Inducing natural invariance in deep segmentation pipelines for generalizable detection of polyps in GI tracts

Any short subtitle

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

Autumn 2021

Inducing natural invariance in deep segmentation pipelines for generalizable detection of polyps in GI tracts

Any short subtitle

Birk Sebastian Frostelid Torpmann-Hagen

Inducing natural invariance in deep segmentation pipelines for generalizable detection of polyps in GI tracts

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

Printed: Reprosentralen, University of Oslo

Abstract

Contents

List of Figures

List of Tables

Preface

Introduction

Colorectal cancer is one of the leading causes of cancer related deaths, causing approximately 9000000 deaths worldwide per year (cite). Early detection thereof is as a consequence of significant importance. Polyps are often an early warning-sign of developing tumor, 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 shape and size 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 treatment.

Clinical applications of deep learning are, however, known to fail in deployment, despite exhibiting excellent performance during development. This is known as generalization failure, and is ubiquitous in the domain. While there has been a growing body of research dedicated to identifying and analyzing the root causes of such failure, most attempts (...)

This thesis presents a novel approach to increasing generalizability, whereby the model is trained to not only minimize segmentation-loss, but also minimize the effects of the data being perturbed by an ensemble of transformations, including color-transformations, geometric transformations, additive noise, and adding extra polyps to the image using a GAN-inpainter. 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.

Generalizability is then measured by evaluating several vanilla-pipelines consisting of several models on a number of separate datasets, which is then compared to the results of the modified pipeline. The results show that (...)

Chapter 2

Background

Medical imaging has in recent years proven to be one of the most promising applications of deep learning, having the capacity to significantly improve both the accuracy and efficiency of detection, diagnosis, and treatment of a wide variety of diseases [4]. 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, despite exhibiting high performance on hold-out sets [1, 2].

2.1 Generalizability

... The term "generalizability" is used widely in literature, despite rarely being particularly well defined. Often, generalizability refers merely to the state of a predictor as "not overfitted". In other words, that it maintains sufficient performance across the training, validation, and test-sets. This, however, typically neglects the more salient aspects of the performance of the pipeline; namely how it behaves when deployed in practical settings, on data that may be distributed differently from the training dataset. Consider for instance the problem of detecting and classifying traffic signs. Though it is relatively trivial to achieve decent performance on such a task when training and evaluating on one specific dataset, it is an entire matter entirely to make sure the resulting predictors are robust to any and all forms of variation one might expect to see when deployed in a practical setting. If the training data was for instance collected from a region with a dry, temperate climate, it might not come as a surprise that it will not perform as well when deployed in an area prone to snowfall, fog, or generally low visibility. Of course, this could be mitigated by ensuring that the training data contains samples from a wide variety of climates, but this really only affects the robustness of the pipeline to differing climates. It does not necessarily ensure that the resulting predictors learn to ignore weather effects entirely, and as such it may nonetheless fail if it encounters something it has not been explicitly trained on. This is made especially evident in the study of black-box adversarial attacks:

In other words, merely being robust to a limited class of perturbations

or variability is not sufficient to deem a pipeline as generalizable. The pipeline must not merely learn to be right, but right for the right reasons. If this is achieved, robustness to distributional shifts follows. The system outlined above should in other words not only be robust to snow or rain or fog, but to be able to ignore the effects thereof entirely. A perfectly generalizable pipeline shouuld return a weather-invariant predictor every time, and for that matter maintain invariance to any and all non-destructive distributional shifts.

The term generalizability will as such in this thesis refer to the ability to infer the right inductive biases from an incomplete dataset. This is as opposed to robustness, which denotes the ability of a pipeline to maintain its performance across certain distributional shifts. Generalizability is as a consequence not as much of a measurable quantity as much as it is an emergent property of a well-designed pipeline.

This section will explore the concept of generalizability in further detail. It will outline how typical deep learning-based systems aim to achieve generalizability, why it nonetheless often fails to do so, and how one can analyse such generalizability failure.

2.1.1 Empirical Risk Minimization

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 $f : X \rightarrow Y$, 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 label. 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 $h : X \rightarrow \hat{Y}$, 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 (henceforth: iid) to $P(x, y)$. This is the so-called iid assumption. Note that 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:

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

Since $P(x, y)$ is not known, however, one cannot compute $R(h)$ explicitly. Instead, the expected risk has to be computed through empirical estimation, 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):

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

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.

Naturally, however, real-world data is rarely neat enough for it to consistently abide by the iid assumption. Commonly encountered variation in real world data such as variable instance lighting conditions, class imbalance, image corruptions, noise, or other more subtle forms of distributional shift all result in structural misalignment of the training and deployment distributions (citation). Ideally, predictors should be robust to these sorts of changes, however evidently this is in not guaranteed by ERM (citation). ERM simply guarantees an iid-optimal predictor. While the difference is subtle, it is worth reemphasizing: empirical risk minimization only generalizes to data which is more or less identically distributed to the training data. Differently distributed or otherwise perturbed data, even that which is near imperceptible or at any rate inconsequentially different to the human eye, violates the iid assumption, and can as such not be expected to be classified correctly given a predictor trained via ERM.

To mitigate this, one could simply add more data to the pipeline through augmentation, or simply collecting more training data. This will lead to a better approximation of the true risk. This does not, however, solve the problem. The variability of the real world is not, unfortunately, easy to model merely through augmentations, and collecting sufficient data to cover every potential source of natural variability is infeasible, especially in medical domains. Consider for example a machine-learning pipeline

wherein a model is trained to classify cows and camels. The dataset consists of cows, pictured in grass fields and pastures, and camels, pictured in deserts. To be generous, let us assume that we have sufficient quantities of data to ensure that the pipeline is perfectly invariant to the pose of the respective animals, to lighting conditions, geometric transforms, etc. One may then expect that the pipeline correctly learns to classify the two, and attains high accuracies, and indeed when evaluated on iid data, this would be entirely correct. However, what would then happen if one such predictor encountered a cow in the desert and a camel in a grass pasture? This constitutes a distributional shift, and as such we cannot expect reliable performance as detailed in 2.1.1. Naturally, the predictor may have learned just fine exactly what constitutes a cow and a camel, but it might just as easily learned to associate deserts with camels and pastures with cows. And from a data perspective, both are equally correct interpretations. The immediate response to this may be to simply add some pictures with more varied backgrounds, but this once again would only serve to make the pipeline more robust to backgrounds. it would not guarantee that the pipeline learns the right inductive biases. The predictor may then for example instead learn that cows typically are black and white and camels usually beige, and then fail when it encounters a brown cow. One could keep adding more and more data, but there is not really any way of knowing when the pipeline is well enough specified by the data such that it starts returning predictors with the desired inductive biases. There are in simpler terms several "correct" interpretations of what separates the classes from a purely data-based perspective, each with their own inductive biases. There are as a consequence not just one risk-minimizing predictor, but a whole family of them. This is referred to as underspecification [2].

2.1.2 Underspecification

2.1.3 Sensitivity

2.1.4 Spurious correlations

2.2 Related work

2.2.1 Model robust deep learning

2.2.2 Divergent Net

2.3 ...

Models of natural variation

GAN-inpainting

Methodology

As described in earlier sections, good generalizability can only be achieved if the pipeline can reliably produce predictors that infer the right inductive biases. Naturally, the set of correct inductive biases are not known, so any such pipeline instead has to learn to not infer the wrong inductive biases. To achieve this, a model of natural variance is constructed, which aims to encapsulate all the variability one might expect to see in the domain. This model can then be leveraged to force the pipeline to be robust to natural variance through contrastive learning. The central idea, then, is that it is more likely that the model learns to infer generalizable inductive biases as opposed to learning to learning to simply be robust to all possible configurations of a large amount of transformations.

To evaluate this, several predictors are trained from several pipelines with and without the influence of (algorithm name). Their performance is then evaluated on both a stress-test, and two separate polyp datasets, namely Etis-larib and EndoCV2021).

3.4 (algorithm name)

3.4.1 Model of natural variability

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, in other words a model of natural variability (MNV) Naturally, there is no way of knowing the full extent thereof, but it may be sufficient to model some subset of the possible distributional shifts. 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, 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 for data-augmentation. Manifold-variability, however, is somewhat more difficult. Similar to how [3] employs cross-dataset style-transfer, 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 in-painting model can be constructed. In particular, a GAN-inpainter.

Gan-based polyp inpainting

3.4.2 Geometric and pixel-wise transformations

3.5 Baselines

Several models were tested (...)

3.6 Datasets

3.7 Metrics and evaluation

Results

Discussion

asdasdf

Bibliography

[1] 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.

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

[3] 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].

[4] 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.