Inspect, Understand, Overcome: A Survey of Practical Methods for AI Safety

Sebastian Houben[1], Stephanie Abrecht[2], Maram Akila[1], Andreas Bär[15], Felix Brockherde[10], Patrick Feifel[8], Tim Fingscheidt[15], Sujan Sai Gannamaneni[1], Seyed Eghbal Ghobadi[8], Ahmed Hammam[8], Anselm Haselhoff[9], Felix Hauser[11], Christian Heinzemann[2], Marco Hoffmann[16], Nikhil Kapoor[7], Falk Kappel[13], Marvin Klingner[15], Jan Kronenberger[9], Fabian Küppers[9], Jonas Löhdefink[15], Michael Mlynarski[16], Michael Mock[1], Firas Mualla[13], Svetlana Pavlitskaya[14], Maximilian Poretschkin[1], Alexander Pohl[16], Varun Ravi-Kumar[4], Julia Rosenzweig[1], Matthias Rottmann[5], Stefan Rüping[1], Timo Sämann[4], Jan David Schneider[7], Elena Schulz[1], Gesina Schwalbe[3], Joachim Sicking[1], Toshika Srivastava[12], Serin Varghese[7], Michael Weber[14], Sebastian Wirkert[6], Tim Wirtz[1], and Matthias Woehrle[2]

[1] *Fraunhofer Institute for Intelligent Analysis and Information Systems*
[2] *Robert Bosch GmbH*
[3] *Continental AG*
[4] *Valeo S.A.*
[5] *University of Wuppertal*
[6] *Bayerische Motorenwerke AG*
[7] *Volkswagen AG*
[8] *Opel Automobile GmbH*
[9] *Hochschule Ruhr West*
[10] *umlaut AG*
[11] *Karlsruhe Institute of Technology*
[12] *Audi AG*
[13] *ZF Friedrichshafen AG*
[14] *FZI Research Center for Information Technology*
[15] *Technische Universität Braunschweig*
[16] *QualityMinds GmbH*

The use of deep neural networks (DNNs) in safety-critical applications like mobile health and autonomous driving is challenging due to numerous model-inherent shortcomings. These shortcomings are diverse and range from a lack of generalization over insufficient interpretability to problems with malicious inputs. Cyber-physical systems employing DNNs are therefore likely to suffer from *safety concerns*. In recent years, a zoo of state-of-the-art techniques aiming to address these safety concerns has emerged. This work provides a structured and broad overview of them. We first identify categories of insufficiencies to then describe research activities aiming at their detection, quantification, or mitigation. Our paper addresses both machine learning experts and safety engineers: The former ones might profit from the broad range of machine learning (ML) topics covered and discussions on limitations of recent methods. The latter ones might gain insights into the specifics of modern ML methods. We moreover hope that our contribution fuels discussions on desiderata for ML systems and strategies on how to propel existing approaches accordingly.

Contents

1 Introduction

Sebastian Houben[1], Michael Mock[1], Timo Sämann[4], Gesina Schwalbe[3], Joachim Sicking[1]

In barely a decade, deep neural networks (DNNs) have revolutionized the field of machine learning by reaching unprecedented, sometimes superhuman, performances on a growing variety of tasks. Many of these neural models have found their way into consumer applications like smart speakers, machine translation engines or content feeds. However, in safety-critical systems, where human life might be at risk, the use of recent DNNs is challenging as various model-immanent insufficiencies are yet difficult to address.
This paper summarizes the promising lines of research in how to identify, address, and at least partly mitigate these DNN insufficiencies. While some of the reviewed works are theoretically grounded and foster the overall understanding of training and predictive power of DNNs, others provide practical tools to adapt their development, training or predictions. We refer to any such method as a *safety mechanism* if it addresses one or several safety concerns in a feasible manner. Their effectiveness in mitigating safety concerns is assessed by *safety metrics* [OOAG19, CNH$^+$18, SS20a, BGS$^+$19]. As most safety mechanisms target only a particular insufficiency, we conclude that a *holistic safety argumentation*[SSH20, SS20a, BGS$^+$19, WSRA20] for a complex DNN-based systems will in many cases rely on a variety of safety mechanisms.
We structure our review of these mechanisms as follows: Chapter 2 focuses on *dataset optimization* for network training and evaluation. It is motivated by the well-known fact that, in comparison to humans, DNNs perform poorly on data that is structurally different from training data. Apart from insufficient generalization capabilities of these models, the data acquisition process and distributional data shifts over time play vital roles. We survey potential counter-measures, e.g., augmentation strategies and outlier detection techniques.
Mechanisms that improve on *robustness* are described in Chapters 3 and 4, respectively. They deserve attention as DNNs are generally not resilient to common perturbations and adversarial attacks.
Chapter 5 addresses incomprehensible network behavior and reviews mechanisms that aim at *explainability*, e.g., a more transparent functioning of DNNs.
Moreover, DNNs tend to overestimate their prediction confidence, especially on unseen data. Straightforward ways to estimate prediction confidence yield mostly unsatisfying results. Among others, this observation fuelled research on more sophisticated *uncertainty estimations* (see Chapter 6), *redundancy mechanisms* (see Chapter 7) and attempts to reach *formal verification* as addressed in Chapter 8.
At last, many safety-critical applications require not only accurate but also near real-time decisions. This is covered by mechanisms on the DNN *architectural level* (see Chapter 9) and furthermore by *compression* and *quantization* methods (see Chapter 10).
We conclude this review of mechanism categories with an outlook on the steps to transfer a carefully arranged combination of safety mechanisms into an actual holistic safety argumentation.

2 Dataset Optimization

Matthias Rottmann[5]

The performance of a trained model inherently relies on the nature of the underlying dataset. For instance, a dataset with poor variability will hardly result in a model ready for real-world applications. In order to approach the latter, data selection processes such as corner case selection and active learning are of utmost importance. These approaches can help to design datasets that contain the most important information, while preventing the so much desired information from getting lost in an ocean of data. For a given dataset and active learning setups, data augmentation techniques are very common aiming at extracting as much model performance out of the dataset as possible.
On the other hand, safety arguments also require the analysis of how a model behaves on out-of-distribution data, data that contains concepts the model has not encountered during training. This is quite likely to happen as our world is under constant change, in other words exposed to a constantly growing domain shift. Therefore, these fields are lately gaining interest, also with respect to perception in automated driving.

2.1 Outlier/Anomaly Detection

Sujan Sai Gannamaneni[1], Matthias Rottmann[5]

The terms anomaly, outlier and *out-of-distribution* (OOD) data detection are often used interchangeably in literature and refer to task of identifying data samples that are not representative of training data distribution. Uncertainty evaluation (cf. Chapter 6) is closely tied to this field as self-evaluation of models is one of the active areas of research for OOD detection. In particular, for image classification problems it has been reported that neural networks often produce high confidence predictions on OOD data [NYC15, HG17]. The detection of such OOD inputs can either be tackled by post-processing techniques that adjust the estimated confidence [LLS18, DT18] or by enforcing low confidence on OOD samples during training [HAB19, HMD19]. Even guarantees that neural networks produce low confidence predictions for OOD samples can be provided under specific assumptions (cf. [MH20b]). More precisely, this work utilizes Gaussian mixture models that, however, may suffer from high-dimensional data and require strong assumptions on the distribution parameters. Some approaches use generative models like *GANs* [SSW+17, AAAB18] and *autoencoders* [ZP17] for outlier detection. The models are trained to learn in-distribution data manifolds and will produce higher reconstruction loss for outliers.
For OOD detection in semantic segmentation, only a few works have been presented so far. Angus et al. [ACS19] present a comparative study of common OOD detection methods, which mostly deal with image-level classification. In addition, they provide a novel setup of relevant OOD datasets for this task. Another work trains a fully convolutional binary

classifier that distinguishes image patches from a known set of classes from image patches stemming from an unknown class [BKOŠ18]. The classifier output applied at every pixel will give the per-pixel confidence value for an OOD object. Both of these works perform at pixel level and without any sophisticated feature generation methods specifically tailored for the detection of entire OOD instances. Up to now, outlier detection has not been studied extensively for object detection tasks based on benchmark object detection datasets. In [GBA+19], two CNNs are used to perform object detection and binary classification (benign or anomaly) in a sequential fashion, where the second CNN takes the localized object within the image as input.
From a safety standpoint, detecting outliers or OOD samples is extremely important and beneficial as training data cannot realistically be large enough to capture all situations. Research in this area is heavily entwined with progress in uncertainty estimation (cf. Chapter 6) and domain adaptation (cf. Sec. 2.3). Extending research works to segmentation and object detection tasks would be particularly significant for leveraging autonomous driving research. In addition to safety, OOD detection can be beneficial in other aspects like when using local expert models. For example, when using an expert model for segmentation of urban driving scenes and another expert model for segmentation of highway driving scenes, an OOD detector could act as trigger on which models can be switched.
With respect to the approaches presented above, uncertainty-based and generative model-based OOD detection methods are currently promising directions of research. However, it remains an open question whether they can unfold their potential well on segmentation and object detection tasks.

2.2 Active Learning

Matthias Rottmann[5]

It is widely known that, as a rule of thumb, for the training of any kind of artificial neural network, an increase of training data leads to increased performance. Obtaining labeled training data, however, is often very costly and time consuming. *Active learning* provides one possible remedy to this problem: Instead of labeling every data point, active learning utilizes a *query strategy* to request labels from a teacher/an oracle which leverage the model performance most. The survey paper by Settles [Set10] provides a broad overview regarding query strategies for active learning methods. However, except for *uncertainty sampling* and *query by committee*, most of them seem to be infeasible in deep learning applications up to now. Hence, most of the research activities in active deep learning focus on these two query strategies, as we outline in the following.
It has been shown [GIG17, RKG18] for image classification that labels corresponding to uncertain samples can leverage the networks' performance significantly and that a combination with semi-supervised learning is promising. In both works, uncertainty of unlabeled samples is estimated via Monte Carlo (MC) dropout inference. MC dropout inference and a chosen number of training epochs are executed alternatingly, after

performing MC dropout inference, the unlabeled samples' uncertainties are assessed by means of sample-wise dispersion measures. Samples for which the DNN model is very uncertain about its prediction are presented to an oracle and labeled.
With respect to object detection, a moderate number of active learning methods has been introduced [BKD19, KLSL19, RUN18, DCG+19]. These approaches include uncertainty sampling [BKD19, KLSL19] and query-by-committee methods [RUN18]. In [KLSL19, DCG+19], additional algorithmic features specifically tailored for object detection networks are presented, i.e., separate treatment of the localization and classification loss [KLSL19], as well as weak and strong supervision schemes [DCG+19]. For semantic segmentation, an uncertainty-sampling-based approach has been presented [MLG+18], which queries polygone masks for image sections of a fixed size (128×128). Queries are performed by means of accumulated entropy in combination with a cost estimation for each candidate image section.
Recently, new methods for estimating the quality of a prediction [DT18, RCH+18] as well as new uncertainty quantification approaches, e.g., gradient-based ones [ORG18], have been proposed. It remains an open question whether they are suitable for active learning. Since most of the conducted studies are rather of academic nature, also their applicability to real-life data acquisition is not yet demonstrated sufficiently. In particular, it is not clear whether the proposed active learning schemes, including the label acquistion, for instance in semantic segmentation, is suitable to be performed by human labelers. Therefore, labeling acquisition with a common understanding of the labelers' convenience and suitability for active learning are a promising direction for research and development.

2.3 Domains

Julia Rosenzweig[1]

The classical assumption in machine learning is that the training and testing data sets are drawn from the same distribution, implying that the model is deployed under the same conditions as it was trained under. However, as [MTRA+12, JDCR12] mention, in real-world applications this assumption is often violated in the sense that the training and the testing set stem from different domains having different distributions. This poses difficulties for statistical models and the performance will mostly degrade when they are deployed on a domain D^{test}, having a different distribution than the training dataset (i.e., generalizing from the training to the testing domain is not possible). This makes the study of domains not only relevant from the machine learning perspective, but also from a safety point of view.
More formally, there are differing notions of a 'domain' in literature. For [Csu17b, MD18], a domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ consists of a feature space $\mathcal{X} \subset \mathbb{R}^d$ together with a marginal probability distribution $P(X)$ with $X \in \mathcal{X}$. In [BCK+08, BDBC+10], a domain is a pair consisting of a distribution over the inputs together with a labeling function. However, instead of a sharp labeling function, it is also widely accepted to define a (training)

domain $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ to consist of n (labeled) samples that are sampled from a joint distribution $P(x, y)$ (cf. [LCWJ18]).
The reasons for *distributional shift* are diverse—as are the names to indicate a shift. For example, if the rate of (class) images of interest is different between training and testing set this can lead to a domain gap and, e.g., result in differing overall error rates. Moreover, as [CLS+18] mentions, changing weather conditions and camera setups in cars lead to a domain mismatch in applications of autonomous driving. In biomedical image analysis, different imaging protocols and diverse anatomical structures can hinder generalization of trained models (cf. [DCO+19, KBL+17]). Common terms to indicate distributional shift are *domain shift, dataset shift, covariate shift, concept drift, domain divergence, data fracture, changing environments* or *dataset bias*. References [Sto09, MTRA+12] provide an overview.
Methods and measures to overcome the problem of domain mismatch between one or more (cf. [ZZW+18]) source domains and target domain(s) and the resulting poor model performance are studied in the field of transfer learning and in particular its subtopic domain adaptation (cf. [MD18]). For instance, adapting a model that is trained on synthetically generated data to work on real data is one of the core challenges, as can be seen [CLS+18, LZG+19, VJB+19]. Furthermore, detecting when samples are out-of-domain or out-of-distribution is an active field of research (cf. [LLLS18] and the outlier/anomaly detection in Sec. 2.1 as well as the topic of observers in the black-box methods in Sec. 8.2 for further reference). This is particularly relevant for machine learning models that operate in the real world: If, e.g., an autonomous vehicle encounters some situation that deviates strongly from what was seen during training (e.g., due to some special event like a biking competition, carnival, etc.) this can lead to wrong predictions and thereby potential safety issues if not detected in time.

2.4 Augmentation

Falk Kappel[13]

Given the need for big amounts of data to train neural networks, one often runs into a situation where data is lacking. This can lead to insufficient generalization and an overfitting to the training data. An overview over different techniques to tackle this challenge can be found in [KGC17]. One approach to try and overcome this issue is the augmentation of data. It aims at optimizing available data and increasing its amount, curating a dataset that represents a wide variety of possible inputs during deployment. Augmentation can as well be of help when having to work with a heavily unbalanced dataset by creating more samples of underrepresented classes. A broad survey on data augmentation is provided by [SK19]. They distinguish between two general approaches to data augmentation with the first one being data warping augmentations that focus on taking existing data and transforming it in a way that does not effect labels. The other option are oversampling augmentations, which create synthetic data that can be used to increase the size of the dataset.

Examples of some of the most basic augmentations are flipping, cropping, rotating, translating, shearing and zooming. These are affecting the geometric properties of the image and are easily implemented [SK19]. The machine learning toolkit `Keras`, for example, provides an easy way of applying them to data using their `ImageDataGenerator` class [C+15]. Other simple methods include adaptations in color space that affect properties such as lighting, contrast and tints, which are common variations within image data. Filters can be used to control increased blur or sharpness [SK19]. In [ZZK+17] random erasing is introduced as a method with similar effect as cropping, aiming at gaining robustness against occlusions. An example for mixing images together as an augmentation technique can be found in [Ino18].
The abovementioned methods have in common that they work on the input data but there are different approaches that make use of deep learning for augmentation. An example for making augmentations in feature space using autoencoders can be found in [DT17]. They use the representation generated by the encoder and generate new samples by interpolation and extrapolation between existing samples of a class. The lack of interpretability of augmentations in feature space in combination with the tendency to perform worse than augmentations in image space present open challenges for those types of augmentations [SK19, WGSM16]. Adversarial training is another method that can be used for augmentation. The goal of adversarial training is to discover cases that would lead to wrong predictions. That means the augmented images won't necessarily represent samples that could occur during deployment but that can help in achieving more robust decision boundaries [SK19]. An example of such an approach can be found in [LCPB18]. Generative modelling can be used to generate synthetic samples that enlarge the dataset in a useful way with GANs, variational autoencoders and the combination of both are important tools in this area [SK19]. Examples for data augmentation in medical context using a CycleGAN [ZPIE17] can be found in [SYPS19] and using a progressively growing GAN [KALL17] in [BCG+18]. Next to neural style transfer [GEB15] that can be used to change the style of an image to a target style, AutoAugment [CZM+19] and population based augmentation [HLS+19] are two more interesting publications. In the latter two, the idea is to search a predefined search space of augmentations to gather the best selection.

2.5 Corner Case Detection

Alexander Pohl[16], Marco Hoffmann[16], Michael Mlynarski[16], Timo Sämann[4]

Ensuring that AI-based applications behave correctly and predictably even in unexpected or rare situations is a major concern that gains importance especially in safety-critical applications such as autonomous driving. In the pursuit of more robust AI corner cases play an important role.
The meaning of the term corner case varies in the literature. Some consider mere erroneous or incorrect behavior as corner cases [ZHML19, TPJR18, PCYJ17]. For example, in [BBLF19] corner cases are referred to as situations in which an object detector fails to

detect relevant objects at relevant locations. Others characterize corner cases mainly as rare combinations of input parameter values [HDHH20, KKB18]. This project adopts the first definition: Inputs that result in unexpected or incorrect behaviour of the AI function are defined as corner cases.
Contingent on the hardware, the AI architecture and the training data, the search space of corner cases quickly becomes incomprehensibly large. While manual creation of corner cases (e.g., constructing or re-enacting scenarios) might be more controllable, approaches that scale better and allow for a broader and more systematic search for corner cases require extensive automation.
One approach to automatic corner case detection is based on transforming the input data. The *DeepTest* framework [TPJR18] uses three types of image transformations: linear, affine and convolutional transformations. In addition to these transformations, metamorphic relations help detect undesirable behaviors of deep learning systems. They allow changing the input while asserting some characteristics of the result [XHM+11]. For example, changing the contrast of input frames should not affect the steering angle of a car [TPJR18]. Input-output pairs that violate those metamorphic relations can be considered as corner cases.
Among other things, the white-box testing framework *DeepXplore* [PCYJ17] applies a method called *gradient ascent* to find corner cases (cf. Sec. 8.1). In the experimental evaluation of the framework, three variants of deep learning architectures were used to classify the same input image. The input image was then changed according to the gradient ascent of an objective function that reflected the difference in the resulting class probabilities of the three model variants. When the changed (now artificial) input resulted in different class label predictions by the model variants, the input was considered as a corner case.
In [BBLF19], corner cases are detected on video sequences by comparing predicted with actual frames. The detector has three components: The first component, semantic segmentation, is used to detect and locate objects in the input frame. As the second component, an image predictor trained on frame sequences predicts the actual frame based on the sequence preceding that frame. An error is determined by comparing the actual with the predicted (i.e., expected) frame, following the idea that only situations that are unexpected for AI-based perception functions may be potentially dangerous and therefore a corner case. Both the segmentation and the prediction error are then fed into the third component of the detector, which determines a corner case score that reflects the extent to which unexpected relevant objects are at relevant locations.
In [HDHH20], a corner case detector based on simulations in a *Carla* environment [DRC+17] is presented. In the simulated world, AI agents control the vehicles. During simulations, state information of both the environment and the AI agents are fed into the corner case detector. While the environment provides the real vehicle states, the AI agents provide estimated and perceived state information. Both sources are then compared to detect conflicts (e.g., collisions). These conflicts are recorded for analysis.
Several ways of automatically generating and detecting corner cases exist. However, corner case detection is a task with challenges of its own: Depending on the operational domain including its boundaries, the space of possible inputs can be very large. Also,

some types of corner cases are specific to the AI architecture, e.g., the network type or the network layout used. Thus, corner case detection has to assume a holistic point of view on both model and input, adding further complexity and reducing transferability of previous insights.
Although it can be argued that rarity does not necessarily characterize corner cases, rare input data might have the potential of challenging the AI functionality (cf. Sec. 2.1). Another research direction could investigate whether structuring the input space in a way suitable for the AI functionality supports the detection of corner cases. Provided that the operational domain is conceptualized as an ontology, ontology-based testing [BMM18] may support automatic detection. A properly adapted generator may specifically select promising combinations of extreme parameter values and, thus, provide valuable input for synthetic test data generation.

3 Robust Training

Nikhil Kapoor[7]

Recent works [AW18, HD19, RSFD16, ETTS19, BRW18, BHSFs19, FF15] have shown that state-of-the-art deep neural networks (DNNs) performing a wide variety of computer vision tasks such as image classification [KSH12, HZRS15, MGR+18], object detection [Gir15, RDGF15, HGDG17] and semantic segmentation [CPSA17, ZSR+19, WSC+19, LBS+19] are not robust to small changes in the input.
Robustness of neural networks is an active and open research field that can be considered highly relevant for achieving safety in autonomous driving. Currently, most of the research is directed towards either improving adversarial robustness [SZS+14] (robustness against carefully designed perturbations that aim at causing misclassifications with high confidence), or improving corruption robustness [HD19] (robustness against commonly occurring augmentations such as weather changes, addition of Gaussian noise, photometric changes, etc.). While adversarial robustness might be more of a security issue than a safety issue, corruption robustness, on the other hand, can be considered highly safety-relevant. Equipped with these definitions, we broadly term *robust training* here as methods or mechanisms that aim at improving either adversarial or corruption robustness of a DNN, by incorporating modifications into the architecture or into the training mechanism itself.

3.1 Hyperparameter Optimization

Seyed Eghbal Ghobadi[8], Patrick Feifel[8]

The final performance of a neural network depends highly on the learning process. The process includes the actual optimization and may additionally introduce training methods

such as dropout, regularization, or parametrization of a multi-task loss.
These methods adapt their behavior for predefined parameters. Hence, their optimal configuration is a priori unknown. We refer to them as *hyperparameters*. Important hyperparameters comprise, for instance, the initial learning rate, steps for learning rate reduction, learning rate decay, momentum, batch size, dropout rate and number of iterations. Their configuration has to be determined according to the architecture and task of the CNN [HKV19]. The search of an optimal hyperparameter configuration is called hyperparameter optimization (HO).
HO is usually described as an optimization problem [HKV19]. Thereby, the combined configuration space is defined as $\mathbf{\Lambda} = \lambda_1 \times \lambda_2 \times \cdots \lambda_N$, according to each domain λ_n. Their individual spaces can be continuous, discrete, categorical or binary.
Hence, we aim to find an optimal hyperparameter configuration $\lambda^\star$ by minimizing an objective function $\mathcal{O}()$, which evaluates a trained model $\mathcal{M}$ on the validation dataset $\mathcal{D}^{\text{val}}$ with the loss $\mathcal{L}$:

$$\lambda^\star = \arg\min_{\lambda \in \mathbf{\Lambda}} \mathcal{O}\left(\mathcal{L}, \mathcal{M}_\lambda, \mathcal{D}^{\text{train}}, \mathcal{D}^{\text{val}}\right) \tag{1}$$

This problem statement is widely regarded in traditional machine learning and primarily based on Bayesian optimization (BO) in combination with Gaussian processes. However, a straightforward application to deep neural networks encounters problems due to a *lack of scalability, flexibility and robustness* [ZCY+19], [FKH18].
To exploit the benefits of BO, many authors proposed different combinations with other approaches. Hyperband [LJD+17] in combination with BO (BOHB) [FKH18] frames the optimization as "... a pure exploration non-stochastic infinite-armed bandit problem ...". The method of BO for iterative learning (BOIL) [NSO19] internalizes iteratively collected information about the learning curve and the learning algorithm itself. The authors of [WTPFW19] introduce the trace-aware knowledge gradient (taKG) as an acquisition function for BO (BO-taKG) which "leverages both trace information and multiple fidelity controls". Thereby BOIL and BO-taKG achieve state-of-research performance regarding CNNs outperforming Hyperband.
Other approaches such as the orthogonal array tuning method (OATM) [ZCY+19] or HO by reinforcement learning (Hyp-RL) [JGST19] turn away from the Bayesian approaches and offer new research directions.
Finally, the insight that many authors include kernel sizes and number of kernels and layers in their hyperparameter configuration should be emphasized. More work should be spent on the distinct integration of HO in the performance estimation strategy of neural architecture search (cf. Sec. 9.3).

3.2 Modification of Loss

Nikhil Kapoor[7]

There exist many approaches that aim at directly modifying the loss function with an objective of improving either adversarial or corruption robustness [Sea20, PLZ19,

XCKW19, IS19, TP18, HK18, WSC+19]. One of the earliest approaches for improving corruption robustness was introduced by Zheng et al. [ZSLG16] called *stability training*, where they introduce a regularization term that penalizes the network prediction to a clean and an augmented image. However, their approach does not scale to many augmentations at the same time. Janocha et al. [JC17] then introduced a detailed analysis on the influence of multiple loss functions to model performance as well as robustness and suggested that expectation-based losses tend to work better with noisy data and squared-hinge losses tend to work better for clean data. Other well-known approaches are mainly based on variations of data augmentation [CZM+19, CZSL19, ZCG+19, LYP+19], which can be computationally quite expensive.

In contrast to corruption robustness, there exist many more approaches based on adversarial examples. We highlight some of the most interesting and relevant ones here. Mustafa et al. [Hao19] proposes to add a loss term that maximally separates class-wise feature map representations, hence increasing the distance from data points to the corresponding decision boundaries. Similarly, Pang et al. [PXD+20] proposed the Max-Mahalanobis center (MMC) loss to learn more structured representations and induce high-density regions in the feature space. Chen et al. [CBLR18] proposed a variation of the well-known cross entropy (CE) loss that not only maximizes the model probabilities of the correct class, but in addition, also minimizes model probabilities of incorrect classes. Cisse et al. [CBG+17] constraints the Lipschitz constant of different layers to be less than one which restricts the error propagation introduced by adversarial perturbations to a DNN. Dezfooli et al. [MDFUF19] proposed to minimize the curvature of the loss surface locally around data points. They emphasize that there exists a strong correlation between locally small curvature and correspondingly high adversarial robustness.

All of these methods highlighted above are evaluated mostly for image classification tasks on smaller datasets, namely CIFAR-10 [Ale09], CIFAR-100 [Ale09], SVHN [Yuv11], and only sometimes on ImageNet [KSH12]. Very few approaches have been tested rigorously on complex safety-relevant tasks such as *object detection* and *semantic segmentation*, etc. Moreover, methods that improve adversarial robustness are only tested on a small subset of attack types under differing attack specifications. This makes comparing multiple methods difficult.

In addition, methods that improve corruption robustness are evaluated over a standard data set of various corruption types which may or may not be relevant to its application domain. In order to assess multiple methods for their effect on safety-related aspects, a thorough robustness evaluation methodology is needed, which is largely missing in the current literature. This evaluation would need to take into account relevant disturbances/corruption types present in the real world (application domain) and had to assess robustness towards such changes in a rigorous manner. Without such an evaluation, we run the risk of being overconfident in our network, thereby harming safety.

3.3 Domain Generalization

Firas Mualla[13]

Domain generalization (DG) can be seen as an extreme case of *domain adaptation* (DA). The latter is a type of transfer learning, where the source and target tasks are the same (e.g., shared class labels) but the source and target domains are different (e.g., another image acquisition protocol or a different background) [Csu17a, WYKN20]. The DA can be either supervised (SDA), where there is little available labeled data in the target domain, or unsupervised (UDA), where data in the target domain is not labeled. The DG goes one step further by assuming that the target domain is entirely unknown. Thus, it seeks to solve the train-test domain shift in general. While DA is already an established line of research in the machine learning community, DG is relatively new [MBS13], though with an extensive list of papers in the last few years.

Probably, the first intuitive solution that one may think of to implement DG is neutralizing the domain-specific features. It was shown in [WHLX19] that the gray-level co-occurrence matrices (GLCM) tend to perform poorly in semantic classification (e.g., digit recognition) but yield good accuracy in textural classification compared to other feature sets such as SURF and LBP. DG was thus implemented by decorrelating the model's decision from the GLCM features of the input image even without the need of domain labels.

Besides the aforementioned intensity-based statistics of an input image, it is known that characterizing image style can be done based on the correlations between the filter responses of a DNN layer [GEB16] (neural style transfer). In [SMK20], the training images are enriched with stylized versions, where a style is defined either by an external style (e.g., cartoon or art) or by an image from another domain. Here, DG is addressed as a *data augmentation* problem.

Some approaches [LTG$^+$18, MH20a] try to learn generalizable latent representations by a kind of adversarial training. This is done by a generator or an encoder, which is trained to generate a hidden feature space that maximizes the error of a domain discriminator but at the same time minimizes the classification error of the task of concern. Another flavor of adversarial training can be seen in [LPWK18], where an adversarial autoencoder [MSJ$^+$16] is trained to generate features, which a discriminator cannot distinguish from random samples drawn from a prior Laplace distribution. This regularization prevents the hidden space from overfitting to the source domains, in a similar spirit to how variational autoencoders do not leave gaps in the latent space. In [MH20a], it is argued that the domain labels needed in such approaches are not always well-defined or easily available. Therefore they assume unknown latent domains which are learned by clustering in a space similar to the style-transfer features mentioned above. The pseudo labels resulting from clustering are then used in the adversarial training.

Autoencoders have been employed for DG not only in an adversarial setup, but also in the sense of *multi-task learning* nets [Car97], where the classification task in such nets is replaced by a reconstruction one. In [GKZB15], an autoencoder is trained to reconstruct not only the input image but also the corresponding images in the other domains.

In the core of both DA and DG we are confronted with a distribution matching problem. However, estimating the probability density in high-dimensional spaces is intractable. The density-based metrics such as Kullback-Leibler divergence are thus not directly applicable. In statistics, the so-called *two-samples tests* are usually employed to measure the distance between two distributions in a point-wise manner, i.e., without density estimation. For deep learning applications, these metrics need not only to be point-wise but also differentiable. The two-samples tests were approached in the machine learning literature using (differentiable) K-NNs [DK17], classifier two-samples tests (C2ST) [LO17], or based on the theory of kernel methods [SGSS07]. More specifically, the *maximum mean discrepancy* (MMD) [GBR+06, GBR+12], which belongs to the kernel methods, is widely used for DA [GKZ14, LZWJ17, YDL+17, YLW+20] but also for DG [LPWK18]. Using the MMD, the distance between two samples is estimated based on pairwise kernel evaluations, e.g., the radial basis function (RBF) kernel.

While the DG approaches generalize to domains from which zero shots are available, the so-called *zero shot learning* (ZSL) approaches generalize to tasks (e.g., new classes in the same source domains) for which zero shots are available. Typically, the input in ZSL is mapped to a semantic vector per class instead of a simple class label. This can be, for instance, a vector of visual attributes [LNH14] or a word embedding of the class name [KXG17]. A task (with zero shots at training time) can be then described by a vector in this space. In [MARC20], there is an attempt to combine ZSL and DG in the same framework in order to generalize to new domains as well as new tasks, which is also referred to as *heterogeneous domain generalization*.

Note that most discussed approaches for DG require non-standard handling, i.e., modifications to models, data, and/or the optimization procedure. This issue poses a serious challenge as it limits the practical applicability of these approaches. There is a line of research which tries to address this point by linking DG to other machine learning paradigms, especially the model-agnostic meta-learning (MAML) [FAL17] algorithm, in an attempt to apply DG in a model-agnostic way. Loosely speaking, a model can be exposed to simulated train-test domain shift by training on a small *support set* to minimize the classification error on a small *validation set*. This can be seen as an instance of a *few shot learning* (FSL) problem [WYKN20]. Moreover, the procedure can be repeated on other (but related) FSL tasks (e.g., different classes) in what is known as *episodic training*. The model transfers its knowledge from one task to another task and learns how to learn fast for new tasks. This can be thus seen as a *meta-learning* objective [HAMS20] (in a FSL setup). Since the goal of DG is to adapt to new domains rather than new tasks, several model-agnostic approaches [LYSH18, LZY+19, BSC18, DdCKG19] try to recast this procedure in a DG setup.

4 Adversarial Attacks

Andreas Bär[15]

Over the last few years, deep neural networks (DNNs) consistently showed state-of-the-art performance across several vision-related tasks. While their superior performance on clean data is indisputable, they show a lack of robustness to certain input patterns, denoted as *adversarial examples* [SZS+14]. In general, an algorithm for creating adversarial examples is referred to as an *adversarial attack* and aims at fooling an underlying DNN, such that the output changes in a desired and malicious way. This can be carried out without any knowledge about the DNN to be attacked (black-box attack) [MDFF16, PMG+17], or with full knowledge about the parameters, architecture, or even training data of the respective DNN (white-box attack) [GSS15, CW17a, MMS+18]. While initially being applied on simple classification tasks, some approaches aim at finding more realistic attacks [TVRG19, JLS+19], which particularly pose a threat to safety-critical applications, such as DNN-based environment perception systems in autonomous vehicles. Altogether, this motivated the research in finding ways of defending against such adversarial attacks [GSS15, MDFUF19, XZZ+19, GRCvdM18]. In this section, we introduce the current state of research regarding adversarial attacks in general, more realistic adversarial attacks closely related to the task of environment perception for autonomous driving, and strategies for detecting or defending adversarial attacks. We conclude each section by clarifying current challenges and research directions.

4.1 Adversarial Attacks and Defenses

Andreas Bär[15], Seyed Eghbal Ghobadi[8], Ahmed Hammam[8]

The term *adversarial example* was first introduced by Szegedy et al. [SZS+14]. From there on, many researchers tried to find new ways of crafting adversarial examples more effectively. Here, the fast gradient sign method (FGSM) [GSS15], DeepFool [MDFF16], least-likely class method (LLCM) [KGB17a, KGB17b], C&W [CW17b], momentum iterative fast gradient sign method (MI-FGSM) [DLP+18], and projected gradient descent (PGD) [MMS+18] are a few of the most famous attacks so far. In general, these attacks can be executed in an iterative fashion, where the underlying adversarial perturbation is usually bounded by some norm and is following additional optimization criteria, e.g., minimizing the number of changed pixels.

The mentioned attacks can be further categorized as image-specific attacks, where for each image a new perturbation needs to be computed. On the other hand, image-agnostic attacks aim at finding a perturbation, which is able to fool an underlying DNN on a set of images. Such a perturbation is also referred to as a *universal adversarial perturbation* (UAP). Here, the respective algorithm UAP [MDFFF17], fast feature fool (FFF) [MGB17], and prior driven uncertainty approximation (PD-UA) [LJL+19] are

a few honorable mentions. Although the creation process of a universal adversarial perturbation typically relies on a white-box setting, they show a high *transferability* across models [HBMF20]. This allows black-box attacks, where one model is used to create a universal adversarial perturbation, and another model is being attacked with the beforehand-created perturbation. Another way of designing black-box attacks is to create a surrogate DNN, which mimics the respective DNN to be attacked and thus can be used in the process of adversarial example creation [PMG+17]. On the contrary, some research has been done to create completely incoherent images (based on evolutionary algorithms or gradient ascent) to fool an underlying DNN [NYC15]. Different from that, another line of work has been proposed to alter only some pixels in images to attack a respective model. Here [NK16] and [NK17] have used optimization approaches to perturb some pixels in images to produce targeted attacks, aiming at a specific class output, or non-targeted attacks, aiming at outputting a class different from the network output or the ground truth. This can be extended up to finding one pixel in the image to be exclusively perturbed to generate adversarial images [SVS19, NK16]. The authors of [BF17, PKGB18, SBMC17] proposed to train generative models to generate adversarial examples. Given an input image and the target label, a generative model is trained to produce adversarial examples for DNNs. However, while the produced adversarial examples look rather unrealistic to a human, they are able to completely deceive a DNN. The existence of adversarial examples not only motivated research in finding new attacks, but also in finding *defense strategies* to effectively defend these attacks. Especially for safety-critical applications, such as DNN-based environment perception for autonomous driving, the existence of adversarial examples needs to be handled accordingly. Similar to adversarial attacks, one can categorize defense strategies into two types: *model-specific* defense strategies and *model-agnostic* defense strategies. The former refers to defense strategies, where the model of interest is modified in certain ways. The modification can be done on the architecture, training procedure, training data, or model weights. On the other hand, model-agnostic defense strategies consider the model to be a black box. Here, only the input or the output is accessible. Some well-known model-specific defense strategies include adversarial training [GSS15, MMS+18], the inclusion of robustness-oriented loss functions during training [CLC+19, MDFUF19, KYL+20], removing adversarial patterns in features by denoising layers [HRF19, MKH+19, XWvdM+19], and redundant teacher-student frameworks [BHSFs19, BKV+20]. The majority of model-agnostic defense strategies primarily focuses on various kinds of (gradient masking) pre-processing strategies [BFW+19, GRCvdM18, GR19, JWCF19, LLL+19, RSFM19, TCBZ19]. The idea is to remove the adversary from the respective image, such that the image is transformed from the adversarial space back into the clean space.

Nonetheless, Athalye et al. [ACW18] showed that gradient masking alone is not a sufficient criterion for a reliable defense strategy. In addition, detection and out-of-distribution techniques have also been proposed as model-agnostic defense strategies against adversarial attacks. Here, the Mahalanobis distance [LLLS18] or area under the receiver operating characteristic curve (AUROC) and area under the precision-recall curve (AUPR) [HG17] are used to detect adversarial examples. The authors of

[HG17, LLLS17, HMCKBF17] on the other hand proposed to train networks to detect, whether the input image is out-of-distribution or not.
Moreover, Feinman et al. [FCSG17] proved that adversarial attacks usually produce high uncertainty on the output of the DNN. As a consequence, they proposed to use the dropout technique to estimate uncertainty on the output to identify a possible adversarial attack.
Regarding adversarial attacks, the majority of the listed attacks are designed for image classification. Only a few adversarial attacks consider tasks that are closely related to autonomous driving, such as bounding box detection, semantic segmentation, instance segmentation, or even panoptic segmentation. Also, the majority of the adversarial attacks rely on a white-box setting, which is usually not present for a potential attacker. Especially universal adversarial perturbations have to be considered as a real threat due to their high model transferability. Generally speaking, the existence of adversarial examples has not been thoroughly studied yet. An analytical interpretation is still missing, but could help in designing more mature defense strategies.
Regarding defense strategies, adversarial training is still considered as one of the most effective ways of increasing the robustness of a DNN. Nonetheless, while adversarial training is indeed effective, it is rather inefficient in terms of training time. In addition, model-agnostic defenses should be favored as once being designed, they can be easily transferred to different models. Moreover, as most model-agnostic defense strategies rely on gradient-masking and it has been shown that gradient-masking is not a sufficient property for a defense strategy, new ways of designing model-agnostic defenses should be taken into account. Furthermore, out-of-distribution and adversarial attacks detection or even correction methods have been a new trend for identifying attacks. However, as the environment perception system of an autonomous driving vehicle could rely on various information sources, including LiDAR, optical flow, or depth from a stereo camera, techniques of information fusion should be further investigated to mitigate or even eliminate the effect of adversarial examples.

4.2 More Realistic Attacks

Svetlana Pavlitskaya[14]

We consider the following two categories of realistic adversarial attacks: (1) image-level attacks, which not only fool a neural network but also pose a provable threat to autonomous vehicles, and (2) attacks which have been applied in a real world or in a simulation environment, such as car learning to act (CARLA) [DRC+17].
Some notable examples in the first category of attacks include attacks on semantic segmentation [HMCKBF17] or person detection [TVRG19].
In the second group of approaches, the attacks are specifically designed to survive real world distortions, including different distances, weather and lighting conditions, as well as camera angles. For this, adversarial perturbations are usually concentrated in a specific image area, called *adversarial patch*. Crafting an adversarial patch involves

specifying a patch region in each training image, applying transformations to the patch, and iteratively changing the pixel values within this region to maximize the network prediction error. The latter step typically relies on an algorithm, proposed for standard adversarial attacks, which aim at crafting invisible perturbations while misleading neural networks, e.g., C&W [CW17b], Jacobian-based saliency map attack (JSMA) [PMG+17], and PGD [MMS+18].

The first printable adversarial patch for image classification was described by Brown et al. [BMR+17]. Expectation over transformations (EOT) [AEIK17] is one of the influential updates to the original algorithm—it permits to robustify patch-based attacks to distortions and affine transformations. Localized and visible adversarial noise (LaVAN) [KZG18] is a further method to generate much smaller patches (up to 2% of the pixels in the image). In general, fooling image classification with a patch is a comparatively simple task, because adversarial noise can mimic an instance of another class and thus lower the prediction probability for a true class.

Recently, patch-based attacks for a more tricky task of object detection have been described [LYL+18, TVRG19]. Also, Lee and Kolter [LK19] generate a patch using PGD [MMS+18], followed by EOT applied to the patch. With this approach, all detections in an image can be successfully suppressed, even without any overlap of a patch with bounding boxes. Furthermore, several approaches for generating an adversarial T-shirt have been proposed, including [XZL+19, WLDG19].

DeepBillboard [ZLZ+18] is the first attempt to attack end-to-end driving models with adversarial patches. The authors propose to generate a single patch for a sequence of input images to mislead four steering models, including DAVE-2 in a drive-by scenario.

Apart from physical feasibility, inconspicuousness is crucial for a realistic attack. Whereas adversarial patches usually look like regions of noise, several works have explored attacks with an inconspicuous patch. In particular, Eykholt et al. [EEF+18] demonstrate the vulnerability of road sign classification to the adversarial perturbations in the form of only black and white stickers. In [BHG+19], an end-to-end driving model is attacked in CARLA by painting of black lines on the road. Also, Kong and Liu [KL19] use a generative adversarial network to get a realistic billboard to attack an end-to-end driving model in a drive-by scenario. In [DMW+20], a method to hide visible adversarial perturbations with customized styles is proposed, which leads to adversarial traffic signs that look unsuspicious to a human.

Current research mostly focuses on attacking image-based perception of an autonomous vehicle. Adversarial vulnerability of further components of an autonomous vehicle, e.g., LiDAR-based perception, optical flow and depth estimation, has only recently gained attention. Furthermore, most attacks consider only a single component of an autonomous driving pipeline, the question whether the existing attacks are able to propagate to further pipeline stages has not been studied yet. The first work in this direction [JLS+19] describes an attack on object detection and tracking. The evaluation is, however, limited to a few clips, where no experiments in the real world have been performed. Overall, the research on realistic adversarial attacks, especially combined with physical tests, is currently in the starting phase.

5 Interpretability

Felix Brockherde[10]

Neural networks are, by their nature, black boxes and therefore intrinsically hard to interpret [Tay06]. Due to their unrivaled performance, they still remain first choice for advanced systems even in many safety-critical areas, such as level 4 automated driving. This is why the research community has invested considerable effort to unhinge the black-box character and make deep neural networks more transparent.
We can observe three strategies that provide different view points towards this goal in the state of the art. First is the most direct approach of opening up the black box and looking at intermediate representations (Sec. 5.2). Being able to interpret individual layers of the system facilitates interpretation of the whole. The second approach tries to provide interpretability by explaining the network's decisions with pixel attributions (Sec. 5.3). Aggregated explanations of decision can then lead to interpretability of the system itself. Third is the idea of approximating the network with interpretable proxies to benefit from the deep neural networks performance while allowing interpretation via surrogate models (Sec. 5.4). Underlying all aspects here is the area of visual analytics (Sec. 5.1).
There exists earlier research in the medical domain to help human experts understand and convince them of machine learning decisions [CLG+15]. Legal requirements in the finance industry gave rise to interpretable systems that can justify their decisions. An additional driver for interpretability research was the concern for Clever Hans predictors [LWB+19].

5.1 Visual Analytics

Elena Schulz[1]

Traditional data science has developed a huge tool set of automated analysis processes conducted by computers, which are applied to problems that are well-defined in the sense that the dimensionality of input and output as well as the size of the data set they rely on is manageable. For those problems that in comparison are more complex, the automation of the analysis process is limited and/or might not lead to the desired outcome. This is especially the case with unstructured data like image or video data in which the underlying information cannot directly be expressed by numbers. Rather, it needs to be transformed to some structured form to enable computers to perform some task of analysis. Additionally, with an ever increasing amount of various types of data being collected, this "information overload" cannot solely be analyzed by automatic methods [KAF+08, KMT10].
Visual analytics addresses this challenge as "the science of analytical reasoning facilitated by interactive visual interfaces" [CT05]. Visual analytics therefore does not only focus

on either computationally processing data or visualizing results but coupling both tightly with interactive techniques. Thus, it enables an integration of the human expert into the iterative visual analytics process: Through visual understanding and human reasoning, the knowledge of the human expert can be incorporated to effectively refine the analysis. This is of particular importance, where a stringent safety argumentation for complex models is required. With the help of visual analytics, the line of argumentation can be built upon arguments that are understandable for humans. To include the human analyst efficiently into this process, a possible guideline is the *visual analytics mantra* by Keim: "Analyze first, show the important, zoom, filter and analyze further, details on demand" [KAF+08][1].

The core concepts of visual analytics therefore rely on well-designed interactive visualizations, which support the analyst in the tasks of, e.g., reviewing, understanding, comparing and inferring not only the initial phenomenon or data but also the computational model and its results itself with the goal of enhancing the analytical process.

Driven by various fields of application, visual analytics is a multidisciplinary field with a wide variety of task-oriented development and research. As follows, recent work has been done in several areas: depending on the task, there exist different pipeline approaches to create whole *visual analytics systems* [WZM+16]; the injection of human expert knowledge into the process of determining trends and patterns from data is the focus of *predictive visual analytics* [LCM+17, LGH+17]; enabling the human to explore *high-dimensional data* [LMW+17] interactively and visually (e.g., via dimensionality reduction [SZS+17]) is a major technique to enhance the understandability of complex models (e.g., neural networks); the iterative improvement and the understanding of machine learning models is addressed by using interactive visualizations in the field of *general machine learning* [LWLZ17] or the other way round: using machine learning to improve visualizations and guidance based on user interactions [ERT+17]. Even more focused on the loop of simultaneously developing and refining machine learning models is the area of *interactive machine learning*, where the topics of *interface design* [DK18] and the *importance of users* [ACKK14, SSZ+17] are discussed. One of the current research directions is using visual analytics in the area of *deep learning* [GTC+18, HKPC18, CL18]. However, due to the interdisciplinarity of visual analytics, there are still open directions and ongoing research opportunities.

Especially in the domain of neural networks and deep learning, visual analytics is a relatively new approach in tackling the challenge of *explainability* and *interpretability* of those often called *black boxes*. To enable the human to better interact with the models, research is done in enhancing the *understandability* of complex deep learning models and their outputs with the use of proper visualizations. Other research directions attempt to achieve improving the *trustability* of the models, giving the opportunity to inspect, diagnose and refine the model. Further, possible areas for research are *online training processes* and the development of *interactive systems* covering the whole process of training, enhancing and monitoring machine learning models. Here, the approach of

[1] Extending the original *visualization mantra* by Shneiderman "Overview first, filter and zoom, details on demand"[Shn96].

mixed guidance, where system-initiated guidance is combined with user-initiated guidance, is discussed among the visual analytics community as well. Another challenge and open question is creating ways of *comparing models* to examine which model yields a better performance, given specific situations and selecting or combining the best models with the goal of increasing performance and overall safety.

5.2 Intermediate Representations

Felix Hauser[11], Jan Kronenberger[9], Seyed Eghbal Ghobadi[8]

In general, representation learning [BCV13] aims to extract lower dimensional features in latent space from higher dimensional inputs. These features are then used as an effective representation for regression, classification, object detection and other machine learning tasks. Preferably, latent features should be disentangled, meaning that they represent separate factors found in the data that are statistically independent. Due to their importance in machine learning, finding meaningful intermediate representations has long been a primary research goal. Disentangled representations can be interpreted more easily by humans and can for example be used to explain the reasoning of neural networks [HDR18].

Among the longer known methods for extracting disentangled representations are principal component analysis (PCA) [FP78, JC16], independent component analysis [HO00], and nonnegative matrix factorization [BBL$^+$07]. PCA is highly sensitive to outliers and noise in the data. Therefore, more robust algorithms were proposed. In [SBS12] already a small neural network was used as an encoder and the algorithm proposed in [FXY12] can deal with high-dimensional data. Some robust PCA algorithms are provided with analytical performance guarantees [XCS10, RA17, RL19].

A popular method for representation learning with deep networks is the variational autoencoder (VAE) [KW14]. An important generalization of the method is the β-VAE variant [HMP$^+$17], which improved the disentanglement capability [FAA18]. Later analysis added to the theoretical understanding of β-VAE [BHP$^+$18, SZYP19, KP20]. Compared to standard autoencoders, VAEs map inputs to a distribution, instead of mapping them to a fixed vector. This allows for additional regularization of the training to avoid overfitting and ensure good representations. In β-VAEs the trade-off between reconstruction quality and disentanglement can be fine-tuned by the hyperparameter β. Different regularization schemes have been suggested to improve the VAE method. Among them are Wasserstein autoencoders [TBGS19, XW19], attribute regularization [PL20] and relational regularization [XLH$^+$20]. Recently, a connection between VAEs and nonlinear independent component analysis was established [KKMH20] and then expanded [SRK20].

Besides VAEs, deep generative adversarial networks can be used to construct latent features [SLY15, CDH$^+$16, MSJ$^+$16]. Other works suggest centroid encoders [GK20] or conditional learning of Gaussian distributions [SYZ$^+$20] as alternatives to VAEs. In [KWG$^+$18] concept activation vectors are defined as being orthogonal to the decision

boundary of a classifier. Apart from deep learning, entirely new architectures, such as capsule networks [SFH17], might be used to disassemble inputs.
While many different approaches for disentangling exist, the feasibility of the task is not clear yet and a better theoretical understanding is needed. The disentangling performance is hard to quantify, which is only feasible with information about the latent ground truth [EW18]. Models that overly rely on single directions, single neurons in fully connected networks or single feature maps in CNNs, have the tendency to overfit [MBRB18]. According to [LBL+19], unsupervised learning does not produce good disentangling and even small latent spaces do not reduce the sample complexity for simple tasks. This is in direct contrast to newer findings that show a decreased sample complexity for more complex visual downstream tasks [vLSB20]. So far, it is unclear if disentangling improves the performance of machine learning tasks.
In order to be interpretable, latent disentangled representations need to be aligned with human understandable concepts. In [EIS+19] training with adversarial examples was used and the learned representations were shown to be more aligned with human perception. For explainable AI, disentangling alone might not be enough to generate interpretable output and additional regularization could be needed.

5.3 Pixel Attribution

Stephanie Abrecht[2], Felix Brockherde[10], Toshika Srivastava[12]

The non-linearity and complexity of DNNs allow them to solve perception problems, like detecting a pedestrian, that cannot be specified in detail. At the same time, the automatic extraction of features given in an input image and the mapping to the respective prediction is counterintuitive and incomprehensible for humans, which makes it hard to argue safety for a neural network-based perception task. Feature importance techniques are currently predominantly used to diagnose the causes of incorrect model behaviors [BXS+20]. So-called *attribution maps* are a visual technique to express the relationship between relevant pixels in the input image and the network's prediction. Regions in an image that contain relevant features are highlighted accordingly. Attribution approaches mostly map to one of three categories.
Gradient-based and activation-based approaches (such as [SDV+16, SVZ14, SGSK17, AS17, STK+17, BBM+15, SDBR14, MBB+15] amongst others) rely on the gradient of the prediction with respect to the input. Regions that were most relevant for the prediction are highlighted. Activation-based approaches relate the feature maps of the last convolutional layer to output classes.
Perturbation-based approaches [FV17, ZF13, ZCAW17, HMK+19] suggest manipulating the input. If the prediction changes significantly, the input may hold a possible explanation at least.
While gradient-based approaches are oftentimes faster in computation, perturbation-based approaches are much easier to interpret.

As many studies have shown [MS17, AGM+18], there is still a lot of research to be done before attribution methods are able to robustly provide explanations for model predictions, in particular erroneous behavior. One key difficulty is the lack of an agreed-upon definition of a good attribution map including important properties. Even between humans, it is hard to agree on what a good explanation is due to its subjective nature. This lack of ground truth makes it hard or even impossible to quantitatively evaluate an explanation method. Instead, this evaluation is done only implicitly. One typical way to do this is the axiomatic approach. Here a set of desiderata of an attribution method are defined, on which different attribution methods are then evaluated. Alternatively, different attribution methods may be compared by perturbing the input features starting with the ones deemed most important and measuring the drop in accuracy of the perturbed models. The best method will result into the greatest overall loss in accuracy as the number of inputs are omitted [ACÖG17]. Moreover, for gradient-based methods it is hard to assess if an unexpected attribution is caused by a poorly performing network or a poorly performing attribution method [FV17]. How to cope with negative evidence, i.e., the object was predicted because a contrary clue in the input image was missing, is an open research question. Additionally, most methods were shown on classification tasks until now. It remains to be seen how they can be transferred to object detection and semantic segmentation tasks. In the case of perturbation-based methods, the high computation time and single-image analysis inhibit wide-spread application.

5.4 Interpretable Proxies

Gesina Schwalbe[3]

Neural networks are capable of capturing complicated logical (cor)relations. However, this knowledge is encoded on a *sub-symbolic* level in the form of learned weights and biases, meaning that the reasoning behind the processing chain cannot be directly read out or interpreted by humans [CPT01]. To explain the sub-symbolic processing, one can either use attribution methods (cf. Sec. 5.3), or lift this sub-symbolic representation to a *symbolic* one [GBY+18], meaning a more interpretable one. Interpretable proxies or surrogate models try to achieve the latter: The DNN behavior is approximated by a model that uses symbolic knowledge representations. Symbolic representations can be linear models like LIME [RSG16] (proportionality), decision trees (if-then-chains) [GBY+18], or loose sets of logical rules. Logical connectors can simply be AND and OR but also more general ones like at-least-M-of-N [CPT01]. The *expressiveness* of an approach refers to the logic that is used: Boolean-only versus first-order logic, and binary versus fuzzy logic truth values [TAGD98]. Other than attribution methods (cf. Sec. 5.3), these representations can capture combinations of features and (spatial) relations of objects and attributes. As an example consider “eyes are closed” as explanation for “person asleep”: Attribution methods only could mark the location of the eyes, dismissing the relations of the attributes [RSS18]. All mentioned surrogate model types (linear, set of rules) require interpretable input features in order to be interpretable themselves. These

features must either be directly obtained from the DNN input or (intermediate) output, or automatically be extracted from the DNN representation. Examples for extraction are the super-pixeling used in LIME for input feature detection, or concept activation vectors [KWG+18] for DNN representation decoding.
Quality criteria and goals for interpretable proxies are [TAGD98]: *accuracy* of the standalone surrogate model on unseen examples, *fidelity* of the approximation by the proxy, *consistency* with respect to different training sessions, and *comprehensibility* measured by the complexity of the rule set (number of rules, number of hierarchical dependencies). The criteria are usually in conflict and need to be balanced: Better accuracy may require a more complex, thus less expressive sets of rules.
Approaches for interpretable proxies differ in the validity range of the representations: Some aim for surrogates that are only valid *locally* around specific samples, like in LIME [RSG16] or in [RSS18] via inductive logic programming. Other approaches try to more *globally* approximate aspects of the model behavior. Another categorization is defined by whether full access (*white-box*), some access (*gray-box*), or no access (*black-box*) to the DNN internals is needed. One can further differentiate between *post-hoc* approaches that are applied to a trained model, and approaches that try to integrate or *enforce symbolic representations* during training. Post-hoc methods cover the wide field of rule extraction techniques for DNNs. The reader may refer to [Hai16, AK12]. Most white- and gray-box methods try to turn the DNN connections into if-then rules that are then simplified, like done in DeepRED [ZLMJ16]. A black-box example is validity interval analysis [Thr95], which refines or generalizes rules on input intervals, either starting from one sample or a general set of rules. Enforcement of symbolic representations can be achieved by enforcing an output structure that provides insights to the decision logic, such as textual explanations, or a rich output structure allowing investigation of correlations [XLZ+18]. An older discipline for enforcing symbolic representations is the field of neural-symbolic learning [SSZ19]. The idea is based on a hybrid learning cycle in which a symbolic learner and a DNN iteratively update each other via rule insertion and extraction.
The comprehensibility of global surrogate models suffers from the complexity and size of concurrent DNNs. Thus, stronger rule simplification methods are required [Hai16]. The alternative direction of local approximations mostly concentrates on linear models instead of more expressive rules [Thr95, RSS18]. Furthermore, balancing of the quality objectives is hard since available indicators for interpretability may not be ideal. And lastly, applicability is heavily infringed by the requirement of interpretable input features. These are usually not readily available from input (often pixel-level) or DNN output. Supervised extraction approaches vary in their fidelity, and unsupervised ones do not guarantee to yield meaningful or interpretable results, respectively, such as the super-pixel clusters of LIME.

6 Uncertainty

Michael Mock[1]

Uncertainty refers to the view that a neural network is not conceived as a deterministic function but as a probabilistic function or estimator, delivering a random distribution for each input point. Ideally, the mean value of the distribution should be as close as possible to the ground truth value of the function being approximated by the neural network and the uncertainty of the neural network refers to its variance when being considered as a random variable, thus allowing to derive a confidence with respect to the mean value. Regarding safety, the variance may lead to estimations about the confidence associated with a specific network output and opens the option for discarding network outputs with insufficient confidence.
There are roughly two broad approaches for training neural networks as probabilistic functions: Parametric approaches [KG17] and Bayesian neural networks on the one hand, such as [BCKW15], where the transitions along the network edges are modeled as probability distributions, and ensemble-based approaches on the other hand [LPB17, SOF+19], where multiple networks are trained and considered as samples of a common output distribution. Apart from training as probabilistic function, uncertainty measures have been derived from single, standard neural networks by post-processing on the trained network logits, leading for example to calibration measures (cf. e.g., [SOF+19]).

6.1 Generative Models

Sebastian Wirkert[6], Tim Wirtz[1]

Generative models belong to the class of unsupervised machine learning models. From a theoretical perspective, these are particularly interesting, because they offer a way to analyze and model the density of data. Given a finite data set $\mathcal{D}$ independently distributed according to some distribution $p(x)$, generative models aim to estimate or enable sampling from the underlying density $p(x)$ in a model $q_\theta(x)$. The resulting model can be used for data indexing [Wes04], data retrieval [ML11], for visual recognition [KSH12], speech recognition and generation [HDY+12], language processing [KM03, CE17] and robotics [T+02]. Following [OE18], we can group generative models into two main classes:

- Cost function-based models such as autoencoder [KW14, Doe16], deep belief networks [Hin09] and generative adversarial networks [Goo16, RMC15, GPAM+14].
- Energy-based models [LCH+06, SH09], where the joint probability density is modeled by an energy function.

Beside these *deep learning* approaches, generative models have been studied in machine learning in general for quite some time (cf. [Wer78, JMS96, Fry77, Sil86, Gra18, Sco15,

She04]). A very prominent example of generative networks are Gaussian processes [Ras03, Mac98, WR96, WB98] and their deep learning extensions [DL13, BHLHL+16] as generative models.
An example of a generative model being employed for image segmentation uncertainty estimation is the probabilistic U-Net [KRPM+18]. Here a variational autoencoder (VAE) conditioned on the image is trained to model uncertainties. Samples from the VAE are fed into a segmentation U-Net which can thus give different results for the same image. This was tested in context of medical images, where inter-rater disagreements lead to uncertain segmentation results and Cityscapes segmentation. For the Cityscapes segmentation the investigated use case was label ambiguity (e.g., is a BMW X7 a car or a van) using artificially created, controlled ambiguities. Results showed that the probabilistic U-Net could reproduce the segmentation ambiguity modes more reliably than competing methods such as a dropout U-Net which is based on techniques elaborated in the next section.

6.2 Monte-Carlo Dropout

Joachim Sicking[1]

A widely used technique to estimate model uncertainty is Monte-Carlo (MC) dropout [GG16], that offers a Bayesian motivation, conceptual simplicity and scalability to application-size networks. This combination distinguishes MC dropout from competing Bayesian neural network (BNN) approximations (like [BCKW15],[RBB18], see Sec. 6.3). However, these approaches and MC dropout share the same goal: to equip neural networks with a *self-assessment* mechanism that detects unknown input concepts and thus potential model insufficiencies.
On a technical level, MC dropout assumes prior distributions on network activations, usually independent and identically distributed (i.i.d.) Bernoulli distributions. Model training with iteratively drawn Bernoulli samples, the so-called *dropout masks*, then yields a data-conditioned posterior distribution within the chosen parametric family. It is interesting to note that this training scheme was used earlier—independent from an uncertainty context—for better model generalization [SHK+14]. At inference, sampling provides estimates of the input-dependent output distributions. The spread of these distributions is then interpreted as the prediction uncertainty that originates from limited knowledge of model parameters. Borrowing 'frequentist' terms, MC dropout can be considered as an implicit network ensemble, i.e., as a set of networks that share (most of) their parameters.
In practice, MC dropout requires only a minor modification of the optimization objective during training and multiple, trivially parallelizable forward passes during inference. The loss modification is largely agnostic to network architecture and does not cause substantial overhead. This is in contrast to the sampling-based inference that increases the computational effort massively—by estimated factors of 20-100 compared to networks without MC dropout. A common practice is therefore the use of *last-layer dropout*

[SOF+19] that reduces computational overhead to estimated factors of 2-10. Alternatively, analytical moment propagation allows sampling-free MC-dropout inference at the price of additional approximations (e.g., [PFC+19]). Further extensions of MC dropout target the integration of data-inherent (aleatoric) uncertainty [KG17] and tuned performance by learning layer-specific dropout rate using concrete relaxations [GHK17].
The quality of MC-dropout uncertainties is typically evaluated using negative log-likelihood (NLL), expected calibration error (ECE) and its variants (cf. Sec. 6.6) and by considering correlations between uncertainty estimates and model errors (e.g., AUSE [ICG+18]). Moreover, it is common to study how useful uncertainty estimates are for solving auxiliary tasks like out-of-distribution classification [LPB17] or robustness w.r.t. adversarial attacks.
MC dropout is a working horse of safe ML, being used with various networks and for a multitude of applications (e.g., [BFS18]). However, several authors pointed out shortcomings and limitations of the method: MC dropout bears the risk of over-confident false predictions ([Osb16]), offers less diverse uncertainty estimates compared to (the equally simple and scalable) deep ensembles ([LPB17], see Sec. 7.1) and provides only rudimentary approximations of true posteriors.
Relaxing these modelling assumptions and strengthening the Bayesian motivation of MC dropout is therefore an important research avenue. Further directions for future work are the development of *semantic uncertainty mechanisms* (e.g., [KRPM+18]), improved local uncertainty calibrations and a better understanding of the outlined sampling-free schemes to uncertainty estimation.

6.3 Bayesian Neural Networks

Maram Akila[1]

As the name suggests, Bayesian neural networks (BNNs) are inspired by a Bayesian interpretation of probability (for an introduction cf. [Mac03]). In essence, it rests on Bayes' theorem,

$$p(x|y)p(y) = p(x,y) = p(y|x)p(x) \quad \Rightarrow \quad p(x|y) = \frac{p(y|x)p(x)}{p(y)}\,, \tag{2}$$

stating that the conditional probability density function (PDF) $p(x|y)$ for x given y may be expressed in terms of the inverted conditional PDF $p(y|x)$. For machine learning, where one intends to make predictions y for unknown x given some training data $\mathcal{D}$, this can be reformulated into

$$y = \mathrm{NN}(x|W) \quad \text{with} \quad p(W|\mathcal{D}) = \frac{p(\mathcal{D}|W)p(W)}{p(\mathcal{D})}\,. \tag{3}$$

Therein NN denotes a conventional (deep) neural network (DNN) with model parameters W, e.g., the set of weights and biases. In contrast to a regular DNN, the weights are given

in terms of a probability distribution $p(W|\mathcal{D})$ turning also the output y of a BNN into a distribution. This allows to study the mean $\mu = \langle y^1 \rangle$ of the DNN for a given x as well as higher moments of the distribution, typically the resulting variance $\sigma^2 = \langle (y-\mu)^2 \rangle$ is of interest, where

$$\left\langle y^k \right\rangle = \int \mathrm{NN}(x|W)^k p(W|\mathcal{D}) \, \mathrm{d}W \, . \tag{4}$$

While μ yields the output of the network, the variance σ^2 is a measure for the uncertainty of the model for the prediction at the given point. Central to this approach is the probability of the data given the model, here denoted by $p(\mathcal{D}|W)$, as it is the key component connecting model and training data. Typically, the prior distribution $p(\mathcal{D})$ is "ignored" as it only appears as a normalization constant within the averages, see (4). In the cases where the data $\mathcal{D}$ is itself a distribution due to inherent uncertainty, i.e., presence of aleatoric risk [KG17], such a concept seems natural. However, Bayesian approaches are also applicable for all other cases. In those, loosely speaking, the likelihood of W is determined via the chosen loss function (for the connection between the two concepts cf. [Bis06]).

On this general level, Bayesian approaches are broadly accepted and also find use for many other model classes besides neural networks. However, the loss surfaces of DNNs are known for their high dimensionality and strong non-convexity. Typically, there are abundant parameter combinations W that lead to (almost) equally good approximations to the training data $\mathcal{D}$ with respect to a chosen loss. This makes an evaluation of $p(W|\mathcal{D})$ for DNNs close to impossible in full generality. At least no (exact) solutions for this case exist at the moment.

Finding suitable approximations to the posterior distribution $p(W|\mathcal{D})$ is an ongoing challenge for the construction of BNNs. At this point we only summarize two major research directions in the field. One approach is to assume that the distribution factorizes. While the full solution would be a joined distribution implying correlations between different weights (etc.), possibly even across layers, this approximation takes each element w_i of W to be independent form the others. Although this is a strong assumption, it is often made, in this case parameters for the respective distributions of each element can be learned via training (cf. [BCKW15]). The second class of approaches focuses on the region of the loss surface around the minimum chosen for the DNN. As discussed, the loss relates to the likelihood and quantities such as the curvature at the minimum, therefore directly connected to the distribution of W. Unfortunately, already using this type of quantities requires further approximations [RBB18]. Alternatively, the convergence of the training process may be altered to sample networks close to the minimum [WT11]. While this approach contains information about correlations among the w_i, it is usually restricted to a specific minimum. For a non-Bayesian ansatz taking into account several minima, see deep ensembles in Sec. 7.1. BNNs also touch other concepts such as MC dropout (cf. Sec. 6.2 or [GG16]), or prior networks, which are based on a Bayesian interpretation but use conventional DNNs with an additional (learned) σ output [MG18].

6.4 Uncertainty Metrics for DNNs in Frequentist Inference

Matthias Rottmann[5]

Classical uncertainty quantification methods in frequentist inference are mostly based on the outputs of statistical models. Their uncertainty is quantified and assessed for instance via dispersion measures in classification (such as entropy, probability margin or variation ratio), or confidence intervals in regression. However, the nature of DNN architectures [RDGF15, CPK+17] and the cutting edge applications tackled by those (e.g., semantic segmentation, cf. [COR+16]) open the way towards more elaborate uncertainty quantification methods. Besides the mentioned classical approaches, intermediate feature representations within a DNN (cf. [OSM20, CZYS19]) or gradients according to self-affirmation that represent re-learning stress (see [ORG18]) reveal additional information. In addition, in case of semantic segmentation, the geometry of a prediction may give access to further information, cf. [RCH+18, RS19, MRG19]. By the computation of statistics of those quantities as well as low-dimensional representations thereof, we obtain more elaborate uncertainty quantification methods specifically designed for DNNs that can help us to detect misclassifications and out-of-distribution objects (cf. [HG17]).
Features gripped during a fordward pass of a data point x through a DNN f can be considered layer-wise, i.e., $f^{(\ell)}(x)$ after the ℓ-th layer. These can be translated into a handful of quantities per layer [OSM20] or further processed by another DNN that aims that detecting errors [CZYS19]. While in particular [OSM20] presents a proof of concept on small scale classification problems, their applicability to large scale datasets and problems such as semantic segmentation and object detection remain open.
The development for gradient-based uncertainty quantification methods [ORG18] is guided by one central question: If the present prediction was true, how much re-learning would this require. The corresponding hypothesis is that wrong predictions would be more in conflict with the knowledge encoded in the deep neural network than correct ones, therefore causing increased re-learning stress. Given a predicted class

$$\hat{y} = \arg\max_y f_y(x) \tag{5}$$

we compute the gradient of layer ℓ corresponding to the predicted label. That is, given a loss function $\mathcal{L}$, we compute

$$\nabla_{w_\ell} \mathcal{L}(\hat{y}, x, w) \tag{6}$$

via backpropagation. The obtained quantities can be treated similarly to the case of forward pass features. While this concept seems to be prohibitively expensive for semantic segmentation (at least when calculating gradients for each pixel of $\hat{y}$), its applicability to object detection might be feasible, in particular with respect to offline applications. Gradients are also of special interest in active learning with query by *expected model change* (cf. [Set10]).
In the context of semantic segmentation, geometrical information on segments shapes as well as neighborhood relations of predicted segments can be taken into account along

side with dispersion measures. It has been demonstrated [RCH+18, RS19, MRG19] that the detection of errors in an in-distribution setting strongly benefits from geometrical information. Recently, this has also been considered in scenarios under moderate domain shift [ORF20]. However, its applicability to out-of-distribution examples and to other sensors than the camera remains subject to further research.

6.5 Markov Random Fields

Seyed Eghbal Ghobadi[8], Ahmed Hammam[8]

Although deep neural networks are currently the state of the art for almost all computer vision tasks, Markov random fields (MRF) remain one of the fundamental techniques used for many computer vision tasks, specifically image segmentation [KK11],[LWZ09]. MRFs hold its power in the essence of being able to model dependencies between pixels in an image. With the use of energy functions, MRFs integrate pixels into models relating between unary and pairwise pixels together [WKP13]. Given the model, MRFs are used to infer the optimal configuration yielding the lowest energy using mainly maximum a posteriori (MAP) techniques. Several MAP inference approaches are used to yield the optimal configuration such as graph cuts [KRBT08] and belief propagation algorithms [FZ10]. However, as with neural networks, MAP inference techniques result in deterministic point estimates of the optimal configuration without any sense of uncertainty in the output. To obtain uncertainties on results from MRFs, most of the work is directed towards modelling MRFs with Gaussian distributions. Getting uncertainties from MRFs with Gaussian distributions is possible by two typical methods: Either approximate models are inferred to the posterior, from which sampling is easy or the variances can be estimated analytically, or approximate sampling from the posterior is used. Approximate models include those inferred using variational Bayesian (VB) methods, like mean-field approximations, and using Gaussian process (GP) models enforcing a simplified prior model [LUAD16], [Bis06]. Examples of approximate sampling methods include traditional Markov chain Monte Carlo (MCMC) methods like Gibbs sampling [GG84]. Some recent theoretical advances propose the perturb-and-MAP framework and a Gumbel perturbation model (GPM) [PY11],[HMJ13] to exactly sample from MRF distributions. Another line of work has also been proposed, where MAP inference techniques are used to estimate the probability of the network output. With the use of graph cuts, [KT08] try to estimate uncertainty using the min-marginals associated with the label assignments of a random field. Here, the work by Kohli and Torr [KT08] was extended to show how this approach can be extended to techniques other than graph cuts [TA12] or compute uncertainties on multi-label marginal distributions [STP17].
A current research direction is the incorporation of MRFs with deep neural networks, along with providing uncertainties on the output [SU15, CPK+17]. This can also be extended to other forms of neural networks such as recurrent neural networks to provide uncertainties on segmentation of streams of videos with extending dependencies of pixels to previous frames [LLL+17], [ZJRP+15].

6.6 Confidence Calibration

Fabian Küppers[9], Anselm Haselhoff[9]

Neural network classifiers output a label $\hat{Y} \in \mathcal{Y}$ on a given input $X \in \mathcal{X}$ with an associated confidence $\hat{P}$. This confidence can be interpreted as a probability of correctness that the predicted label matches the ground truth label $Y \in \mathcal{Y}$. Therefore, these probabilities should reflect the "self-confidence" of the system. If the empirical accuracy for any confidence level matches the predicted confidence, a model is called *well calibrated*. Therefore, a *classification* model is perfectly calibrated if

$$\underbrace{\mathbb{P}(\hat{Y} = Y | \hat{P} = p)}_{\text{accuracy given } p} = \underbrace{p}_{\text{confidence}} \qquad \forall p \in [0, 1] \tag{7}$$

is fullfilled [GPSW17]. For example, assume 100 predictions with confidence values of 0.9. We call the model well calibrated if 90 out of these 100 predictions are actually correct. However, recent work has shown that modern neural networks tend to be too overconfident in their predictions [GPSW17]. The deviation of a model to the perfect calibration can be measured by the expected calibration error (ECE) [NCH15]. It is possible to recalibrate models as a post-processing step after classification. One way to get a calibration mapping is to group all predictions into several bins by their confidence. Using such a binning scheme, it is possible to compute the empirical accuracy for certain confidence levels, as it is known for a long time already in reconstructing confidence outputs for Viterbi decoding [HR90]. Common methods are histogram binning [ZE01], isotonic regression [ZE02] or more advanced methods like Bayesian binning into quantiles (BBQ) [NCH15] and ensembles of near-isotonic regression (ENIR) [NC16]. Another way to get a calibration mapping is to use scaling methods based on logistic regression like Platt scaling [Pla99], temperature scaling [GPSW17] and beta calibration [KSFF17].
In the setting of probabilistic regression, a model is calibrated if, e.g., 95% of the true target values are below or equal to a credible level of 95% (so called *quantile-calibrated regression*) [GBR07, KFE18, SDKF19]. A regression model is usually calibrated by fine-tuning its predicted CDF in a post-processing step to match the empirical frequency. Common approaches utilize isotonic regression [KFE18], logistic and beta calibration [SKF18], as well as Gaussian process models [SKF18, SDKF19] to build a calibration mapping. In contrast to quantile-calibrated regression, [SDKF19] have recently introduced the concept of distribution calibration, where calibration is applied on a distribution level and naturally leads to calibrated quantiles.
Recent work has shown that miscalibration in the scope of *object detection* also depends on the position and scale of a detected object [KKSH20]. The additional box regression output is denoted by $\hat{R}$ with J as the size of the used box encoding. Furthermore, if we have no knowledge about all anchors of a model (which is a common case in many applications), it is not possible to determine the accuracy. Therefore, Küppers et al. [KKSH20] use the precision as a surrogate for accuracy and propose that an *object*

detection model is perfectly calibrated if

$$\underbrace{\mathbb{P}(M = 1|\hat{P} = p, \hat{Y} = y, \hat{R} = r)}_{\text{precision given } p,y,r} = \underbrace{p}_{\text{confidence}} \qquad \forall p \in [0, 1], y \in \mathcal{Y}, r \in \mathbb{R}^J \tag{8}$$

is fulfilled, where $M = 1$ denotes a correct prediction that matches a ground-truth object with a chosen IoU threshold and $M = 0$ denotes a mismatch, respectively. The authors propose the detection-expected calibration error (D-ECE) as the extension of the ECE to object detection tasks in order to measure miscalibration also by means of the position and scale of detected objects. Other approaches try to fine-tune the regression output in order to obtain more reliable object proposals [JLM+18, RTG+19] or to add a regularization term to the training objective such that training yields models that are both well-performing and well-calibrated [PTC+17, SSH19].

7 Aggregation

Maram Akila[1]

From a high-level perspective, a *neural network* is based on processing inputs and coming to some output conclusion, e.g., mapping incoming image data onto class labels. Aggregation or collection of non-independent information on either the input or output side of this network function can be used as a tool to leverage its performance and reliability. Starting with the input, any additional "dimension" to add data can be of use. For example, in the context of autonomous vehicles this might be input from any further sensor measuring the same scene as the original one, e.g., stereo cameras or LiDAR. Combining those sensor sets for prediction is commonly referred to as *sensor fusion* [CBSW19]. Staying with the example, the scene will be monitored consecutively providing a whole (temporally ordered) stream of input information. This may be used either by adjusting the network for this kind of input [KLX+17] or in terms of a post-processing step, in which the predictions are aggregated by some measure of temporal consistency.

Another more implicit form of aggregation is training the neural network on several "independent" tasks, e.g., segmentation and depth regression. Although the individual task is executed on the same input, the overall performance can still benefit from the correlation among all given tasks. We refer to the discussion on *multi-task networks* in Sec. 9.2. By extension, solving the same task in multiple different ways, can be beneficial for performance and provide a measure of redundancy. In this survey, we focus on single-task systems and discuss *ensemble methods* in the next section and the use of *temporal consistency* in the one thereafter.

7.1 Ensemble Methods

Joachim Sicking[1]

Training a neural network is optimizing its parameters to fit a given training data set. The commonly used gradient-based optimization schemes cause convergence in a 'nearby' local minimum. As the loss landscapes of neural networks are notoriously non-convex [CHM+15], various locally optimal model parameter sets exist. These local optima differ in the degree of optimality ("deepness"), qualitative characteristics ("optimal for different parts of the training data") and their generalizability to unseen data (commonly referred to by the geometrical terms of "sharpness" and "flatness" of minima [KMN+16]).
A single trained network corresponds to one local minimum of such a loss landscape and thus captures only a small part of a potentially diverse set of solutions. *Network ensembles* are collections of models and therefore better suited to reflect this multi-modality. Various modelling choices shape a loss landscape: the selected model class and its meta-parameters (like architecture and layer width), the training data and the optimization objective. Accordingly, approaches to diversify ensemble components range from combinations of different model classes over varying training data (bagging) to methods that train and weight ensemble components to make up for the flaws of other ensemble members (boosting) [Bis06].
Given the millions of parameters of application-size networks, ensembles of NNs are resource-demanding w.r.t. computational load, storage and runtime during training and inference. This complexity increases linearly with ensemble size for naïve ensembling. Several approaches were put forward to reduce some dimensions of this complexity: *snapshot ensembles* [HLP+17] require only one model optimization with a cyclical learning-rate schedule—leading to an optimized training runtime. The resulting training trajectory passes through several local minima. The corresponding models compose the ensemble. On the contrary, *model distillation* [HVD15a] tackles runtime at inference. They 'squeeze' a NN ensemble into a single model that is optimized to capture the gist of the model set. However, such a compression goes along with reduced performance compared to the original ensemble.
Several hybrids of single model and model ensemble exist: Multi-head networks [AJD18] share a backbone network that provides inputs to multiple prediction networks. Another variant are mixture-of-expert models that utilize a gating network to assign inputs to specialized expert networks [SMM+17]. Multi-task networks (cf. Sec. 9.2) and Bayesian approximations of NNs (cf. Sec. 6.3 and Sec. 6.2) can be seen as implicit ensembles.
NN ensembles (or deep ensembles) are not only used to boost model quality. They pose the frequentist's approach to estimating NN uncertainties and are state-of-the-art in this regard [LPB17, SOF+19]. The emergent field of federated learning is concerned with the integration of decentrally trained ensemble components [MMR+16] and safety-relevant applications of ensembling range from autonomous driving [Zha12] to medical diagnostics [RRMH17]. Taking this safe-ML perspective, promising research directions comprise a more principled and efficient composition of model ensembles, e.g., by application-driven

diversification, as well as improved techniques to miniaturize ensembles, e.g., by gaining a better understanding of methods like distillation. In the long run, better designed, more powerful learning systems might partially reduce the need for combining weaker models in a network ensemble.

7.2 Temporal Consistency

Timo Sämann[4]

The focus of previous DNN development for semantic segmentation has been on single image prediction. This means that the final and intermediate results of the DNN are discarded after each image. However, the application of a computer vision model often involves the processing of images in a sequence, i.e., there is a temporal consistency in the image content between consecutive frames (for a metric, cf., e.g., [VBB+20]). This consistency has been exploited in previous work to increase quality and reduce computing effort. Furthermore, this approach offers the potential to improve the robustness of DNN prediction by incorporating this consistency as a-priori knowledge into DNN development. The relevant work in the field of video prediction can be divided in two major approaches:

1. DNNs are specially designed for video prediction. This usually requires training from scratch and the availability of training data in a sequence.

2. A transformation from single prediction DNNs to video prediction DNNs takes place. Usually no training is required, i.e., the existing weights of the model can be used unaltered.

The first set of approaches often involves *conditional random fields* (CRF) and its variants. CRFs are known for their use as postprocessing step in the prediction of semantic segmentation, in which their parameters are learned separately or jointly with the DNN [ZJRP+15]. Another way to use spatiotemporal features is to include 3D convolutions, which add an additional dimension to the conventional 2D convolutional layer. Tran et al. [TBF+15] use 3D convolution layers for video recognition tasks such as action and object recognition. One further approach to use spatial and temporal characteristics of the input data is to integrate *long short-term memory* (LSTM) [HS97], a variant of the *recurrent neural network* (RNN). Fayyaz et al. [FSS+16] integrate LSTM layers between the encoder and decoder of their convolutional neural network for semantic segmentation. The significantly higher GPU memory requirements and computational effort are a disadvantage of this method. More recently, Nilsson and Sminchisescu [NS18] deployed *gated recurrent units*, which generally requires significantly less memory. An approach to improve temporal consistency of automatic speech recognition outputs is known as a posterior-in-posterior-out (PIPO) LSTM "sequence enhancer", a postfilter which could be applicable to video processing as well [LSF19]. A disadvantage of the described methods is that sequential data for training must be available, which may be limited or show a lack of diversity.

The second class of approaches has the advantage that it is model-independent most of the time. Shelhamer et al. [SRHD16] found that the deep feature maps within the network change only slightly with temporal changes in video content. Accordingly, [GJG17] calculate the optical flow of the input images from time steps t_0 and t_{-1} and convert it into the so-called *transform flow* which is used to transform the feature maps of the time step t_{-1} so that an aligned representation to the feature map t_0 is achieved. Sämann et al. [SAMG19] use a confidence-based combination of feature maps from previous time steps based on the calculated optical flow.

8 Verification

Gesina Schwalbe[3]

Verification and validation is an integral part of the safety assurance for any safety critical systems. As of the functional safety standard for automotive systems [3218], *verification* means to determine whether given requirements are met [3218, 3.180], such as performance goals. *Validation* on the other side tries to assess whether the given requirements are sufficient and adequate to guarantee safety [3218, 3.148], e.g., whether certain types of failures or interactions simply were overlooked. The latter is usually achieved via extensive testing in real operation conditions of the integrated product. This differs from the notion of validation used in the machine learning community in which it usually refers to simple performance tests on a selected dataset. In this section, we want to concentrate on general verification aspects for deep neural networks.
Verification as in the safety domain encompasses (manual) inspection and analysis activities, and testing. However, the contribution of single processing steps within a neural network to the final behavior can hardly be assessed manually (compare to the problem of interpretability in Sec. 5). Therefore, we here will concentrate on different approaches to verification testing. Section 8.1 covers approaches to systematic test data selection. While the suggested methods assume full access to the model internals for coverage measurement, this is not in all cases available. Therefore, Sec. 8.2 highlights assessment approaches that consider the DNN as a black-box component. Also, the topic of verification activity during operation with the help of observers is discussed.

8.1 Formal Testing

Christian Heinzemann[2], Gesina Schwalbe[3], Matthias Woehrle[2]

Formal testing refers to testing methods that include formalized and formally verifiable steps, e.g., for test data acquisition, or for verification in the local vicinity of test samples. For image data, local testing around valid samples is usually more practical than fully formal verification: (Safety) properties are not expected to hold on the complete input

space but only on the much smaller unknown lower-dimensional manifold of real images [WGH19]. Sources of such samples can be real ones or generated ones using an input space formalization or a trained generative model.
Coverage criteria for the data samples are commonly used for two purposes: (a) deciding when to stop testing or (b) identifying missing tests. For CNNs, there are at least three different approaches towards coverage: (1) approaches that establish coverage based on a model with semantic features of the input space [GHHW20], (2) approaches trying to semantically cover the latent feature space of neural network or a proxy network (e.g., an autoencoder) [SS20a], and (3) approaches trying to cover neurons and their interactions, inspired by classical software white-box analysis [PCYJ17, SHK+19].
Typical types of properties to verify are simple test performance, local stability (robustness), a specific structure of the latent spaces like embedding of semantic concepts [SS20b], and more complex logical constraints on inputs and outputs, which can be used for testing when fuzzified [RDG18]. Most of these properties require in-depth semantic information about the DNN inner workings, which is often only available via interpreting intermediate representations [KWG+18], or interpretable proxies / surrogates (cf. Sec. 5.4), which do not guarantee fidelity.
There exist different testing and formal verification methods from classical software engineering that have already been applied to CNNs. *Differential testing* as used by DeepXPlore [PCYJ17] trains n different CNNs for the same task using independent data sets and compares the individual prediction results on a test set. This allows to identify inconsistencies between the CNNs but no common weak spots. *Data augmentation* techniques start from a given data set and generate additional transformed data. *Generic data augmentation* for images like rotations and translation are state-of-the-art for training but may also be used for testing. *Concolic testing* approaches incrementally grow test suites with respect to a coverage model to finally achieve completeness. Sun et al. [SWR+18] use an adversarial input model based on some norm (cf. Sec. 4.1), e.g., an L_p-norm, for generating additional images around a given image using concolic testing. *Fuzzing* generates new test data constrained by an input model and tries to identify interesting test cases, e.g., by optimizing white-box coverage mentioned above [OOAG19]. Fuzzing techniques may also be combined with the differential testing approach discussed above [GJZ+18]. In all these cases it needs to be ensured that the image as well as its meta-data remain valid for testing after transformation. Finally, *proving methods* surveyed by Liu et al. [LAL+19] try to formally prove properties on a trained neural network, e.g., based on satisfiablity modulo theories (SMT). These approaches require a formal characterization of an input space and the property to be checked, which is hard for non-trivial properties like contents of an image.
Existing formal testing approaches can be quite costly to integrate into testing workflows (cf. [WGH19]): Differential testing and data augmentation require several inferences per initial test sample; concolic and fuzzy testing apply an optimization to each given test sample, while convergence towards the coverage goals is not guaranteed; also, the iterative approaches need tight integration into the testing workflow; and lastly, proving methods usually have to balance computational efficiency against the precision or completeness of the result [LAL+19]. Another challenge of formal testing is that machine learning

applications usually solve problems for which no (formal) specification is possible. This makes it hard to find useful requirements for testing [ZHML19] and properties that can be formally verified. Even partial requirements such as specification of useful input perturbations, specified corner cases, and valuable coverage goals are typically difficult to identify [SS20a, BTLF20].

8.2 Black Box Methods

Jonas Löhdefink[15], Julia Rosenzweig[1]

In machine learning literature, neural networks are often referred to as black boxes due to the fact that their internal operations and their decision making are not completely understood [SZT17], hinting at a lack of interpretability and transparency. However, in this survey we consider a black box to be a machine learning model to which we only have oracle (query) access [PMG+17, TZJ+16]. That means we can query the model to get input-output pairs, but we do not have access to the specific architecture (or weights, in case of neural networks). As [OAFS18] describes, black boxes are increasingly wide-spread, e.g., healthcare, autonomous driving or ML as a service in general, due to proprietary, privacy or security reasons.

As deploying black boxes gains popularity, so do methods that aim to extract internal information such as architecture and parameters or to find out, whether a sample belongs to the training dataset. These include *model extraction attacks* [KTP+20, TZJ+16], *membership inference attacks* [SSSS17], general attempts to reverse-engineer the model [OAFS18] or to attack it adversarially [PMG+17]. Protection and counter-measures are also actively researched: [KMAM18] proposes a warning system that estimates how much information an attacker could have gained from queries. The authors of [ABC+18] use watermarks for models to prevent illegal re-distribution and to identify intellectual property.

Many papers in these fields make use of so-called *surrogate*, *avatar* or *proxy models* that are trained on input-output pairs of the black box. In case the black-box output is available in soft form (e.g., logits), distillation as first proposed by [HVD15b] can be applied to train the surrogate (student) model. Then, any white-box analysis can be performed on the surrogates (cf. e.g., [PMG+17]) to craft adversarial attacks targeted at the black box. More generally, (local) surrogates as for example in [RSG16] can be used to (locally) explain its decision-making. Moreover, these techniques are also of interest if one wants to compare or test black-box models (cf. Sec. 8.1, formal verification). This is the case, among others, in ML marketplaces, where you wish to buy a pre-trained model [ABC+18], or if you want to verify or audit that a third-party black-box model obeys regulatory rules (cf. [CH19]).

Another topic of active research are so-called *observers*. The concept of observers is to evaluate the interface of a black-box module to determine if it behaves as expected within a given set of parameters. The approaches can be divided into *model-explaining* and *anomaly-detecting* observers. First, model explanation methods answer the question of

which input characteristic is responsible for changes at the output. The observer is able to alter the inputs for this purpose. If the input of the model under test evolves only slightly but the output changes drastically, this can be a signal that the neural network is mislead, which is also strongly related to adversarial examples (cf. Chapter 4). Hence, the reason for changes in the classification result via the input can be very important. In order to figure out in which region of an input image the main reason for the classification is located, [FV17] "delete" information from the image by replacing regions with generated patches until the output changes. This replaced region is likely responsible for the decision of the neural network. Building upon this, [UEKH19] adapt the approach to medical images and generate "deleted" regions by a variational autoencoder (VAE). Second, anomaly-detecting observers register input and output anomalies, either examining input and output independently or as an input-output pair, and predict the black-box performance in the current situation. In contrast to model-explaining approaches, this set of approaches has high potential to be used in an online scenario since it does not need to modify the model input. The maximum mean discrepancy (MMD) [BGR+06] measures the domain gap between two data distributions independently from the application and can be used to raise a warning if input or output distributions during inference deviate too strongly from their respective training distributions. By use of a GAN-based autoencoder [LFK+20] perform a domain shift estimation using neural networks in conjunction with the Wasserstein distance as domain mismatch metric. This metric can also be evaluated by use of a casual time-variant aggregation of distributions during inference time.

9 Architecture

Michael Weber[14]

In order to solve a specific task, the architecture of a CNN and its building blocks play a significant role. Since the early days of using CNNs in image processing, when they were applied to handwriting recognition [LBD+89] and the later breakthrough in general image classification [KSH12], the architecture of the networks has changed radically. Did the term of *deep learning* for these first convolutional neural networks imply a depth of approximately four layers, their depth increased significantly during the last years and new techniques had to be developed to successfully train and utilize these networks [HZRS16]. In this context, new activation functions [RZL18] as well as new loss functions [LGG+17] have been designed and new optimization algorithms [KB15] were investigated.
With regard to the layer architecture, the initially alternating repetition of convolution and pooling layers as well as their characteristics have changed significantly. The convolution layers made the transition from a few layers with often large filters to many layers with small filters. A further trend was then the definition of entire modules, which were used repeatedly within the overall architecture as so-called *network in network* [LCY14].
In areas such as autonomous driving, there is also a strong interest in the simultaneous

execution of different tasks within one single convolutional neural network architecture. This kind of architecture is called *multi-task learning* (MTL) [Car97] and can be utilized in order to save computational resources and at the same time to increase performance of each task [KTMFs20]. Within such multi-task networks, usually one shared feature extraction part is followed by one separate so-called head per task [TWZ+18].
In each of these architectures, manual design using expert knowledge plays a major role. The role of the expert is the crucial point here. In recent years, however, there have also been great efforts to automate the process of finding architectures for networks or, in the best case, to learn them. This is known under the name *neural architecture search* (NAS).

9.1 Building Blocks

Michael Weber[14]

Designing a convolutional neural network typically includes a number of design choices. The general architecture usually contains a number of convolutional and pooling layers which are arranged in a certain pattern. Convolutional layers are commonly followed by a non-linear activation function. The learning process is based on a loss function which determines the current error and an optimization function that propagates the error back to the single convolution layers and its learnable parameters.
When CNNs became state of the art in computer vision [KSH12], they were usually built using a few alternating convolutional and pooling layers having a few fully connected layers in the end. It turned out that better results are achieved with deeper networks and so the number of layers increased [SZ15] over the years. To deal with these deeper networks, new architectures had to be developed. In a first step, to reduce the number of parameters, the convolutional layers with partly large filter kernels were replaced by several layers with small 3×3 kernels. Today, most architectures are based on the *network in network* principle [LCY14], where more complex modules are used repeatedly. Examples of such modules are the *inception module* from GoogleNet [SWY+15] or the *residual block* from ResNet [HZRS16]. While the inception module consists of multiple parallel strings of layers, the residual blocks are based on the *highway network* [SGS15], which means that they can bypass the original information and the layers in between are just learning residuals. With ResNeXt [XGD+17] and Inception-ResNet [SIVA17] there already exist two networks that combine both approaches. For most tasks, it turned out that replacing the fully connected layers by convolutional layers is much more convenient making the networks fully convolutional [LSD15]. These so-called *fully convolutional networks* (FCN) are no longer bound to fixed input dimensions. Note that with the availability of convolutional long short-term memory (ConvLSTM) structures also fully convolutional recurrent neural networks (FCRNs) became available for fully scalable sequence-based tasks [SCW+15, SDF+20].
Inside the CNNs, the *rectified linear unit* (ReLU) has been the most frequently used activation function for a long time. However, since this function suffers from problems

related to the mapping of all negative values to zero like the vanishing gradient problem, new functions have been introduced in recent years. Examples are the *exponential linear unit* (ELU), *swish* [RZL18] and the *non-parametric linearly scaled hyperbolic tangent* (LiSHT) [RMDC19]. In order to be able to train a network consisting of these different building blocks, the loss function is the most crucial part. This function is responsible for how and what the network ultimately learns and how exactly the training data is applied during the training process to make the network train faster or perform better. So the different classes can be weighted in a classification network with fixed values or so-called α-balancing according to their probability of occurrence. Another interesting approach is weighting training examples according to their easiness for the current network [LGG+17], [WFZ19]. For multi-task learning also weighting tasks based on their uncertainty [KGC18] or gradients [CBLR18] can be done as further explained in Sec. 9.2. A closer look on how a modification of the loss function might affect safety-related aspects is given in Sec. 3.2.

9.2 Multi-Task Networks

Marvin Klingner[15], Varun Ravi-Kumar[4], Timo Sämann[4], Gesina Schwalbe[3]

Multi-task learning (MTL) in the context of neural networks describes the process of optimizing several tasks simultaneously by learning a unified feature representation [Car97, GHL+20, RBV18, KBFs20] and coupling the task-specific loss contributions, thereby enforcing cross-task consistency [CPMA19, LYW+19, KTMFs20].
Unified feature representation is usually implemented by sharing the parameters of the initial layers inside the encoder (also called feature extractor). It not only improves the single tasks by more generalized learned features but also reduces the demand for computational resources at inference. Not an entirely new network has to be added for each task but only a task-specific decoder head. It is essential to consider the growing amount of visual perception tasks in autonomous driving, e.g., depth estimation, semantic segmentation, motion segmentation, and object detection. While the parameter sharing can be soft, as in *cross stitch* [MSGH16] and *sluice networks* [RBAS17], or hard [TWZ+18, Kok17], meaning ultimately sharing the parameters, the latter is usually preferred due to its straightforward implementation and lower computational complexity during training and inference.
Compared to implicitly coupling tasks via a shared feature representation, there are often more direct ways to optimize the tasks inside cross-task losses jointly. It is only made possible as, during MTL, there are network predictions for several tasks, which can be enforced to be consistent. As an example, sharp depth edges should only be at class boundaries of semantic segmentation predictions. Often both approaches to MTL are applied simultaneously [CLLW19, YZS+18] to improve a neural network's performance as well as to reduce its computational complexity at inference.
While the theoretical expectations for MTL are quite clear, it is often challenging to find a good weighting strategy for all the different loss contributions as there is no theoretical

basis on which one could choose such a weighting with early approaches either involving heuristics or extensive hyperparameter tuning. The easiest way to balance the tasks is to use uniform weight across all tasks. However, the losses from different tasks usually have different scales, and uniformly averaging them suppresses the gradient from tasks with smaller losses. Addressing these problems, Kendall et al. [KGC18] propose to weigh the loss functions by the *homoscedastic uncertainty* of each task. One does not need to tune the weighting parameters of the loss functions by hand, but they are adapted automatically during the training process. Concurrently Chen et al. [CBLR18] propose *GradNorm*, which does not explicitly weigh the loss functions of different tasks but automatically adapts the gradient magnitudes coming from the task-specific network parts on the backward pass. Liu et al. [LJD19] proposed dynamic weight average (DWA), which uses an average of task losses over time to weigh the task losses.

9.3 Neural Architecture Search

Patrick Feifel[8], Seyed Eghbal Ghobadi[8]

In the previous sections we saw manually engineered modifications of existing CNN architectures proposed by ResNet [HZRS16] or Inception [SWY+15]. They are results of human design and showed their ability to improve performance. ResNet introduces a *skip connection* in building blocks and Inception makes use of its specific *inception module*. Hereby, the intervention by an expert is crucial. The approach of *neural architecture search* (NAS) aims to automate this time-consuming and manual design of neural network architectures.

NAS is closely related to hyperparameter optimization (HO), which is described in Sec. 3.1. Originally, both tasks were solved simultaneously. Consequently, the kernel size or number of filters were seen as additional hyperparamters. Nowadays, the distinction between HO and NAS should be stressed. The concatenation of complex building blocks or modules cannot be accurately described with single parameters. This simplification is no longer suitable.

To describe the NAS process, the authors of [EMH19b] define three steps: (1) definition of search space, (2) search strategy and (3) performance estimation strategy.

The majority of search strategies take advantage of the *NASNet search space* [ZVSL18] which arranges various operations, e.g., convolution, pooling within a single cell. However, other spaces based on a chain or multi-branch structure are possible [EMH19b]. The search strategy comprises advanced methods from sequential model-based optimization (SMBO) [LZN+18], Bayesian optimization [KNS+18], evolutionary algorithms [RAHL19, EMH19a], reinforcement learning [ZVSL18, PGZ+18] and gradient descent [LSY19, SDW+19]. Finally, the performance estimation describes approximation techniques due to the impracticability of multiple evaluation runs. For a comprehensive survey regarding the NAS process we refer to [EMH19b].

Recent research has shown that reinforcement learning approaches such as NASNet-A [ZVSL18] and ENAS [PGZ+18] are partly outperformed by evolutionary algorithms, e.g.,

AmoebaNet [RAHL19] and gradient-based approaches, e.g., DARTS [LSY19].
Each of these approaches focuses on different optimization aspects. Gradient-based methods are applied to a continuous search space and offer faster optimization. On the contrary, the evolutionary approach LEMONADE [EMH19a] enables multi-object optimization by considering the conjunction of resource consumption and performance as the two main objectives. Furthermore, single-path NAS [SDW+19] extends the multi-path approach of former gradient-based methods and proposes the integration of 'over-parameterized superkernels', which significantly reduces memory consumption.
The focus of NAS is on the optimized combination of humanly predefined CNN elements with respect to objectives such as resource consumption and performance. NAS offers automation, however, the realization of the objectives is strongly limited by the potential of the CNN elements.

10 Model Compression

Serin Varghese[7]

Recent developments in CNNs have resulted in neural networks being the state-of-the-art in computer vision tasks like image classification [KSH12, HZRS15, MGR+18], object detection [Gir15, RDGF15, HGDG17] and semantic segmentation [CPSA17, ZSR+19, WSC+19, LBS+19]. This is largely due to the increasing availability of hardware computational power and an increasing amount of training data. We also observe a general upwards trend of the complexities of the neural networks along with their improvement in state-of-the-art performance. These CNNs are largely trained on back-end servers with significantly higher computing capabilities. The use of these CNNs in real-time applications are inhibited due to the restrictions on hardware, model size, inference time, and energy consumption. This led to an emergence of a new field in machine learning, commonly termed as model compression. Model compression basically implies reducing the memory requirements, inference times and model size of DNNs to eventually enable the use of neural networks on edge devices. This is tackled by different approaches such as *network pruning* (identifying weights or filters that are not critical for network performance), *weight quantizations* (reducing the precision of the weights used in the network), *knowledge distillation* (a smaller network is trained with the knowledge gained by a bigger network), and *low-rank factorization* (decomposing a tensor into multiple smaller tensors). In this section, we introduce some of these methods for model compression and discuss in brief the current open challenges and possible research directions with respect to its use in automated driving applications.

10.1 Pruning

Falk Kappel[13], Serin Varghese[7]

Pruning has been used as a systematic tool to reduce the complexity of deep neural networks. The redundancy in DNNs may exist on various levels, such as the individual weights, filters, and even layers. All the different methods for pruning try to take advantage of these available redundancies on various levels. Two of the initial approaches for neural networks proposed weight pruning in the 1990s as a way of systematically damaging neural networks [CDS90, Ree93]. As these weight pruning approaches do not aim at changing the structure of the neural network, these approaches are called *unstructured pruning*. Although there is reduction in the size of the network when it is saved in sparse format, the acceleration depends on the availability of hardware that facilitate sparse multiplications. As pruning filters and complete layers aim at exploiting the available redundancy in the architecture or structure of neural networks, these pruning approaches are called *structured pruning*. Pruning approaches can also be broadly classified into: data-dependent and data-independent methods. Data-dependent methods [LLS+17, LWL17, HZS17] make use of the training data to identify filters to prune. Theis et al. [TKTH18] and Molchanov et al. [MTK+17] propose a greedy pruning strategy that identifies the importance of feature maps one at a time from the network and measures the effect of removal of the filters on the training loss. This means that filters corresponding to those feature maps that have least effect on training loss are removed from the network. Within data-independent methods [LKD+17, HKD+18, YLLW18, ZQF+18], the selection of CNN filters to be pruned are based on the statistics of the filter values. Li et al. [LKD+17] proposed a straightforward method to calculate the rank of filters in a CNN. The selection of filters are based on the ℓ_1-norm, where the filter with the lowest norm is pruned away. He et al. [HZS17] employ a LASSO regression-based selection of filters to minimize the least squares reconstruction.
Although the above-mentioned approaches demonstrated that a neural network can be compressed without affecting the accuracy, the effect on robustness is largely unstudied. Dhillon et al. [DAL+18] proposed pruning a subset of activations and scaling up the survivors to show improved adversarial robustness of a network. Lin et al. [LGH19] quantize the precision of the weights after controlling the Lipschitz constant of layers. This restricts the error propagation property of adversarial pertubations within the neural network. Ye et al. [YLX+19] evaluated the relationship between adversarial robustness and model compression in detail and show that naive compression has a negative effect on robustness. Gui et al. [GWY+19] co-optimize robustness and compression constraints during the training phase and demonstrate improvement in the robustness along with reduction in the model size. However, these approaches have mostly been tested on image classification tasks and on smaller datasets only. Their effectiveness on safety-relevant automated driving tasks such as object detection and semantic segmentation tasks are not studied and remains an open research challenge.

10.2 Quantization

Firas Mualla[13]

Quantization of a random variable x having a probability density function $f(x)$ is the process of dividing the range of x into intervals, each is represented using a single value (also called reconstruction value), such that the following reconstruction error is minimized:

$$\sum_{i=1}^{L} \int_{b_i}^{b_{i+1}} (q_i - x)^2 f(x) dx, \tag{9}$$

where b_i is the left-side border of the i-th interval, q_i is its reconstruction value, and L is the number of intervals, e.g., $L = 8$ for a 3-bit quantization. This definition can be extended to multiple dimensions as well.

Quantization of neural networks has been around since the 1990s [Guo18], however, with a focus in the early days on improving the hardware implementations of these networks. In the deep learning literature, a remarkable application of quantization combined with unstructured pruning can be found in the approach of *deep compression* [HMD16], where 1-dimensional k-means is utilized to cluster the weights per layer and thus finding the L cluster centers (q_i values in (9)) iteratively. This procedure conforms to an implicit assumption that $f(x)$ has the same spread inside all clusters. Deep compression can reduce the network size needed for image classification by a factor of 35 for AlexNet and a factor of 49 for VGG-16 without any loss in accuracy. However, as pointed out in [JKC+18], these networks from the early deep learning days are over-parameterized and a less impressing compression factor is thus expected when the same technique is applied to lightweight architectures such as MobileNet and SequeezeNet. For instance, considering SqueezeNet (50 times smaller than AlexNet), the compression factor of deep compression without accuracy loss drops to about 10.

Compared to scalar quantization used in deep compression, there were attempts to exploit the structural information by applying variants of vector quantization of the weights [GLYB14, CEL20, SJG+20]. Remarkably, in the latter (i.e., [SJG+20]), the reconstruction error of the activations (instead of the weights) is minimized in order to find an optimal codebook for the weights, as the ultimate goal of quantization is to approximate the network's output not the network itself. This is performed in a layer-by-layer fashion (as to prevent error accumulation) using activations generated from unlabeled data.

Other techniques [MDSN17, JKC+18] apply variants of so-called "linear" *quantization*, i.e., the quantization staircase has a fixed interval size. This paradigm conforms to an implicit assumption that $f(x)$ in (9) is uniform and is thus also called *uniform quantization*. The uniform quantization is widely applied both in specialized software packages such as the `Texas Instruments Deep Learning Library` (automotive boards) [MDS+18] and in general-purpose libraries such as the `Tensorflow Lite`. The linearity assumption enables practical implementations, as the quantization and dequantization

can be implemented using a scaling factor and an intercept, whereas no codebook needs to be stored. In many situations, the intercept can be omitted by employing a symmetric quantization mapping. Moreover, for power of 2 ranges, the scaling ends up being a bitwise shift operator, where quantization and dequantization differ only in the shift direction. It is also straightforward to apply this scheme *dynamically*, i.e., for each tensor separately using a tensor-specific multiplicative factor. This can be easily applied not only to filters (weight tensors) but also to activation tensors (see for instance [MDSN17]). Unless the scale factor in the linear quantization is assumed constant by construction, it is computed based on the statistics of the relevant tensor and can be thus sensitive to outliers. This is known to result in a low precision quantization. In order to mitigate this issue, the original range can be *clipped* and thus reduced to the most relevant part of the signal. Several approaches are proposed in the literature for finding an optimal clipping threshold: simple percentile analysis of the original range (e.g., clipping 2% of the largest magnitude values), minimizing the mean square error between the quantized and original range in the spirit of (9) [BNS19], or minimizing the Kullback-Leibler divergence between the original and the quantized distributions [Mig17]. While the clipping methods trade off large quantization errors of outliers against small errors of inliers [WJZ+20], other methods tackle the outliers problem using a different trade-off, see for instance the outlier channel splitting approach in [ZHD+19].

An essential point to consider when deciding for a quantization approach for a given problem is the allowed or intended interaction with the training procedure. The so-called *post-training quantization*, i.e., quantization of a pre-trained network, seems to be attractive from a practical point of view: No access to training data is required and the quantization and training toolsets can be independent from each other. On the other hand, the training-aware quantization methods often yield higher inference accuracy and shorter training times. The latter is a serious issue for large complicated models which may need weeks to train on modern GPU clusters. The training-aware quantization can be implemented by inserting fake quantization operators in the computational graph of the forward-pass during training (simulated quantization), whereas the backward pass is done as usual in floating-point resolution [JKC+18]. Other approaches [ZGY+19, ZWN+16] go a step further by quantizing the gradients as well. This leads to much lower training time, as the time of the often computationally expensive backward pass is reduced. The gradient's quantization, however, is not directly applicable as it requires the derivative of the quantization function (staircase-like), which is zero almost everywhere. Luckily, this issue can be handled by employing a *straight-through estimator* [BLC13] (approximating the quantization function by an identity mapping). There are also other techniques proposed recently to mitigate this problem [UMY+19, LM19].

11 Discussion

We have presented an extensive overview of approaches to effectively handle safety concerns accompanying deep learning: lack of generalization, robustness, explainability, plausibility, and efficiency. It has been described which lines of research we deem prevalent,

important, and promising for each of the individual topics and categories into which the presented methods fall.
The reviewed methods alone will not provide safe ML systems as such – and neither will their future extensions. This is due to the limitations of quantifying complex real-world contexts. A complete and plausible *safety argumentation* will, thus, require more than advances in methodology and theoretical understanding of neural network properties and training processes. Apart from methodological progress, it will be necessary to gain practical experience in using the presented methods to gather evidence for overall secure behavior, using this evidence to construct a tight safety argument, and testing its validity in various situations.
In particular, each autonomously acting robotic system with state-of-the-art deep-learning-based perception and non-negligible actuation may serve as an object of study and is, in fact, in need of this kind of systematic reasoning before being transferred to widespread use or even market entry. We strongly believe that novel scientific insights, the potential market volume, and public interest will drive the arrival of reliant and trustworthy AI technology.

Acknowledgment

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project "KI Absicherung – Safe AI for Automated Driving". The authors would like to thank the consortium for the successful cooperation. Furthermore, this research has been funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01IS18038B).

References

[3218] ISO/TC 22/SC 32. *ISO 26262-1:2018(en): Road Vehicles — Functional Safety — Part 1: Vocabulary*, volume 1. International Organization for Standardization, second edition, 2018.

[AAAB18] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon. Ganomaly: Semi-Supervised Anomaly Detection via Adversarial Training. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 622–637, 2018.

[ABC+18] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring. In *Proceedings of the USENIX Security Symposium*, pages 1615–1631. USENIX Association, 2018.

[ACKK14] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, 35(4):105–120, 2014.

[ACÖG17] Marco Ancona, Enea Ceolini, A. Cengiz Öztireli, and Markus H. Gross. A Unified View of Gradient-Based Attribution Methods for Deep Neural Networks. *Computing Research Repository (CoRR)*, arXiv:1711.06104, 2017.

[ACS19] Matt Angus, Krzysztof Czarnecki, and Rick Salay. Efficacy of Pixel-Level OOD Detection for Semantic Segmentation. *Computing Research Repository (CoRR)*, arXiv:1911.02897, 2019.

[ACW18] A. Athalye, N. Carlini, and D. Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 274–283, 2018.

[AEIK17] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing Robust Adversarial Examples. *Computing Research Repository (CoRR)*, arXiv:1707.07397, 2017.

[AGM+18] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. *Computing Research Repository (CoRR)*, arXiv:1810.03292, 2018.

[AJD18] Sercan Ö Arık, Heewoo Jun, and Gregory Diamos. Fast Spectrogram Inversion using Multi-Head Convolutional Neural Networks. *IEEE Signal Processing Letters*, 26(1):94–98, 2018.

[AK12] M. G. Augasta and T. Kathirvalavakumar. Rule Extraction from Neural Networks — A Comparative Study. In *Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME)*, pages 404–408, 2012.

[Ale09] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. `https://www.cs.toronto.edu/~kriz/cifar.html`, 2009.

[AS17] Anshul Kundaje Avanti Shrikumar, Peyton Greenside. Learning Important Features Through Propagating Activation Differencess. *Computing Research Repository (CoRR)*, arXiv:1704.02685, 2017.

[AW18] Aharon Azulay and Yair Weiss. Why Do Deep Convolutional Networks Generalize so Poorly to Small Image Transformations? *Computing Research Repository (CoRR)*, arXiv:1805.12177, 2018.

[BBL+07] Michael W. Berry, Murray Browne, Amy N. Langville, V. Paul Pauca, and Robert J. Plemmons. Algorithms and Applications for Approximate Nonnegative Matrix Factorization. *Computational Statistics & Data Analysis*, 52(1):155–173, 2007.

[BBLF19] Jan-Aike Bolte, Andreas Bär, Daniel Lipinski, and Tim Fingscheidt. Towards Corner Case Detection for Autonomous Driving. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 438–445, 2019.

[BBM+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLoS ONE*, 10(7):e0130140, 2015.

[BCG+18] Christopher Bowles, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger N. Gunn, Alexander Hammers, David Alexander Dickie, Maria del C. Valdés Hernández, Joanna M. Wardlaw, and Daniel Rueckert. GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks. *Computing Research Repository (CoRR)*, arXiv:1810.10863, 2018.

[BCK+08] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning Bounds for Domain Adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 129–136, 2008.

[BCKW15] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1613–1622, 2015.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1798–1828, 2013.

[BDBC+10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Vaughan. A Theory of Learning from Different Domains. *Machine Learning*, 79:151–175, 2010.

[BF17] Shumeet Baluja and Ian Fischer. Adversarial Transformation Networks: Learning to Generate Adversarial Examples. *Computing Research Repository (CoRR)*, arXiv:1703.09387, 2017.

[BFS18] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-Term On-Noard Prediction of People in Traffic Scenes under Uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4194–4202, 2018.

[BFW+19] Yang Bai, Yan Feng, Yisen Wang, Tao Dai, Shu-Tao Xia, and Yong Jiang. Hilbert-Based Generative Defense for Adversarial Examples. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4784–4793, 2019.

[BGR+06] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. Integrating Structured Biological Data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[BGS+19] Simon Burton, Lydia Gauerhof, Bibhuti Bhusan Sethy, Ibrahim Habli, and Richard Hawkins. Confidence Arguments for Evidence of Performance in Machine Learning for Highly Automated Driving Functions. In *Computer Safety, Reliability, and Security (SAFECOMP)*, volume 11699, pages 365–377. Springer International Publishing, 2019.

[BHG+19] Adith Boloor, Xin He, Christopher Gill, Yevgeniy Vorobeychik, and Xuan Zhang. Simple Physical Adversarial Examples against End-To-End Autonomous Driving Models. In *Proceedings of the IEEE International Conference on Embedded Software and Systems (ICESS)*, pages 1–7, 2019.

[BHLHL+16] Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep Gaussian Processes for Regression using Approximate Expectation Propagation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1472–1481, 2016.

[BHP+18] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding Disentangling in β-VAE. *Computing Research Repository (CoRR)*, arXiv:1804.03599, 2018.

[BHSFs19] Andreas Bär, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. On the Robustness of Redundant Teacher-Student Frameworks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1380–1388, 2019.

[Bis06] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[BKD19] Clemens Alexander Brust, Christoph Käding, and Joachim Denzler. Active Learning for Deep Object Detection. *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, 5:181–190, 2019.

[BKOŠ18] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Discriminative Out-of-Distribution Detection for Semantic Segmentation. *Computing Research Repository (CoRR)*, arXiv:1808.07703, 2018.

[BKV+20] Andreas Bär, Marvin Klingner, Serin Varghese, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. Robust Semantic Segmentation by Redundant Networks With a Layer-Specific Loss Contribution and Majority Vote. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1348–1358, 2020.

[BLC13] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *Computing Research Repository (CoRR)*, arXiv:1308.3432, 2013.

[BMM18] Gerrit Bagschik, Till Menzel, and Markus Maurer. Ontology based Scene Creation for the Development of Automated Vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1813–1820, 2018.

[BMR+17] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial Patch. *Computing Research Repository (CoRR)*, arXiv:1712.09665, 2017.

[BNS19] Ron Banner, Yury Nahshan, and Daniel Soudry. Post Training 4-bit Quantization of Convolutional Networks for Rapid-Deployment. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7948–7956, 2019.

[BRW18] Charlotte Bunne, Lukas Rahmann, and Thomas Wolf. Studying Invariances of Trained Convolutional Neural Networks. *Computing Research Repository (CoRR)*, arXiv:1803.05963, 2018.

[BSC18] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. MetaReg: Towards Domain Generalization using Meta-Regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1006–1016, 2018.

[BTLF20] Jasmin Breitenstein, Jan-Aike Termöhlen, Daniel Lipinski, and Tim Fingscheidt. Systematization of Corner Cases for Visual Perception in Automated Driving. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8, 2020.

[BXS+20] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. Explainable Machine Learning in Deployment. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2020.

[C+15] François Chollet et al. Keras. `https://keras.io`, 2015.

[Car97] Rich Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, 1997.

[CBG+17] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval Networks: Improving Robustness to Adversarial Examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1–10, 2017.

[CBLR18] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 794–803, 2018.

[CBSW19] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. LIDAR–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019.

[CDH+16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2172–2180, 2016.

[CDS90] Yann Le Cun, John S. Denker, and Sara A. Solla. Optimal Brain Damage. In *Advances in Neural Information Processing Systems (NIPS)*, pages 598–605, 1990.

[CE17] Ryan Cotterell and Jason Eisner. Probabilistic Typology: Deep Generative Models of Vowel Inventories. *Computing Research Repository (CoRR)*, arXiv:1705.01684, 2017.

[CEL20] Y. Choi, M. El-Khamy, and J. Lee. Universal deep neural network compression. *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[CH19] Jack Clark and Gillian K. Hadfield. Regulatory Markets for AI Safety. *Computing Research Repository (CoRR)*, arXiv:2001.00078, 2019.

[CHM+15] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The Loss Surfaces of Multilayer Networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

[CL18] Jaegul Choo and Shixia Liu. Visual Analytics for Explainable Deep Learning. *Computing Research Repository (CoRR)*, arXiv:1804.02527, 2018.

[CLC+19] Hao-Yun Chen, Jhao-Hong Liang, Shih-Chieh Chang, Jia-Yu Pan, Yu-Ting Chen, Wei Wei, and Da-Cheng Juan. Improving Adversarial Robustness via Guided Complement Entropy. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4881–4889, 2019.

[CLG+15] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible Models for Healthcare: Predicting Pneumonia Risk and Hospital 30-day Readmission. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1721–1730, 2015.

[CLLW19] Po-Yi Chen, Alexander H. Liu, Yen-Cheng Liu, and Yu-Chiang F. Wang. Towards Scene Understanding: Unsupervised Monocular Depth Estimation With Semantic-Aware Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2624–2632, 2019.

[CLS+18] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain Adaptive Faster R-CNN for Object Detection in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[CNH+18] Chih-Hong Cheng, Georg Nührenberg, Chung-Hao Huang, Harald Ruess, and Hirotoshi Yasuoka. Towards Dependability Metrics for Neural Networks. In *Proceedings of the ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE)*, pages 43–46, 2018.

[COR+16] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[CPK+17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2017.

[CPMA19] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8001–8008, 2019.

[CPSA17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *Computing Research Repository (CoRR)*, arXiv:1706.05587, 2017.

[CPT01] Alexandra Cristea, Cristea P, and Okamoto T. Neural Network Knowledge Extraction. *Revue Roumaine des ScienceTechnique, Série Électrotechnique et Énergétique*, 2001.

[Csu17a] Gabriela Csurka. A Comprehensive Survey on Domain Adaptation for Visual Applications. In *Domain Adaptation in Computer Vision Applications*, pages 1–35. Springer, 2017.

[Csu17b] Gabriela Csurka. Domain Adaptation for Visual Applications: A Comprehensive Survey. *Advances in Computer Vision and Pattern Recognition*, 2017.

[CT05] Kristin A. Cook and James J. Thomas. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Press, 2005.

[CW17a] Nicholas Carlini and David Wagner. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In *Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISec)*, pages 3–14, 2017.

[CW17b] Nicholas Carlini and David A. Wagner. Towards Evaluating the Robustness of Neural Networks. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[CZM+19] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Strategies from Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019.

[CZSL19] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical data augmentation with no separate search. *Computing Research Repository (CoRR)*, arXiv:1909.13719, 2019.

[CZYS19] Feiyang Cheng, Hong Zhang, Ding Yuan, and Mingui Sun. Leveraging Semantic Segmentation with Learning-based Confidence Measure. *Neurocomputing*, 329:21–31, 2019.

[DAL+18] Guneet Dhillon, Kamyar Azizzadenesheli, Zachary Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Animashree Anandkumar. Stochastic Activation Pruning for Robust Adversarial Defense. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–11, 2018.

[DCG+19] Sai Vikas Desai, Akshay L. Chandra, Wei Guo, Seishi Ninomiya, and Vineeth N. Balasubramanian. An Adaptive Supervision Framework for Active Learning in Object Detection. *Computing Research Repository (CoRR)*, arXiv:1908.02454:1–13, 2019.

[DCO+19] Qi Dou, Cheng Chen, Cheng Ouyang, Hao Chen, and Pheng Ann Heng. *Unsupervised Domain Adaptation of ConvNets for Medical Image Segmentation via Adversarial Learning*, pages 93–115. Springer International Publishing, 2019.

[DdCKG19] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain Generalization via Model-Agnostic Learning of Semantic Features. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6447–6458, 2019.

[DK17] Josip Djolonga and Andreas Krause. Learning Implicit Generative Models Using Differentiable Graph Tests. *Computing Research Repository (CoRR)*, arXiv:1709.01006, 2017.

[DK18] John J. Dudley and Per Ola Kristensson. A Review of User Interface Design for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2), 2018.

[DL13] Andreas Damianou and Neil Lawrence. Deep Gaussian Processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

[DLP+18] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting Adversarial Attacks With Momentum. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9185–9193, 2018.

[DMW+20] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, AK Qin, and Yun Yang. Adversarial Camouflage: Hiding Physical-World Attacks with Natural Styles. *Computing Research Repository (CoRR)*, arXiv:2003.08757, 2020.

[Doe16] Carl Doersch. Tutorial on Variational Autoencoders. *Computing Research Repository (CoRR)*, arXiv:1606.05908, 2016.

[DRC+17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. *Computing Research Repository (CoRR)*, arXiv:1711.03938, 2017.

[DT17] Terrance DeVries and Graham W. Taylor. Dataset Augmentation in Feature Space, 2017.

[DT18] Terrance DeVries and Graham W. Taylor. Learning Confidence for Out-of-Distribution Detection in Neural Networks. *Computing Research Repository (CoRR)*, arXiv:1802.04865, 2018.

[EEF+18] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust Physical-World Attacks on Deep Learning Visual Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1625–1634, 2018.

[EIS+19] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial Robustness As a Prior for Learned Representations. *Computing Research Repository (CoRR)*, arXiv:1906.00945, 2019.

[EMH19a] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Efficient Multi-Objective Neural Architecture Search Via Lamarckian Evolution. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019.

[EMH19b] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural Architecture Search: A Survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

[ERT+17] A. Endert, W. Ribarsky, C. Turkay, B.L. William Wong, I. Nabney, I. Díaz Blanco, and F. Rossi. The State of the Art in Integrating Machine Learning into Visual Analytics. *Computer Graphics Forum*, 36(8):458–486, 2017.

[ETTS19] Logan Engstrom, Brandon Tran, Dimitris Tsipras, and Ludwig Schmidt. A Rotation and a Translation Suffice : Fooling CNNs with Simple Transformations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–21, 2019.

[EW18] Cian Eastwood and Christopher K. I. Williams. A Framework for the Quantitative Evaluation of Disentangled Representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, page 15, 2018.

[FAA18] Emily Fertig, Aryan Arbabi, and Alexander A. Alemi. β-VAEs Can Retain Label Information Even at High Compression. *Computing Research Repository (CoRR)*, arXiv:1812.02682, 2018.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 70, pages 1126–1135. PMLR, 2017.

[FCSG17] Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. Detecting Adversarial Samples from Artifacts. *Computing Research Repository (CoRR)*, arXiv:1703.00410, 2017.

[FF15] Alhussein Fawzi and Pascal Frossard. Manitest: Are Classifiers Really Invariant? *Computing Research Repository (CoRR)*, arXiv:1507.06535, 2015.

[FKH18] Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. In *Proceedings of the*

International Conference on Machine Learning (ICML), volume 80, pages 1436–1445. PMLR, 2018.

[FP78] DF Frey and RA Pimentel. *Principal Component Analysis and Factor Analysis*. John Wiley & Sons, Inc., 1978.

[Fry77] MJ Fryer. A Review of some Non-Parametric Methods of Density Estimation. *IMA Journal of Applied Mathematics*, 20(3):335–354, 1977.

[FSS+16] Mohsen Fayyaz, Mohammad Hajizadeh Saffar, Mohammad Sabokrou, Mahmood Fathy, Reinhard Klette, and Fay Huang. STFCN: Spatio-Temporal FCN for Semantic Video Segmentation. *Computing Research Repository (CoRR)*, arXiv:1608.05971, 2016.

[FV17] Ruth C. Fong and Andrea Vedaldi. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[FXY12] Jiashi Feng, Huan Xu, and Shuicheng Yan. Robust PCA in High-dimension: A Deterministic Approach. *Computing Research Repository (CoRR)*, arXiv:1206.4628, 2012.

[FZ10] Pedro F. Felzenszwalb and Ramin Zabih. Dynamic Programming and Graph Algorithms in Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(4):721–740, 2010.

[GBA+19] Yona Falinie A. Gaus, Neelanjan Bhowmik, Samet Akçay, Paolo M. Guillén-Garcia, Jack W. Barker, and Toby P. Breckon. Evaluation of a Dual Convolutional Neural Network Architecture for Object-Wise Anomaly Detection in Cluttered X-Ray Security Imagery. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[GBR+06] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A Kernel Method for the Two-Sample-Problem. In *Advances in Neural Information Processing Systems (NIPS)*, pages 513–520. MIT Press, 2006.

[GBR07] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic Forecasts, Calibration and Sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.

[GBR+12] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A Kernel Two-Sample Test. *Machine Learning Research*, 13:723–773, 2012.

[GBY+18] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018.

[GEB15] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style. *Computing Research Repository (CoRR)*, arXiv:1508.06576, 2015.

[GEB16] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer using Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[GG84] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 6(6):721–741, 1984.

[GG16] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 48, pages 1050–1059, 2016.

[GHHW20] Christoph Gladisch, Christian Heinzemann, Martin Herrmann, and Matthias Woehrle. Leveraging Combinatorial Testing for Safety-Critical Computer Vision Datasets. In *Proceedings of the Workshop on Safe Artificial Intelligence for Automated Driving (SAIAD) at IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[GHK17] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete Dropout. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3581–3590, 2017.

[GHL+20] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-Guided Representation Learning for Self-Supervised Monocular Depth. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–14, 2020.

[GIG17] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian Active Learning with Image Data. *Proceedings of the International Conference on Machine Learning (ICML)*, 3:1923–1932, 2017.

[Gir15] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–21, 2015.

[GJG17] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic Video CNNs through Representation Warping. In *Proceedings of the IEEE*

International Conference on Computer Vision (ICCV), pages 4453–4462, 2017.

[GJZ+18] Jianmin Guo, Yu Jiang, Yue Zhao, Quan Chen, and Jiaguang Sun. DLFuzz: Differential fuzzing testing of deep learning systems. In *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on Foundations of Software Engineering (ESEC/FSE)*, pages 739–743. ACM, 2018.

[GK20] Tomojit Ghosh and Michael Kirby. Supervised Dimensionality Reduction and Visualization Using Centroid-encoder. *Computing Research Repository (CoRR)*, arXiv:2002.11934, 2020.

[GKZ14] Muhammad Ghifary, W. Bastiaan Kleijn, and Mengjie Zhang. Domain Adaptive Neural Networks for Object Recognition. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence (PRICA) - Trends in Artificial Intelligence*, volume 8862, pages 898–904. Springer, 2014.

[GKZB15] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain Generalization for Object Recognition with Multi-task Autoencoders. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2551–2559, 2015.

[GLYB14] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir D. Bourdev. Compressing Deep Convolutional Networks using Vector Quantization. *Computing Research Repository (CoRR)*, arXiv:1412.6115, 2014.

[Goo16] Ian Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. *Computing Research Repository (CoRR)*, arXiv:1701.00160, 2016.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 70, pages 1321–1330. PMLR, 2017.

[GR19] Puneet Gupta and Esa Rahtu. CIIDefence: Defeating Adversarial Attacks by Fusing Class-Specific Image Inpainting and Image Denoising. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6708–6717, 2019.

[Gra18] Artur Gramacki. *Nonparametric Kernel Density Estimation and its Computational Aspects*. Springer, 2018.

[GRCvdM18] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering Adversarial Images using Input Transformations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–12, 2018.

[GSS15] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–10, 2015.

[GTC+18] Rafael Garcia, Alexandru C. Telea, Bruno Castro da Silva, Jim Tørresen, and João Luiz Dihl Comba. A Task-and-technique Centered Survey on Visual Analytics for Deep Learning Model Engineering. *Computers & Graphics*, 77:30–49, 2018.

[Guo18] Yunhui Guo. A Survey on Methods and Theories of Quantized Neural Networks. *Computing Research Repository (CoRR)*, arXiv:1808.04752, 2018.

[GWY+19] Shupeng Gui, Haotao Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model Compression with Adversarial Robustness: A Unified Optimization Framework. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–16, 2019.

[HAB19] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Hai16] Tameru Hailesilassie. Rule Extraction Algorithm for Deep Neural Networks: A Review. *CoRR*, abs/1610.05267:555555, 2016.

[HAMS20] Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. Meta-learning in Neural Networks: A Survey. *Computing Research Repository (CoRR)*, arXiv:2004.05439, 2020.

[Hao19] Hao-Yun Chen, Pei-Hsin Wang, Chun-Hao Liu, Shih-Chieh Chang, Jia-Yu Pan, Yu-Ting Chen, Wei Wei, Da-Cheng Juan. Complement Objective Training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–10, 2019.

[HBMF20] Atiye Sadat Hashemi, Andreas Bär, Saeed Mozaffari, and Tim Fingscheidt. Transferable Universal Adversarial Perturbations Using Generative Models. *Computing Research Repository (CoRR)*, arXiv:2010.14919, 2020.

[HD19] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–15, 2019.

[HDHH20] Jussi Hanhirova, Anton Debner, Matias Hyyppä, and Vesa Hirvisalo. A Machine Learning Environment for Evaluating Autonomous Driving Software, 2020.

[HDR18] Michael Harradon, Jeff Druce, and Brian Ruttenberg. Causal Learning and Explanation of Deep Neural Networks Via Autoencoded Activations. *Computing Research Repository (CoRR)*, arXiv:1802.00541, 2018.

[HDY+12] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[HG17] Dan Hendrycks and Kevin Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[HGDG17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[Hin09] Geoffrey E Hinton. Deep Belief Networks. *Scholarpedia*, 4(5):5947, 2009.

[HK18] P.S. Sastry Himanshu Kumar. Robust Loss functions for Learning Multi-Class Classifiers. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1–6, 2018.

[HKD+18] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 2234–2240, 2018.

[HKPC18] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *Computing Research Repository (CoRR)*, arXiv:1801.06889, 2018.

[HKV19] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges*. Springer International Publishing, 2019.

[HLP+17] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q Weinberger. Snapshot Ensembles: Train 1, Get m for Free. *Computing Research Repository (CoRR)*, arXiv:1704.00109, 2017.

[HLS+19] Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules. *Computing Research Repository (CoRR)*, arXiv:1905.05393, 2019.

[HMCKBF17] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal Adversarial Perturbations Against Semantic Image Segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2774–2783, 2017.

[HMD16] Song Han, Huizi Mao, and William J. Dally. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[HMD19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep Anomaly Detection with Outlier Exposure. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[HMJ13] Tamir Hazan, Subhransu Maji, and Tommi Jaakkola. On Sampling from the Gibbs Distribution with Random Maximum A-Posteriori Perturbations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1268–1276, 2013.

[HMK+19] Lukas Hoyer, Mauricio Muñoz, Prateek Katiyar, Anna Khoreva, and Volker Fischer. Grid Saliency for Context Explanations of Semantic Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[HMP+17] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[HO00] A. Hyvärinen and E. Oja. Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 13(4-5):411–430, 2000.

[HR90] J. Huber and A. Rüppel. Zuverlässigkeitsschätzung für die Ausgangssymbole von Trellis-Decodern. *Archiv für Elektronik und Übertragung (AEÜ) (in German)*, 44(1):8–21, 1990.

[HRF19] Zhezhi He, Adnan S. Rakin, and Deliang Fan. Parametric Noise Injection: Trainable Randomness to Improve Deep Neural Network Robustness Against Adversarial Attack. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 588–597, 2019.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[HVD15a] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *Computing Research Repository (CoRR)*, arXiv:1503.02531, 2015.

[HVD15b] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the Knowledge in a Neural Network. In *Proceedings of the Deep Learning and Representation Learning Workshop at Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[HZRS15] K. He, X. Zhang, S. Ren, and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9):1904–1916, 2015.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[HZS17] Yihui He, Xiangyu Zhang, and Jian Sun. Channel Pruning for Accelerating Very Deep Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1398–1406, 2017.

[ICG+18] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty Estimates and Multi-Hypotheses Networks for Optical Flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 652–667, 2018.

[Ino18] Hiroshi Inoue. Data Augmentation by Pairing Samples for Images Classification. *Computing Research Repository (CoRR)*, arXiv:1801.02929, 2018.

[IS19] Stephane Canu Ismaïla Seck, Gaëlle Loosli. L1-norm double backpropagation adversarial defense. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 1–6, 2019.

[JC16] Ian T. Jolliffe and Jorge Cadima. Principal Component Analysis: A Review and Recent Developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[JC17] Katarzyna Janocha and Wojciech Marian Czarnecki. On Loss Functions for Deep Neural Networks in Classification. *Schedae Informaticae*, 25(9):49–49, 2017.

[JDCR12] Mahesh Joshi, Mark Dredze, William W. Cohen, and Carolyn Penstein Rosé. Multi-domain Learning: When Do Domains Matter? In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1302–1312, 2012.

[JGST19] Hadi Samer Jomaa, Josif Grabocka, and Lars Schmidt-Thieme. Hyp-RL : Hyperparameter Optimization by Reinforcement Learning. *Computing Research Repository (CoRR)*, arXiv:1906.11527, 2019.

[JKC+18] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2704–2713, 2018.

[JLM+18] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of Localization Confidence for Accurate Object Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–799, 2018.

[JLS+19] Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Hao Chen, Zhenyu Zhong, and Tao Wei. Fooling Detection Alone is not Enough: Adversarial Attack against Multiple Object Tracking. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[JMS96] M. Chris Jones, James S. Marron, and Simon J. Sheather. A Brief Survey of Bandwidth Selection for Density Estimation. *Journal of the American statistical association*, 91(433):401–407, 1996.

[JWCF19] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. ComDefend: An Efficient Image Compression Model to Defend Adversarial Examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6084–6092, 2019.

[KAF+08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. *Visual Analytics: Definition, Process, and Challenges*, pages 154–175. Springer Berlin Heidelberg, 2008.

[KALL17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *Computing Research Repository (CoRR)*, arXiv:1710.10196, 2017.

[KB15] Diederik P. Kingma and Jimmy Ba. ADAM: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[KBFs20] Marvin Klingner, Andreas Bär, and Tim Fingscheidt. Improved Noise and Attack Robustness for Semantic Segmentation by Using Multi-Task Training with Self-Supervised Depth Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1299–1309, 2020.

[KBL+17] Konstantinos Kamnitsas, Christian Baumgartner, Christian Ledig, Virginia Newcombe, Joanna Simpson, Andrew Kane, David Menon, Aditya Nori, Antonio Criminisi, Daniel Rueckert, and Ben Glocker. Unsupervised Domain Adaptation in Brain Lesion Segmentation with Adversarial Networks. In *Proceedings of the International Conference on Information Processing in Medical Imaging (IPMI)*, pages 597–609. Springer International Publishing, 2017.

[KFE18] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate Uncertainties for Deep Learning Using Calibrated Regression. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2801–2809, 2018.

[KG17] Alex Kendall and Yarin Gal. What Uncertainties do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems (NIPS)*, pages 5574–5584, 2017.

[KGB17a] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial Examples in the Physical World. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, pages 1–14, 2017.

[KGB17b] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial Machine Learning at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–17, 2017.

[KGC17] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for Deep Learning: A Taxonomy. *Computing Research Repository (CoRR)*, arXiv:1710.10686v1, 2017.

[KGC18] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018.

[KK11] Philipp Krähenbühl and Vladlen Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *Advances in Neural Information Processing Systems (NIPS)*, pages 109–117, 2011.

[KKB18] Philip Koopman, Aaron Kane, and Jen Black. Credible Autonomy Safety Argumentation. In *Proceedings of the Safety-Critical Systems Symposium (SSS)*, 2018.

[KKMH20] Ilyes Khemakhem, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. *Computing Research Repository (CoRR)*, arXiv:1907.04809, 2020.

[KKSH20] Fabian Küppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate Confidence Calibration for Object Detection. In *Proceedings of the Workshop on Safe Artificial Intelligence for Automated Driving (SAIAD) at IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. Manuscript under review.

[KL19] Zelun Kong and Cong Liu. Generating Adversarial Fragments with Adversarial Networks for Physical-world Implementation. *Computing Research Repository (CoRR)*, arXiv:1907.04449, 2019.

[KLSL19] Chieh Chi Kao, Teng Yok Lee, Pradeep Sen, and Ming Yu Liu. Localization-Aware Active Learning for Object Detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11366 LNCS:506–522, 2019.

[KLX+17] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object Detection in Videos with Tubelet Proposal Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[KM03] Dan Klein and Christopher D Manning. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3–10, 2003.

[KMAM18] Manish Kesarwani, Bhaskar Mukhoty, Vijay Arya, and Sameep Mehta. Model Extraction Warning in MLaaS Paradigm. In *Proceedings of the Annual Computer Security Applications Conference*, pages 371–380. Association for Computing Machinery, 2018.

[KMN+16] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *Computing Research Repository (CoRR)*, arXiv:1609.04836, 2016.

[KMT10] Daniel A. Keim, Florian Mansmann, and Jim Thomas. Visual Analytics: How Much Visualization and How Much Analytics? *ACM SIGKDD Explorations Newsletter*, 11(2):5–8, 2010.

[KNS+18] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabás Póczos, and Eric P. Xing. Neural Architecture Search with Bayesian Optimisation and Optimal Transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2020–2029, 2018.

[Kok17] Iasonas. Kokkinos. Ubernet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-level Vision Using Diverse Datasets and Limited Memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5454–5463, 2017.

[KP20] Abhishek Kumar and Ben Poole. On Implicit Regularization in *beta*-VAEs. *Computing Research Repository (CoRR)*, arXiv:2002.00041, 2020.

[KRBT08] Pushmeet Kohli, Jonathan Rihan, Matthieu Bray, and Philip HS Torr. Simultaneous Segmentation and Pose Estimation of Humans using Dynamic Graph Cuts. *International Journal of Computer Vision*, 79(3):285–298, 2008.

[KRPM+18] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R. Ledsam, Klaus Maier-Hein, S.M. Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A Probabilistic U-Net for Segmentation of Ambiguous Images. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6965–6975, 2018.

[KSFF17] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta Calibration: A Well-Founded and Easily Implemented Improvement on Logistic Calibration for Binary Classifiers. In *Artificial Intelligence and Statistics*, pages 623–631, 2017.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[KT08] Pushmeet Kohli and Philip HS Torr. Measuring Uncertainty in Graph Cut Solutions. *Computer Vision and Image Understanding*, 112(1):30–38, 2008.

[KTMFs20] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[KTP+20] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on Sesame Street! Model Extraction of BERT-based APIs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[KW14] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *Computing Research Repository (CoRR)*, arXiv:1312.6114, 2014.

[KWG+18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2668–2677, 2018.

[KXG17] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic Autoencoder for Zero-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4447–4456, 2017.

[KYL+20] Nikhil Kapoor, Chun Yuan, Jonas Löhdefink, Roland Zimmermann, Serin Varghese, Fabian Hüger, Nico Schmidt, Peter Schlicht, and Tim Fingscheidt. A Self-Supervised Feature Map Augmentation (FMA) Loss and Combined Augmentations Finetuning to Efficiently Improve the Robustness of CNNs. In *Proceedings of the ACM Computer Science in Cars Symposium*, 2020.

[KZG18] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and Visible Adversarial Noise. *Computing Research Repository (CoRR)*, arXiv:1801.02608, 2018.

[LAL+19] Changliu Liu, Tomer Arnon, Christopher Lazarus, Clark Barrett, and Mykel J. Kochenderfer. Algorithms for Verifying Deep Neural Networks. *Computing Research Repository (CoRR)*, arXiv:1903.06758, 2019.

[LBD+89] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 1989.

[LBL+19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. *Computing Research Repository (CoRR)*, arXiv:1811.12359, 2019.

[LBS+19] Jonas Löhdefink, Andreas Bär, Nico M. Schmidt, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. On Low-Bitrate Image Compression for Distributed Automotive Perception: Higher Peak SNR Does Not Mean Better Semantic Segmentation. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 352–359, 2019.

[LCH+06] Yann LeCun, Sumit Chopra, Raia Hadsell, M. Ranzato, and F. Huang. A Tutorial on Energy-Based Learning. *Predicting Structured Data*, 1(0), 2006.

[LCM+17] Junhua Lu, Wei Chen, Yuxin Ma, Junming Ke, Zongzhuang Li, Fan Zhang, and Ross Maciejewski. Recent Progress and Trends in Predictive Visual Analytics. *Frontiers of Computer Science*, 11(2):192–207, 2017.

[LCPB18] Shuangtao Li, Yuanke Chen, Yanlin Peng, and Lin Bai. Learning More Robust Features with Adversarial Training. *Computing Research Repository (CoRR)*, arXiv:1804.07757, 2018.

[LCWJ18] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional Adversarial Domain Adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1640–1650, 2018.

[LCY14] Min Lin, Qiang Chen, and Shuicheng Yan. Network In Network. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[LFK+20] Jonas Löhdefink, Justin Fehrling, Marvin Klingner, Fabian Hüger, Peter Schlicht, Nico M. Schmidt, and Tim Fingscheidt. Self-Supervised Domain Mismatch Estimation for Autonomous Perception. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1359–1368, 2020.

[LGG+17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. In *International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[LGH+17] Yafeng Lu, Rolando Garcia, Brett Hansen, Michael Gleicher, and Ross Maciejewski. The State-of-the-Art in Predictive Visual Analytics. *Computer Graphics Forum*, 36(3):539–562, 2017.

[LGH19] Ji Lin, Chuang Gan, and Song Han. Defensive Quantization: When Efficiency Meets Robustness. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–15, 2019.

[LJD+17] Lisha Li, Kevin G. Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *Machine Learning Research*, 18:185:1–185:52, 2017.

[LJD19] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end Multi-Task Learning with Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019.

[LJL+19] Hong Liu, Rongrong Ji, Jie Li, Baochang Zhang, Yue Gao, Yongjian Wu, and Feiyue Huang. Universal Adversarial Perturbation via Prior Driven Uncertainty Approximation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2941–2949, 2019.

[LK19] Mark Lee and Zico Kolter. On Physical Adversarial Patches for Object Detection. *Computing Research Repository (CoRR)*, arXiv:1906.11897, 2019.

[LKD+17] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning Filters for Efficient ConvNets. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–27, 2017.

[LLL+17] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, and Xiaoou Tang. Deep Learning Markov Random Field for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(8):1814–1828, 2017.

[LLL+19] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature Distillation: DNN-Oriented JPEG Compression Against Adversarial Examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868, 2019.

[LLLS17] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples. *Computing Research Repository (CoRR)*, arXiv:1711.09325, 2017.

[LLLS18] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7167–7177, 2018.

[LLS+17] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning Efficient Convolutional Networks through Network Slimming. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2755–2763, 2017.

[LLS18] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the Reliability of Out-of-distribution Image Detection in Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[LM19] Zhi Gang Liu and Matthew Mattina. Learning Low-precision Neural Networks without Straight-through Estimator (STE). In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3066–3072. ijcai.org, 2019.

[LMW+17] S. Liu, D. Maljovec, B. Wang, P. Bremer, and V. Pascucci. Visualizing High-Dimensional Data: Advances in the Past Decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, 2017.

[LNH14] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-Based Classification for Zero-Shot Visual Object Categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(3):453–465, 2014.

[LO17] David Lopez-Paz and Maxime Oquab. Revisiting Classifier Two-Sample Tests. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2017.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6402–6413, 2017.

[LPWK18] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain Generalization With Adversarial Feature Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5400–5409, 2018.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[LSF19] Timo Lohrenz, Maximilian Strake, and Tim Fingscheidt. On Temporal Context Information for Hybrid BLSTM-Based Phoneme Recognition. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 516–523, 2019.

[LSY19] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable Architecture Search. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019.

[LTG+18] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep Domain Generalization via Conditional Invariant Adversarial Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11219, pages 647–663. Springer, 2018.

[LUAD16] Matthieu Lê, Jan Unkelbach, Nicholas Ayache, and Hervé Delingette. Sampling Image Segmentations for Uncertainty Quantification. *Medical Image Analysis*, 34:42–51, 2016.

[LWB+19] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking Clever Hans Predictors and Assessing what Machines Really Learn. *Nature Communications*, 10(1):1–8, 2019.

[LWL17] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5068–5076, 2017.

[LWLZ17] Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. Towards Better Analysis of Machine Learning Models: A Visual Analytics Perspective. *Visual Informatics*, 1(1):48–56, 2017.

[LWZ09] Ming Li, Yan Wu, and Qiang Zhang. SAR Image Segmentation based on Mixture Context and Wavelet Hidden-Class-Label Markov Random Field. *Computers & Mathematics with Applications*, 57(6):961–969, 2009.

[LYL+18] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. DPATCH: An Adversarial Patch Attack on Object Detectors. *Computing Research Repository (CoRR)*, arXiv:1806.02299, 2018.

[LYP+19] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D. Cubuk. Improving Robustness Without Sacrificing Accuracy with Patch Gaussian Augmentation. *Computing Research Repository (CoRR)*, arXiv:1906.02611, 2019.

[LYSH18] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to Generalize: Meta-Learning for Domain Generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI), the Innovative Applications of Artificial Intelligence (IAAI), and the AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI)*, pages 3490–3497. AAAI Press, 2018.

[LYW+19] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ramkant Nevatia, and Alan Yuille. Every Pixel Counts ++: Joint Learning of Geometry and Motion with 3D Holistic Understanding. *Computing Research Repository (CoRR)*, 2019.

[LZG+19] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a Closer Look at Domain Shift: Category-Level Adversaries for Semantics Consistent Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[LZN+18] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy.

Progressive Neural Architecture Search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[LZWJ17] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep Transfer Learning with Joint Adaptation Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 70, pages 2208–2217. PMLR, 2017.

[LZY+19] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic Training for Domain Generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1446–1455, 2019.

[Mac98] David J.C. MacKay. Introduction to Gaussian Processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166, 1998.

[Mac03] David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[MARC20] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards Recognizing Unseen Categories in Unseen Domains. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[MBB+15] Grégoire Montavon, Sebastian Bach, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition. *Computing Research Repository (CoRR)*, arXiv:1512.02479, 2015.

[MBRB18] Ari S. Morcos, David G. T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. On the Importance of Single Directions for Generalization. *Computing Research Repository (CoRR)*, arXiv:1803.06959, 2018.

[MBS13] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain Generalization via Invariant Feature Representation. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 28, pages 10–18. JMLR.org, 2013.

[MD18] Wang Mei and Weihong Deng. Deep Visual Domain Adaptation: A Survey. *Neurocomputing*, 2018.

[MDFF16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

[MDFFF17] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal Adversarial Perturbations. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1765–1773, 2017.

[MDFUF19] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via Curvature Regularization, and Vice Versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9078–9086, 2019.

[MDS+18] Manu Mathew, Kumar Desappan, Pramod Kumar Swami, Soyeb Nagori, and Biju Moothedath Gopinath. Embedded Low-Power Deep Learning with TIDL. Technical report, Texas Instruments Technical Report, 2018.

[MDSN17] Manu Mathew, Kumar Desappan, Pramod Kumar Swami, and Soyeb Nagori. Sparse, Quantized, Full Frame CNN for Low Power Embedded Devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 328–336, 2017.

[MG18] Andrey Malinin and Mark Gales. Predictive Uncertainty Estimation via Prior Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7047–7058, 2018.

[MGB17] Konda Reddy Mopuri, Utsav Garg, and R. Venkatesh Babu. Fast Feature Fool: A Data Independent Approach to Universal Adversarial Perturbations. In *Proceedings of the British Machine Vision Conference (BMCV)*, pages 1–12, 2017.

[MGR+18] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the Limits of Weakly Supervised Pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–23, 2018.

[MH20a] Toshihiko Matsuura and Tatsuya Harada. Domain Generalization using a Mixture of Multiple Latent Domains. In *Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI), the Innovative Applications of Artificial Intelligence (IAAI), and the AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI)*, pages 11749–11756. AAAI Press, 2020.

[MH20b] Alexander Meinke and Matthias Hein. Towards Neural Networks that Provably Know when They don't Know. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[Mig17] Szymon Migacz. 8-bit Inference with TensorRT. *GPU Technology Conference*, 2017.

[MKH+19] Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial Defense by Restricting the Hidden Space of Deep Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3385–3394, 2019.

[ML11] Riccardo Miotto and Gert Lanckriet. A Generative Context Model for Semantic Music Annotation and Retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 20(4):1096–1108, 2011.

[MLG+18] Radek Mackowiak, Philip Lenz, Omair Ghori, Ferran Diego, Oliver Lange, and Carsten Rother. CEREALS - Cost-Effective REgion-based Active Learning for Semantic Segmentation. *Proceedings of the British Machine Vision Conference (BMCV)*, 2018.

[MMR+16] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. *Computing Research Repository (CoRR)*, arXiv:1602.05629, 2016.

[MMS+18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–28, 2018.

[MRG19] Kira Maag, Matthias Rottmann, and Hanno Gottschalk. Time-Dynamic Estimates of the Reliability of Deep Semantic Segmentation Networks. *CoRR*, abs/1911.05075, 2019.

[MS17] Qiqi Yan Mukund Sundararajan, Ankur Taly. Axiomatic Attribution for Deep Networks. *Computing Research Repository (CoRR)*, arXiv:1703.01365, 2017.

[MSGH16] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-Stitch Networks for Multi-task Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[MSJ+16] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian J. Goodfellow, and Brendan Frey. Adversarial Autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, 2016.

[MTK+17] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning Convolutional Neural Networks for Resource Efficient Inference. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–27, 2017.

[MTRA+12] Jose Moreno-Torres, Troy Raeder, Rocío Alaiz, Nitesh Chawla, and Francisco Herrera. A Unifying View on Dataset Shift in Classification. *Pattern Recognition*, 45:521–530, 2012.

[NC16] Mahdo Naeini and Gregory Cooper. Binary Classifier Calibration Using an Ensemble of Near Isotonic Regression Models. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 360–369, 2016.

[NCH15] Mahdo Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2901–2907, 2015.

[NK16] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple Black-Box Adversarial Perturbations for Deep Networks. *Computing Research Repository (CoRR)*, arXiv:1612.06299, 2016.

[NK17] Nina Narodytska and Shiva Kasiviswanathan. Simple Black-Box Adversarial Attacks on Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1310–1318, 2017.

[NS18] David Nilsson and Cristian Sminchisescu. Semantic Video Segmentation by Gated Recurrent Flow Propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6819–6828, 2018.

[NSO19] Vu Nguyen, Sebastian Schulze, and Michael A. Osborne. Bayesian Optimization for Iterative Learning. *Computing Research Repository (CoRR)*, arXiv:1909.09593, 2019.

[NYC15] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015.

[OAFS18] Seong Joon Oh, Max Augustin, Mario Fritz, and Bernt Schiele. Towards Reverse-Engineering Black-Box Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[OE18] Achraf Oussidi and Azeddine Elhassouny. Deep Generative Models: Survey. In *Proceedings of the International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–8, 2018.

[OOAG19] Augustus Odena, Catherine Olsson, David Andersen, and Ian Goodfellow. TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4901–4911, 2019.

[ORF20] Philipp Oberdiek, Matthias Rottmann, and Gernot A. Fink. Detection and Retrieval of Out-of-Distribution Objects in Semantic Segmentation, 2020.

[ORG18] Philipp Oberdiek, Matthias Rottmann, and Hanno Gottschalk. Classification Uncertainty of Deep Neural Networks Based on Gradient Information. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11081 LNAI:113–125, 2018.

[Osb16] Ian Osband. Risk versus Uncertainty in Deep Learning: Bayes, Bootstrap and the Dangers of Dropout. In *Proceedings of the Workshop on Bayesian Deep Learning at Advances in Neural Information Processing Systems (NIPS)*, volume 192, 2016.

[OSM20] Hirono Okamoto, Masahiro Suzuki, and Yutaka Matsuo. Out-of-Distribution Detection Using Layerwise Uncertainty in Deep Neural Networks, 2020.

[PCYJ17] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. DeepXplore. *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, abs/1705.06640:1–18, 2017.

[PFC+19] Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free Epistemic Uncertainty Estimation Using Approximated Variance Propagation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2931–2940, 2019.

[PGZ+18] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient Neural Architecture Search Via Parameter Sharing. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 80, pages 4092–4101. PMLR, 2018.

[PKGB18] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative Adversarial Perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4422–4431, 2018.

[PL20] Ashis Pati and Alexander Lerch. Attribute-based Regularization of VAE Latent Spaces. *Computing Research Repository (CoRR)*, arXiv:2004.05485, 2020.

[Pla99] John Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in Large Margin Classifiers*, pages 61–74, 1999.

[PLZ19] Bowen Zhou Pengcheng Li, Jinfeng Yi and Lijun Zhang. Improving the Robustness of Deep Neural Networks via Adversarial Training with Triplet Loss . In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1–7, 2019.

[PMG+17] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACSS)*, pages 1–12, 2017.

[PTC+17] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing Neural Networks by Penalizing Confident Output Distributions. *CoRR*, 2017.

[PXD+20] Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking Softmax Cross-Entropy Loss for Adversarial Robustness. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–19, 2020.

[PY11] George Papandreou and Alan L Yuille. Perturb-and-Map Random Fields: Using Discrete Optimization to Learn and Sample from Energy Models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 193–200, 2011.

[RA17] Mostafa Rahmani and George Atia. Coherence Pursuit: Fast, Simple, and Robust Principal Component Analysis. *IEEE Transactions on Signal Processing*, 65(23):6260–6275, 2017.

[RAHL19] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized Evolution for Image Classifier Architecture Search. In *Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI), the Innovative Applications of Artificial Intelligence (IAAI), and the AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI)*, pages 4780–4789. AAAI Press, 2019.

[Ras03] Carl Edward Rasmussen. Gaussian Processes in Machine Learning. In *Summer School on Machine Learning*, pages 63–71, 2003.

[RBAS17] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Learning what to Share between Loosely Related Tasks. *Computing Research Repository (CoRR)*, arXiv:1705.08142, 2017.

[RBB18] Hippolyt Ritter, Aleksandar Botev, and David Barber. A Scalable Laplace Approximation for Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[RBV18] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient Parametrization of Multi-Domain Deep Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8119–8127, 2018.

[RCH+18] Matthias Rottmann, Pascal Colling, Thomas-Paul Hack, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities. *Computing Research Repository (CoRR)*, arXiv:1811.00648, 2018.

[RDG18] Soumali Roychowdhury, Michelangelo Diligenti, and Marco Gori. Image Classification Using Deep Learning and Prior Knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence Workshops*, volume WS-18, pages 336–343. AAAI Press, 2018.

[RDGF15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2015.

[Ree93] Russell Reed. Pruning Algorithms-A Survey. *IEEE Transactions on Neural Networks*, 04(5):740–747, 1993.

[RKG18] Matthias Rottmann, Karsten Kahl, and Hanno Gottschalk. Deep Bayesian Active Semi-Supervised Learning. *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 158–164, 2018.

[RL19] Mostafa Rahmani and Ping Li. Outlier Detection and Data Clustering Via Innovation Search. *Computing Research Repository (CoRR)*, arXiv:1912.12988, 2019.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *Computing Research Repository (CoRR)*, arXiv:1511.06434, 2015.

[RMDC19] Swalpa Kumar Roy, Suvojit Manna, Shiv Ram Dubey, and Bidyut B. Chaudhuri. LiSHT: Non-Parametric Linearly Scaled Hyperbolic Tangent Activation Function for Neural Networks. *CoRR*, abs/1901.05894, 2019.

[RRMH17] Reza Rasti, Hossein Rabbani, Alireza Mehridehnavi, and Fedra Hajizadeh. Macular OCT Classification using a Multi-Scale Convolutional Neural Network Ensemble. *IEEE Transactions on Medical Imaging*, 37(4):1024–1034, 2017.

[RS19] Matthias Rottmann and Marius Schubert. Uncertainty Measures and Prediction Quality Rating for the Semantic Segmentation of Nested Multi Resolution Street Scene Images. *Computing Research Repository (CoRR)*, arXiv:1904.04516, 2019.

[RSFD16] Erik Rodner, Marcel Simon, Robert B. Fisher, and Joachim Denzler. Fine-grained Recognition in the Noisy Wild: Sensitivity Analysis of Convolutional Neural Networks Approaches. In *Proceedings of the British Machine Vision Conference (BMCV)*, pages 1–13, 2016.

[RSFM19] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. Barrage of Random Transforms for Adversarially Robust Defense. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6528–6537, 2019.

[RSG16] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144. ACM, 2016.

[RSS18] Johannes Rabold, Michael Siebers, and Ute Schmid. Explaining Black-box Classifiers with ILP – Empowering LIME with Aleph to Approximate Non-linear Decisions with Relational Rules. In *Proceedings of the International Conference on Inductive Logic Programming (ILP)*, pages 105–117. Springer International Publishing, 2018.

[RTG+19] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, 2019.

[RUN18] Soumya Roy, Asim Unmesh, and Vinay P. Namboodiri. Deep Active Learning for Object Detection. *Proceedings of the British Machine Vision Conference (BMCV)*, pages 1–12, 2018.

[RZL18] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for Activation Functions. In *International Conference on Learning Representations (ICLR) - Workshops*, 2018.

[SAMG19] Timo Sämann, Karl Amende, Stefan Milz, and Horst-Michael Groß. Robust Semantic Video Segmentation through Confidence-based Feature Map Warping. In *Proceedings of the ACM Computer Science in Cars Symposium (CSCS)*, pages 1–9, 2019.

[SBMC17] Sayantan Sarkar, Ankan Bansal, Upal Mahbub, and Rama Chellappa. UPSET and ANGRI: Breaking High Performance Image Classifiers. *Computing Research Repository (CoRR)*, arXiv:1707.01159, 2017.

[SBS12] Pablo Sprechmann, Alex M. Bronstein, and Guillermo Sapiro. Learning Robust Low-Rank Representations. *Computing Research Repository (CoRR)*, arXiv:1209.6393, 2012.

[Sco15] David W Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 2015.

[SCW+15] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems (NIPS)*, pages 802–810, 2015.

[SDBR14] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. *Computing Research Repository (CoRR)*, arXiv:1412.6806, 2014.

[SDF+20] Maximilian Strake, Bruno Defraene, Kristoff Fluyt, Wouter Tirry, and Tim Fingscheidt. Fully Convolutional Recurrent Networks for Speech Enhancement. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6674–6678, 2020.

[SDKF19] Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. Distribution calibration for regression. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97, pages 5897–5906. PMLR, 2019.

[SDV+16] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. *Computing Research Repository (CoRR)*, arXiv:1610.02391, 2016.

[SDW+19] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu. Single-Path NAS: Designing Hardware-Efficient ConvNets in Less Than 4 Hours. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 481–497, 2019.

[Sea20] Sean Saito and Sujoy Roy . Effects of Loss Functions And Target Representations on Adversarial Robustness . In *Proceedings of the Conference on Machine Learning and Systems (MLSys) Workshops*, pages 1–10, 2020.

[Set10] Burr Settles. Active Learning Literature Survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2010.

[SFH17] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic Routing between Capsules. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3856–3866, 2017.

[SGS15] Rupesh K. Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[SGSK17] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not Just a Black Box: Learning Important Features Through Propagating Activation Differences. *Computing Research Repository (CoRR)*, arXiv:1605.01713, 2017.

[SGSS07] Alexander J. Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert Space Embedding for Distributions. In *Proceedings of the International Conference Algorithmic Learning Theory (ALT)*, volume 4754, pages 13–31. Springer, 2007.

[SH09] Ruslan Salakhutdinov and Geoffrey Hinton. Deep Boltzmann Machines. In *Artificial Intelligence and Statistics*, pages 448–455, 2009.

[She04] Simon J. Sheather. Density Estimation. *Statistical Science*, pages 588–597, 2004.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Machine Learning Research*, 15(1):1929–1958, 2014.

[SHK+19] Youcheng Sun, Xiaowei Huang, Daniel Kroening, James Sharp, Matthew Hill, and Rob Ashmore. Structural Test Coverage Criteria for Deep Neural Networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s):1–23, 2019.

[Shn96] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[Sil86] Bernard W Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26. CRC press, 1986.

[SIVA17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[SJG+20] Pierre Stock, Armand Joulin, Rémi Gribonval, Benjamin Graham, and Hervé Jégou. And The Bit Goes Down: Revisiting The Qunatization Of Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[SK19] Connor Shorten and Taghi M. Khoshgoftaar. A Survey on Image Data Augmentation for Deep Learning. *Big Data*, 2019.

[SKF18] Hao Song, Meelis Kull, and Peter Flach. Non-Parametric Calibration of Probabilistic Regression. *CoRR*, 2018.

[SLY15] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3483–3491, 2015.

[SMK20] Nathan Somavarapu, Chih-Yao Ma, and Zsolt Kira. Frustratingly Simple Domain Generalization via Image Stylization. *Computing Research Repository (CoRR)*, arXiv:2006.11207, 2020.

[SMM+17] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *Computing Research Repository (CoRR)*, arXiv:1701.06538, 2017.

[SOF+19] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you Trust your Model's Uncertainty? Evaluating Predictive Uncertainty under Dataset Shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13969–13980, 2019.

[SRHD16] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork Convnets for Video Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 852–868, 2016.

[SRK20] Peter Sorrenson, Carsten Rother, and Ullrich Köthe. Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (GIN). *Computing Research Repository (CoRR)*, arXiv:2001.04872, 2020.

[SS20a] Gesina Schwalbe and Martin Schels. A Survey on Methods for the Safety Assurance of Machine Learning Based Systems. In *Proceedings of the European Congress Embedded Real Time Software and Systems (ERTS)*, 2020.

[SS20b] Gesina Schwalbe and Martin Schels. Concept Enforcement and Modularization As Methods for the ISO 26262 Safety Argumentation of Neural Networks. In *Proceedings of the European Congress Embedded Real Time Software and Systems (ERTS)*, 2020.

[SSH19] Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han. Learning for Single-Shot Confidence Calibration in Deep Neural Networks Through Stochastic Inferences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[SSH20] Timo Sämann, Peter Schlicht, and Fabian Hüger. Strategy to Increase the Safety of a DNN-Based Perception for HAD Systems. *Computing Research Repository (CoRR)*, arXiv:2002.08935, 2020.

[SSSS17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership Inference Attacks against Machine Learning Models. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[SSW+17] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In *Proceedings of the International Conference on Information Processing in Medical Imaging (IPMI)*, pages 146–157, 2017.

[SSZ+17] Dominik Sacha, Michael Sedlmair, Leishi Zhang, John A. Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. What You See Is What You Can Change: Human-centered Machine Learning by Interactive Visualization. *Neurocomputing*, 268:164–175, 2017. Advances in artificial neural networks, machine learning and computational intelligence.

[SSZ19] Martin Svatos, Gustav Sourek, and Filip Zelezny. Revisiting Neural-Symbolic Learning Cycle. In *Proceedings of the International Workshop on Neural-Symbolic Learning and Reasoning*, 2019.

[STK+17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. SmoothGrad: Removing Noise by Adding Noise. *Computing Research Repository (CoRR)*, arXiv:1706.03825, 2017.

[Sto09] Amos J. Storkey. When Training and Test Sets are Different: Characterising Learning Transfer. In *Dataset Shift in Machine Learning*, pages 3–28. MIT Press, 2009.

[STP17] Brian Summa, Julien Tierny, and Valerio Pascucci. Visualizing the Uncertainty of Graph-based 2D Segmentation with Min-path Stability. *Computer Graphics Forum*, 36(3):133–143, 2017.

[SU15] Alexander G. Schwing and Raquel Urtasun. Fully Connected Deep Structured Networks. *Computing Research Repository (CoRR)*, arXiv:1503.02351, 2015.

[SVS19] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: VisualisingImage Classification Models and Saliency Maps. *Computing Research Repository (CoRR)*, arXiv:1312.6034, 2014.

[SWR+18] Youcheng Sun, Min Wu, Wenjie Ruan, Xiaowei Huang, Marta Kwiatkowska, and Daniel Kroening. Concolic Testing for Deep Neural Networks. In *Proceedings of the ACM/IEEE International Conference Automated Software Engineering (ASE)*, pages 109–119. ACM, 2018.

[SWY+15] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[SYPS19] Veit Sandfort, Ke Yan, Perry Pickhardt, and Ronald Summers. Data Augmentation using Generative Adversarial Networks (CycleGAN) to Improve Generalizability in CT Segmentation Tasks. *Scientific Reports*, 9, 2019.

[SYZ+20] Xin Sun, Zhenning Yang, Chi Zhang, Guohao Peng, and Keck-Voon Ling. Conditional Gaussian Distribution Learning for Open Set Recognition. *Computing Research Repository (CoRR)*, arXiv:2003.08823, 2020.

[SZ15] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[SZS+14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing Properties of Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–10, 2014.

[SZS+17] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):241–250, 2017.

[SZT17] Ravid Shwartz-Ziv and Naftali Tishby. Opening the Black Box of Deep Neural Networks Via Information. *Computing Research Repository (CoRR)*, 2017.

[SZYP19] Harshvardhan Sikka, Weishun Zhong, Jun Yin, and Cengiz Pehlevan. A Closer Look at Disentangling in *beta*-VAE. *Computing Research Repository (CoRR)*, arXiv:1912.05127, 2019.

[T+02] Sebastian Thrun et al. Robotic Mapping: A Survey. *Exploring Artificial Intelligence in the New Millennium*, 1(1-35):1, 2002.

[TA12] Daniel Tarlow and Ryan P Adams. Revisiting Uncertainty in Graph Cut Solutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2440–2447, 2012.

[TAGD98] A. B. Tickle, R. Andrews, M. Golea, and J. Diederich. The Truth Will Come to Light: Directions and Challenges in Extracting the Knowledge Embedded within Trained Artificial Neural Networks. *IEEE Tansactions on Neural Networks*, 9(6):1057–1068, 1998.

[Tay06] Brian J Taylor. *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*. Springer Science & Business Media, 2006.

[TBF+15] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features with 3d Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[TBGS19] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein Auto-Encoders. *Computing Research Repository (CoRR)*, arXiv:1711.01558, 2019.

[TCBZ19] Rajkumar Theagarajan, Ming Chen, Bir Bhanu, and Jing Zhang. ShieldNets: Defending Against Adversarial Attacks Using Probabilistic Adversarial Robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6988–6986, 2019.

[Thr95] Sebastian Thrun. Extracting Rules from Artificial Neural Networks with Distributed Representations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 505–512. MIT Press, 1995.

[TKTH18] Lucas Theis, Iryna Korshunova, Alykhan Tejani, and Ferenc Huszár. Faster Gaze Prediction with Dense Networks and Fisher Pruning. *Computing Research Repository (CoRR)*, arXiv:1801.05787, 2018.

[TP18] Jun Zhu Tianyu Pang, Chao Du. Max-Mahalanobis Linear Discriminant Analysis Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1–13, 2018.

[TPJR18] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars. In *Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE)*, pages 303–314. Association for Computing Machinery, 2018.

[TVRG19] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 0–0, 2019.

[TWZ+18] Marvin Teichmann, Michael Weber, Marius Zollner, Roberto Cipolla, and Raquel Urtasun. MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[TZJ+16] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing Machine Learning Models via Prediction APIs. In *Proceedings of the USENIX Security Symposium*, pages 601–618. USENIX Association, 2016.

[UEKH19] Hristina Uzunova, Jan Ehrhardt, Timo Kepp, and Heinz Handels. Interpretable Explanations of Black Box Classifiers Applied on Medical Images by Meaningful Perturbations using Variational Autoencoders. In *Medical Imaging 2019: Image Processing*, volume 10949, 2019.

[UMY+19] Stefan Uhlich, Lukas Mauch, Kazuki Yoshiyama, Fabien Cardinaux, Javier Alonso García, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Differentiable Quantization of Deep Neural Networks. *Computing Research Repository (CoRR)*, arXiv:1905.11452, 2019.

[VBB+20] Serin Varghese, Yasin Bayzidi, Andreas Bär, Nikhil Kapoor, Sounak Lahiri, Jan David Schneider, Nico Schmidt, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. Unsupervised Temporal Consistency Metric for Video Segmentation in Highly-Automated Driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1369–1378, 2020.

[VJB+19] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Perez. ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[vLSB20] Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are Disentangled Representations Helpful for Abstract Visual Reasoning? *Computing Research Repository (CoRR)*, arXiv:1905.12506, 2020.

[WB98] Christopher KI Williams and David Barber. Bayesian Classification with Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(12):1342–1351, 1998.

[Wer78] Wolfgang Wertz. *Statistical Density Estimation: A Survey*. Vandenhoeck & Ruprecht, 1978.

[Wes04] Thijs Henk-Willem Westerveld. *Using Generative Probabilistic Models for Multimedia Retrieval*. PhD thesis, University of Twente, 2004.

[WFZ19] Michael Weber, Michael Fürst, and J. Marius Zöllner. Automated Focal Loss for Image based Object Detection. *CoRR*, abs/1904.09048, 2019.

[WGH19] Matthias Woehrle, Christoph Gladisch, and Christian Heinzemann. Open Questions in Testing of Learned Computer Vision Functions for Automated Driving. In *Computer Safety, Reliability, and Security (SAFECOMP)*, pages 333–345. Springer International Publishing, 2019.

[WGSM16] Sebastien C. Wong, Adam Gatt, Victor Stamatescu, and Mark D. McDonnell. Understanding Data Augmentation for Classification: When to Warp? *Computing Research Repository (CoRR)*, arXiv:1609.08764, 2016.

[WHLX19] Haohan Wang, Zexue He, Zachary C. Lipton, and Eric P. Xing. Learning Robust Representations by Projecting Superficial Statistics Out. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019.

[WJZ+20] Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation. *Computing Research Repository (CoRR)*, arXiv:2004.09602, 2020.

[WKP13] Chaohui Wang, Nikos Komodakis, and Nikos Paragios. Markov Random Field Modeling, Inference & Learning in Computer Vision & Image Understanding: A Survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013.

[WLDG19] Zuxuan Wu, Ser-Nam Lim, Larry Davis, and Tom Goldstein. Making an Invisibility Cloak: Real World Adversarial Attacks on Object Detectors. *Computing Research Repository (CoRR)*, arXiv:1910.14667, 2019.

[WR96] Christopher KI Williams and Carl Edward Rasmussen. Gaussian Processes for Regression. In *Advances in Neural Information Processing Systems (NIPS)*, pages 514–520, 1996.

[WSC+19] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep High-Resolution Representation Learning for Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–17, 2019.

[WSRA20] Oliver Willers, Sebastian Sudholt, Shervin Raafatnia, and Stephanie Abrecht. Safety Concerns and Mitigation Approaches Regarding the Use of Deep Learning in Safety-Critical Perception Tasks. *Computing Research Repository (CoRR)*, arXiv:2001.08001, 2020.

[WT11] Max Welling and Yee Whye Teh. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 681–688, 2011.

[WTPFW19] Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, and Andrew Gordon Wilson. Practical Multi-Fidelity Bayesian Optimization for Hyperparameter Tuning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, page 284. AUAI Press, 2019.

[WYKN20] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Computing Surveys*, 53(3):63:1–63:34, 2020.

[WZM+16] Xu-Meng Wang, Tian-Ye Zhang, Yu-Xin Ma, Jing Xia, and Wei Chen. A Survey of Visual Analytic Pipelines. *Journal of Computer Science and Technology*, 31(4):787, 2016.

[XCKW19] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_{DMI} : A Novel Information-theoretic Loss Function for Training Deep Nets Robust to Label Noise. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–16, 2019.

[XCS10] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA Via Outlier Pursuit. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2496–2504, 2010.

[XGD+17] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[XHM+11] Xiaoyuan Xie, Joshua W. K. Ho, Christian Murphy, Gail Kaiser, Baowen Xu, and Tsong Yueh Chen. Testing and Validating Machine Learning Classifiers by Metamorphic Testing. *Systems and Software*, 84(4):544–558, 2011.

[XLH+20] Hongteng Xu, Dixin Luo, Ricardo Henao, Svati Shah, and Lawrence Carin. Learning Autoencoders with Relational Regularization. *Computing Research Repository (CoRR)*, arXiv:2002.02913, 2020.

[XLZ+18] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified Perceptual Parsing for Scene Understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11209, pages 432–448. Springer International Publishing, 2018.

[XW19] Yijun Xiao and William Yang Wang. Disentangled Representation Learning with Wasserstein Total Correlation. *Computing Research Repository (CoRR)*, arXiv:1912.12818, 2019.

[XWvdM+19] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature Denoising for Improving Adversarial Robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 501–507, 2019.

[XZL+19] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial T-Shirt! Evading Person Detectors in a Physical World. *Computing Research Repository (CoRR)*, arXiv:1910.11099, 2019.

[XZZ+19] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving Transferability of Adversarial Examples With Input Diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2730–2739, 2019.

[YDL+17] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the Class Weight Bias: Weighted Maximum Mean Discrepancy for Unsupervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 945–954, 2017.

[YLLW18] Jianbo Ye, Xin Lu, Zhe Lin, and James Wang. Rethinking the Smaller-Norm-Less-Informative Assumption in Channel Pruning of Convolution Layers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–11, 2018.

[YLW+20] Hongliang Yan, Zhetao Li, Qilong Wang, Peihua Li, Yong Xu, and Wangmeng Zuo. Weighted and Class-Specific Maximum Mean Discrepancy for Unsupervised Domain Adaptation. *IEEE Transactions on Multimedia*, 22(9):2420–2433, 2020.

[YLX+19] Shaokai Ye, Xue Lin, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, and Yetang Wang. Adversarial Robustness vs. Model Compression, or Both? In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 111–120, 2019.

[Yuv11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, Andrew Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *Advances in Neural Information Processing Systems (NIPS) Workshops*, pages 1–18, 2011.

[YZS+18] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. SegStereo: Exploiting Semantic Information for Disparity Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–651, 2018.

[ZCAW17] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. *Computing Research Repository (CoRR)*, arXiv:1702.04595, 2017.

[ZCG+19] Barret Zoph, Ekin D. Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning Data Augmentation Strategies for Object Detection. *Computing Research Repository (CoRR)*, arXiv:1906.11172, 2019.

[ZCY+19] Xiang Zhang, Xiaocong Chen, Lina Yao, Chang Ge, and Manqing Dong. Deep Neural Network Hyperparameter Optimization with Orthogonal Array Tuning. In *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, volume 1142, pages 287–295. Springer, 2019.

[ZE01] Bianca Zadrozny and Charles Elkan. Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 609–616, 2001.

[ZE02] Bianca Zadrozny and Charles Elkan. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 694–699, 2002.

[ZF13] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. *Computing Research Repository (CoRR)*, arXiv:1311.2901, 2013.

[ZGY+19] Feng Zhu, Ruihao Gong, Fengwei Yu, Xianglong Liu, Yanfei Wang, Zhelong Li, Xiuqi Yang, and Junjie Yan. Towards Unified INT8 Training for Convolutional Neural Network. *Computing Research Repository (CoRR)*, arXiv:1912.12607, 2019.

[Zha12] Bailing Zhang. Reliable Classification of Vehicle Types Based on Cascade Classifier Ensembles. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):322–332, 2012.

[ZHD+19] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Christopher De Sa, and Zhiru Zhang. Improving Neural Network Quantization without Retraining using Outlier Channel Splitting. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97, pages 7543–7552. PMLR, 2019.

[ZHML19] Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine Learning Testing: Survey, Landscapes and Horizons, 2019.

[ZJRP+15] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional Random Fields as Recurrent Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537, 2015.

[ZLMJ16] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. DeepRED – rule extraction from deep neural networks. In *Proceedings of the International Conference on Discovery Science (DS)*, pages 457–473. Springer International Publishing, 2016.

[ZLZ+18] Husheng Zhou, Wei Li, Yuankun Zhu, Yuqun Zhang, Bei Yu, Lingming Zhang, and Cong Liu. Deepbillboard: Systematic Physical-World Testing of Autonomous Driving Systems. *Computing Research Repository (CoRR)*, arXiv:1812.10812, 2018.

[ZP17] Chong Zhou and Randy C. Paffenroth. Anomaly Detection with Robust Deep Autoencoders. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 665–674, 2017.

[ZPIE17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *Computing Research Repository (CoRR)*, arXiv:1703.10593, 2017.

[ZQF+18] Huiyuan Zhuo, Xuelin Qian, Yanwei Fu, Heng Yang, and Xiangyang Xue. SCSP: Spectral Clustering Filter Pruning with Soft Self-adaption Manners. *Computing Research Repository (CoRR)*, arXiv:1806.05320, 2018.

[ZSLG16] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving The Robustness of Deep Neural Networks via Stability Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4480–4488, 2016.

[ZSR+19] Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–14, 2019.

[ZVSL18] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning Transferable Architectures for Scalable Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8697–8710, 2018.

[ZWN+16] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients. *Computing Research Repository (CoRR)*, arXiv:1606.06160, 2016.

[ZZK+17] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random Erasing Data Augmentation. *Computing Research Repository (CoRR)*, arXiv:1708.04896, 2017.

[ZZW+18] Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, Joao P. Costeira, and Geoffrey J. Gordon. Adversarial Multiple Source Domain Adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8559–8570, 2018.