.4 Overall Process

The overall process of the IFMix algorithm is summarized in Algorithm 1. The algorithm starts with creating the intermediate domains in each iteration. Then two networks are trained with the new intermediate domains using the defined loss functions. The co-convergence term is added after the warm-up period to allow the models to develop unique characteristics without the influence of the other model. In experiments, the mixup ratio  $H$  is updated every few epochs to prevent potential divergence of models.

Algorithm 1: IFMix Algorithm.

---

**Require:** Source dataset  $\mathcal{D}^s$ , Labeled Target subset  $\mathcal{D}_l^t$ , Number of epochs  $T$ , Batch size  $B$ , Warm-up period  $W$ , Mixup ratio  $H$ , Mixup increment rate  $\alpha$

- 1: **for**  $t \in 1, \dots, T$  **do**
- 2:   Select samples from same category in  $\mathcal{D}^s$  and  $\mathcal{D}_l^t$
- 3:   Create intermediate domains  $\mathcal{D}_H^{hi}$  and  $\mathcal{D}_H^{lo}$  using Eq. 1
- 4:   **for**  $b \in 1, \dots, B$  **do**
- 5:     Update loss functions  $\mathcal{L}_{cce}^{lo}$  and  $\mathcal{L}_{cce}^{hi}$  using Eq. 3
- 6:     **if**  $i \geq W$  **then**
- 7:       Update co-convergence term  $\mathcal{L}_{cct}$  using Eq. 4
- 8:     **end if**
- 9:   **end for**
- 10:   Update the mixup ratio using Eq. 2
- 11: **end for**

---

## 4 EXPERIMENTS & EVALUATION

To evaluate the proposed method, three different domain adaptation benchmarks are chosen so that the performance of the proposed method can be compared with state-of-the-art methods. In each experiment, 5% of samples from the target domain are selected as labeled target subsets for the proposed method, and the remaining 95% of samples are left as test data.

### 4.1 Office-31

Office-31 (Saenko et al., 2010), a domain adaptation benchmark, provides samples for 31 categories from three domains. These domains are denoted as  $A$  for images taken from Amazon.com,  $D$  for images taken with a DSLR camera, and  $W$  for images taken with a webcam. The dataset has around 4000 samples, making it a perfect benchmark for proof of concept.

### 4.2 Office-Home

Office-Home (Venkateswara et al., 2017), a domain adaptation benchmark, provides samples for 65 categories from four domains. These domains are denoted as  $A$  for arts and paintings,  $C$  for clipart images,  $P$  for product images without a background, and  $R$  for real-

Table 1: Accuracy (%) on the Office-31 dataset. The best accuracy is indicated in bold, and the second best is underlined.

Method	A → D	A → W	D → A	D → W	W → A	W → D	Average
GSDA (Hu et al., 2020)	94.8	95.7	73.5	99.1	74.9	<b>100</b>	89.7
SRDC (Tang et al., 2020)	95.8	95.7	76.7	99.2	77.1	<b>100</b>	90.8
RSDA (Gu et al., 2020)	95.8	96.1	77.4	<b>99.3</b>	78.9	<b>100</b>	91.1
FixBi (Na et al., 2021b)	95	96.1	<b>78.7</b>	<b>99.3</b>	79.4	<b>100</b>	91.4
CoVi (Na et al., 2021a)	<b>98</b>	<b>97.6</b>	77.5	<b>99.3</b>	78.4	<b>100</b>	91.8
IFMix (Ours)	<u>97.6</u>	<u>97.5</u>	<u>77.9</u>	<b>99.3</b>	<b>79.7</b>	<b>100</b>	<b>92</b>

Table 2: Accuracy (%) on the Office-Home dataset. The best accuracy is indicated in bold, and the second best is underlined.

Method	A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	Average
MetaAlign (Wei et al., 2021)	59.3	76	80.2	65.7	74.7	75.1	65.7	56.5	81.6	74.1	61.1	85.2	71.3
FixBi (Na et al., 2021b)	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
CoVi (Na et al., 2021a)	58.5	78.1	80	68.1	80	77	66.4	60.2	82.1	76.6	<b>63.6</b>	86.5	73.1
CDTrans (Xu et al., 2021)	60.6	79.5	82.4	75.6	81	82.3	72.5	56.7	84.4	77	59.1	85.5	74.7
WinTR (Ma et al., 2021)	65.3	<b>84.1</b>	<b>85</b>	<b>76.8</b>	<b>84.5</b>	84.4	73.4	60	85.7	<u>77.2</u>	63.1	86.8	<u>77.2</u>
IFMix (Ours)	<b>66.1</b>	<u>84</u>	<b>86.6</b>	<b>77.4</b>	<u>84.1</u>	<b>86.1</b>	<b>75.2</b>	<b>61.1</b>	<b>86.5</b>	<b>78.4</b>	62.8	<b>87.4</b>	<b>78</b>

Table 3: Accuracy (%) on the VisDa-2017 dataset. The best accuracy is indicated in bold, and the second best is underlined.

Method	Plane	Bike	Bus	Car	Horse	Knife	Motor	Human	Plant	Skate	Train	Truck	Average
CAN (Kang et al., 2019)	97	87.2	82.5	74.3	97.8	<u>96.2</u>	90.8	80.7	96.6	96.3	87.5	59.9	87.2
FixBi (Na et al., 2021b)	96.1	87.8	90.5	90.3	96.8	95.3	92.8	<b>88.7</b>	97.2	<u>94.2</u>	90.9	25.7	87.2
CDTrans (Xu et al., 2021)	97.1	90.5	82.4	77.5	96.6	96.1	93.6	<u>88.6</u>	<u>97.9</u>	86.9	90.3	<b>62.8</b>	88.4
CoVi (Na et al., 2021a)	96.8	85.6	88.9	88.6	97.8	93.4	91.9	87.6	96	93.8	93.6	48.1	88.5
WinTR (Ma et al., 2021)	<b>98.7</b>	<u>91.2</u>	<b>93</b>	<u>91.9</u>	<u>98.1</u>	96.1	<b>94</b>	72.7	97	<b>95.5</b>	<b>95.3</b>	57.9	90.1
IFMix (Ours)	<u>98.2</u>	<b>91.7</b>	<u>92.9</u>	<b>92.2</b>	<b>98.5</b>	<b>96.5</b>	<u>93.7</u>	88	<b>98</b>	<b>95.5</b>	<u>94.8</u>	<u>61.8</u>	<b>91.8</b>

world images taken with a camera. The dataset has around 15000 samples, making it a more challenging task than Office-31.

### 4.3 VisDa-2017

VisDa-2017 (Peng et al., 2017), a domain adaptation benchmark, provides samples for 12 categories from two domains, simulated and real-world. The dataset has around 280000 samples, making it a complex and realistic benchmark for domain adaptation problems.

### 4.4 Hyper-Parameters

In the experiments with Office datasets, ResNet-50 with stochastic gradient descent (SGD) is used as the base model. The initial learning rate is 0.001, the momentum is 0.9, the weight decay is 0.005, the initial mixup ratio is 0.05 with a 0.05 increment every 10 epochs, and the total number of epochs is 200. In the experiments with VisDA dataset, the base model is swapped to ResNet-101. The initial learning rate is 0.0001, the initial mixup ratio is 0.1 with a 0.1 increment every 5 epochs, and the total number of epochs is 50. In all experiments, the models utilize pre-trained weights on ImageNet (Russakovsky et al., 2015).

### 4.5 Results and Comparison

Table 1 holds the results for the Office-31 dataset. Six different tasks are experimented upon, and the results are compared with state-of-the-art methods.

The accuracy of state-of-the-art methods is obtained from their respective published papers. The results from each task indicate that the proposed method is competitive. The average accuracy of the proposed method is 92%, which is a slight improvement over the previous best method, CoVi (Na et al., 2021a). As stated before, the Office-31 dataset was utilized to prove that the proposed method works as intended, even if the improvement is slight and negligible.

Table 2 holds the results for the Office-Home dataset. Twelve different tasks are experimented upon, and the results are compared with state-of-the-art methods. Similar to previous experiments, the accuracy of state-of-the-art methods is obtained from their respective published papers. The results from each task indicate that the proposed method is still competitive. The average accuracy of the proposed method is 78%, which is an improvement over the previous best method, WinTR (Ma et al., 2021). Note that the proposed method outperformed CoVi (Na et al., 2021a), the previous best method on the Office-31 dataset, by 4.9% on average. This experiment offers more insight into the value of utilizing the proposed method. While the proposed method slightly outperforms the alternatives in this case, it also offers a more robust solution that works on different datasets.

Table 3 holds the results for the VisDa-2017 dataset. The results are compared on category and overall level. Similar to previous experiments, the accuracy of state-of-the-art methods is obtained from their respective published papers. The results from each category indicate that the proposed method is

operating as intended. The average accuracy of the proposed method is 91.8%, which is a significant improvement over the previous best method, WinTR (Ma et al., 2021). The proposed method offers a noticeable improvement in this experiment.

## 5 CONCLUSION

This paper proposed a practical domain adaptation method that utilizes a labeled subset from the target domain and low- and high-pass filters to create intermediate domains. The iterative creation of intermediate domains helps the model quickly adapt despite a significant gap between domains. The effectiveness of the proposed method is shown with empirical experiments on public benchmark datasets. The proposed method outperforms the current state-of-the-art methods by a noticeable margin while maintaining robustness over different datasets.

## ACKNOWLEDGEMENT

This research is part of a Ph.D. study co-funded by Tampere University and Forum for Intelligent Machines ry (FIMA).

## REFERENCES

- Abnar, S., Berg, R. v. d., Ghiasi, G., Dehghani, M., Kalchbrenner, N., and Sedghi, H. (2021). Gradual domain adaptation in the wild: When intermediate distributions are absent.
- Adhikari, B. and Huttunen, H. (2021). Iterative bounding box annotation for object detection. In *International Conference on Pattern Recognition (ICPR)*.
- Bakhshi Gerami, S. and Rahtu, E. (2022a). Enhanced data-recalibration: Utilizing validation data to mitigate instance-dependent noise in classification. In *Image Analysis and Processing (ICIAP)*.
- Bakhshi Gerami, S. and Rahtu, E. (2022b). A practical overview of safety concerns and mitigation methods for visual deep learning algorithms. In *Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI)*.
- Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*.
- Chen, H.-Y. and Chao, W.-L. (2021). Gradual domain adaptation without indexed intermediate domains. In *Advances in Neural Information Processing Systems*.
- Chen, M., Zhao, S., Liu, H., and Cai, D. (2020). Adversarial-learned loss for domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4).
- Choi, J., Choi, Y., Kim, J., Chang, J., Kwon, I., Gwon, Y., and Min, S. (2020). Visual domain adaptation by consensus-based transfer to intermediate domain. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(7).
- Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., and Tian, Q. (2020). Gradually vanishing bridge for adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Dai, Y., Liu, J., Sun, Y., Tong, Z., Zhang, C., and Duan, L.-Y. (2021). Idm: An intermediate domain module for domain adaptive person re-id. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Gu, X., Sun, J., and Xu, Z. (2020). Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hsu, H.-K., Yao, C.-H., Tsai, Y.-H., Hung, W.-C., Tseng, H.-Y., Singh, M., and Yang, M.-H. (2020). Progressive domain adaptation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.
- Hu, L., Kan, M., Shan, S., and Chen, X. (2020). Unsupervised domain adaptation with hierarchical gradient synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kalantidis, Y., Sariyildiz, M. B., Pion, N., Weinzaepfel, P., and Larlus, D. (2020). Hard negative mixing for contrastive learning. In *Advances in Neural Information Processing Systems*.
- Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. (2019). Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kumar, A., Ma, T., and Liang, P. (2020). Understanding self-training for gradual domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning (PMLR)*.
- Li, S., Liu, C. H., Lin, Q., Wen, Q., Su, L., Huang, G., and Ding, Z. (2021a). Deep residual correction network for partial domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7).
- Li, S., Xie, M., Lv, F., Liu, C. H., Liang, J., Qin, C., and Li, W. (2021b). Semantic concentration for domain adaptation. In *Proceedings of the IEEE/ICCV International Conference on Computer Vision (ICCV)*.
- Liang, J., Hu, D., and Feng, J. (2020). Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning (PMLR)*.

- Liang, J., Hu, D., and Feng, J. (2021). Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Liu, H., Long, M., Wang, J., and Jordan, M. (2019). Transferable adversarial training: A general approach to adapting deep classifiers. In *Proceedings of the 36th International Conference on Machine Learning (PMLR)*.
- Liu, H., Wang, J., and Long, M. (2021). Cycle self-training for domain adaptation. In *Advances in Neural Information Processing Systems*.
- Ma, W., Zhang, J., Li, S., Liu, C. H., Wang, Y., and Li, W. (2021). Exploiting both domain-specific and invariant knowledge via a win-win transformer for unsupervised domain adaptation.
- Na, J., Han, D., Chang, H. J., and Hwang, W. (2021a). Contrastive vicinal space for unsupervised domain adaptation.
- Na, J., Jung, H., Chang, H. J., and Hwang, W. (2021b). Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. (2019). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. (2017). Visda: The visual domain adaptation challenge.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., and et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3).
- Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*.
- Sagawa, S. and Hino, H. (2022). Gradual domain adaptation via normalizing flows.
- Tang, H., Chen, K., and Jia, K. (2020). Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Wang, X., Li, L., Ye, W., Long, M., and Wang, J. (2019). Transferable attention for domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1).
- Wei, G., Lan, C., Zeng, W., and Chen, Z. (2021). Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Wilson, G. and Cook, D. J. (2020). A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology*, 11(5).
- Wu, Y., Inkpen, D., and El-Roby, A. (2020). Dual mixup regularized learning for adversarial domain adaptation. In *European Conference on Computer Vision (ECCV)*.
- Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., and Zhang, W. (2020). Adversarial domain adaptation with domain mixup. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4).
- Xu, T., Chen, W., Wang, P., Wang, F., Li, H., and Jin, R. (2021). Cdtrans: Cross-domain transformer for unsupervised domain adaptation.
- Yan, S., Song, H., Li, N., Zou, L., and Ren, L. (2020). Improve unsupervised domain adaptation with mixup training.
- Yang, G., Tang, H., Zhong, Z., Ding, M., Shao, L., Sebe, N., and Ricci, E. (2021a). Transformer-based source-free domain adaptation.
- Yang, L., Wang, Y., Gao, M., Shrivastava, A., Weinberger, K. Q., Chao, W.-L., and Lim, S.-N. (2021b). Deep co-training with task decomposition for semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). Mixup: Beyond empirical risk minimization.
- Zhang, Y., Deng, B., Jia, K., and Zhang, L. (2021a). Gradual domain adaptation via self-training of auxiliary models.
- Zhang, Y., Li, J., and Wang, Z. (2022). Low-confidence samples matter for domain adaptation.
- Zhang, Y., Wang, Z., and Mao, Y. (2021b). Rpn prototype alignment for domain adaptive object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhu, R., Zhao, B., Liu, J., Sun, Z., and Chen, C. W. (2021). Improving contrastive learning by visualizing feature transformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2021). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1).





