Improving the Generalizability of Deep Learning Models for Polyp Segmentation by Optimizing for Consistency

Birk Sebastian Frostelid Torpmann-Hagen

Thesis submitted for the degree of
Master in Robotics and Intelligent Systems
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2022

Improving the Generalizability of Deep Learning Models for Polyp Segmentation by Optimizing for Consistency

Birk Sebastian Frostelid Torpmann-Hagen

Improving the Generalizability of Deep Learning Models for Polyp Segmentation by Optimizing for Consistency

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

Contents

List of Figures

List of Tables

Preface

Chapter 1

Introduction

Colorectal cancer is one of the leading causes of cancer related deaths, causing approximately 900 thousand deaths worldwide per year (cite). Early detection thereof is as a consequence of significant importance. Polyps are often an early warning-sign of developing tumour, and early detection thereof can as a result significantly reduce fatality rates. Polyps are, however, often missed during colonoscopies, owing to the significant variability in the shapes and sizes of polyps, as well as the high degrees of similarity to surrounding tissue. Automatic segmentation of polyps via deep learning has the potential to significantly increase the likelihood of early detection and thus effective treatment.

Though there has been a wealth of work dedicated to developing such systems, with promising results, recent work in the field has highlighted that deep neural networks (DNNs) readily fail to maintain performance when deployed outside of lab-conditions. This is known as generalization failure, and has been shown to be ubiquitous across practically every application of deep learning. The deep learning community is still in the early stages of understanding exactly how and why such generalization failure is so ubiquitous. Consqeuently, developing methods and frameworks to combat generalisation remains an open problem.

This thesis attempts to address this problem by synthesizing and systematizing recent work in generalizability, generalizable methods, and recent attempts at inducing generalizability in polyp segmentation as presented in the EndoCV2021 challenge. A novel approach to increasing generalizability, based on recent work in the field, is also presented. The approach, named PLACEHOLDER ALGORITHM NAME, works by employing specific augmentation strategies to produce a so-called model of natural variation, intended to encapsulate the variability one might expect across different datasets and hence assist in inducing invariances in the model. This is acheived through the use of a specifically tailored loss, referred to as consistency loss, which punishes inconsistent predictions across the augmented and unaugmented folds irrespective of the correctness of the predictions. This endows the pipeline with the ability to more readily infer causally viable inductive biases by explicitly forcing the model to be robust to any combination of the aforementioned

transformations.

(...)

Chapter 2

Background

2.1 Colorectal Polyps, Medical Imaging, and Deep Learning

Polyps are small growths found in and around the inner lining of the large intestine. These polyps, also referred to as adenomas, can in time develop into cancerous tumours, or carcinomas, in a process known as the adenoma-carcinoma sequence [19]. Though the majority of polyps do not undergo this process, identifying polyps nonetheless constitutes an important step towards preventing colorectal cancer. Indeed, resection of these polyps has been shown to reduce the incidence of colorectal cancer by a significant margin [27].

Though colorectal cancer remains as one of the leading causes of cancer-related death worldwide (source), mortality rates have nonetheless declined in large part to increased use of screening colonoscopy, which in turn has facilitated the use of more preemptive treatment. Polyps are, however, by nature somewhat difficult to detect and are routinely missed by clinicians, with miss rates ranging upwards of 27% for diminutive (<2.5mm) polyps [14, 21].

Reducing this miss rate has the potential to further reduce incidence rates. As a result, there has been a significant body of work dedicated to developing systems and techniques to aid in optimizing and effectivizing the screening procedure. One such example, referred to as chromoendoscopy, has been shown to reduce miss rates by (...) merely by employing the use of specific dyes prior to the colonoscopy. Similarly, the use of narrow-band imaging techniques, wherein light of specific wavelengths specifically designed to highlight the textural differences between the polyps and the surrounding tissue, has been shown to reduce miss rates by (...)

These systems do, however, require more equipment, training and expertise to effectively employ. Thus, automatic polyp segmentation using deep learning and convolutional neural networks (CNNs) has been identified as a possible diminutive detection method. This requires minimal training time on the part of the clinician, no additional equipment, and has been show to significantly increase detection rates when deployed in a clinical setting [4].

This has spurred on a large body of research dedicated to improving on the performance and expanding the capabilities of deep-learning based systems for polyp detection and segmentation. Several challenges been also held, namely the Endotect challenge [15], EndoCV2020 [2], EndoCV2021 [1], and more.

There are, however, still several hurdles to overcome; recent research has shown that even state of the art deep-learning pipelines are prone to generalization failure when deployed in practical settings, particularly when exposed to distributional shifts such as changes in demographics, imaging equipment, noise, and more despite exhibiting high performance on hold-out sets [5, 7, 11, 29]. As a result, the EndoCV2021 challange emloyed training data from several centers, with the data from one of the centers being hidden and used as generalization test data. The results from this challange demomstrated the pervasiveness of generalization failure, with every submitted model exhibiting significant performance reductions when evaluated on their hidden dataset (cite summary here).

Naturally, automatic segmentation systems are rendered practically useless should they fail to perform sufficiently outside of the very carefully controlled conditions upon which they are trained. Consequently, for any such system to have any practical merit, it has to have the capacity to infer causally reasonable patterns in the data that generalize well to other hospitals, demographics, imaging equipment, resolutions, and so on.

2.2 Generalization failure in broader contexts

2.2.1 Generalization failure in Medical Imaging

Generalization failure is not, of course, unique to the gastrointestinal domain. Indeed, though medical imaging has in recent years proven to be one of the most promising applications of artificial intelligence and deep learning, having the capacity to significantly improve both the accuracy and efficiency of detection, diagnosis, and treatment of a wide variety of diseases [25], they are nonetheless highly prone to generalization failure. In addition to the already limited capabilities of deep neural networks to generalise, medical domains are subject to a number of other exacerbating factors that make generalization all the more diffcult. Training data is often scarce, the pathologies that constitute the classification targets are unevenly distributed and often exhibit high degrees of within-class variability. Moreover, due to the sheer scope of the data involved, there are inevitably a significant number of confounding variables both during training and in deployment.

For instance, a deep-learning based classifier which successfully detected pneumonia in X-ray scans across a number of hospitals with striking accuracy was determined to be basing its predictions not on any lesions or otherwise pathologically relevant features in the images, but rather on a hospital-specific metal token that was on every image, which it used in conjunction with learning the prevalence rate of pneumonia for the hos-

pitals from which the data was collected. As a result, when deployed on data from hospitals that it had not seen during training, the system failed to generalize [29]. In another study, it was shown that a classifier intended to detect diabetic retinopathy exhibited significant variability in performance depending on the type of camera used. The same study also showed that the same type of performance variability could be found when detecting skin-conditions across demographics with differing skin tones. [7]. Finally, a model trained to detect and diagnose melanomas was shown to in large part be basing its predictions on whether or not it could detect any pre-surgical markings, used by doctors to assist in surgery preparation, in the vinicity of the lesion[28].

2.2.2 Generalization failure in other domains

Naturally, non-medical domains are in no way immune to generalization failure. In fact, one could easily argue that the vast majority of deep-learning pipelines fail to generalize altogether, and instead merely infer some set of inductive biases that, although perhaps causally incorrect, perform sufficiently well for general use. It has for instance been shown that CNNs trained on imagenet, one of the largest and most diverse datasets in the domain of computer vision, are heavily biased towards textural features[10]. Naturally, this is not necessarily causally accurate; a cat is not a cat because it has cat-like fur; nor is an elephant an elephant only because it has skin of an elephant. By manually increasing shape bias, it has been shown that the performance of such CNNs improves both in robustness to perturbations and iid accuracy.

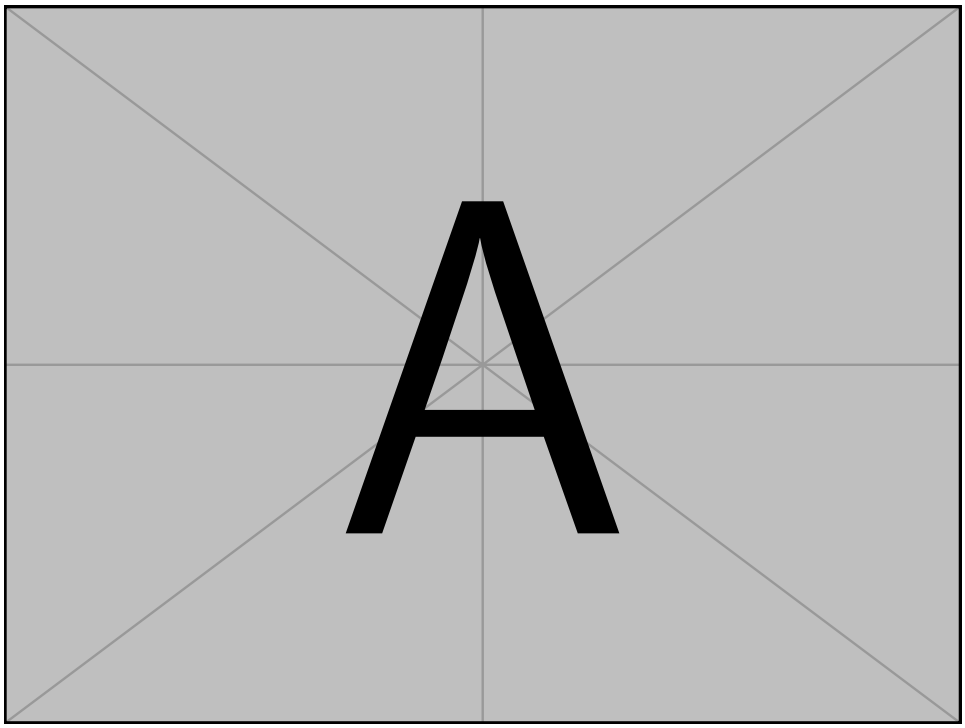

Figure 2.1:

Another characteristic of deep learning that supports this argument is the effectiveness of adversarial attacks [17], which specifically target weaknesses in the inductive biases within DNNs through any number of means in an attempt to induce high rates of incorrect, yet highly confident predictions. Gradient-based adversarial attacks, for instance, use the gradients of the model to break even the most sophisticated and well-trained pipelines merely by adding some carefully crafted, yet visually imperceptible noise to the inputs [6]. Even without access to the gradients, there exists a multitude of so-called black-box attacks that only use output samples to generate similarly effective attacks (cite). Finally, it has been shown that adding minor visual distractions to objects, for example adding bits of tape or graffiti to stop signs, dramatically increases misclassification rates [9].

Even benign, but nonetheless confounding perturbations also have the potential to induce failure. It has for instance been shown that sophisticated natural language processing models can and readily do fail if one adds peripheral information to the input. (Example, citation)

2.3 Generalisability Theory

Exactly why and how DNNs seem to so persistently fail to generalize is a topic of ongoing research, and the available literature seems to suggest that the problem is multifaceted. This section is an attempt to summarize and distill the findings and analysis performed in the field. It will cover

the theoretical basis of generalization and why one might expect DNNs to generalize, discuss the key characteristics of generalization failure and their origins, and finally introduce a probabilistic perspective of generalization.

2.3.1 Generalization through Empirical Risk Minimization

Naturally, deep learning would not have experienced as much of a revolution in the last decade or so if there was not some semblance of an expectation that their striking performance was generalisable and performant also outside the idealized settings typically involved in research. The theoretical basis that informs this belief in (most) modern deep learning pipelines is the idea of so-called empirical risk minimization, wherein it is assumed that the dataset upon which the model is trained is a representative sample of the distribution of all possible samples in the relevant domain. In other words, it assumes that the dataset is independently and identically distributed (iid) to the domain distribution. To better understand this assumption, it is beneficial to consider the it from first principles:

At the most fundamental level, the goal of machine learning is to learn a mapping between two spaces of objects X and Y. This mapping, namely the function $f : X \to Y$, maps some input object $x \in X$, an image for example, to a corresponding and application-relevant output object $y \in Y$, for instance a segmentation mask or a class probabilities. It is worth noting, however, that f is not as much a function in the mathematical sense as much as it is an abstraction of whatever ground-truth relationship that the deep learning system is intended to capture, and consequently cannot typically be modelled explicitly. Instead, machine learning systems aim to find a representation of this mapping automatically by leveraging a training set $\{x_i, y_i\}_{0...n}$ to find a sufficiently performant approximation of f. This is referred to as supervised learning, and the resulting approximation found using the training set is denoted by $h : X \to \hat{Y}$, and typically referred to as a hypothesis.

To find such an approximation, we assume that there exists a joint probability distribution over X and Y, namely $P(x, y)$, and that the training data $\{x_i, y_i\}_{0...n}$ is drawn from this probability distribution such that the resulting sample distribution is independent and identically distributed to $P(x, y)$. This is the so-called iid assumption. By modelling the mapping as a joint probability distribution, one can model uncertainty in the predictions by expressing the output as a conditional probability $P(y|x)$. In conjunction with a loss-function $L(h(x), y)$ which measures the discrepancy between the hypothesis and the ground truth, these assumptions allows us to quantify the expected performance of a given hypothesis:

$$R(h) = \boldsymbol{E}[L(h(x), y)] = \int L(h(x), y) dP(x, y) \tag{2.1}$$

Using this framework, one can then find an iid-optimal hypothesis, often called a predictor, by finding the predictor h^* among a fixed class of functions (defined by network architecture) $\mathcal{H}$ that minimizes risk:

$$h^* = \underset{h \in \mathcal{H}}{\arg\min}\, R(h) \tag{2.2}$$

Since $P(x, y)$ is not known, however, one cannot compute $R(h)$ explicitly. Instead, the expected risk has to be estimated empirically, i.e by finding the arithmetic average of the risk associated with each prediction by the hypothesis over the training set:

$$R_{emp}(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), y_i) \tag{2.3}$$

This risk can in turn be minimized with respect to the hypothesis class. This is called empirical risk minimization (ERM):

$$\hat{h} = \arg\min_{h \in \mathcal{H}} R_{emp}(h) \tag{2.4}$$

To reiterate, the central idea with this approach to machine learning is that the training data can be considered a finite iid sampling of the underlying distribution. As such, by the central limit theorem, the hold-out performance of the computed hypothesis will approach iid-optimal performance given a sufficient amount of training data and some sufficiently capable and regularized training procedure. This should in theory allow deep learning systems to be able to generalize, since the empirical risk in theory can approximate the true risk arbitrarily well given sufficient training data support.

2.3.2 Understanding Generalisation

As the analysis in 2.2 shows, ERM nonetheless readily fails to generate generalizable predictors with respect to out-of-distribution data). Understanding exactly why this is the case is a subject of ongoing study, and cannot be uniquely attributed to any one property of the machine learning pipeline. In broad strokes, the literature attributes generalization failure to two key properties of modern deep learning pipeline, namely that DNNs readily learn non-robust features, and that DNNs are underspecified by the training data. Naturally, there is some degree of overlap between these two views, and both phenomena induce similar behaviour. In an attempt to systematize the To fully understand the nuances that distinguish the respective arguments, it is useful to first consider the assumptions upon which ERM is based, namely that: TODO: FIX

1. f exists in $\mathcal{H}$
2. R_{emp} is sufficiently sampled
3. $\{x_i, y_i\}$ is an IID sampling of $P(x, y)$. This is the aforementioned iid assumption.
4. $\hat{h}$ is unique in $\mathcal{H}$
5. The optimizer consistently finds $\hat{h}$ (given that it exists and is unique)

As the following sections will show, violations of any one of these assumptions can and typically will result in generalization failure.

2.3.3 Underfitting, Overfitting and Regularization

Violations of assumptions 1 and 2 correspond to well known and fairly well understood forms of generalization failure, namely underfitting and overfitting. One can however argue that these factors can be all but discounted as plausible explanations for the pervasiveness of generalization failure.

Underfitting occurs when the model simply lacks the complexity required to encapsulate the patterns necessary to form generalizable predictions. To give a simple example - consider the problem of trying to fit a linear model to the following dataset:

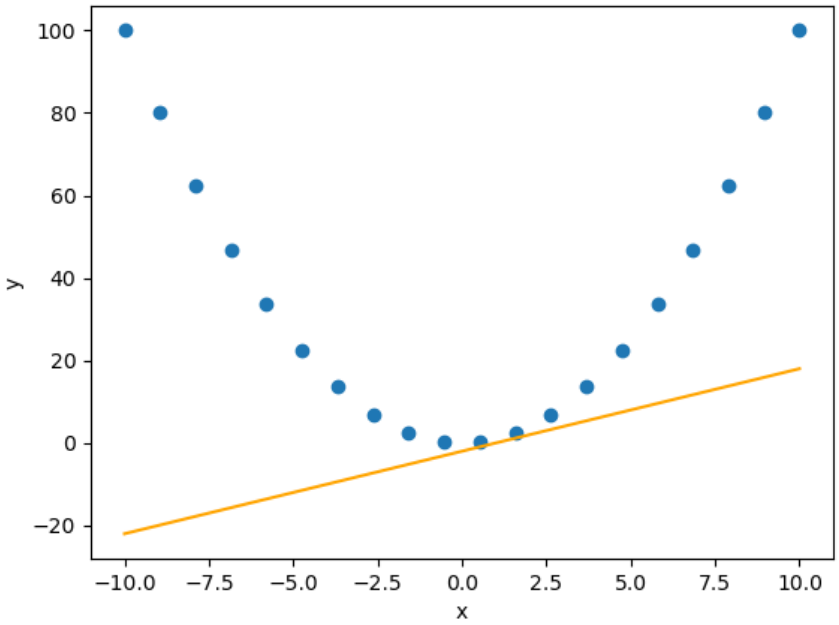

Figure 2.2: A linear model cannot fit polynomial data

Obviously, no amount of optimization of the parameters in the linear model can ever result in a sufficient description of the underlying data and the function it follows, namely $y = x^2$.

It is commonly accepted that DNNs are not particularly susceptible to underfitting. Modern DNNs, as it turns out, can after all be shown to have infinite effective capacity for practical purposes. It can for instance be shown that even a 2-layer feedforward neural network is capable of fitting noise to random labels with 100% accuracy [30]. Assuming this ability translates to a capacity to generalize (which may not necessarily be the case, as will be discussed in chapter 6), it is fairly reasonable to expect that the hypothesis space of the highly complex models used today contains a generalizable predictor.

This is also evidenced by the fact that practically all deep learning

pipelines take considerable steps towards avoiding the exact opposite of underfitting - overfitting. Overfitting occurs when the model learns patterns that by conventional wisdom are too complex to be likely to actually describe the process(es) that generate them. Once again, to give a somewhat simple example:

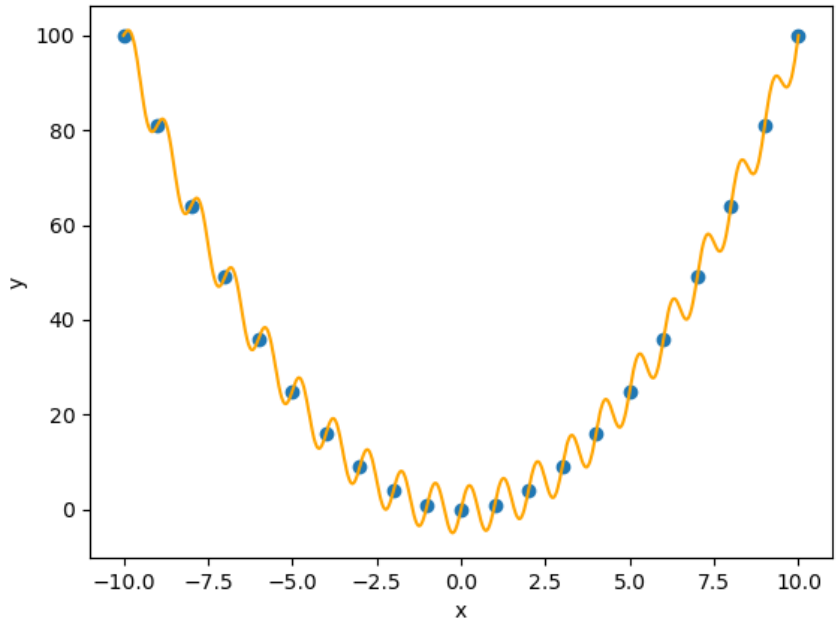

Figure 2.3: A model with excessive capacity interpolates unneccesarily complex patterns

The underlying function is once again $y = x^2$. The sinusoidal pattern in the interpolated function is obviously erroneous, but this is impossible to determine with complete certainty given only a finite sampling of the underlying function. More generally, every finite sampling of any function can be interpolated as any one of an uncountably infinite set of functions. To prove this, consider the task of finding a function that fits the series $(1, 4, 9)$. Though ones first instinct would be to assume the pattern is described by the function $y = x^2$, and that the next number in the sequence thus is 16, the next number can really be anything at all. Using Newton interpolation, or for that matter any arbitrary interpolation scheme, one can easily find a polynomial that fits the sequence $(1, 4, 9, n, n \in \mathbb{R}$. Since the set of real numbers is uncountably infinite, it follows that the space of functions that fits this new sequence also is uncountably infinite, depending on n. Thus, there is an uncountably infinite number of functions that describe $1, 4, 9$ as well. This of course does not only apply to extrapolating the next element in a sequence, but also any interpolating between consecutive elements. TODO clarify Naturally, this also applies to neural networks. Though instead of discrete sequences, it is probability

distributions that are being modelled. Just like how extrapolation and interpolation between points is ill-defined for sequences, it is ill-defined for probability distributions. Consequently, it is necessary to introduce a number of assumptions and constraints to the problem. (CLARIFY)

Obviously, the minimum level of generalization one should achieve is generalization to iid data. To this end, it is necessary to incorporate an iid evaluation method, such as keeping hold-out sets, cross validation, etc. This is of course ubiquitous in modern deep learning. Moreover, it is necessary to make some assumptions regarding the simplicity - or rather, complexity - of the resulting predictor. Certain neurons should for instance not dominate the predictions, the weights should have reasonable values, etc. This is what motivates so-called regularization, whereby different methods - for instance batch normalization, drop-out, L2 loss, weight decay, etc - are used to bias the search towards areas in the loss landscape that are more credible from a perspective of model complexity. This is of course only necessary because assumptions 2 and 4 do not hold. Given a perfect (or at least very well sampled) representation of the risk and consequently the loss-landscape and a perfect optimizer, the model would not overfit in any meaningful way.

Regularization has of course for this reason been extensively studied, and for most purposes the existing regularization techniques suffice just fine for IID data, and typically yield highly performant predictors so long as it is not subjected to any form of distributional shift. Consequently, Overfitting, though naturally still a factor that needs to be accounted for when designing deep learning pipelines, is not at fault for the generalization failure that is exhibited in modern deep learning systems.

2.3.4 Structural Misalignment and dataset bias

Generalization failure is often attributed to structural misalignment between the predictor as generated by ERM and the causal structure which it ideally should encode [3, 11, 17, 24]. Generally, this misalignment occurs as a result of the predictor learning spurious or otherwise causally unrepresentative features that nonetheless perform well within the training distribution. This if of course made evident as soon as the predictor is exposed to any form of distributional shift, at which point it will (typically) fail to generalize. These distributional shifts can range in magnitude, from changes in imaging modalities, common corruptions such as noise or blurs [12] or spatial transforms [8] to practically imperceptible perturbations, typically exemplified by adversarial attacks [6]. ERM does not and cannot guarantee invariance to distributional shifts, as it assumes that the training data is IID to $R(h)$. This is not, however, necessarily as much of a flaw with ERM inasmuch as it is a flaw in the reasoning behind our expectations.

To illustrate, consider the rather pertinent example of training a model exclusively on either white-light or narrow-band endoscopy. Assume that there are two datasets, each containing samples depicting identical scenes, with the only difference being that dataset A employs white-light

endoscopy, whereas dataset B employs narrow-band endoscopy. Ideally, a model trained on either dataset should generate predictors that can generalize to the other, and though one may be optimistic and hope this is the case, this is in no way guaranteed. The causal structure behind the decisions - i.e what exactly makes a polyp a polyp - is never considered at any point in the training process. Instead, the models will simply try to leverage whatever predictive patterns can be found the training data. The model trained on narrow-band images may for instance principally consider the textural characteristic of the polyps, which narrow-band endoscopy enhances. Conversely, the model trained on white-light, lacking access to these textural characteristics, may instead consider more color- or shape-based features. Naturally, if this narrow-band-texture-biased model is deployed in white-light endoscopy, it is not likely to succeed since its principal discriminative features no longer are particularly useful. Similarly, the color-biased model would fail when deployed in narrow-band endoscopy since the colours it once used to distinguish polyps are no longer predictive in narrow-band images.

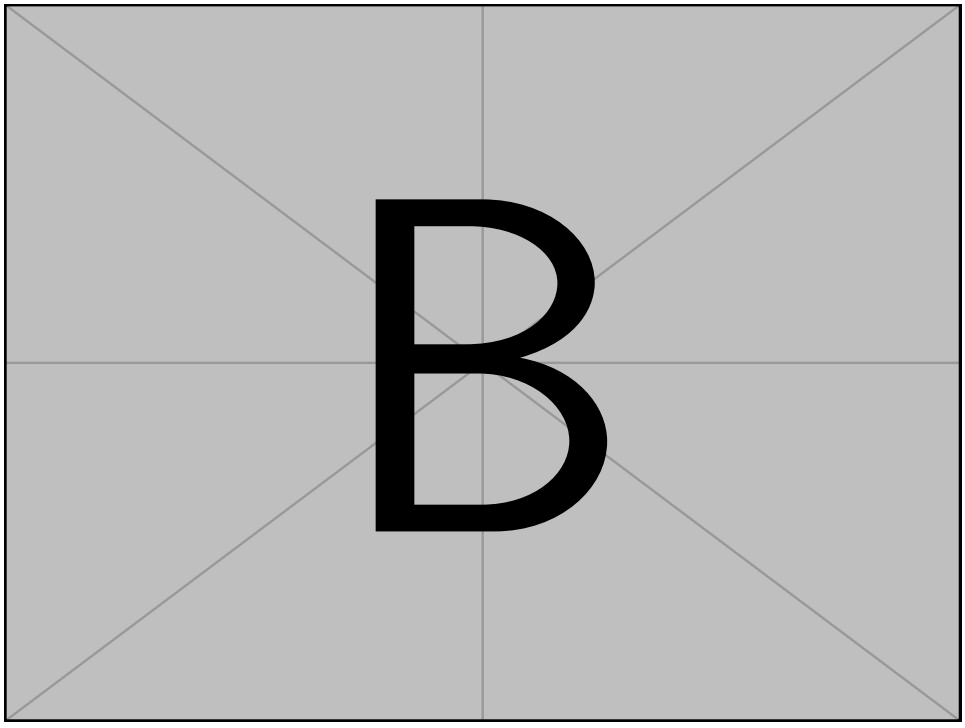

Figure 2.4:

Though the features each model learns are not particularly representative of the broader context of what makes a polyp a polyp, they make sense when considered from the perspective of either of the two hypothetical modalities. When considering only narrow-band imaging, it makes some sense to heavily weigh the texture of the polyps. When considering only white-light imaging, it makes some sense to heavily weight the shape and colour of the polyps. Though humans are capable of appreciating broader

context and subconsciously know that certain features are ancilliary rather than causal (and perhaps more importantly: know the strengths and weaknesses of each modality), DNNs lack the inductive biases needed to take this into account. Once again, DNNs merely leverage the first and best predictive patterns found during the training process, and cannot be expected to optimize for specific invariances on their own, irrespective of how self-evident these invariances may be for humans. This predilection towards dataset-specific features is aptly referred to as dataset bias.

Shortcut Learning

In the aforementioned example, tough the features each predictor learns is not robust to dataset shift, they nevertheless have causal explanations. The causal structure that they correspond to is of course not dataset-agnostic, and as a result flawed in their own way, but the patterns the respective models leverage to interpret the data are not particularly irrational. As it turns out, however, DNNs are unlikely to learn such causally viable features in the first place. In other words, the predictors would not necessarily learn to consider texture in narrow-band images - it could learn any arbitrary pattern so long as it is predictive. Moreover, if such interpretable distributional shifts were the principal cause of generalization failure, generalizability could be induced by explicitly modelling the effects such shifts induce and taking this into account in the pipeline. In the aforementioned example, one could for instance train some model to map from one lighting environment to the other. Though this would imbue the model with an inherent invariance to the choice of lighting, it is nonetheless not given that the resulting model will be perfectly generalizable.

Consequently, though these detectable forms of distributional shifts also hold some importance when designing generalizable models, a more pervasive and substantially more significant issue is the fact that many of the distributional shifts encountered in clinical settings are not necessarily considered significant or for that matter at all perceptible to a human observer. A human would for instance not be significantly affected by slightly noisy, blurry, rotated, or compressed images, nor would they in all likelihood notice these perturbations. DNNs, on the other hand, have been shown to be highly sensitive to these and several other forms of minor perturbations [12, 13, 16, 26]. Moreover, a human would likely not pick up on the subtle phenotypic cues that may exist in the colon during endoscopy, whereas a DNN may leverage some of these cues to inform their decisions.

It is important to note, however, that despite how these two forms of distributional shift may at surface level appear as completely seperate classes of problems, they can both be traced to the same phenomena - namely that DNNs do not leverage any form of causal logic to inform their decisions and, as mentioned previously, simply exploit any sufficiently predictive pattern they may observe in the data. This phenomenon has been shown to be pervasive across all manner of domains, from natural language processing and computer vision to reinforcement learning and algorithmic decision making. This is referred to as shortcut learning [11]

or the Clever Hans effect [18]. Shortcut learning and the brittle features it corresponds to have also been identified as one of the key properties that explains the effectiveness and pervasiveness of adversarial attacks [17]. Naturally, a generalizable predictor should be robust to such minor perturbations, as the model should not in the first place be learning features that get perturbed to any significant degree by adding such high-frequency, low-amplitude noise. Adversarial attacks simply leverage the high degrees of sensitivity inherent to shortcut features, and construct perturbations according the direction in the search space that corresponds to the principal component of this sensitivity [20].

2.3.5 Underspecification

Closely related to shortcut learning is underspecification [7]. A machine learning pipeline can be considered underspecified when it can return any number of risk-equivalent predictors when evaluated on an iid holdout set, dependent only on the random variables used within the training procedure - i.e dropout, seed initialization, and so on. Even with identical hyperparameters, a given training procedure can return any number of predictors each having learned different patterns. One predictor may have learned one shortcut, another may have learned a different shortcut, and one may have fully learned the actual causal relationships it is intended to. With ERM, and in particular with iid-oriented evaluation procedures, these are all erroneously considered equivalent.

Note that this does not however presuppose anything about the relative occurrence rates of generalizable and non-generalizable predictors. It may not necessarily even be the case that the pipeline can return a generalizable predictor at all. Only that there exists a multitude of risk-equivalent predictors in the search space the optimizer typically explores. Nor does it presuppose anything about the distribution of predictors, only that there is indeed a distribution.

This is evidenced by the fact that generalizability can vary greatly depending on the choice of random seed used during training. In the foundational paper on underspecification in deep learning, for instance, it was highligted that certain classification pipelines can produce predictors that vary in ood accuracy by up to 10% [7]. This is a function of the robustness of the learned features and how likely the pipeine is to return the corresponding predictors.

2.3.6 A Bayesian perspective of generalization

With this in mind, one can start considering the statistical properties of a pipeline. In particular, one can leverage

2.4 Generating generalizable predictors

In light of the growing interest in generalization failure, there has been several attempts at increasing generalizability through modifications to the

typical ERM pipeline. This section will survey some of these methodss, and put them in context with the theory as outlined above.

2.4.1 Adversarial Training

2.4.2 Adversarial attacks and defences

2.4.3 Improved Risk estimation

Data augmentation

Distributional modelling

2.4.4 Bayesian Learning

2.5 Related work on Generalizable Polyp Segmentation

The EndoCV2021 challenge focused primarily on addressing methods to increase the robustness and generalizability. The approaches utilized by the submissions can by and large be assigned one of the following categories:

- Generalisation through regularization
- Generalisation through ensembles
- Generalisation through feature strengthening

...

2.6 Putting it all together

...c

Chapter 3

Methods and Implementation

3.1 PLACEHOLDER ALGORITHM NAME

Summarizing the key points made in chapter 2, generalization is in large part a function of the causal robustness of the features it learns. The problem then boils down to the following questions:

1. How can behaviour consistent with the causal structure of a problem be rewarded?

2. How can the pertinent inductive biases that underpin this causal structure be expressed?

3. How can we optimize for causally consistent inductive biases?

(Justifications here)

3.1.1 Consistency Loss

In order to bias the pipeline towards inferring causally reasonable inductive biases, the model needs to be able to learn to be robust to distributional shifts that are known to do not affect the causal structure of the problem. These types of distributional shift, henceforth referred to as perturbations, should not affect the predictions of the model beyond what one would expect from the nature of the perturbation. I.e, if an image is rotated, the only change one should expect in the output segmentation is a corresponding rotation. If an image is exposed to some form of distortion, the segmentation mask should be distorted accordingly. If an image is exposed to low-amplitude additive noise, it should not really be affected at all, and so on. This property will be referred to as the consistency of the model. Consistency should then be maximized in order for the model to learn causally robust features -or more accurately *not* learn causally spurious features. Expressed as a minimization problem, which of course is necessary for gradient descent, this corresponds to minimizing inconsistency. Thus, a loss function that can describe inconsistent behaviour is necessary. Qualitatively, This loss function needs to be able to numerically express the discrepancy between the expected change in the segmentation and the actual

change in the segmentation when subjected to some perturbation. Numerically, this can be expressed as follows:

Let $Y := \{y, \hat{y} := f(x)\}$ be the set consisting of the segmentation labels (masks) and predictions for the unperturbed samples. Let $\epsilon(\cdot)$ be some perturbation function. Then, let $A := \{a := \epsilon(y), \hat{a} := f(\epsilon(x))\}$ be the set consisting of segmentation predictions and masks when the input is subjected to this perturbation. Consistency can then be expressed as follows:

$$L_c = \frac{1}{\sum\{y \cup a\}} \sum\{y \ominus \hat{y} \ominus a \ominus \hat{a}\} \tag{3.1}$$

Where $\ominus$ denotes the symmetric difference. This corresponds to counting the number of pixels that change after the input is subjected to a perturbation - $\hat{a} \ominus \hat{y}$, but discounting those we expect to change, $a \ominus y$. The attentive reader may have noticed that this loss is minimized not only if the predictions are both correct and consistent with one another, but also if the predictions are both incorrect, so long as they are consistent with the expected change:

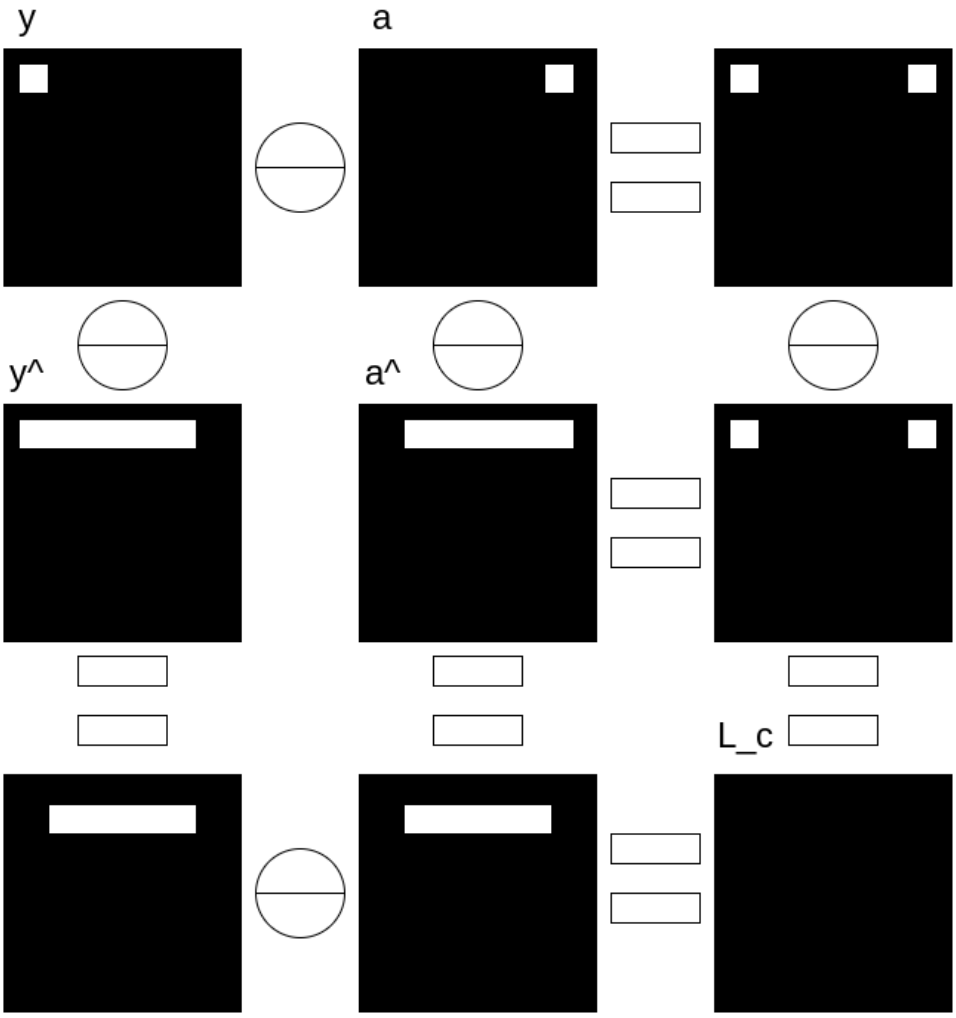

Figure 3.1: Visualisation of consistency loss sets, where white is a positive prediction. Note that loss is zero regardless of prediction correctness so long as it changes in the expected manner.

The reasoning behind this is that consistent behaviour should be rewarded even if the model has not quite learned how to perform to an adequate standard. To illustrate, consider once more the example from chapter 2 generalisation from narrow-band to white-light datasets and vice versa, and assume that the perturbation function simply maps between the respective lighting modalities. In this case, the loss will reward the model if it predicts identical segmentations regardless of lighting conditions. Even if these predictions are incorrect, the model will nevertheless be trying to leverage features which are invariant to lighting, and consequently be causally stronger than a pipeline wherein the model is permitted to be leverage lighting-dependent features.

Note, however, that the loss does not presuppose what transformation has occurred. In Figure 3.1, for instance, the change induced by the perturbation may correspond to simply moving the polyp in the image (and replacing the empty space with a believable background), or it

may correspond to a rotation by 90 degrees. How this should affect the segmentations is up to interpreation - one can argue that a rotation should rotate the incorrect predictions as well, or one can argue that it should only rotate the correct component of the prediction. For simplicity, consistency loss adheres to the latter argument, though it should be noted that designing loss functions that account for the former case is also an interesting direction to pursue.

Moreover, using this loss in isolation is not really practical. For one, the model will have no way of knowing what the actual intent behind it is, and moreover the model will most likely learn the simplest possible interpretation of consistency and simply predict the same segmentation every time, with the only difference being whatever it learns constitutes expected change. This constitutes fairly broad local minima, and there would naturally be a significant number of risk-equivalent predictors, which of course in accordance with the analysis in 2 constitutes generalisation failure on its own. Thus, it has to be combined with a task-specific loss, which for the polyp-segmentation task could be Dice-loss, Jaccard-Loss, binary cross entropy, etc.

Naturally, jointly optimizing for these two often conflicting objectives - overall task performance vs consistency - is not as straight forward as it may seem. The naive approach would be to simply add the task-loss and the naked consistency:

$$L = L_{task} + L_c \tag{3.2}$$

This, however, leads to unstable behaviour and often inhibits convergence, since the consistency term quickly starts gaining precedence if the segmentation task is difficult (see appendix). Consequently, adaptive weighing is required. To avoid getting stuck in broad local minima early, the training should in the early stages be biased towards acheiving semi-decent segmentation performance. Later on, when segmentation performance is starting to become reasonably high, the pipeline should shift towards trying to optimize for consistency. This can be acheived by weighing each term according to a desired performance metric, for instance intersection over union (IoU):

$$L = (1 - IoU) \times L_{task} + IoU \times L_c \tag{3.3}$$

If Jaccard loss is used, this is also equivalent to:

$$L = {L_{jac}}^2 + (1 - L_{jac}) \times L_c \tag{3.4}$$

3.1.2 Model of natural variation

In order to account for any natural variation one may expect to find in deployment, it is necessary to construct a model which can parameterize the variability that is encountered. This model of natual variation, or MNV, can then be leveraged in conjunction with consistency loss to facilitate the learning of features that are robust to the types of variation the MNV defines. Naturally, there is no way of knowing the full extent of all the types

of variability one may find in the wild, but it may nonetheless be sufficient to model some subset thereof. This, naturally, requires some knowledge of the domain from which the dataset is collected. Similarly to how adding rotational augmentations is a bad idea for classification of hand-written numbers, certain transformations may or may not be suitable for use within a MNV.

In the case of polyp-segmentation, it is clear that it is necessary to account for variability in for instance lighting, image-resolution, polyp-size, polyp-shape, polyp-location, camera-quality, color-shifts, blurs, optical distortions, and affine transformations. Thus, a model is required that can (more or less) parametrize this variability. Broadly speaking, these transformations can be categorized as follows:

- Pixel-wise variability, which affect only the image, i.e color-shifts, brightness shifts, contrast-shifts, lighting, blurs etc

- Geometric variability, which affect both the image and the segmentation mask by some parametrizable quantity, i.e affine transforms and distortions

- Manifold variablity, which affects both the image and the segmentation mask depending on a learned model of the distribution, i.e the size, shape and location of polyps

Pixel-wise variability and geometric variability can be modeled fairly trivially through the use of the same transformations typically used in conventional data-augmentation. Manifold-variability, however, is somewhat more difficult. Similar to how [22] and [23] employ cross-dataset style-transfer to account for more implicit distributional shift, it is necessary to find some way to model the distributional properties of the data, and then apply perturbations using the resulting model. Since both the size, shape, and position of polyps can be expected to vary, a model that can change all these factors is necessary. To this end, an inpainting model can be constructed. In particular, a GAN-inpainter.

Gan-based polyp inpainting

Geometric and pixel-wise transformations

3.1.3 Training methods

Consistency Training

3.1.4 Adversarial Consistency Training

3.1.5 Augmentation only training

3.2 Baselines and Metrics

3.2.1 Baseline Models

3.2.2 Performance Metrics

3.2.3 Datasets

3.3 Experiments

3.3.1 MNV-testing

3.3.2 Training methods

Chapter 4

Results

Chapter 5

Analysis

5.1 Augmentation Robustness and Consistency Loss

As the results show, the performance of the pipeline that merely used augmentations is more or less equivalent to the performance exhibited by the modified pipeline. There is a very good reason for this: Consistency loss is mathematically equivalent to data augmentation, up to the choice of hyperparemeters - i.e augmentation probability, learning rates, etc. This section presents a proof of this fact, along with a theoretical analysis of how data augmentation affects the pipeline.

5.1.1 Data augmentation

Let $Y := \{y, \hat{y} := f(x)\}$ be the set consisting of the segmentation predictions and masks for the unaugmented samples, and $A := \{a := MNV(y), \hat{a} := f(MNV(x))\}$ be the set consisting of segmentation predictions and masks for the augmented samples. Finally, let $Z := \{z, \hat{z}\} \in_R \{Y, A\}$. The loss function subject to data augmentation can then be expressed as $L(Z \in_R Y, A)$, where L is any loss function. For the sake of simplicity in remaining calculations, this will be treated as the Jaccard loss, i.e $L(y, \hat{y}) := 1 - \sum y \cap \hat{y} / \sum y \cup \hat{y}$

$$L(Z \in_R Y, A)$$

5.1.2 Consistency loss

$$L_s = \frac{1}{\sum \hat{y} \cup \hat{a}} \sum \hat{y} \ominus y \tag{5.1}$$

$$L_c = \frac{1}{\sum \hat{y} \cup \hat{a}} \sum [\hat{y} \ominus y \ominus \hat{a} \ominus a] \tag{5.2}$$

$$L_{c+s} = L_c(Y, A) + L_s(Y) \tag{5.3}$$

$$= \frac{1}{\sum \hat{y} \cup \hat{a}} \left[\sum \{\hat{y} \ominus y \ominus \hat{a} \ominus a\} + \sum \{\hat{y} \ominus y\} \right] \tag{5.4}$$

$$\begin{aligned} &= \frac{1}{\sum \hat{y} \cup \hat{a}} \Bigg[\sum \{\hat{y} \ominus y\} + \sum \{\hat{a} \ominus a\} \\ &- \sum \{\backslash y \cap \hat{y} \cap a \cap \hat{a}\} \cup \{y \cap \backslash \hat{y} \cap a \cap \hat{a}\} \cup \\ &\{y \cap \hat{y} \cap \backslash a \cap \hat{a}\} \cup \{y \cap \hat{y} \cap a \cap \backslash \hat{a}\} \\ &- \sum \{\backslash y \cap \hat{y} \cap \backslash a \cap \hat{a}\} \cup \{y \cap \backslash \hat{y} \cap \backslash a \cap \hat{a}\} - \\ &\cup \{\backslash y \cap \hat{y} \cap a \cap \backslash \hat{a}\} \cup \{y \cap \backslash \hat{y} \cap a \cap \backslash \hat{a}\} \\ &+ \sum \{\hat{y} \ominus y\} \Bigg] \end{aligned} \tag{5.5}$$

$$\begin{aligned} &= 2L_s(y, \hat{y}) + L_s(a, \hat{a}) + \frac{1}{\sum \hat{y} \cup \hat{a}} \Bigg[\\ &- \sum \{\backslash y \cap \hat{y} \cap a \cap \hat{a}\} \cup \{y \cap \backslash \hat{y} \cap a \cap \hat{a}\} \cup \\ &\{y \cap \hat{y} \cap \backslash a \cap \hat{a}\} \cup \{y \cap \hat{y} \cap a \cap \backslash \hat{a}\} \\ &- \sum \{\backslash y \cap \hat{y} \cap \backslash a \cap \hat{a}\} \cup \{y \cap \backslash \hat{y} \cap \backslash a \cap \hat{a}\} - \\ &\cup \{\backslash y \cap \hat{y} \cap a \cap \backslash \hat{a}\} \cup \{y \cap \backslash \hat{y} \cap a \cap \backslash \hat{a}\} \Bigg] \end{aligned} \tag{5.6}$$

The non-loss terms in equation 5.6 are proper subsets of the symmetric difference of the mask and segmentation across either dataset. The component of the loss that corresponds to these terms consequently grows in proportion to both $L_s(y, \hat{y})$ and $L_s(a, \hat{a})$. L_{c+s} and L_{sy+sa} are therefore monotonically correlated - i.e, when one grows, the other grows with it, and when one falls, the other one falls with it.

5.1.3 Adversarial Dice

$$L = \frac{1}{2} L(a, \hat{a}) + \frac{1}{2} L(y, \hat{y})$$

This should be asymptotically equivalent to data augmentation wit p=0.5

Chapter 6

Discussion

asdasdf

Bibliography

[1] Sharib Ali et al. 'EndoCV 2021 3rd International Workshop and Challenge on Computer Vision in En-doscopy'. In: ().

[2] Sharib Ali et al. 'Preface to: EndoCV2020Computer Vision in Endoscopy'. In: *CEUR Workshop Proceedings*. Vol. 2595. CEUR Workshop Proceedings. 2020.

[3] Martin Arjovsky et al. *Invariant Risk Minimization*. 2020. arXiv: 1907.02893 [stat.ML].

[4] Ishita Barua et al. 'Artificial intelligence for polyp detection during colonoscopy: a systematic review and meta-analysis'. en. In: *Endoscopy* 53.3 (Mar. 2021), pp. 277–284.

[5] Emma Beede et al. 'A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–12. ISBN: 9781450367080. DOI: 10.1145/3313831.3376718. URL: https://doi.org/10.1145/3313831.3376718.

[6] Battista Biggio et al. 'Evasion Attacks against Machine Learning at Test Time'. In: *Lecture Notes in Computer Science* (2013), 387–402. ISSN: 1611-3349. DOI: 10.1007/978-3-642-40994-3_25. URL: http://dx.doi.org/10.1007/978-3-642-40994-3_25.

[7] Alexander D'Amour et al. *Underspecification Presents Challenges for Credibility in Modern Machine Learning*. 2020. arXiv: 2011.03395 [cs.LG].

[8] Logan Engstrom et al. *Exploring the Landscape of Spatial Robustness*. 2019. arXiv: 1712.02779 [cs.LG].

[9] Ivan Evtimov et al. 'Robust Physical-World Attacks on Machine Learning Models'. In: *CoRR* abs/1707.08945 (2017). arXiv: 1707.08945. URL: http://arxiv.org/abs/1707.08945.

[10] Robert Geirhos et al. *ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness*. 2019. arXiv: 1811.12231 [cs.CV].

[11] Robert Geirhos et al. 'Shortcut learning in deep neural networks'. In: *Nature Machine Intelligence* 2.11 (2020), 665–673. ISSN: 2522-5839. DOI: 10.1038/s42256-020-00257-z. URL: http://dx.doi.org/10.1038/s42256-020-00257-z.

[12] Dan Hendrycks and Thomas Dietterich. *Benchmarking Neural Network Robustness to Common Corruptions and Perturbations*. 2019. arXiv: 1903.12261 [cs.LG].

[13] Dan Hendrycks and Thomas Dietterich. *Benchmarking Neural Network Robustness to Common Corruptions and Perturbations*. 2019. arXiv: 1903.12261 [cs.LG].

[14] D Heresbach et al. 'Miss rate for colorectal neoplastic polyps: a prospective multicenter study of back-to-back video colonoscopies'. en. In: *Endoscopy* 40.4 (Apr. 2008), pp. 284–290.

[15] Steven Hicks et al. 'The EndoTect 2020 Challenge: Evaluation and Comparison of Classification, Segmentation and Inference Time for Endoscopy'. In: Feb. 2021, pp. 263–274. ISBN: 978-3-030-68792-2. DOI: 10.1007/978-3-030-68793-9_18.

[16] Hossein Hosseini, Baicen Xiao and Radha Poovendran. *Google's Cloud Vision API Is Not Robust To Noise*. 2017. arXiv: 1704.05051 [cs.CV].

[17] Andrew Ilyas et al. *Adversarial Examples Are Not Bugs, They Are Features*. 2019. arXiv: 1905.02175 [stat.ML].

[18] Jacob Kauffmann et al. *The Clever Hans Effect in Anomaly Detection*. 2020. arXiv: 2006.10609 [cs.LG].

[19] A Leslie et al. 'The colorectal adenoma-carcinoma sequence'. en. In: *Br. J. Surg.* 89.7 (July 2002), pp. 845–860.

[20] Roman Novak et al. *Sensitivity and Generalization in Neural Networks: an Empirical Study*. 2018. arXiv: 1802.08760 [stat.ML].

[21] D K Rex et al. 'Colonoscopic miss rates of adenomas determined by back-to-back colonoscopies'. en. In: *Gastroenterology* 112.1 (Jan. 1997), pp. 24–28.

[22] Alexander Robey, Hamed Hassani and George J. Pappas. *Model-Based Robust Deep Learning: Generalizing to Natural, Out-of-Distribution Data*. 2020. arXiv: 2005.10247 [cs.LG].

[23] Veit Sandfort et al. 'Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks'. In: *Scientific Reports* 9 (Nov. 2019). DOI: 10.1038/s41598-019-52737-x.

[24] Bernhard Schölkopf. *Causality for Machine Learning*. 2019. arXiv: 1911.10500 [cs.LG].

[25] Dinggang Shen, Guorong Wu and Heung-Il Suk. 'Deep Learning in Medical Image Analysis'. In: *Annual Review of Biomedical Engineering* 19.1 (2017). PMID: 28301734, pp. 221–248. DOI: 10.1146/annurev-bioeng-071516-044442. eprint: https://doi.org/10.1146/annurev-bioeng-071516-044442. URL: https://doi.org/10.1146/annurev-bioeng-071516-044442.

[26] Ashish Shrivastava et al. 'Learning From Simulated and Unsupervised Images Through Adversarial Training'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[27] Sidney J. Winawer et al. 'Prevention of Colorectal Cancer by Colonoscopic Polypectomy'. In: *New England Journal of Medicine* 329.27 (1993). PMID: 8247072, pp. 1977–1981. DOI: 10.1056/NEJM199312303292701. eprint: https://doi.org/10.1056/NEJM199312303292701. URL: https://doi.org/10.1056/NEJM199312303292701.

[28] Julia Winkler et al. 'Association Between Surgical Skin Markings in Dermoscopic Images and Diagnostic Performance of a Deep Learning Convolutional Neural Network for Melanoma Recognition'. In: *JAMA Dermatology* 155 (Aug. 2019). DOI: 10.1001/jamadermatol.2019.1735.

[29] John R. Zech et al. 'Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study'. In: *PLOS Medicine* 15.11 (Nov. 2018), pp. 1–17. DOI: 10.1371/journal.pmed.1002683. URL: https://doi.org/10.1371/journal.pmed.1002683.

[30] Chiyuan Zhang et al. *Understanding deep learning requires rethinking generalization*. 2017. arXiv: 1611.03530 [cs.LG].