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 Boult 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 Germi, 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 Boult, 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.; Heinzemann, 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 Woehrle, M. 2021. Inspect, 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)*.

Kläs, 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.; Cârlan, 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.