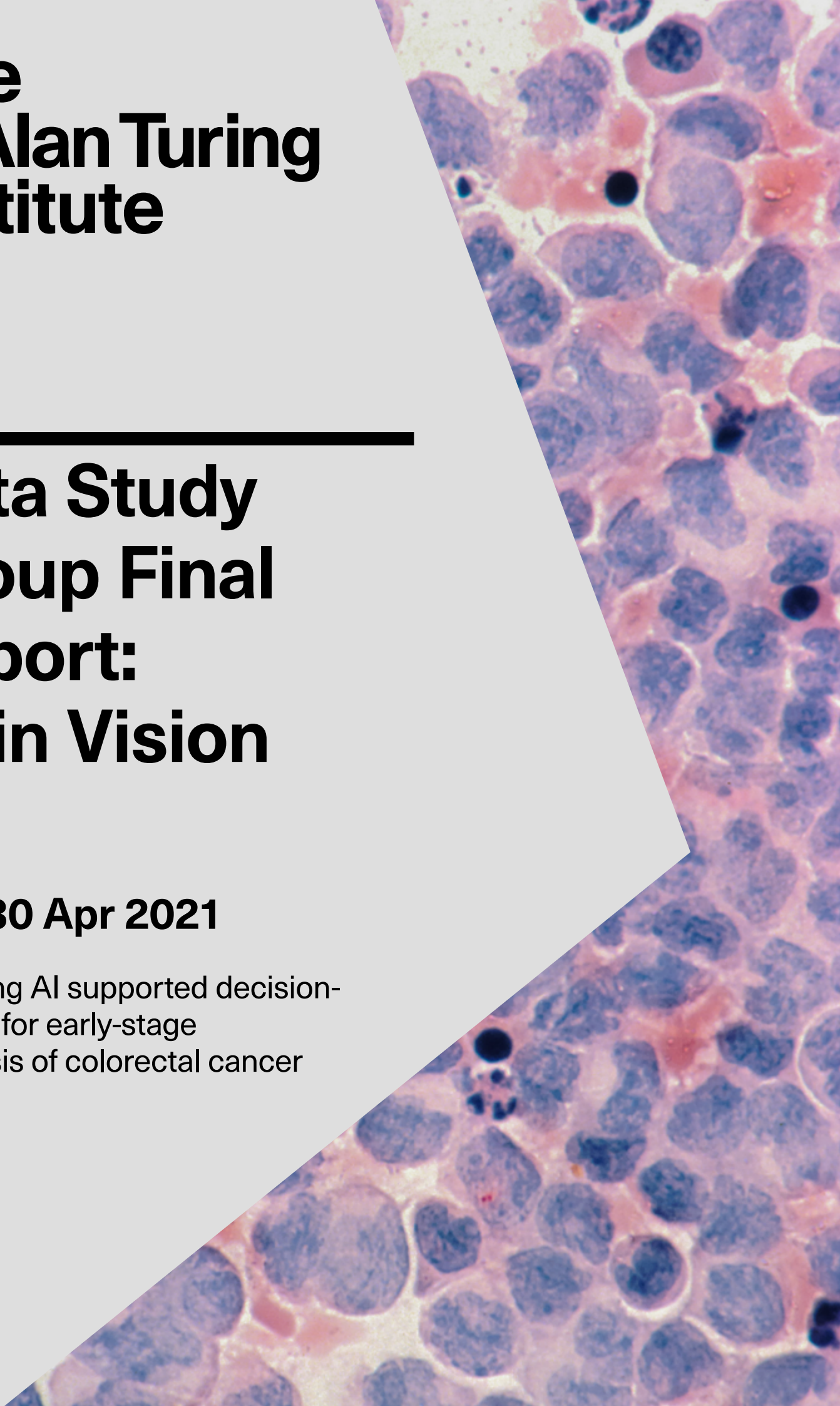
The Alan Turing Institute

Data Study Group Final Report: Odin Vision



19 – 30 Apr 2021

Exploring AI supported decision-making for early-stage diagnosis of colorectal cancer

https://doi.org/10.5281/zenodo.5729340

Executive summary

Objectives & Approaches

The aim of this challenge is to explore methods that enhance the explainability of Odin-Vision's current machine learning models to aid clinical decision-making. Their current capabilities include a real-time detection and classification model, deployed in a clinical setting, where a polyp is first imaged by a clinician and automatically classified as adenoma or non-adenoma; a binary classification task. The procedure is time sensitive and each polyp gets imaged approximately only once, with clinicians taking a few seconds for image capture and decision making. The aim of the machine learning model is to aid the clinician's decision process, providing confidence in more ambiguous cases and substantially increase the reproduciblity of those decisions. The clinician's trust in the model is also particularly important to encourage widespread uptake and acceptance of automated methods in a clinical setting.

We approach the main objective of this challenge in three main workstreams:

1. **Uncertainty estimation:** first we enhance the classification process with uncertainty bounds. The aim is to make model predictions more interpretable and build clinician's trust in the ML models. We explore adding a measure of predictive uncertainty along with class prediction;
2. **Attribution methods:** we investigate multiple attribution methods to make the classifications more understandable to clinicians. The aim is to output an image heatmap that describes which pixels or regions in the image are most important to the model's classification;
3. **Automatic representation:** in order to move beyond the standard optical feature representation NICE (described in Section 1.3), we explore automatic feature representation learning methods based on self-supervised generative models. We aim to learn a disentangled (non-overlapping) representation of features, and interpret what individual features represent.

Contributions

Uncertainty estimation

We adopted a Monte Carlo (MC) Dropout approach to provide a level of uncertainty in the model's classification. This approach has been shown to closely approximate Bayesian inference [1], and was considered the best approach given the time available. The quantification of the model uncertainty was addressed by combining a collection of diagnostics. Our approach allowed us to (i) gain insight into the model's confidence by providing a distribution over predicted classifications and (ii) relate model confidence to model performance. We successfully applied this approach to several model architectures. Finally, a framework was developed to translate the statistical outputs of the model into a widely understandable "certainty" score that presents the certainty of the model on an ordinal 1-5 scale for greater interpretability.

Attribution methods

We explored a variety of methods for interpreting neural network classifications. We focused on gradient-based and perturbation-based attribution methods. To summarise, we provide the relative influence of input features (pixels or regions) in the classification process, based on multiple

techniques. We found that Guided GradCam is the most insightful method, based on discussions with Odin-Vision clinicians. We combined these attribution methods with a predicted level of uncertainty (from our work on uncertainty estimation) to increase the trustability in the model and identify potential failure modes such as false negative predictions. Further, both measurements have been incorporated into a bespoke interactive dashboard, ideal to provide to clinicians, illustrated in Figure 1.

Representation learning

We successfully implemented a Resnet Variational Autoencoder (VAE) and a Beta-VAE to explore self-supervised automatic representations of a polyp image dataset. We demonstrate this is a viable direction of research and show that we can reconstruct polyp images after passing them through a low-dimensional bottleneck representation. Further, we explore what the individual features in the latent code represent. We do this by providing a sequence of reconstructed images, generated by smoothly varying just one latent dimension and decoding the image.

Integrated Dashboard:

We demonstrate the integration of an interpretable measure of predictive uncertainty with the output of an attribution method, applied to two test images in Figure 1. The left image demonstrates a false positive classification, and right, a true positive case. The attribution method used is Guided GradCAM. We can see for the false negative, the attribution method highlights specular reflections in the image, an unrelated feature, whereas in the true positive case, it highlights vein structure and meaningful patches within the polyp image. We also provide a measure of predictive uncertainty (right panel of each image) based on MC Dropout. We can see that the true positive is more confident about the prediction (location of peak), and more certain (width of distribution). We could also present other metrics and store them in the cloud.

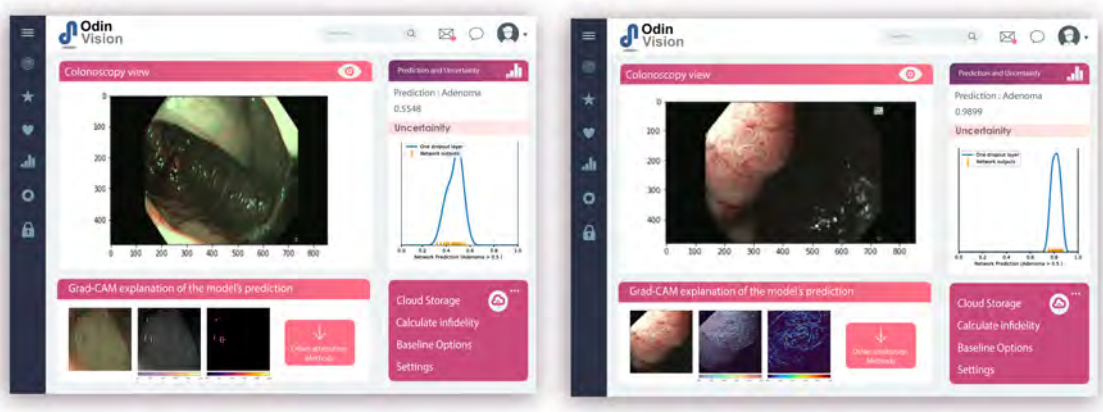


Figure 1: A proposed bespoke interactive dashboard: To identify potential failure modes and provide real-time rational of model predictions to clinicians using automated software. Two test cases are provided: (left) a false positive classification, (right) a true positive classification. Guided Gradcam provides the attributions (lower panel), and MC Dropout provides the uncertainty measure (right pannel). Dashboard was designed using Adobe illustrator and resources from Freepik.com

Conclusions

We have provided multiple tools, based on machine learning techniques, that make it possible to interpret Odin-Vision's so-called, "black-box" prediction models for colorectal polyp classification. This challenge was predominantly exploratory, and as a team we have made progress across all three proposed workstreams. Ultimately, the combination of three directions: i) approximate Bayesian models, ii) attribution methods, and iii) representation learning, has lead to a better understanding of Odin-Vision's current deep learning models and will have tangible outcomes for both clinicians, patients and Odin researchers. Specifically:

1. Monte Carlo dropout has been shown to accurately incorporate model prediction uncertainty, and highlight potential false negative classifications. This is a particular highlight for Odin-Vision researchers.
2. Attribution and occlusion methods have been shown to highlight certain image structures, such as vein patterns, as highly influential and attributed to positive classification classes. However, there were challenges regarding suitable summary metrics to use, and parameters such as appropriate image baselines.
3. A variational auto-encoder was successfully developed that provided a proof-of-concept that a low-dimensional bottleneck (latent space) can be found and queried in order to represent an image dataset of colorectal polyps. By varying individual features in this low-dimensional representation, and decoding back into images it is a huge first step towards understanding automatic feature representations, along with implications for better automated classification and diagnosis.

Limitations and Future work

All datasets used during the challenge are discussed in Section 2.2. One major limitation during this challenge was the poor quality of the open source colorectal polyp datasets (discussed in Section 2.3.) As this was the only publicly available data for this challenge, one solution was for researchers at Odin-Vision to train models we developed on their "in-house" dataset. This allowed us to better understand the effectiveness of different methods on real data, whilst limiting access to their sensitive data. This is described in more detail in Section 2.4. A second limitation was time. We did not have sufficient time to train very large or complex models, as we were required to rapidly prototype ideas. As such, techniques that require large computational resources or time for classification were not explored. Finally, computational power within a clinical setting is restrictive. For example, full Bayesian models to provide an accurate uncertainty measure would be too computationally expensive to run inbetween the time a clinician takes an image, and waits for an automated classification. Clinicians at Odin-Vision suggested there should be no more than a one second delay.

Overall, the three pronged approach was able to effectively explore the research question, and provide valuable and insightful findings. Future work could further explore the combination of the attribution methods with uncertainty quantification, e.g. provided by the same network. Secondly, the uncertainty quantification could be upgraded to a full Bayesian neural network. Finally, future work could develop further the automatic feature representation learning findings, providing a real alternative to the currently hand-labelled optical visual features (NICE). We expand upon future directions in Section 5.

Contents

1 Introduction

1.1 Background

In the UK, there are over 42,000 new cases of colorectal cancer (CRC) and 16,000 related deaths per year, making it the second leading cause of cancer deaths [2]. The number of CRC related deaths is predicted to increase by 51% over the next 15 years [3], with increased prevalence in young people [4]. CRC is treatable and curable: nearly everyone survives if diagnosed at the earliest stage but this drops significantly as the disease develops [2].

During colonoscopy, a camera is used to inspect the large bowel and rectum for small growths called polyps. Some polyps are harmless ('benign') whereas others can develop into cancer ('pre-cancerous'). Most polyps are resected and diagnosed by histopathology examination. Histopathology reports are costly, time-consuming and only 3% of NHS departments have enough staff to meet future clinical demand [5].

There is a drive to replace routine histopathology examination with real-time optical diagnosis which is cost-effective, streamlines follow-up care and reduces patient anxiety due to same-day results [6]. There are however, barriers to clinical adoption. For example, a large UK multi-centre clinical trial, inclusive of non-expert endoscopy centres, found test sensitivity for human optical diagnosis fell below the required threshold to gain regulatory approval [7].

Recent breakthroughs in computer-aided diagnosis for colonoscopy are gaining attention and could increase the viability of automatic optical diagnosis [8]. Odin-Vision has developed technology to detect and characterise colorectal polyps in real-time using deep learning algorithms. We believe that the best way to overcome the challenges of optical diagnosis is to combine artificial intelligence with the experience of clinicians, to provide better, automated tools to work with.

Even with the use of computer-aided diagnosis, it remains the responsibility of the endoscopist to make a final diagnostic call. Neural networks often produce high predictive probabilities even for incorrect predictions [9, 10] and offer no measure of uncertainty over their prediction. This can be confusing and harmful to users who generally have little machine learning background. In this challenge we also encourage the use of probabilistic methods, namely Bayesian models, to help quantify the predictive uncertainty along with innovative ways to communicate this valuable information to clinicians. The inclusion of a measure of uncertainty alongside the model prediction would enable clinicians to discard uncertain predictions, and increase reproducibility.

1.2 The Challenge

In this challenge we explore methods to facilitate AI supported decision-making in an automated colonoscopy procedure, with a focus on delivering interpretable predictions to clinicians to support diagnosis. We are primarily interested in two measures necessary for the acceptance of optical diagnosis in clinical practice; 1) standardised optical diagnosis criteria and 2) level of diagnostic confidence [6].

Odin-Vision have developed neural networks that predict polyp histopathology with high accuracy and which create internal representations that capture the optical features overlooked by current diagnostic criteria. In this challenge we provide a deeper understanding of these automatic feature representations and explore methods to interpret them. We explore three avenues to achieve this: Approximate Bayesian methods that provide uncertainties; Gradient and perturbation-based attribution methods; and automatic representation learning using self-supervised methods.

1.3 Current Odin-Vision ML Capability

We begin with a brief technical overview of Odin-Visions's current capabilities, and a recount of some commonly used terminology.

The current approach at Odin-Vision involves adopting a state-of-the-art binary classifier to produce an output label for a polyp optically identified by instance detection techniques during a colonoscopy procedure. Specifically, the neural network outputs a score (a float) $\hat{y} \in [0, 1]$. A threshold τ is selected for determining positive and negative classes, and a prediction $\hat{y} > \tau$ identifies image $\mathbf{x}$ as belonging to the positive class (adenoma), or to the negative class (non-adenoma). The predictive distribution can be considered as a Bernoulli distribution:

$$B(p, k) = q^k (1 - q)^{1-k} \tag{1}$$

with $q = \hat{y}$ and where $k = 0$ and $k = 1$ correspond to the negative and positive class respectively.

The existing pre-trained neural network model, made available during the challenge, is based on a resnet101 architecture [11] implemented in Pytorch [12]. Residual networks belong to the class of DAG networks with residual connections that bypass the main network layers. Residual connections enable the parameter gradients to propagate more easily from the output layer to the earlier layers of the network, making it possible to train deeper networks [13]. This increased network depth can result in higher accuracy. The depth of the network is defined as the largest number of sequential convolutional or fully connected layers on a path from the input layer to the output layer. The depth of ResNet-101 is 101, although in total, it has 347 layers [13, 14].

A major limitation of this type of "deep" model is that the predictions are difficult to interpret. First, this approach does not provide information about the uncertainty of the prediction, which is crucial to assess the model confidence in classifying a particular instance of a polyp image. Moreover, the assistive technology developed by Odin-Vision does not currently provide any insight about which features (pixels or patches) of the image are most informative to the prediction method, i.e. which areas informed the output prediction of a certain class label. It is essential that these two issues are addressed to build trust in the automatic classification models, so it may be safely deployed in clinical practice.

Terminology

Throughout this report various clinical, and machine learning terms are used. In this brief section we define some of the common terminology used:

- True Positive (TP): A prediction from a binary classification that correctly matches the positive class label.
- True Negative (TN): A prediction from a binary classification that correctly matches the negative/null class label.
- False Positive (FP): A prediction from a binary classification that incorrectly predicts a positive class when the class label is negative/null.
- False Positive (FN): A prediction from a binary classification that incorrectly predicts a negative class when the class label is positive.
- Precision: TP / (TP+FP) (rate of occurrences summed over a dataset)

- Recall: TP / (TP+FN) (rate of occurrences summed over a dataset)
- F1: 2 $\times$ (Precision $\times$ Recall)/(Precision + Recall)
- Accuracy: (TP+TN)/(TP+TN+FP+FN)

Standardised Optical Diagnosis Criteria

When examining colorectal polyps, clinicians focus on image characteristics that have been shown to be most relevant to discriminate between adenoma and non-adenoma polyps. The most widely used optical diagnosis criterion is the NBI International Colorectal Endoscopic (NICE) classification. It describes the real-time classification of hyperplastic (mostly benign) and adenomatous (pre-cancerous) polyps using three hand-crafted image features: 1) polyp colour, 2) vessel structure and 3) surface pattern. Table 1 provides the NICE polyp classification system. More details regarding the NICE classification system can be found in [15–17]. However, in the aforementioned clinical trial [7], Rees et al. found that 35% of adenomatous polyps did not exhibit any of these features.

Feature	**Type 1**	**Type2**	**Type3**
Colour (Relative to Background)	Similar	Brownish	Dark brown with white spots
Vessel structure	None	Brown vessels	Disrupted or Missing vessels
Surface pattern	Uniform spots	Brown Vessels and white structures	Lack of surface pattern
Pathology	Hyperplastic	Adenoma	Submucosal invasive cancer

Table 1: NICE Classification System for images containing Polyps

2 Data overview

Odin-Vision collated seven public datasets of colorectal polyp images. The images were taken by either white light endoscopy (WLE) or narrow-band-imaging (NBI), and were merged into three datasets available for the challenge. We describe the data modalities and then each dataset.

2.1 Data Modalities

White light endoscopy uses a light source that comprises the full spectrum of visible light meaning a large number of wavelengths are emitted and reflected by the tissue. In contrast, narrow band imaging (NBI) technology passes light through a narrow band filter before it illuminates the tissue, allowing only two specific bands of light to pass (See Figure 2). The frequencies that pass match the absorption spectrum of hemoglobin contained in blood which improves the visibility of blood vessels and mucosal structures. NBI light is absorbed by vessels but reflected by mucosa achieving a maximum contrast between vessels and the surrounding mucosa (Figure 3). The shorter of the two NBI light wavelengths only penetrates the superficial layers of the mucosa and is absorbed by capillary vessels on the surface. This facilitates the detection of tumors, as they are often highly vascularised. The longer NBI wavelength penetrates deeper and is absorbed by blood vessels located within the mucosal layer. Thus, it is particularly helpful to display the deeper vasculature of suspect lesions.

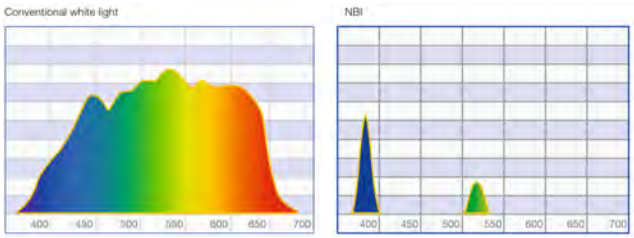


Figure 2: In contrast to white light, NBI light is composed of only two specific bands of light. Image taken from [18].

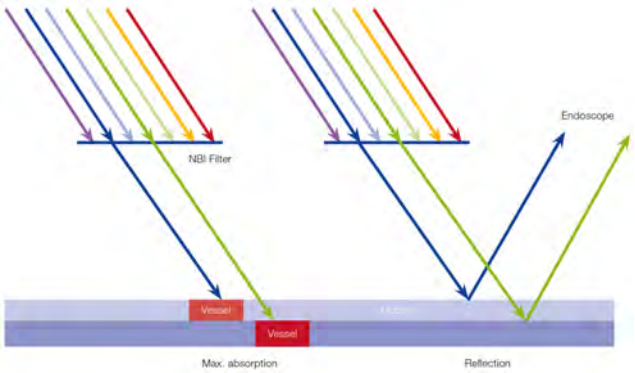


Figure 3: Absorption of narrow band light by capillaries on the mucosal surface (blue) and veins in the submucosa (green). Image taken from [18].

2.2 Dataset Details

In general, RGB images containing polyps are first cropped around the polyp area and then resized to 244×244 pixels. The images are then normalised for processing, e.g. output[channel] = (input[channel] - mean[channel]) / std[channel].

Where labels are available, a one or zero associated the polyp image with adenoma or non-adenoma, and was the focus of the binary classification challenge.

For preliminary testing purposes we also made use of a standard handwritten digits image dataset, MNIST [19].

Database	**Identifier**
1. CVC-ClinicVideoDB [20]	cvc-12k-wle
2. CVC-ClinicDB [21]	cvc-612-wle
3. Kvasir-Seg Dataset / Hyper-Kvasir [22]	kva-seg-wle / kva-unl-wle
4. PICCOLO RGB/NBI (WIDEFIELD) [23]	piccolo-nbi / piccolo-wle
5. Colonoscopic Dataset (Depeca) [24]	depeca-nbi / depeca-wle
6. SUN Colonoscopy Video Database [25, 26]	sun-wle
7. Kansas [27]	kansas-nbi / kansas-wle

Table 2: Publicly available polyp databases

Dataset 1: Narrow Band Imaging (NBI)

This dataset consists of the pooled, publicly available, narrow band image datasets. All provided with binary classification labels Adenoma (1) vs Non-adenoma (0):

Database	Labels	Adenoma	Non-adenoma	Total Images
piccolo-nbi	Labelled	765 (1)	2,872 (0)	3637
depeca-nbi	Labelled	16,317 (1)	11,053 (0)	27,370
kansas-nbi	Labelled	1,835 (1)	1,343 (0)	3,178
Total		18,917 (1)	15,268 (0)	31,600

Table 3: Dataset 1

Dataset 2: White Light Endoscopy (WLE)

This dataset is a collection of publicly available white light endoscopy datasets; four provided with binary labels and four provided without labels.

Database	Labels	Adenoma	Non-adenoma	Total Images
cvc-612-wle	Unlabelled	-	-	648
kva-seg-wle	Unlabelled	-	-	1,068
kva-unl-wle	Unlabelled	-	-	1,803
cvc-12k-wle	Unlabelled	-	-	10,025
total		-	-	13,544
piccolo-wle	Labelled	1,423 (1)	420 (0)	1,843
depeca-wle	Labelled	0	3,572 (0)	3,572
kansas-wle	Labelled	1,200 (1)	367 (0)	1,567
sun-wle	Labelled	43,887 (1)	4,557 (0)	48,444
Total		46,510 (1)	8,916 (0)	55,426

Table 4: Dataset 2

Dataset 3: Odin-Vision In-house Data

This is a high-quality, non-public, dataset of colorectal polyp images. This dataset was unavailable during the challenge, but researchers at Odin-Vision were able to use it to train our developed models.

Database	Labels	Adenoma	Non-adenoma	Total Images
internal	Labelled	85,893 (1)	32,887 (0)	118,780

Table 5: Odin-Vision In-house Dataset

2.3 Data Quality Issues

Positive Bias

One issue with the relatively small datasets described in Section 2.2 is the unbalanced classes that is typical in real-world medical datasets. Here we see a positive-case bias in both Dataset 2 and Dataset 3, i.e. they have many more Adenoma labelled images than Non-adenoma images, and this is likely to effect the downstream analysis.

Blurry image

Whilst significant effort was made to collate high quality images into the three datasets described above, unfortunately, they included many blurry images. An example is shown in Figure 4.

One method used by Odin-Vision to tackle this was to associate each images with a "blur" score ranging from -200 to 0, where smaller numbers correlated to a lower quality image [28]. See Figure 5 for plots applying this metric to two of the publicly available datasets: sun-wle (top) and piccolo-nb (bottom).

In order to maintain model accuracy and confidence in the training process, we limit the blur score to values ≥ -120 and ≤ -60 to threshold accepting images. For example, a commonly used test dataset in our Experiments and Results Section 4 was the Piccolo-nbi. Applying this blur-score filter, reduced the set of images from $1,052$ to 900, of which 252 are labelled as non-adenoma, and 648 are adenoma.

Figure 4: Examples of blurry image.

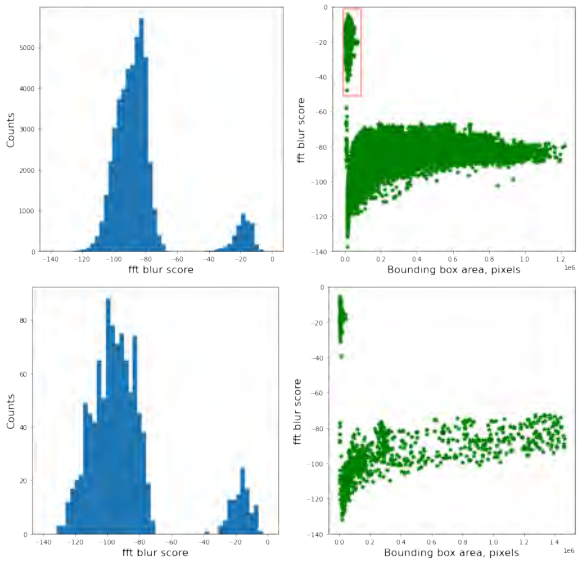


Figure 5: Applying the “Blur Score” from [28] to the sun-wle (top) and picollo-nbi (bottom) datesets.

Post-cleaning Datasets

We reduced Dataset 1 and Dataset 2 using the blur score. The new number of images are described in Table 6 and Table 7.

Database	Labels	Adenoma	Non-adenoma	Total Images
piccolo-nbi	Labelled	648 (1)	252 (0)	900
depeca-nbi	Labelled	4,685 (1)	2,359 (0)	7,044
kansas-nbi	Labelled	501 (1)	686 (0)	1,187
Total		5,834 (1)	3,297 (0)	9,131

Table 6: Dataset 1 after applying the Blur score.

Database	Labels	Adenoma	Non-adenoma	Total Images
cvc-612-wle	Unlabelled	-	-	597
kva-seg-wle	Unlabelled	-	-	1,057
kva-unl-wle	Unlabelled	-	-	1,797
cvc-12k-wle	Unlabelled	-	-	3,706
total		-	-	7,157
piccolo-wle	Labelled	1,139 (1)	347 (0)	1,486
depeca-wle	Labelled	0	2,400 (0)	2,400
kansas-wle	Labelled	318 (1)	202 (0)	520
sun-wle	Labelled	28,451 (1)	3,842 (0)	32,293
Total		29,908 (1)	6,791 (0)	36,699

Table 7: Dataset 2 after applying the Blur score.

Reflective spots

Many images also contained reflective white spots due to the endoscope light reflecting off the polyp (specular reflection). An example is shown in Figure 6 (left).

The reflective light spots cause particular problems with the attribution methods used to interpret the model's predictions. Some methods showed that the model's gradients' would activate very strongly in regions of the input images corresponding to these reflections. As such the interpretability would be significantly less useful to the clinician.

To alleviate this issue we sought to either remove or reduce the difference between these regions and the rest of the image. We altered the pre-processing pipeline provided for the extraction of these images to remove the normalisation process, since some methods similar to those used here only work on images with pixel values in the range 0-255. We converted the images into grayscale and then used binary thresholding to create a mask of the original image with the reflective regions' pixels set to 1 and everywhere else set to 0 as shown in Figure 6.

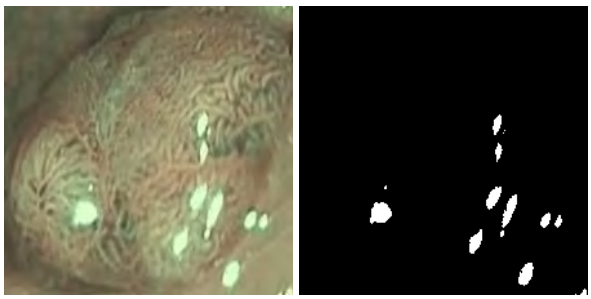


Figure 6: Example of polyp image with specular reflections (left) and generated mask (right).

We used a variety of inpainting methods to restore these regions based on their neighbouring pixels. We tested three algorithms for this including the Telea method, a method based on the Navier-Stokes equations and a method that uses biharmonic equations [29–31]. Examples of the effect these had on the original image are included in Figure 7.

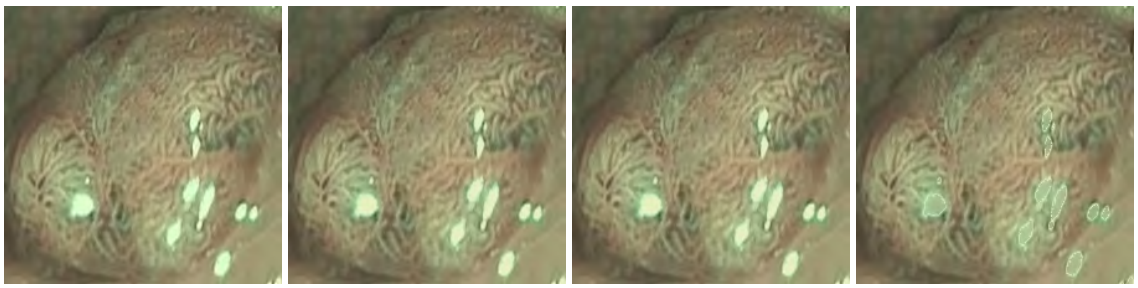


Figure 7: Original image with specular reflections (far left), image after inpainting with biharmonic equations (centre left), image after inpainting with Telea method (centre right), image after inpainting with Navier-Stokes equations (far right).

Ultimately, this issue can be eliminated with higher quality data. If this was found to be a frequent problem, then a simple approach would be to require another picture to be taken by the practitioner. However, if it was desired for the existing or another dataset to be cleaned, removing this issue it is likely a more sophisticated methods such as Belief Propagation [32] and Generative Adversarial Networks (GANs) would be required to have better results [33].

2.4 Training vs Test Dataset Caveat

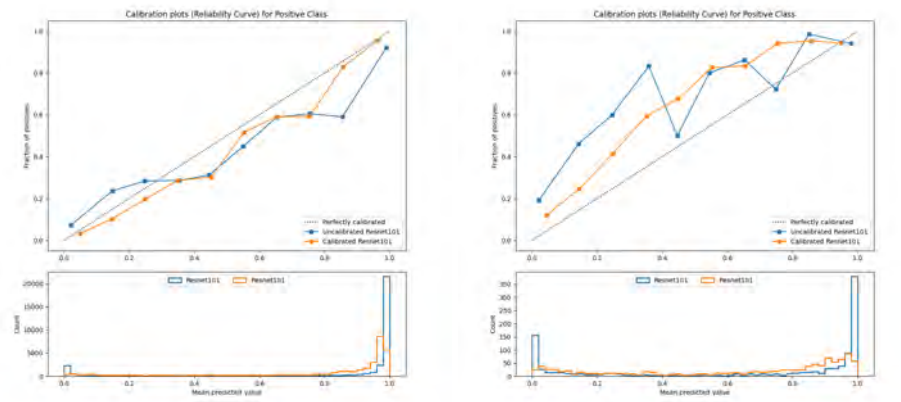


Figure 8: (Left): Calibration curve for the internal Odin Vision dataset. (Right): Calibration curve for the Piccolo dataset.

To alliviate some of the challenges presented in this section, the models developed for uncertainty quantification and attribution methods were predominantly trained on the internal Odin-Vision dataset (Dataset 3), whose individual images were not made available to the team. However, to

properly understand the approaches we were investigating, we had the requirement to use a test set of images that were publicly available, most notably from either the Piccolo or Depeca datasets (see Section 2.2). In principle these datasets may not be drawn from the same data distribution as the training set, and are generally considered lower quality image datasets than the internal dataset.

To investigate this, we ran a standard confidence calibration algorithm based on temperature scaling – the problem of predicting probability estimates representative of the true correctness likelihood [34]. Calibration was found to be very good for the Odin-Vision internal dataset (see Figure 8 (left) while that is not the case for the Piccolo dataset (shown in Figure 8 (right)).

Therefore, any particular quantitative results shown in this report might be affected by the distributional shift between training and test datasets. However, we believe it was the best way of training our models using high-quality, not publicly available data, and presenting results in this report using the publicly available images. While we believe in the exploratory methodology set out in this report, we recommend that it should be thoroughly tested on the internal Odin-Vision dataset (Dataset 3) prior to adopting more widely.

3 Methodology

This section contains an in-depth description of the techniques and metrics used during the study group and throughout the remainder of the report. The chapter is split into the three workflows used to tackle the challenge objectives. We describe the methodology used to provide uncertainty quantification in Section 3.1, attribution methods in Section 3.2, and finally representation learning methods in Section 3.3.

3.1 Uncertainty Quantification

As discussed in Section 1.3, a major hindrance to Odin-Vision's current neural network capability is that they are based upon standard deep learning tools that do not provide a measure of predictive uncertainty or model confidence with the classification prediction. Often in classification settings, predictive probabilities obtained at the end of the pipeline (the softmax output) are erroneously interpreted as model confidence. However, a model can be uncertain in its predictions even with a high softmax output.

Here we describe a probabilistic method, based on MC Dropout, to approximate a Bayesian posterior predictive distribution over the predicted class label, and provide extra information to the clinician.

Probabilistic models

A quantification of the model uncertainty in deep learning settings can be achieved if the neural network produces a distribution of output predictions rather than a single value. Bayesian Neural Networks (e.g., [35]) are a class of deep learning models that satisfies this requirement by computing a posterior distribution of predictions, $p(\hat{y}|\mathbf{x}, \theta)$, given an input example image $\mathbf{x}$ and model parameters θ.

Inference on a full Bayesian NN can be approximated using the Monte Carlo Dropout methodology [1]. Briefly, standard dropout [36] consists of setting a random selection (a fixed proportion) of the network weights to zero at each iteration during the training of the network. This discouraging any single neuron to heavily rely on a particular configuration in the previous layers. This is well known to improve generalisation capabilities of the network.

Monte Carlo dropout is an extension for computing model uncertainty at inference time. The use of dropout can be interpreted as a Bayesian approximation of a well known probabilistic model: the Gaussian process (GP) [37]. This interpretation suggests that dropout approximately integrates over the models' weights [1].

This is performed by providing an example image multiple times into the network at inference time. Given the random nature of Dropout, a different selection of neurons is set to zero during each forward pass, resulting in a distribution of output predictions. Further, each forward pass is independent and so this process can be performed by parallel computations for improved efficiency.

Quantifying uncertainty

We first quantify the confidence with which the model makes a decision by following the approach adopted in [38, 39]. In the context of a probabilistic methodology such as MC Dropout, we provide

the prediction $\hat{y}$ that is now computed as the value where the output distribution over possible y's is maximised, i.e. $\hat{y} = \mathrm{argmax}\ \mathrm{p_y}(\mathrm{y}|\mathbf{x}, \theta)$.

Intuitively, high values of $\hat{y}$ are associated with high confidence. Given a confidence threshold τ, the model performance can thus be evaluated in terms of Accuracy, Precision and Recall for subsets of test data classified at least with confidence τ. See Section 4.1 for our results in this area.

On the other hand, an alternate estimation of the model uncertainty for a given input example is given by the variance of the output distribution $p(y|\mathbf{x}, \theta)$, which we can approximate using the standard deviation of the multiple predictions. Moreover, once $\hat{y}$ is computed following the approach highlighted above, the entropy [40] of the predictive distribution (see eq. 1) is also used as a measure/metric of certainty [41]. In Section 4.1 we evaluate uncertainty using confidence, variance and predictive entropy.

3.2 Attribution Methods

Attribution methods (AMs) aim to improve the explainability of a model by assigning scalar values to the different factors contributing to a model's prediction. These scalar values, usually referred to as attributions, reflect the influence of each factor in the final prediction. Given the exploratory nature of this challenge, we explore many different AMs and classify the approaches according to whether they base their outcomes on:

1. The input features (primary attributions)
2. Any other model component, such as a specific layer etc.

AMs can be further classified into **gradient-based** or **perturbation-based**, depending on whether they obtain the attributions via the model's gradient or by assessing the relative changes in the model's prediction when different input features are perturbed.

We used the Captum [42] library, built on PyTorch [12], in this challenge. It provides a built-in implementation of many of these methods. Below, we give a technical overview of these methods, and summarise in Table 8.

3.2.1 Primary attribution methods

i) Perturbation-based methods

As mentioned above, perturbation-based AMs produce their attributions by comparing changes in the model's prediction when different input features are perturbed. Consequently, perturbation-based AMs do not depend on the implementation of the model used to compute the prediction or even the model itself. In fact, perturbation-based AMs can be applied to models other than neural networks, such as decision trees or SVMs. For this reason, they were an ideal starting point for our exploration.

Occlusion methods: Occlusion-based methods calculates attributions as the change in the model's prediction caused by occluding a single feature (one pixel in the case of image inputs) or a group of features. By occluding the features one by one, a very high level of granularity is achieved in the attributions. However, this approach only characterises independent attributions. On the other hand, occluding several features at a time reduces the levels of granularity but allows us to capture the complex correlations among features.

In order to provide attributions that cover the whole input, a model prediction needs to be computed for each occluded input feature(s). In order to reduce the computation time, larger occlusion patches can be used. Choosing the size of the occlusion patch is therefore a trade-off between the desired level of detail in the attributions and the computation times.

Much like the size of the occluding patches, the choice of a baseline value also presents some challenges (see Section 4.2.3). A naive choice of baseline (e.g. black baseline) can lead to occluded inputs that no longer follow the same data distribution as the training data set, hence invalidating the model's predictions. One approach to mitigate the 'distribution shift' effect is to simply blur out the patch to be occluded [43]. Although there are more efficient approaches, which involve inferring the information in the occluded patch based on the values of the neighbouring features, these approaches are costly and more difficult to implement.

ii) Gradient-based methods

For the next sections, we consider a function $F : \mathbb{R}^n \to [0, 1]$ that represents a neural network, and an input $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$, where n is the number of features/pixels.

Saliency, Deconvolution and Guided Backpropagation: While these three methods have been developed independently, they all share the same rationale. Out of the three, Saliency is the simplest: each attribution ϕ_i is computed as a backward pass through the network, i.e. as the gradient of the output with respect to an input feature: $\frac{\partial F(\mathrm{x})}{x_i}$, for $i = 1, ..., n$.

The magnitude of these attributions can therefore be interpreted as a measure of the amount of change in the input features needed to affect the model's prediction [44]. Saliency applies the usual backpropagation procedure, where gradients are set to zero if, when propagating through ReLU activations, the input data in the forward pass was negative.

Deconvolution and Guided Backpropagation are equivalent to Saliency in the sense that ϕ_i are also computed as a backward pass through the network, but with small modifications. In Deconvolution, ReLU activations only mask gradients if their value is negative, regardless of the sign of the input data in the forward pass. Guided Backpropagation is a combination of Saliency and Deconvolution: for example, when backpropagating through a ReLU activation, either the input data in the forward pass or the input gradient in the backward pass are negative, the value of the gradient is set to zero [45].

Integrated Gradients and GradientSHAP: Consider now a baseline input $\mathbf{x}'$ and the straight-line path in $\mathbb{R}^n$ from $\mathbf{x}'$ to the input $\mathbf{x}$. The attributions given by Integrated Gradients correspond to integrating the gradient of F with respect to an input feature along this path:

$$\phi_i = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' - \alpha(x - x'))}{\partial x_i} d\alpha. \tag{2}$$

Integrated Gradients satisfies the sensitivity axiom, whereby for every input and baseline that differ in one feature but have different predictions, there is a non-zero attribution associated with that feature. The method also satisfies the axiom of implementation invariance, i.e. two equivalent networks that can be represented with the same function F should have the same attributions for a given input, despite having different implementations. More details can be found in the original paper [46].

GradientSHAP [47] is similar to Integrated Gradients in that it also requires to consider a baseline $\mathbf{x}'$ and the linear path between the baseline and the input image. To compute the attributions we need

the quantity $(\mathbf{x}-\mathbf{x}'')\times\frac{\partial\mathbf{F}}{\partial\mathbf{x}}$, where now $\mathbf{x}$ and $\mathbf{x}''$ are treated as random variables. The random input $\mathbf{x}$ is equal to the original input plus white noise, the new baseline $\mathbf{x}''$ is obtained by first, choosing a baseline image $\mathbf{x}'$ from a randomly generated distribution of baselines and then, choosing a random value α between 0 and 1 so that $\mathbf{x}'' = \mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}')$.

The attributions ϕ_i are computed as the expected value of $(\mathrm{x}-\mathrm{x}'')\times\frac{\partial\mathrm{F}}{\partial\mathrm{x}}$, which can be approximated by averaging over multiple random samples of this quantity.

Guided GradCAM: The attributions of Guided GradCAM are based on those corresponding to another two methods: GradCAM and Guided Backpropagation. The GradCAM attributions ψ_j^k for layer k, which is usually chosen to be the last layer of the model, are based on the feature map activations A^k of layer k and the gradients of F with respect to these activations [48]. The number of GradCAM activations is not equal to the number of input features, but to the number of activation units in layer k. While GradCAM is able to localise the most relevant regions of input features, it lacks the ability to highlight fine-grained details like pixels in the case of image inputs. The authors of GradCAM propose Guided GradCAM as a solution to this problem. The final attributions ϕ_k are obtained by upsampling the GradCAM attributions so they match the dimensions of the input and then, performing an element-wise multiplication with the attributions from Guided Backpropagation.

3.2.2 Layer attribution Methods

While primary attribution methods evaluate the contribution of each input feature to the output of a model, layer attribution methods estimate the contribution of each layer to the output of the model, hence providing further information about the model's behaviour [42, 49].

As described in Section 1.3, Odin-Vision's existing model is based on a resnet101 model. A resnet101 model is composed of multiple "residual" blocks. Each block consists of three sets of convolution operations with varying numbers of kernels, strides and sizes each followed by a BatchNorm operation and finally a non-linear ReLU operation with a residual connection. For full details we refer the reader to the original resnet paper [11]. A resnet101 has 33 of these blocks organised into four "layers". The first layer has 3 blocks, the second 4, the third 23 and the fourth has 3. We explored the attribution outputs for all of these 33 layers but we present our results here as the attributions after each of these four layers for display purposes. We evaluated the use of layer integrated gradient, layer GradientSHAP and both guided and layer GradCAM.

Layer Integrated Gradient. Layer Integrated Gradients is a variant of Integrated Gradients (described in Section 3.2.1), that assigns an importance score to layer inputs or outputs. This relies on whether we attribute to the former or to the latter. Similarly, layer GradientSHAP is the analogue of GradientSHAP applied to a particular layer in the network. As an overview, it adds Gaussian noise to each input sample multiple times, then chooses a random point on the path between the baseline and input image, and calculates the gradient of the output for the identified layer [46, 50]. We show that it produces similar results to Layer Integrated Gradient.

Layer and Guided GradCAM. As explained previously, GradCAM is a class activation maps (or CAMs), i.e. it is class discriminative and solely highlights the regions that contribute to a class prediction. Traditional CAMs can only be used by a small class of ConvNets or those without densely connected layers, directly passing forward the convolutional feature maps to the output layer [42, 49].

Layer GradCAM calculates the gradients of the target output for the provided layer, aggregates for

each output channel (dimension 2 of output), and multiplies the average gradient for every channel by the layer activations. The results are summed over all channels. GradCAM attributions are usually upsampled and can be seen as a mask to the input since a convolutional layer output equals the input image spatially. This upsampling can be achieved using interpolation [42, 50].

On the other hand, Guided GradCAM, as described above, can also be computed on a specific layer. It shows specifically where the activations within the pixel of the image happen (see Section 4.2.1 for our results on this area). The structural parts of a convolutional neural network are its filters. These filters can recognise from simple features to more complex features as we go up the convolutional layer stack [42, 50, 51].

Algorithm	Type	Model Passes	Requires Baseline
Integrated Gradients	Gradient	Forward / Backward	Yes (Single Baseline Per Input)
GradientSHAP	Gradient	Forward / Backward	Yes (Multiple Baselines Per Input)
Deconvolution	Gradient	Forward / Backward	No
Saliency	Gradient	Forward / Backward	No
Guided BackProp	Gradient	Forward / Backward	No
Guided GradCam	Gradient	Forward / Backward	No
Occlusion	Perturbation	Forward	Yes (usually, zero baseline is used)

Table 8: A comparison between the various attribution methods used in the project. This comparison is taken directly from the documentation of Captum [42, 52]

3.2.3 Attribution performance metrics

It is a desirable quality of AMs that, if the input is modified to the extent that it significantly affects the model's prediction, this should also be manifested through changes in the attributions. However, if a modification of the input does not affect the prediction, the attributions should remain consistent.

In [53] the authors propose a metric for how "good" an AM performs, to attempt to capture this notion of consistency. The metric, called infidelity, requires performing a perturbation of the original input. It is defined as the mean square error between i) the dot product of the modified input and its corresponding attributions and ii) the change in the prediction for the original and the modified inputs. This definition allows for a variety of modified inputs but here, we opt for a perturbation that consists of adding Gaussian noise to the original input.

With this definition, we believe that AMs that minimise the infidelity metric will be more faithful to the predictive model. In this context, the idea of faithfulness corresponds to an adequate amount of change in the attributions when two inputs that produce significantly different predictions are considered.

3.3 Representation Learning

In the previous two sections, we have described methods we explored to better communicate model predictions to clinicians. The aim is to provide more interpretation and move away from "black-box" prediction models, and in turn build trust in the models. This third workflow based on representation learning is more exploratory, and not directly intended to present directly to clinicians. We are interested in discovering new optical features derived directly from image data to aid understanding of the models themselves.

Currently, the most widely used optical diagnosis criterion, the NBI International Colorectal Endoscopic (NICE) classification, describes the real-time classification of hyperplastic and adenomatous polyps using three hand-crafted image features: 1) polyp colour, 2) vessel structure and 3) surface pattern. However, in the aforementioned clinical trial [7], Rees et al found that 35% of adenomatous polyps did not exhibit any of these features.

With appropriate models, optical features can be derived directly from the polyp image datasets without any labels of the polyp class. This is advantageous on two fronts: 1) since acquiring accurate class labels is often an expensive and labour intensive task; 2) data-driven feature representations can lead to new understanding of the polyp images and may provide insights to improve the current diagnosis procedure, beyond the current state-of-the-art.

We focus on deriving a set of important features using a generative model, called variational autoencoders (VAEs) [54, 55]. In this section, we will introduce VAEs and why it can discover interesting features in the data. We will also explain how to interpret the derived features once we have trained a VAE model for a given dataset. Amongst different types of generative models including GANs [56], flow models [57, 58] and autoregressive models [59], VAEs have been favoured for their training stability, strong theoretical grounding and ability to learn well-structured latent representations.

3.3.1 Variational autoencoders (VAEs)

Variational autoencoders (VAEs) are probabilistic latent variable models that learn a low dimensional representation of some complex, often high dimensional data in an unsupervised manner. As shown in Figure 9, a VAE consists of two parts: 1) a generative decoder that generates a data sample from a latent code (a feature vector) and 2) a variational encoder that maps a data sample to an approximate posterior distribution over latent variables. The choice of the specific function to use to parameterise the encoder and decoder module is quite flexible. In most of the machine learning contexts, the encoder and the decoder are parameterised by neural networks for their capacity to approximate any complex functions.

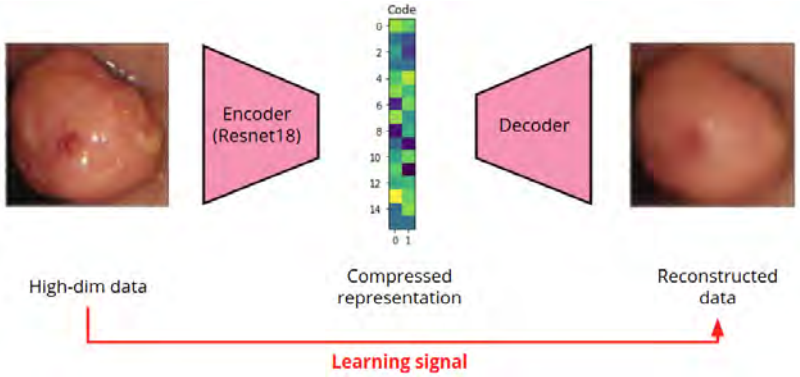


Figure 9: VAE concept sketch: A VAE utilises an encoder to map the high dimensional input image (left) to a low dimensional latent code (centre). A decoder then attempts to reconstruct the original image from the latent code alone (right). A VAE is trained under a regularised reconstruction error.

To train a VAE, we optimise an evidence lower bound (ELBO) loss, defined in Equation 3 below, with respect to the parameters of the encoder (ϕ) and decoder (θ). The ELBO loss consists of two

components: ① a reconstruction likelihood that encourages good reconstruction through the auto-encoding process, ② a regulariser that requires the encoding distributions $q_\phi(\mathbf{z}|\mathbf{x})$ stay close to a prior distribution $p(\mathbf{z})$, which is often specified as a unit Gaussian distribution as an uninformative prior. These two terms together act as a regularised reconstruction error objective. Because of the regularisation in ②, the derived encodings are confined to a local region around the zero latent code, rather than being scattered around in the entire latent space as a normal autoencoder would do (whose learning objective does not include the regularisation term). Hence, VAEs are more likely to provide a smooth latent space than a normal autoencoder.

As the ② term has the effect of confining the encoding distribution $q_\phi(\mathbf{z}|\mathbf{x})$, we can derive disentangled encodings by choosing the prior $p(\mathbf{z})$ to be statistically independent across its dimensions, such as a diagonal Gaussian. To be clear, here we define disentanglement as statistically independent, i.e. two factors are disentangled if there is no (or little) correlations between the them. As the disentanglement is resulted from the prior constraint specified in the ② term, a natural question arises: is it possible to encourage stronger disentanglement if we scale up the impact of the regularisation ② term?

This idea was first proposed in [60] where a hyper-parameter β is introduced in the ELBO loss, as shown in Equation 4. Having a large β has the effect of favouring the prior constraint more than the objective to purely minimise the reconstruction error. As a result, β becomes a control parameter that balances the two often conflicting objectives of ① and ②. More recent paper [61] has shown that a smaller β, on the other hand, can have a beneficial effect in preserving the information stored in the original input data $\mathbf{x}$, i.e. minimising the distortion resulted from the compression process in the autoencoder. As a result, the optimal value of β depends on the task at hand. In this project, we have some initial exploration on the impact of β in the derived latent factors. We demonstrate the results in Section 4.4.

$$\mathcal{L}(\mathbf{x};\theta,\phi) \triangleq \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\big[\ \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{① reconstruction likelihood}} - \underbrace{D_{\mathrm{KL}}[\,q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]}_{\text{② prior constraint}}\big] \tag{3}$$

$$\mathcal{L}(\mathbf{x};\beta,\theta,\phi) \triangleq \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\big[\ \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{① reconstruction likelihood}} - \underbrace{\beta}_{\text{③}}\underbrace{D_{\mathrm{KL}}[\,q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]}_{\text{② prior constraint}}\big] \tag{4}$$

3.3.2 Latent space interpolation

Once a VAE is trained on a given dataset, we have a mapping between an input image, a low-dimensional latent feature vector, and its reconstructed image. We then want to find out what each dimension of the latent features actually represent.

To understand the latent code, we utilise a latent space interpolation scheme, as in [60]. The procedure can be summarised as follows. We focus on one latent feature dimension at a time. Starting at latent dimension i, we replace the corresponding latent feature value z_i with a number in the range of -3 to 3, while keeping the rest of the latent features unchanged. This gives us a perturbed latent code. Then we "decode" the perturb latent code using the trained decoder and observe any changes in the reconstructed image.

An example of such procedure is shown in Figure 10. Here we examine the information encoded in the first latent dimension z_1. We replace the original latent feature value -1.19 with a number

equidistant in the range -3 and 3. The ten corresponding changes are shown in the decoded images at the bottom of the figure. As you can see, this latent feature seems to control the shape (changing from a round polyp with clear boundary to a blob that submerges into the background tissue) and the texture (changing from a smooth surface to a less smooth surface). We present our latent space interpolation result in Section 4.4.3.

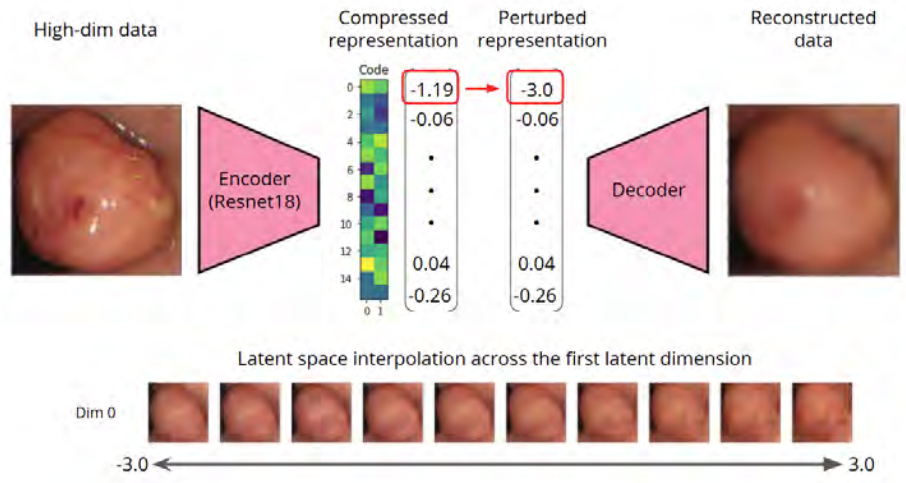


Figure 10: Latent space interpolation demo: To understand what information each latent factor encodes, we perturb the feature value of a single latent dimension at a time and observe the decoded image of the perturbed latent code. For example, here we perturb the first latent feature -1.19 and replace it with a equi-distant values in the range of -3 to 3. The corresponding changes in the image space are shown in the bottom row.

4 Experiments & Results

In the following we present all the experiments and results obtained during the challenge across the three workstreams. The outline of this section is as follows:

In Section 4.1 we discuss the results of the uncertainty quantification workstream. We initially developed methods using the MNIST dataset as a proof-of-concept (see Section 4.1.1) and then applied our findings to the colorectal polyp image datasets (trained in-house by Odin-Vision). We evaluate on the Piccolo and Depeca datasets in Section 4.1.2. Experiments with a thorough quantification of model uncertainty are presented in Section 4.1.3.

Experiments with the various attribution methods are presented in Section 4.2, here again we use the publicly available Piccolo dataset to provide insights into the possible polyps structure, or image features, that are highly influential for the model classification.

Finally, in Section 4.4 we present our investigations into automatic feature representations. First we present the training strategy of the VAE in Section 4.4.1. We query the latent space interpolation in Section 4.4.3, and provide reconstructed polyp images from varying individual latent codes.

4.1 Uncertainty Quantification Experiments & Results

4.1.1 Proof-of-concept: Reporting Uncertainty on MNIST

As an initial proof-of-concept we modified the standard MNIST dataset into a binary classification task, by only using images representing the numbers one and seven. A ResNet18 model was trained using similar input architecture to the Odin-Vision provided model (see Section 1.3). We added a dropout layer in the final layer of the network to obtain a probability distribution over the output prediction (scalar). We provide some example predictions on test-set images in Figure 11 (top), where a good predictive model is achieved, and additionally very little variance in the predicted probability distributions is demonstrated. This is as expected since this is an easy task where the model is certain about every prediction.

In order to test the model's ability to provide uncertainty, out of distribution data (OOD) was used. This means that other digits of the MNIST dataset were used as test examples. Digits that the algorithm has not seen during training.

We present the distribution of probabilities predicted for 10 randomly selected OOD images in Figure 11 (bottom). Ideally, as these are out of distribution data, the figure would show results with high variance and hence a low level of confidence, with class determination very close to the 0.5 probability threshold, indicating a quasi-random determination. This is indeed what we find for some of the test samples (seen in column 2, 3 and 9 for digits zero, four and six respectively). However, we find that the model classifies the other test samples with a fairly high level of confidence as indicated by the low variance, i.e. each test image is predicted as class zero or class one with high probability.

Thus, the proof of principle model on the MNIST dataset shows MC dropout layers can approximate probabilistic modelling, with the distribution of outputs aiding in the understanding of model certainty. The model performed well on the test dataset, showing near negligible variance in output probabilities and correct classification. For the OOD dataset, the limitations of the model becomes visible, where incorrect classification is made with high levels of certainty. Further

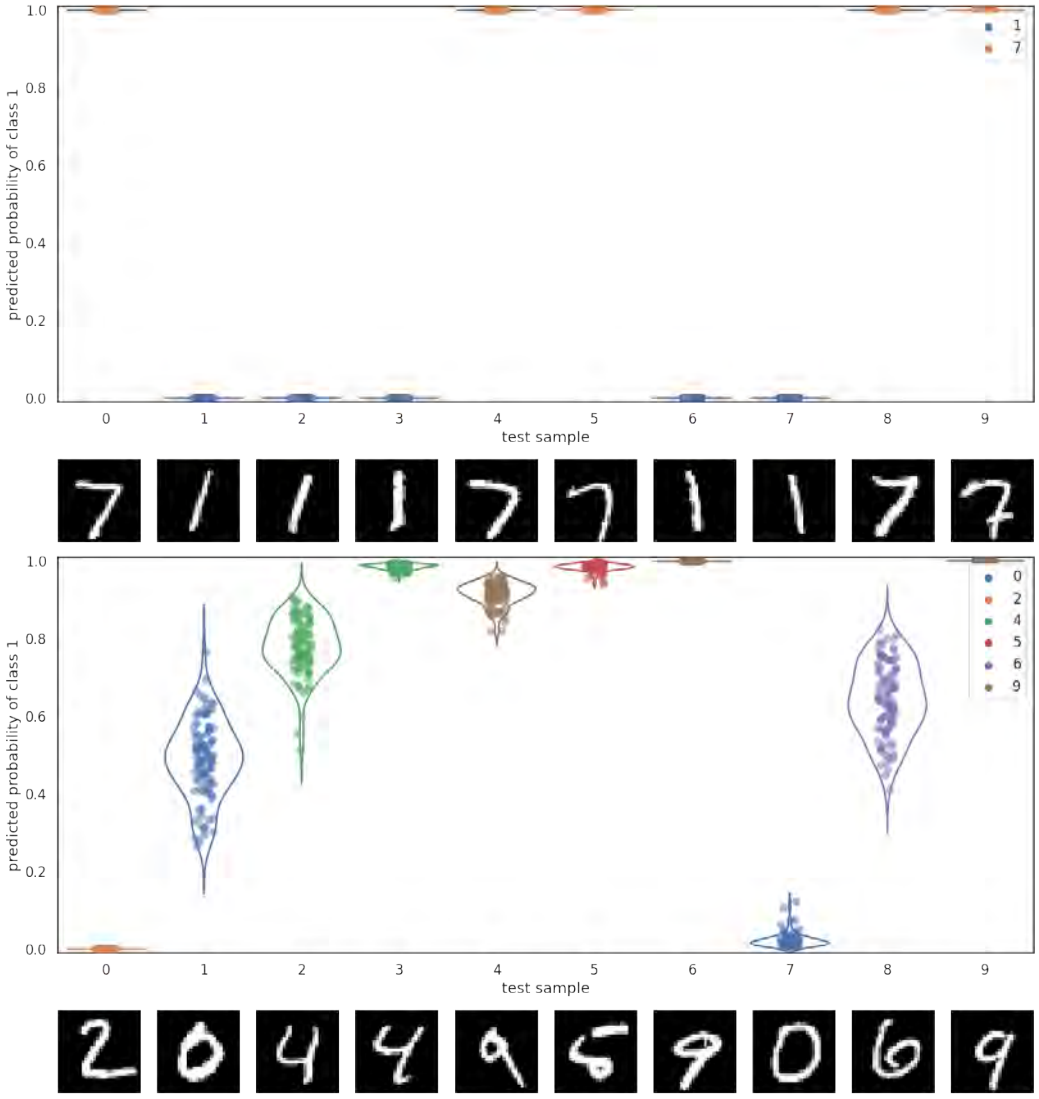


Figure 11: Distribution of predictions for sample images in the binary MNIST test set (top). Distribution of predictions for sample out-of-distribution images (bottom).

training and parameter tuning is needed to improve the performance of the model, but provides a suitable basis for a proof of concept experiment.

4.1.2 Reporting Uncertainty on Polyp Image Data

In this section we define multiple model architectures in order to investigate the performance of the proposed MC Dropout method. We first describe the architecture choices and provide a qualitative view of how the network and dropout performs. In the subsequent section, Section 4.1.3, we provide a quantitative summary of the different architectures.

The models presented in this section were trained on the internal Odin-Vision dataset (Dataset 3). In order to benchmark them, and to provide images for this report, we used the publicly available Piccolo dataset as a test set (see Section 2 for more details on datasets).

Architectures

Since Odin-Vision's current ML capability is based on a ResNet101 model, we also used this as a backbone to train three parent architectures by adding sequential fully connected layers after the ResNet features:

- *OneDrop*: with a single additional fully connected layer with dropout (code shown in Listing 1);
- *ThreeDrop*: with three additional fully connected layers as shown in code Listing 2;
- *Resnet-VAE* with two additional layers as shown in code Listing 3, to mimic the VAE representation investigated in Section 4.4.1, i.e. the final fully connected layer matches the dimensionality of the latent code used in our VAE implementation.

Each of the above models were trained with two different values of Dropout, 0.2 and 0.5. We found a dropout value of 0.2 worked best, and is therefore used for the remainder of this section. In line with the literature on this method, preliminary studies seemed to be largely insensitive to choice of dropout fraction [1].

The code listings for these architectures are provided in Appendix A.

Qualitative Findings

As a qualitative example, we discuss the results of two test images shown in Figure 12, using the OneDrop network. The test images are provided, along with a panel showing the ensemble prediction results, from running multiple (in this instance 20) forward passes of the test image through the neural network with dropout enabled at evaluation time. In the figure, the right hand panel shows along the x-axis the predicted softmax probability for the binary classes (1 = adenoma, 0 = non-adenoma).

Without any calibration we can make the simple assumption that any network prediction greater than 0.5 is identifying the image as adenoma, and otherwise is predicted to be non-adenoma. Each individual network sample is shown as an orange check mark along the x-axis. The distribution of these predictions is then estimated using a Gaussian kernel density estimate. This profile is then visualised in blue on the same axes. No units are displayed on the y-axis of this distribution as it is intended for visualization purposes only, to display the shape, or spread, of the 1D distribution of predictions.

As a simple outcome of this project this is the kind of feedback metric that can be intuitively displayed alongside an image taken as a snapshot during an endoscopy. Where a traditional computer vision model would simply quote a value between 0 and 1 as the prediction of the network (analogous to the network mean quoted in the figure heading), by including a spread of predictions and visualising them in this manner, we achieve clear feedback of how certain the model is of its prediction, beyond just how far the mean is away from the 0.5 decision boundary.

We can interpret this as follows. A narrow spread of predictions represents a more confident prediction, giving a sharper peaked profile. Meanwhile a less confident prediction is represented

by a large spread. The spread of the distribution can also be mapped to a single number on a user-defined scale that could be presented as a traffic light system which can provide immediate and easily interpretable feedback to the clinician. These communication methods are further discussed in Section 4.1.4.

This methodology is intended to provide near real-time feedback to the clinician. For reference, a timing estimate to produce 20 samples from a single image on a non-optimised CPU machine is shown in Table 9. This table was generated using a standard D4s_v4 VM from Microsoft Azure Cloud computing suite.

In Figure 13 we show the KDE predicted profiles for the same two test images as above, but using

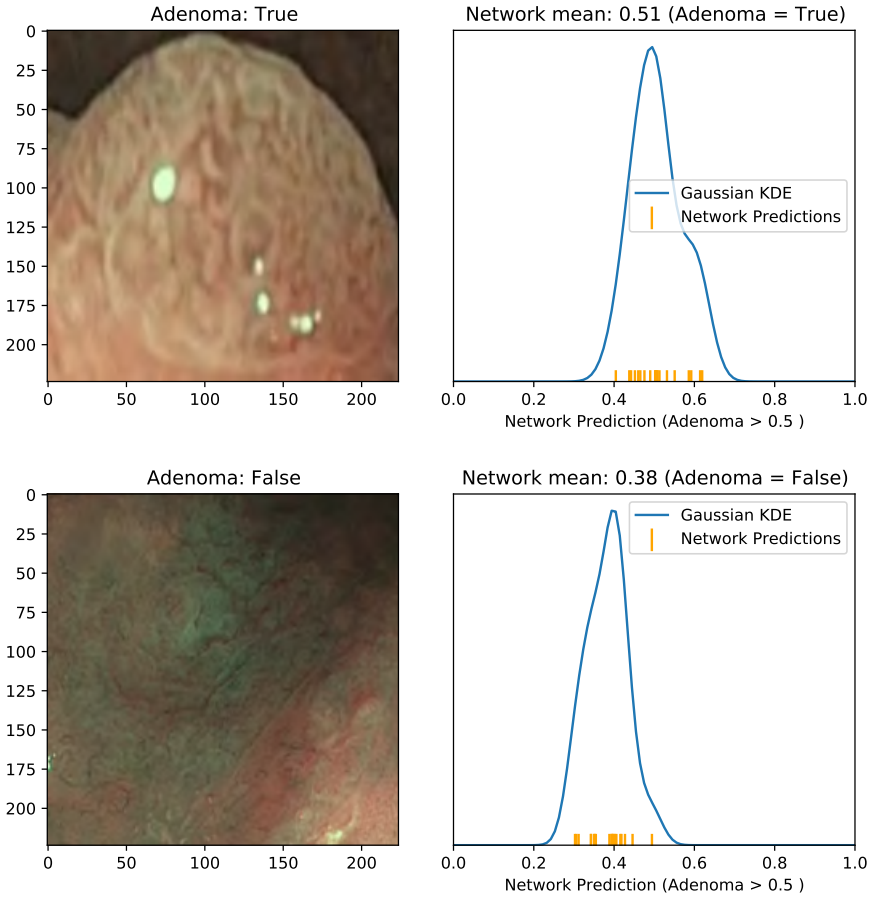


Figure 12: An example predicted output distribution by the OneDrop model for a True Positive test image (left) and a True Negative test image (right). The true class of the test image is shown in the title of the image panel, and the classification resulting from the mean of the model predictions is shown in the title of the profile panel. The orange check marks show the 20 different predictions generated by using MC Dropout. The blue profile is a Gaussian kernel density estimate (KDE) of this distribution.

Network	One Forward pass (s)	20 Dropout samples (s)
ThreeDrop	0.12	2.25
OneDrop	0.12	2.24

Table 9: Average wall-clock computation time for producing samples and single forward passes for the OneDrop and ThreeDrop architectures. All times are taken by running 100 repeats and averaging. All times are taken from non-optimised evaluations on the Turing Safe Haven VM machine (standard D4s_v4 from Microsoft Azure Cloud).

all three different model architectures proposed above (each using dropout = 0.2). In both test cases the deeper ThreeDrop network, with vastly more parameters, produces both more confident (more extreme values near zero and one) and more certain (lower variance) predictions.

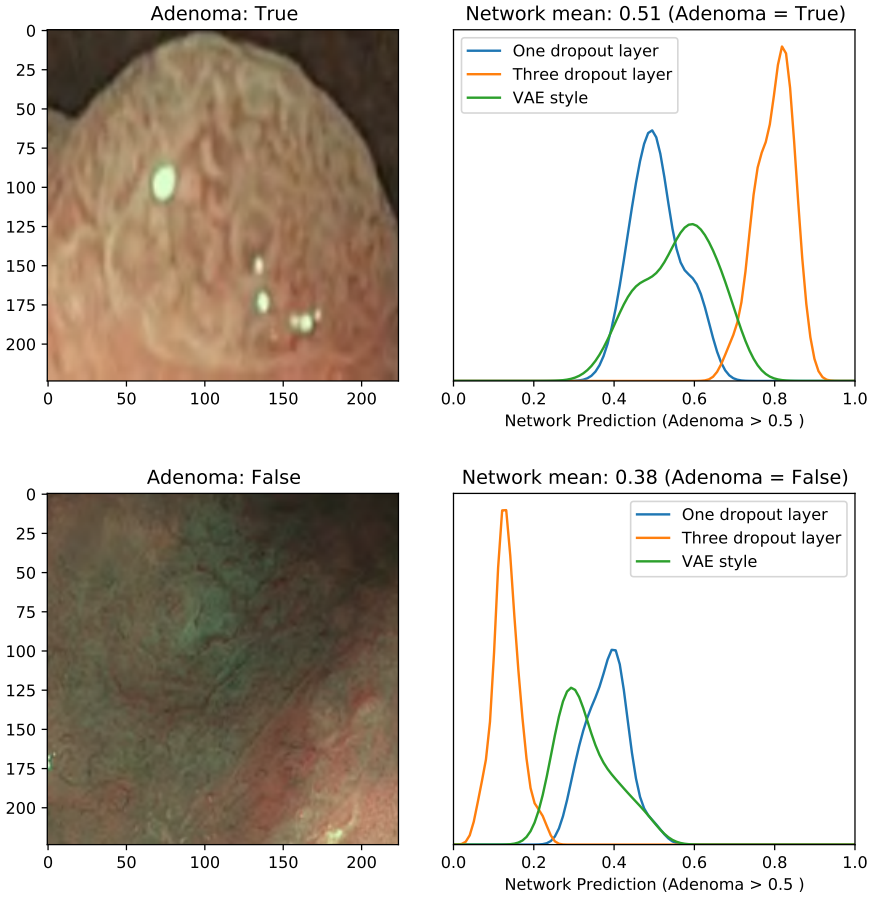


Figure 13: Comparing three different model architectures for the two example test images, a True Positive case (left) and a True Negative case (right). The Network mean and resulting classification in the title are based off the OneDrop (blue profile).

The confidence is intuitively represented as the position of the peak along the x-axis, with more extreme predictions (further from 0.5 decision boundary) being a more confident prediction. The certainty (variance) is intuitively captured in both the width and height of the peak. Since the KDE is a unit area density function, taller and narrower peaks are more certain, whereas wider more spread distributions are less certain. More quantitative statements comparing these models and quantifying more precisely this certainty and confidence are covered in Section 4.1.3.

In Figure 14 we show other example test images of adenoma polyp (top) and non-adenoma polyp (bottom). Here we present the KDE profiles from the three different architectures separated into three separate panels, each displaying their own model mean and classification in their panel title. These two images are taken as a qualitative example to illustrate that although the ThreeDrop network appears to perform well it might be making overconfident predictions. In this case a clinical expert would perhaps think a high confidence is justified, but in both cases for the TP and TN example, the network makes a prediction at the extreme of its range with no variance. This is perhaps to be expected from a model with so many parameters being run on a small dataset, and is indicative of potential overfitting.

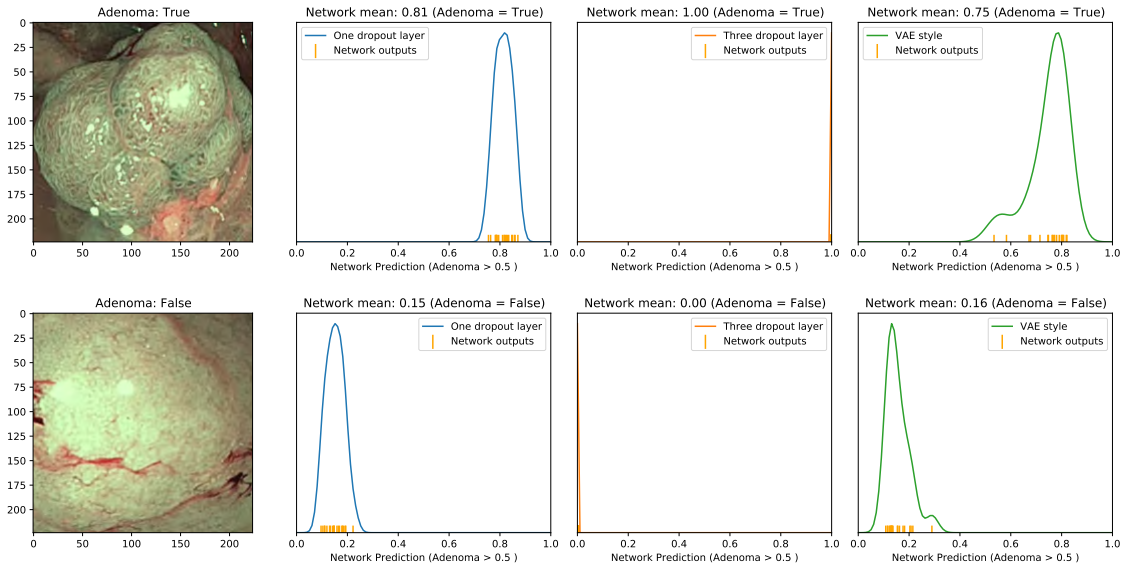


Figure 14: A True Positive and True Negative example test image with each of the three network predictions displayed separately, showing the 20 sample predicted scores and the KDE plot. Note, that the ThreeDrop network makes predictions with almost no uncertainty.

To conclude this qualitative review of the dropout models, Figure 15 displays a set of three False Positive images, with the individual panels for each model as described previously. Figure 16 shows the same setup for three False Negative test examples. Here the decision as to whether this is a False classification is taken based on the mean of the OneDrop network.

We have provided a random selection of examples here, but the miss-classified images are an important aspect of this work. We would hope to demonstrate that any miss-classification could be provided with a high uncertainty score, so not to persuade a clinician in favour of a false classification. In reality there are a number of issues encountered, such as mislabelling or image

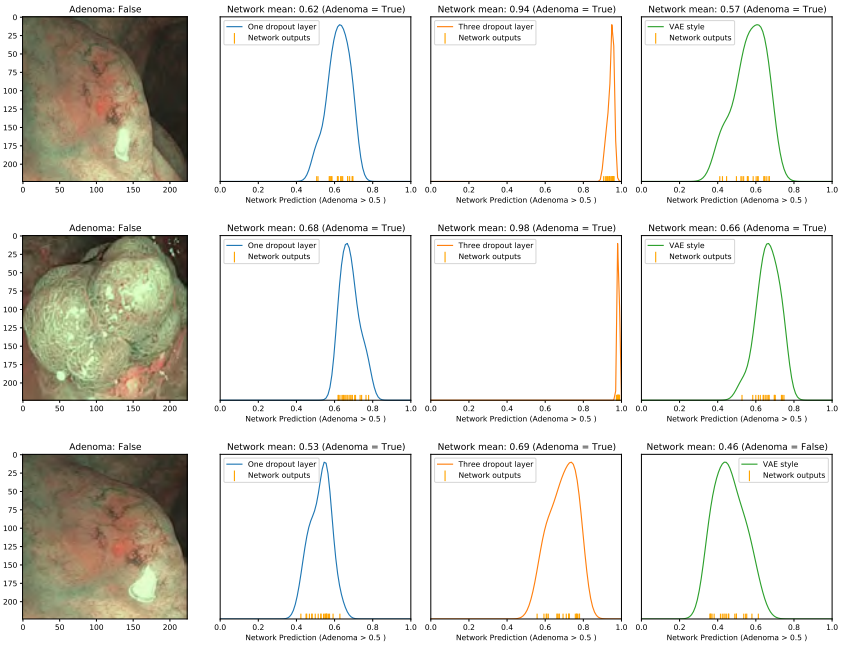


Figure 15: A selection of false positive classified images (based on OneDrop mean classification).

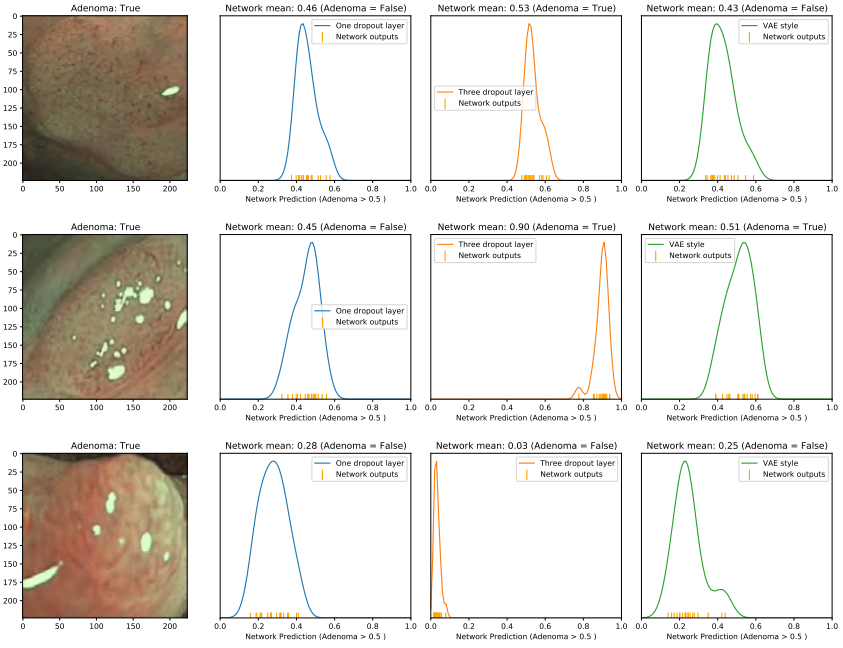


Figure 16: A selection of false negative classified images (based on OneDrop mean classification).

quality still driving network output. In the next section we quantitatively investigate if factors such as image blurriness do indeed drive a higher predictive uncertainty as we would desire. One interesting feature to point out from Figure 15 is that two of these images are sequential shots of the same polyp with similar quality and framing. The fact that the ThreeDrop network displays a very unstable confidence in its prediction, again indicative of an overfitted network.

4.1.3 Quantifying Dropout Model Performance

Classification accuracy

In Table 10, we present three scores each derived from classifying a cleaned subset of the Piccolo dataset, based on the "blur score" described in Section 2.3. In total 900 images form this set of test images, of which 252 are labelled as non-adenoma, and 648 are adenoma. The precision, recall and F1 for this dataset is shown for both classes and the weighted average. Although not reproduced in their entirety here the ThreeDrop model has an average precision and recall of 0.87 and 0.86 respectively. The VAE style model reports 0.85 and 0.82 for these metrics respectively. All models seem to perform reasonably well, and similarly, from this high level perspective. The ThreeDrop architecture seems to increase both precision and recall based on this test dataset. The confusion matrices for the three models are shown in Figure 17. This displays the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) rates for the three architectures.

class	precision	recall	f1-score	# examples
Non-adenoma (0)	0.63	0.86	0.73	252
Adenoma (1)	0.94	0.81	0.87	648
weighted avg	0.85	0.82	0.83	900

Table 10: Classification matrix information, with standard metrics for the OneDrop classifier on the entire selected subset of the Piccolo data used as test. The mean of a sample of 20 predictions is taken as the network classification. In total 900 images are included.

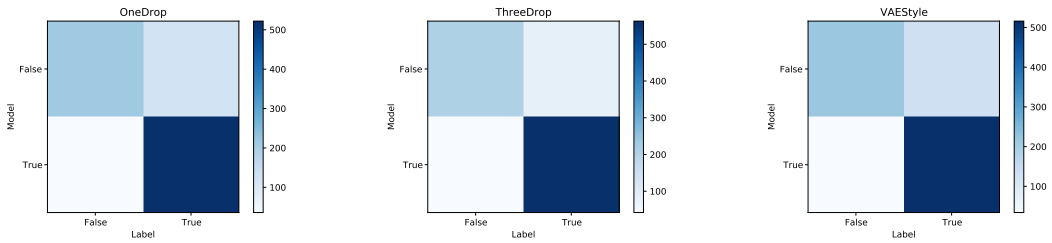


Figure 17: The confusion matrix for the three model architectures explored, run on the entire Piccolo (N=900) image dataset.

Quality of Uncertainty Estimates

To quantitatively assess the quality of the uncertainty estimates, we computed the precision and recall of the model at different uncertainty thresholds on the Piccolo dataset, i.e. removing those

predictions that were not confidently predicted as adenoma (0.5 or higher). This is shown in Figure 18 (top). Ideally, we would expect the performance to increase with increasing confidence, indicating that the model does not make confident false predictions. The performance of the two OneDrop models were similar; both showed increasing precision and recall with increasing confidence. Both achieved relatively good performance even at low confidence level, indicating that they are somewhat under-confident. The performance of the ThreeDrop model was similar but slightly lower at high confidence levels, indicating over-confidence; this is further supported by the high fraction of samples where the model makes extremely confident predictions ($\gtrsim$95% confidence on 60% of the data). The precision of the narrow model is similar to the OneDrop models, however, the recall decreases with increasing confidence. This could be caused by overfitting.

We plot the distribution of the predictive entropy for subsets of sharp and blurry images in Figure 18 (bottom). This provides further insight into the behaviour of the uncertainty estimates. Predictions based on blurry images should be less certain on average than for sharp images, resulting in higher average predictive entropy. This is true for both the OneDrop and the ThreeDrop models, however, the uncertainty of the OneDrop models is higher on average than ThreeDrop. This supports the findings that the ThreeDrop model is somewhat over-confident, while the OneDrop model is slightly under-confident. Since all models achieved relatively high performance even on blurry images, we performed further experiments by artificially blurring data (for the OneDrop models only due to time and computational resource limits). The results are shown in Figure 19).

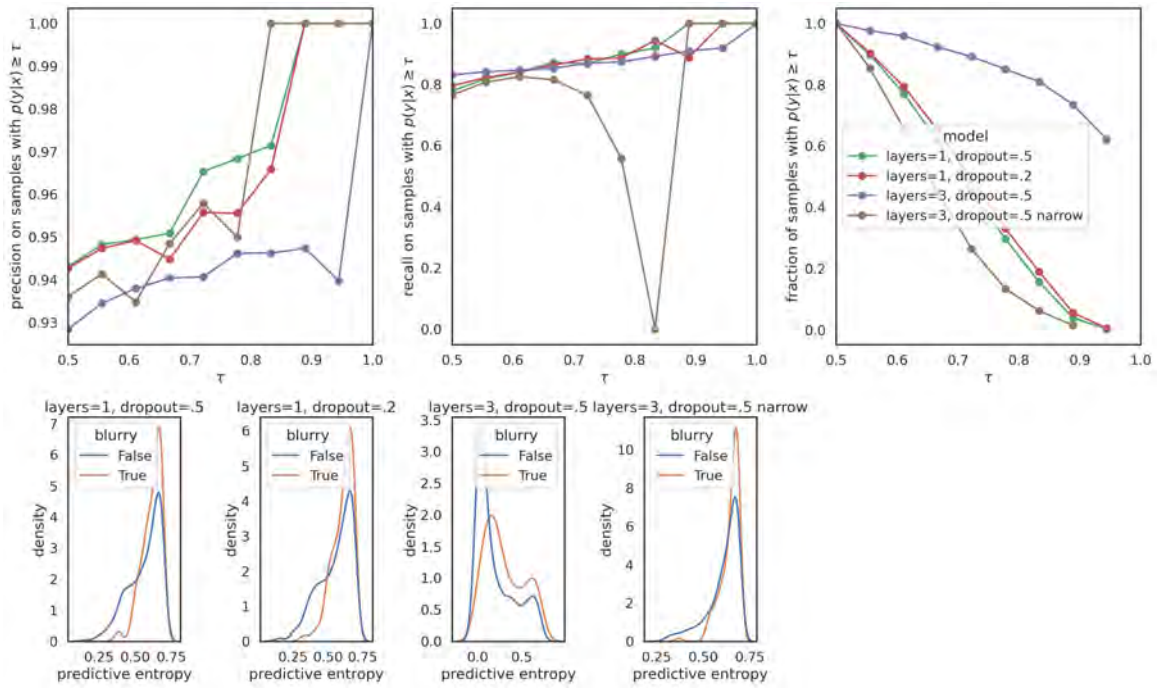


Figure 18: (Top) Model performance at different confidence thresholds τ. For each confidence threshold, we computed the precision, recall and percentage of dataset using the samples for which the model is more confident than the threshold. (Bottom) Distribution of the predictive entropy of each of the models for blurry (orange) and sharp (blue) images.

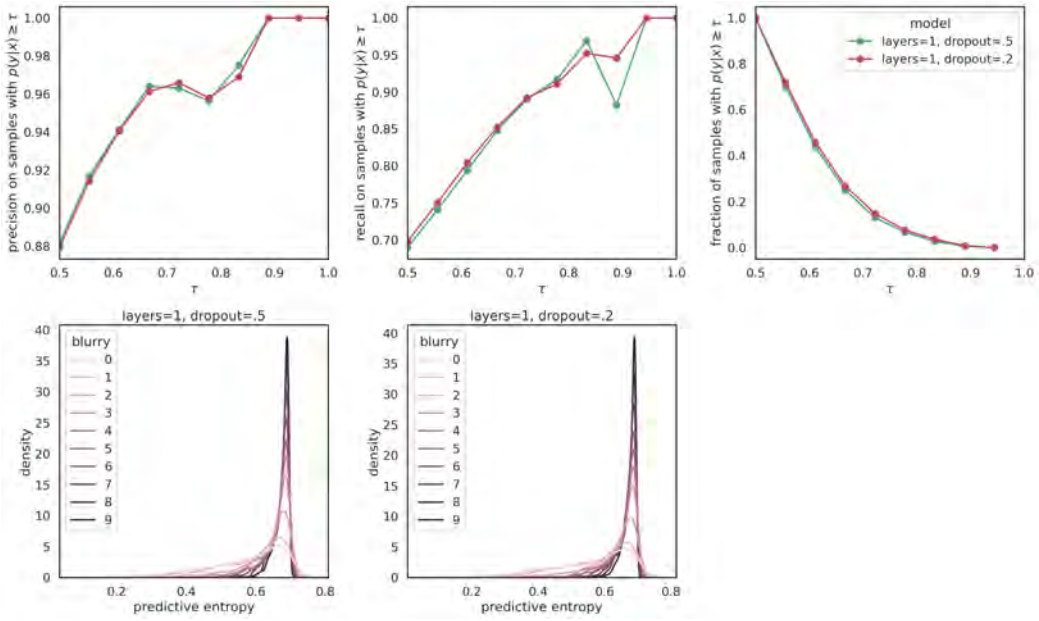


Figure 19: (Top) Performance of the OneDrop models at different confidence thresholds τ for artificially blurred images. Each image in the Piccolo dataset was blurred using a Gaussian filter with varying σ. For each confidence threshold, we computed the precision, recall and percentage of dataset. (Bottom) Distribution of the predictive entropy as a function of blur strength σ.

We further evaluated the quality of the MC dropout uncertainty estimates using the entropy of the correct and incorrect predictions, shown in Figure 20. The OneDrop models show somewhat lower entropy values for correct predictions, being the most confidently correct for true negatives. The ThreeDrop model has very low entropy for true positives and negatives, but also makes some highly confident predictions for false positives and false negatives. Using variance as the measure of uncertainty, Figure 21 further highlights the differences: the OneDrop models have a long-tailed distribution of confidence and variance for the correct predictions, while the ThreeDrop model has a more concentrated distribution for correct predictions, but also a non-trivial proportion of high confidence, low variance incorrect predictions.

The overall picture emerging from the quantitative analysis shows that MC dropout can provide good uncertainty estimates. The number of dropout layers (but not dropout strength) is an important hyperparameter, with the OneDrop models being somewhat under-confident and the ThreeDrop model being over-confident.

Finally, we explored whether rejecting high-uncertainty predictions could lead to improved model performance. We show this in Figures 22 and 23 for the Piccolo and Depeca datasets respectively. We observed an increase in performance with increasing proportion of highest entropy (or variance) samples, e.g. rejecting 20% highest-entropy samples on the Piccolo data leads to 7% increase in recall. Additional experiments on the Depeca dataset with lower quality images indicates that entropy might be a better measure for this purpose. The results here serve as a proof-of-concept, and in real clinical practice the performance should be balanced against the number of rejections to prevent unnecessary repetition of image acquisition by clinicians.

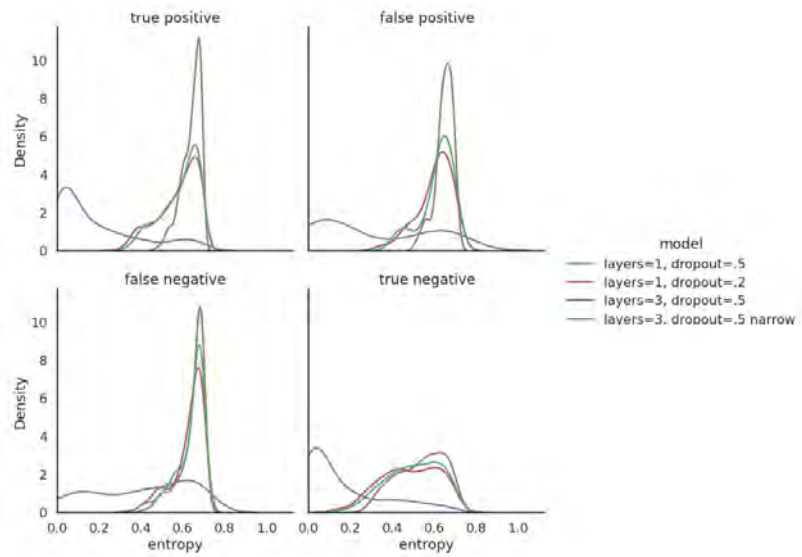


Figure 20: Predictive entropy for true positives, true negatives, false positives and false negatives for each of the models. Rows correspond to model predictions and columns to ground truth labels.

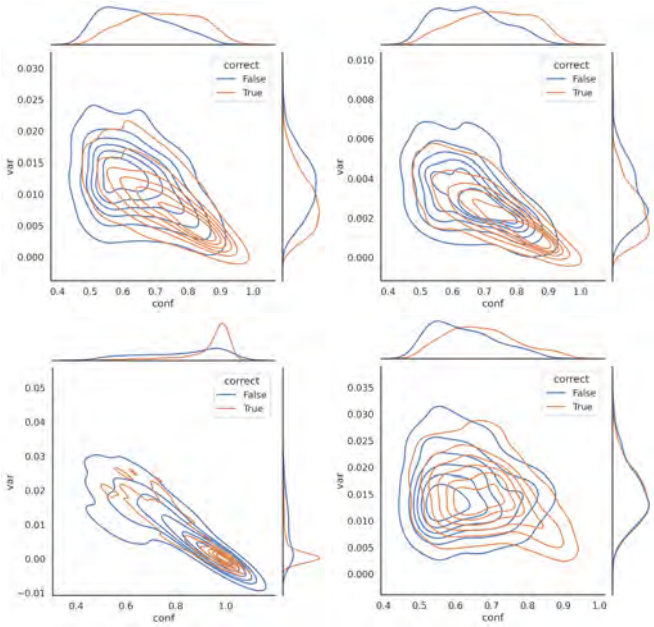


Figure 21: Joint distribution of model confidence and variance for correct (orange) and incorrect (blue) predictions for (clockwise from top): OneDrop model with $p = .5$, OneDrop model with $p = .2$, ThreeDrop model with $p = .5$ and the VAE two-layer architecture.

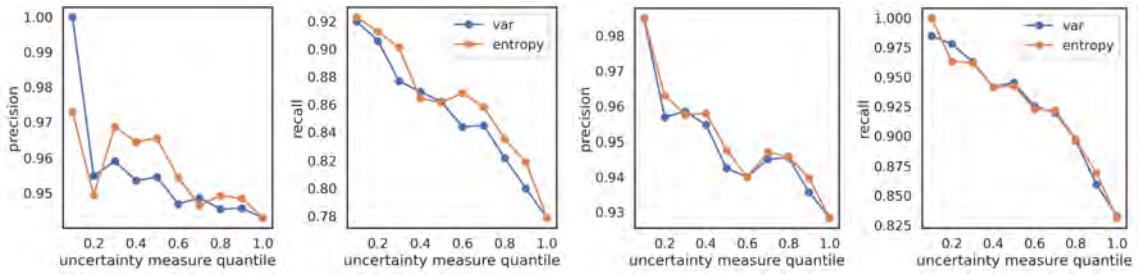


Figure 22: Performance of OneDrop model ($p = .5$) and ThreeDrop model ($p = .5$) on Piccolo dataset. Entropy is shown in orange and variance in blue.

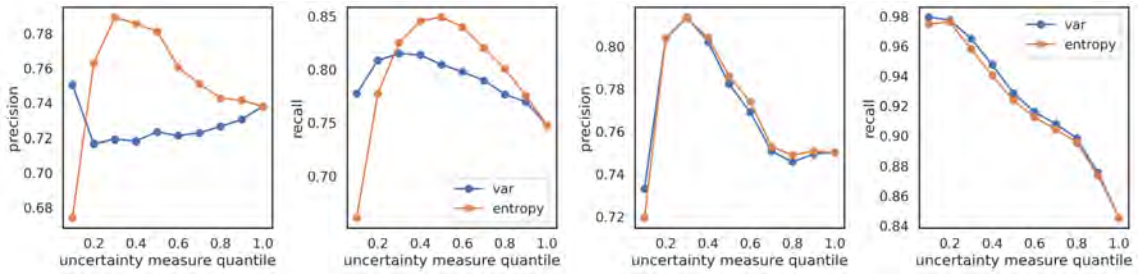


Figure 23: Performance of OneDrop model ($p = .5$) and ThreeDrop model ($p = .5$) on Depeca dataset.

4.1.4 Interpretable Metric: Classify the level of uncertainty

One key aim of the project was to develop an intuitive metric for communicating the certainty/uncertainty of a model prediction to a non-technical audience, e.g. a clinician. We propose one potential metric to classify a model's uncertainty on a user-defined scale of N_{conf} integers (e.g., 1-5, where 1 is very confident and 5 is maximally un-confident). This approach avoids introducing concepts such as the entropy of a predictive distribution to an end user.

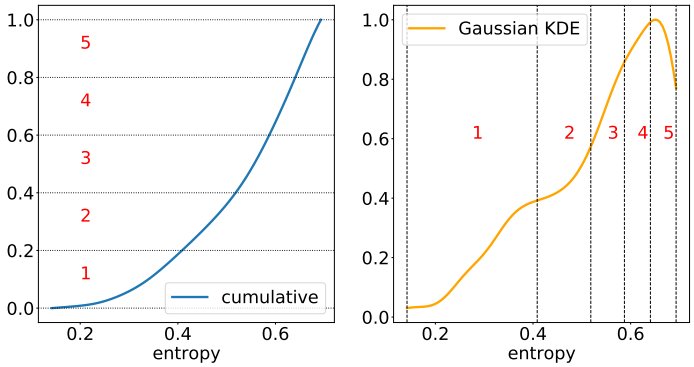


Figure 24: A metric to assign a confidence score to a given model prediction, for $N_{\text{conf}} = 5$ confidence levels. The Piccolo dataset and the OneDrop model (p=0.2) are used in this example.

This approach amounts to numbering/classifying the quantiles of the uncertainty metric -those for example in Figure 22. In Figure 24) we demonstrate the predictive entropy for the Piccolo dataset split into 5 certainty levels. The left panel shows the cumulative distribution function of the entropy distribution over the whole test dataset, and shown on the right as a Gaussian KDE. The horizontal lines in the left panel identify equally probable intervals of the uncertainty metric (entropy of the prediction in this case), which are replicated in the right panel as vertical lines. Low entropy values (certain predictions) correspond to the number 1, while high entropy values correspond to the number 5. In a clinical setting, a quick decision could be made to take a second image in the case of a high uncertainty score.

4.2 Attribution Experiments & Results

4.2.1 Initial Attribution Method Comparison

First, we performed simple sanity checks on various gradient-based and perturbation-based attribution methods. We present this in Figure 25. This figure shows a test image in the first column (lower left), and a normalised version (upper left). The subsequent columns demonstrate the following methods applied to this image: integrated gradient, saliency, deconvolution, Guided GradCAM, guided backpropagation and occlusion. Each method is described in Section 3.2.

Some of the methods investigated are independent of the model weights or the class labels and require a baseline which can significantly affect the heatmap and the activated regions [42, 62]. The difference between the methods in terms of type, baselines and model passes is described in Table 8. We discuss hyper-parameter optimisation, which includes the baseline, in more details in section 4.2.3.

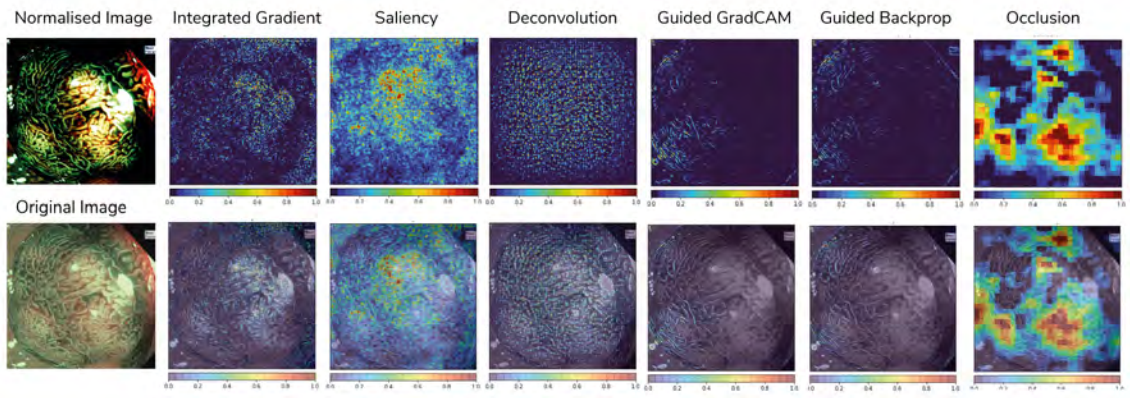


Figure 25: Applying various attribution methods on True positive adenoma polyp image (first column). We compare: Integrated Gradient, Saliency, Deconvolution, Guided Grad-CAM, Guided backpropagation and Occlusion based methods.

After multiple discussions with Odin-Vision clinical researchers, we found that Guided GradCAM, among the investigated methods, highlighted the most informative and meaningful features from the polyp images. In particular it was able to highlight detailed structure of vessels in particular regions of the polyp relevant to clinicians' diagnosis. For this reason we use it in much of the subsequent analysis.

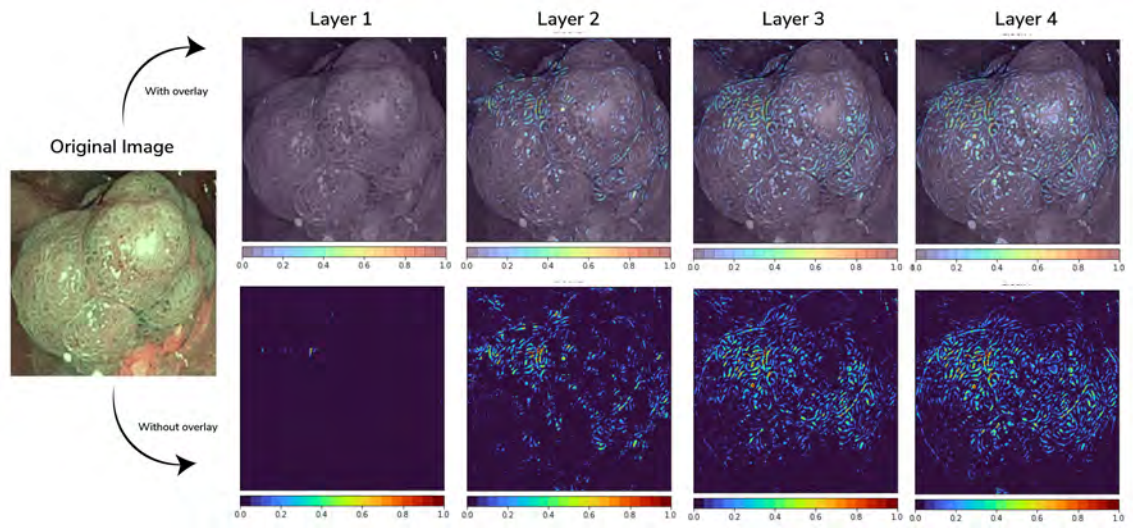


Figure 26: Guided GradCAM applied to the final output of each of the four “layers” of the Odin-Vision resnet101 model as detailed in Section 3.2.2

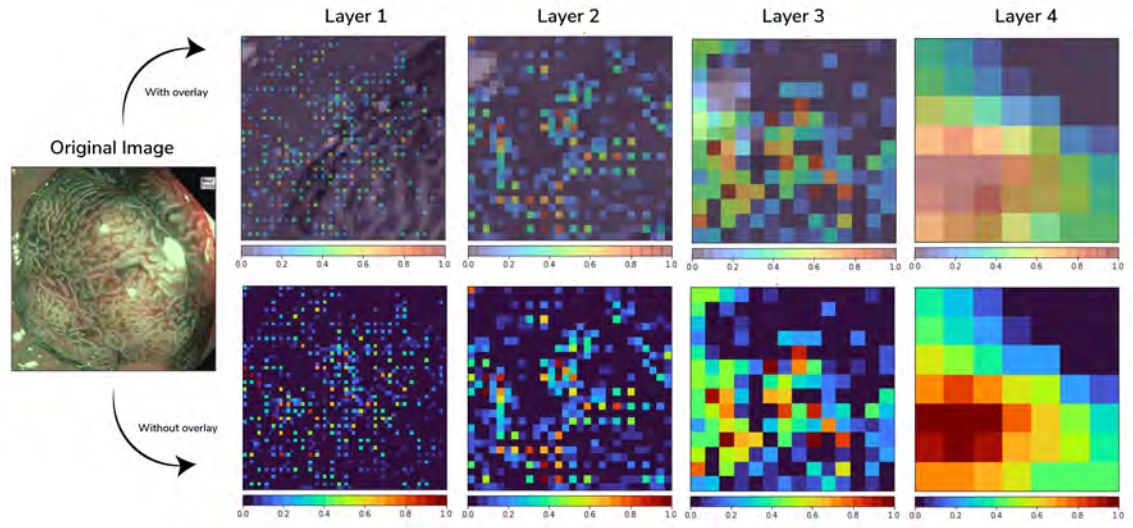


Figure 27: Layer Integrated gradient applied to the output of the four “layers” of resnet101 blocks in the Odin-Vision model.

The methods we compared in Figure 25 are all primary attribution methods that evaluate each input feature’s contribution to the output of the model. However, we can also estimate the contribution of each layer to the output of the model and gain more insight into the behaviour of the model. For this we use layer attribution methods including layer Guided GradCam and layer integrated gradient, as described in Section 3.2. We present the findings in Figures 26 and Figures 27.

We find that GradCAM maps become progressively less defined as we move to earlier convolutional

layers since they have smaller receptive area. This trend was also found in the original GradCAM paper where the most interpretable visualisations were obtained in the deeper layers of the network with worse localistions found in the shallower layers [51]. In the last two layers, the features of the polyps are more defined. While being simple to implement, Guided GradCAM produced sharper and more descriptive visualisations while relatively preserving both the spatial information and high-level features compared to layer GradCam and layer integrated gradient. In addition, GradCAM does not require a baseline input, which is one of the limitations of some of the attribution methods, discussed further in the limitations Section 4.2.3.

Further, we used guided GradCAM to try to detect the vessel structure of the test image polyp as described by the NICE classification. We present one true positive and one false negative in Figure 28.

This analysis also highlighted that the performance of AMs depend on the quality of the input images. As it can be seen on the right panel of Figure 28, the resulting attributions are not relevant due to the blurriness of the image (top right).

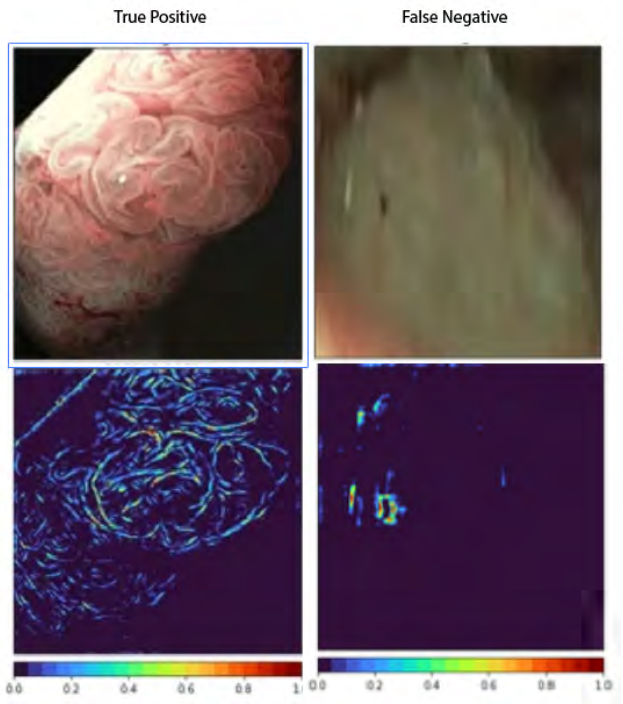


Figure 28: Attribution Heatmaps on a good and poor quality image

4.2.2 Computational Considerations

We compare the computational efficiency of the different attribution methods. Table 11 shows the wallclock run time on the CPU DSVM provided by the Alan Turing Institute, on 900 images from the Piccolo test dataset. This is not a perfect comparison, as run time can vary depending on the amount of computational resources that were available at the specific time, but gives a good estimate of the time to produce the output.

Model	Runtime
Occlusion with Sliding Window (3,8,8) and Stride (3,4,4)	262min28s
Integrated Gradients	31min 3s
GradientSHAP	34min 8s
Guided Backpropagation	29min 18s
Guided GradCAM	34min 22s
Saliency	27min 30s
Deconvolution	30min 14s

Table 11: CPU Runtime for different attribution methods on 900 images.

Table 12 shows the runtime of GuidedGradCam for the original Odin-Vision model, plus the OneDrop model, ThreeDropModel introduced in Section 4.1.2. Similarly, this test was run on the DSVM provided by the Turing Institute on 900 images from the Piccolo test dataset.

Model	Runtime
Original Model	39m14s
One Drop Model	48m3s
Three Drop Model	37m45s

Table 12: CPU Runtime for Guided GradCAM on the Original Model on 900 images. In addition to the OneDropModel and ThreeDropModel model defined in Section 4.1.2.

If attribution method outputs are going to be supplied during an endoscopy, they need to be easy to integrate into the clinician's workflow and take into account the patient's comfort. For this reason, methods with a computation time above 10 seconds are not realistically viable in the clinical setting. Taking into account these practical considerations, gradient-based methods are superior to perturbation-based methods.

An alternative approach is to run the attribution methods on the polyp image after the endoscopy has taken place, when time is no longer a constraint. Clinicians could then use the outcome as an additional tool to support the model's diagnosis, once the initial prediction has been made.

4.2.3 Limitations & Challenges

Specular Reflections in Images

As described in Section 2, many of the polyp images contained spots or regions of white light caused by specular reflections from the endoscope light during the operation. During our early experiments we found these to be frequently highlighted as significant by many of the AMs. However, these are an artefact of the data collection process and represent no physical feature within the polyp itself.

None of the pre-processing methods we tried were able to successfully eliminate this issue and the generated heat maps from the AMs did not differ significantly from the image without this pre-processing approach.

Hyperparameters and baselines

It was particularly challenging to set the hyperparameters and baseline for some of the attribution methods we explored during the project. Here we will discuss Occlusion, Integrated Gradients and GradientSHAP. We start by discussing the limitations and impact of the selection of baseline images, since all three methods require them as an input.

Initially, we noticed that some gradient-based methods had attributions that were focused on areas of the images that are brighter or have higher pixel intensity, caused by the reflection of the light from the endoscope. As explained above, we have experimented with several approaches to mitigate the glare effect (see 4.2.3). But we find that the choice of baseline image or images also impacts this effect.

Captum uses a black image by default when only one baseline is required. Given that the baseline serves as a reference value against which input features are compared, the specular reflections receive high attributions. This problem was also discussed by the authors of Integrated Gradients in [46]. An example of this is shown in Figure 29.

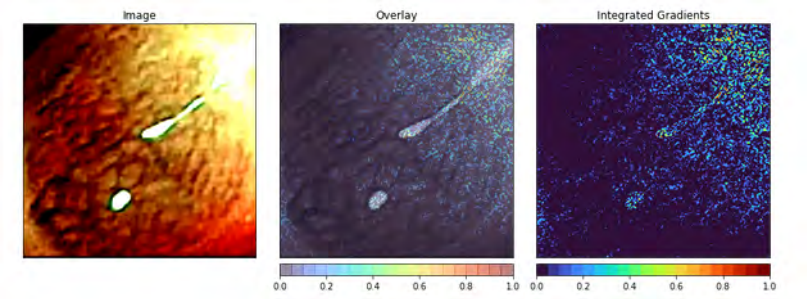


Figure 29: The default Captum choice of baseline for Integrated Gradients attributes the model's outcome to the brighter regions of the input image. This effect is caused by a biased choice of baseline.

The challenge of selecting an unbiased baseline is still an ongoing area of research. Many alternative baselines have been suggested to tackle the problem, examples of which are blurry versions of the input image, the input image plus Gaussian noise or the image that results from averaging over multiple random baselines. The impact of baselines and alternatives are discussed extensively in [43] and [63]. However, to our knowledge none of the solutions has been formally proven to be unbiased.

Figure 30 shows the results from AMs when using a baseline generated as Gaussian noise and a baseline generated as a blurred version of the input image (Gaussian blur). The following parameters are used to create the baselines:

- Gaussian Noise: $\mu = 0$, $\sigma = 0.2$
- Gaussian Blur: kernel size = 35, $\mu = 0.1$, $\sigma = 0.5$

These choice of μ and σ above are arbitrary and we would recommend a more extensive evaluation to compare different parameter choices.

Lastly, besides the baseline image, Occlusion has two additional hyperparameters which are the

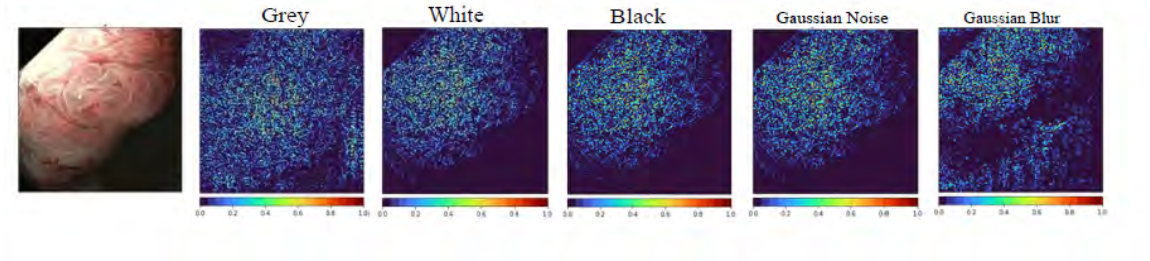


Figure 30: Impact of different baselines images on attributions using Integrated Gradients method.

size of the occluding window and the size of the strides, which dictate the number of pixels that the occluding window moves in each iteration. The smaller the sliding windows and strides are, the larger the computational time will be. We were able to test occlusion parameters: Size of Sliding Windows: (3, 8, 8) and (3, 40, 40), and Strides: (3,4,4), (3,8,8), (3,40,40).

All the hyperparameters above were tested via Odin Vision's GPUs, which helped us speed up the computations and allowed us to perform an evaluation of the different AMs on a large number of images. Due to the time and computational constraints during the project, we did not conduct a comprehensive exploration for the optimization of all the above hyperparameters. This is an area that Odin Vision could explore.

Rotational Variance of Attribution Methods

During our experiments we found that some of the AMs, particularly Occlusion, changed which regions were highlighted as significant depending on the orientation of the original input image. An example of this is shown in Figure 31.

It has been well known that CNNs are translation-invariant due to the pooling operations which disregard spatial information within their neighbourhood, i.e. if an input image is shifted in the x or y direction this is equivalent to shifting the resulting feature, and consequently, the feature extraction process is independent of spatial position [64]. However, this is not the case for the rotation transformation and it has been established in the literature that CNNs are not rotation-invariant. Attempts to rectify this problem have been proposed in computational pathology and other fields with good success [65–67]. However, given the time constraints of this project these were out of scope but it is likely that they would significantly alleviate this issue if applied in future models.

4.3 Integration of Workflows

4.3.1 Attribution applied to Uncertainty Quantification models

We took our findings from the Attribution experiments in Sections 3.2, and integrated it with the Uncertainty quantification Dropout models detailed in Section 4.1.2. The idea is to implement the most successful attribution method, guided GradCAM, applied to the MC dropout model. We approximated the mean of the attributions corresponding to the distribution of predictions obtained from repeated MC dropout iterations, we turned dropout off when completing the forward pass, required for the guided GradCAM algorithm.

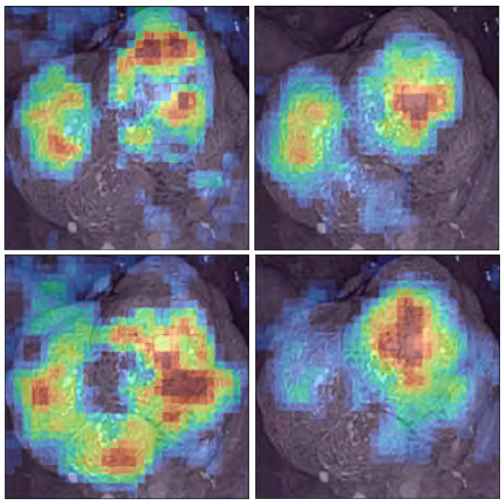


Figure 31: Occlusion maps for a window size of (15x15) with a black baseline for the same test image. Rotation is applied: 0 degrees, 90, 180 and 270 to top left, top right, bottom right and bottom left respectively. Images have been rotated back to the same orientation to demonstrate the difference in the attributions. The model correctly predicts this polyp is an adenoma in each case with confidence: 0.998, 0.99983, 0.9984, 0.9993, going clockwise from the top left image.

We present the results on two test images shown in Figure 32, a true positive (left two columns), and a false positive (right two columns). We observe that when the model is highly certain, as in the true positive case, shown left in Figure 32, the attributions do not vary significantly between the original ResNet 101 model and the MC dropout models. However, when the model is less certain (right), there are differences between the attributions, especially when the number of dropout layers is increased as shown in the final row of Figure 32. We found this to be particularly prominent in the case of poor quality images.

We also computed the infidelity and sensitivity of the guided GradCAM attributions for each image in the Piccolo test data set, as described in Section 3.2.3. This was used to measure how "useful" the guided GradCAM interpretation inside the model's "black-box" was for a given test image. We calculated three metrics to measure how uncertain the MC model was about its prediction for the given image. These metrics included the entropy and variance of the MC model's distribution of predictions and the confidence of the MC model calculated via

$$\tau = \begin{cases} p, & \text{if } p > 0.5 \\ 1 - p, & \text{otherwise} \end{cases} \tag{5}$$

where p is the mean confidence of the model calculated as the average of the model's prediction after the softmax operation over the MC samples and τ is the model's confidence in this mean prediction. If the predicted class was adenoma, i.e. label=1, τ would be equal to p and equal to $1 - p$ if the predicted class was non-adenoma, i.e. label=0.

To explore how the utility of the attribution methods varied with the MC model's uncertainty across

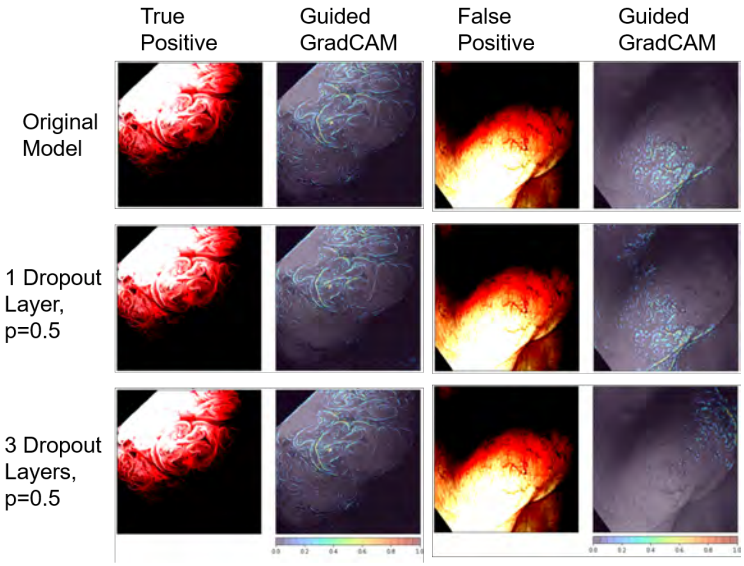


Figure 32: Guided GradCAM applied to the final non-linear layer of each model.

the test images, we randomly sampled 25 images from the test data for each prediction type made by the model (true positive, false positive, true negative and false negative) and plotted how useful the interpretation was against how uncertain the MC model was in its prediction.

The overall performance of the model was very good: there were 571 true positive predictions, 210 true negative, 42 false positive and 77 false negative across all 900 test images. The random sampling enabled us to keep the sample size the same for each type of prediction for ease of comprehension and a fairer comparison, a plot of all 900 examples is included in Figure 33 (right). After excluding one false negative case as a major outlier, Figure 33 shows that while the model had a wide range of uncertainty in its predictions, the attributions consistently had a very low infidelity corresponding to a useful interpretation of the model.

Furthermore, we present the distribution for each prediction type in Figure 34. The median values here show that the model is on average more confident in its correct predictions than the incorrect predictions. However, its confidence varies more with its correct predictions than its incorrect predictions. One potential reason for this is that the labels in the test data contain noise. We suspected several examples to be incorrect and this was then further verified by the clinician from Odin Vision, who suggested that this could be due to the polyp extraction process. This process can sometimes corrupt the polyp sample prior to being sent to the lab for histopathological examination, where the ground truth label is sourced from.

Overall, it was encouraging that the model was on average more confident in its correct predictions. From a clinical perspective, this could give greater trust in the application of AI-enabled systems. However, it is still desirable to reduce this variation in uncertainty as much as possible.

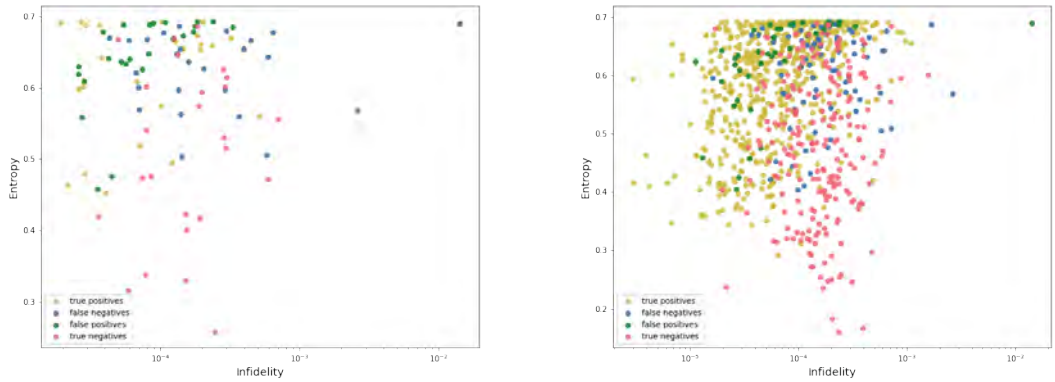


Figure 33: (left) Scatter plot of the one dropout layer MC model's uncertainty, represented by the entropy of MC model's distribution of predictions, against the infidelity of the guided Gradcam heat map from the last non-linear layer for a sample of 25 images from each prediction type. (right) Entropy against infidelity for all 900 test images in the piccolo dataset for the 1 dropout layer MC model.

4.3.2 Integrated Visualisation Dashboard

The interpretability of results predicted by deep learning models is crucial in the field of healthcare. Clinicians should be able to recognise what the model has based its decision on in order to facilitate their own decision-making process. For instance, it can help reject false-negative prediction easily and ultimately increase the trust in the model prediction.

The software package Captum provides a web interface called Insights for easy visualisation and access to some of the interpretability algorithms described in this project. Captum Insights is built in the Captum library and allows sample-based model debugging and visualisation using feature importance metrics. We suggest that this interactive visualisation tool can potentially be used in clinical settings[42]. The tool allows to sub-sample input images and provides interactive options to assess the results from different attribution methods.

An example is depicted in Figure 35. It can also be embedded in a notebook to help data scientists to debug their machine learning models. However, it only allows to modify a limited selection of hyperparameters and does not provide a some relevant information such as a measure of the uncertainty in the prediction.

In order to overcome the limitations of Captum Insights, we suggest building a bespoke interactive dashboard would be the most useful way for Odin Vision to analyse and explain model predictions intuitively. Such a tool could be built using streamlit or plotly and incorporate several features which are missing in Captum Insights. An example potential dashboard is shown in Figure 35

On top of the model prediction, this dashboard could show information about the uncertainty of the prediction and attributions. An example of such a dashboard is provided in Figure 1 (in Section).

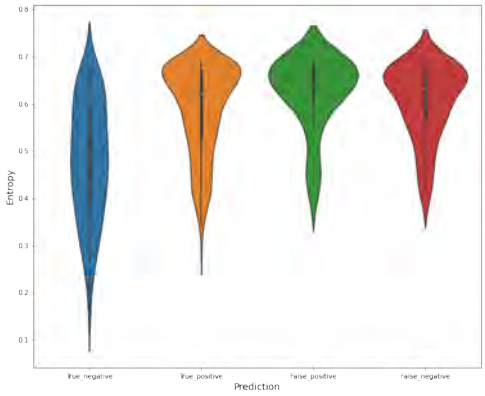


Figure 34: Violin plot demonstrating the distribution of model uncertainty, represented by entropy, of the one dropout layer MC model in its predictions for each type of prediction, across all 900 test images in the piccolo dataset.

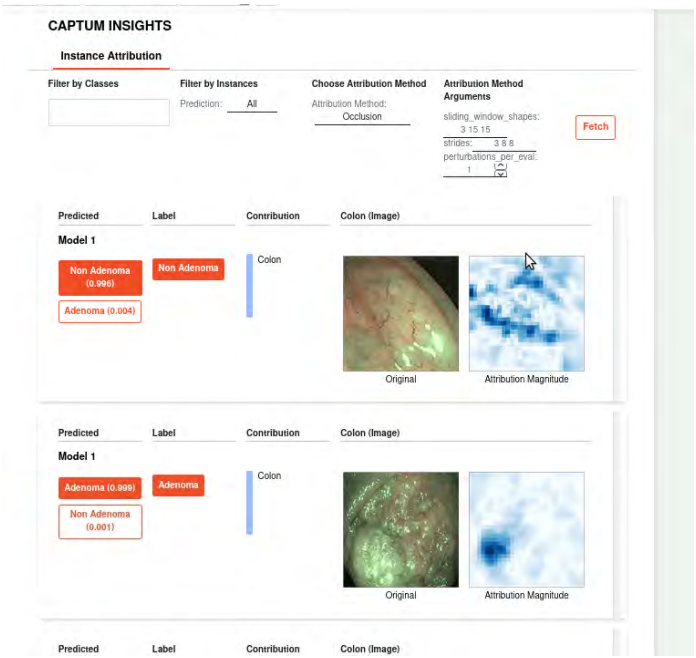


Figure 35: Captum Insight: An interactive visualisation tool displaying the application of occlusion attribution metod to both adenoma and non-adenoma polyp images. The tool also visualises aggregated attribution magnitudes of each modality.

4.4 Representation Learning Experiments & Results

In this section, we present the details and results of our training of a VAE model on the White Light Endoscopy (WLE) Dataset (referred as Dataset 2 in this report; see details of the dataset in Section 2.2). We first provide details of the training setup in Section 4.4.1. We then show the performance of the trained model in Section 4.4.2. We evaluate the model based on it reconstructing the original polyp images, as a proxy metric to indicate the quality of the derived encoding function, which maps a high-dimensional input image to a very low-dimensional latent feature vector. Finally, in Section 4.4.3, we try to interpret what information each latent feature has encoded using a latent space interpolation technique that was introduced in Section 3.3.2.

4.4.1 Training Setup for the VAE on polyp images

We use a CNN based neural network for the encoder and decoder in our VAE. Specifically, we adopt a ResNet18 [68] for the encoder and a model with 5-layer deconvolution for the decoder. Three fully connected layers convert the spatial feature map from the ResNet encoder to a latent space with 16 dimensions, which are parameterised by the prediction of a 16-dimensional mean vector and a 16-dimensional variance vector. The latent feature vector is then sampled from the Gaussian distributions specified with the mean and variance vector using the reparameterisation trick. To decode the latent feature vector back to the original image space, we convert the latent feature vector to a spatial feature map using a fully connected layer and a reshape.

The total number of parameters for the encoder and decoder was around 11M and 5M respectively. Details of the architecture can be found in our project Github here: [69].

We train this network for 50 epochs with the loss in Equation (4) and a batch size of 512. We use an Adam optimizer [70] with learning rate of $1e-4$. We explore the control parameter β at two different values $\beta = 1,$ or 10 and start a training for each β value. We use Dataset 2 for training the VAE model (see Section 2.2 for more details of this dataset).

4.4.2 Examining the training losses and reconstruction quality

We monitor three losses - reconstruction error, prior regularisation and negative ELBO loss - as the training proceeds. The losses for both β values are shown in Figure 36. It can be seen that the VAE model with a smaller β, i.e. $\beta = 1$, manages to produce better reconstruction quality, as the reconstruction error is around 10% lower than the model with a higher β, i.e. $\beta = 10$. On the other hand, the benefit of having a strong prior regularisation pays off in terms of the level of disentanglement being reached in the derived latent features. As shown in the middle column, the higher β model has a much lower prior regularisation penalty (60% lower), indicating the derived latent features are much more independent from each other than the features derived with the VAE with smaller β.

It is worth noting that despite of producing more disentangled latent factors, the VAE model with higher β reaches a lower ELBO than the model with smaller β. Considering that ELBO is a lower bound of the log likelihood of the current VAE model fitting to the observed dataset, we would prefer the model with higher ELBO, i.e. the model with smaller β, in general.

We also include some examples of the reconstruction from both trained models with different β values in Figure 37. The reconstruction quality from both models are quite good. The overall shape, colour and important features of polyps are clearly captured. On the downside, both models exhibit some level of blurriness. This might be a limiting factor in the downstream polyp classification task,

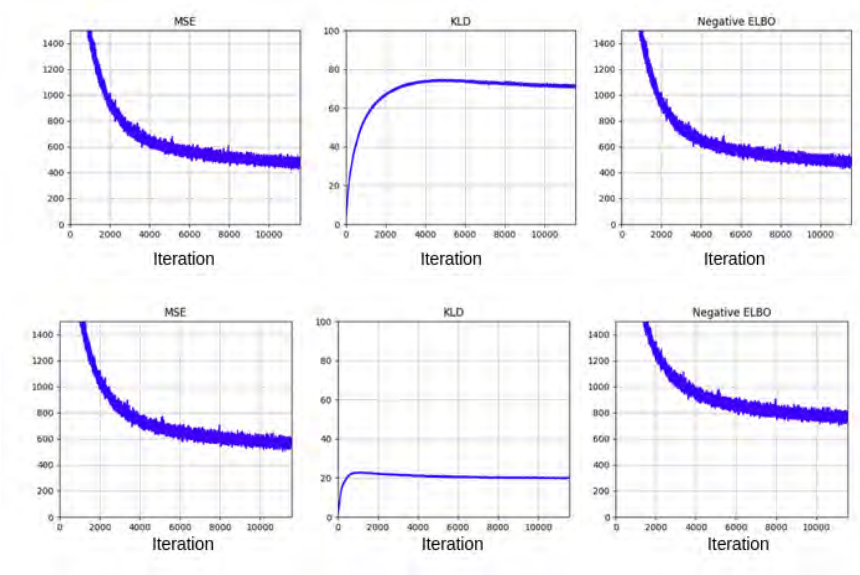


Figure 36: We monitor three losses: reconstruction error (left), prior regularisation penalty (middle) and ELBO loss (right), for two β values: $\beta = 1$ (top) and $\beta = 10$ (bottom).

as some fine-grain vessel structures might not be preserved in the mapping to the latent features. Comparing the reconstruction quality of the two models against each other, we can notice that the model with smaller β (i.e. $\beta = 1$), gives better reconstruction, as the reconstructed images exhibit less artefacts (see the third example from left in the first row) and preserve the details, such as reflections, better (see the second example from right in the first row).

It is worth emphasising that producing a perfect reconstruction is not the ultimate goal of the task. Despite the desire of having a high quality reconstruction from the autoencoder, we need to separate this task from our end goal, i.e. to derive a set of features that can be useful for the downstream classification task. This is the reason that we are now going to examine the information encoded in each of the latent features in the following section.

4.4.3 Understanding the derived latent features

As introduced in Section 3.3.2, we follow the procedure in [60] to examine the information encoded in different latent features. Specifically, we observe the change in the decoded images by perturbing some latent feature. Here we present two sets of results: 1) given the same input image, we perturb one of its latent feature at a time and observe the corresponding changes in the decoded images; 2) given the interest in a specific latent feature (i.e. a fixed latent dimension), we perturb this feature across a set of different images and observe the changes resulted. Intuitively, the first experiment allows us to gain an understanding on what each latent feature precisely encodes for a particular image, whereas the second experiment illustrates whether the encoding of a specific latent feature is shared globally across the entire population of the polyp images.

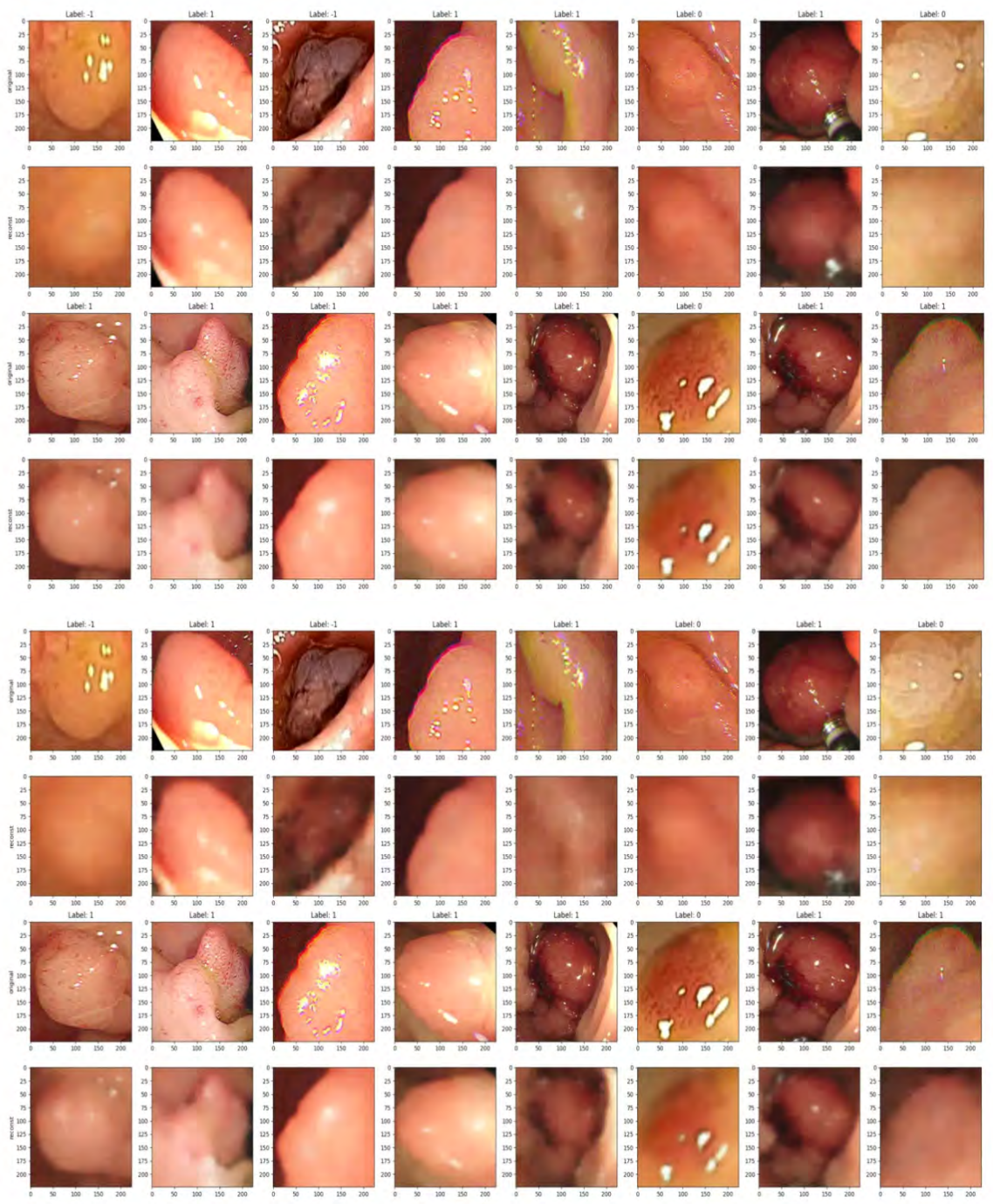


Figure 37: We show the reconstruction results from the two trained VAE models with different β values - $\beta = 1$ (top) and $\beta = 10$ (bottom).

Examining different latent features of the same image

We present example results in Figure 38 for the VAE model with $\beta = 1$ and Figure 39 for the VAE model with $\beta = 10$. As can be seen in Figure 38 (top), different latent features control different aspects of the visual information. For instance, z_5 (row b) seems to control the redness of the polyp and some texture, z_7 (row c) seems to control the whiteness and reflection and z_{15} (row g) seems to control the shape and the contrast of the polyp's colour in comparison to the background.

We investigate the impact of β in this example by comparing Figure 38 and 39. Higher β (Figure 39) seems to result in a clearer separation of different factors among different latent dimensions. For example, z_5 (row b in Figure 39) seems to control the shape as well as the appearance of mucus (the control of mucus is only marginally shown in the row b in Figure 38) and z_8 clearly controls the brightness of the polyp from purely white to very dark. However, the higher β also has a side effect that some latent dimensions are left unused (being purely white noise) or end up with encoding very similar information (see row e and f in Figure 39).

Examining the same latent feature across different images

To explore whether the encoding of certain information is shared among all input images, we also carry out the latent space interpolation on the same latent dimension across different images. Figure 40, 41 and 42 show the effects as we perturb three latent feature across five different images respectively. In Figure 40, β is set to be 1. We can see that this latent feature (z_0) controls the appearance of the reflection in the polyp image. The shiny spots on the polyps gradually fade when the feature value is changed from -3 to 3 across all images.

It is worth mentioning that different latent feature has a different level of control over the amount of changes to the corresponding visual features. This effect can be seen in the comparison between Figure 41 and 42, where we perturb the feature value in z_7 and z_15 respectively on a VAE with $\beta = 10.0$. In Figure 41, we can see a slight change in the polyp colour from left to right, whereas in Figure 42 the change is very more obvious varying from yellow on the left to pink on the right. According to this result, we can conclude that the latent feature z_15 has a stronger control over the colour change in comparison to z_7.

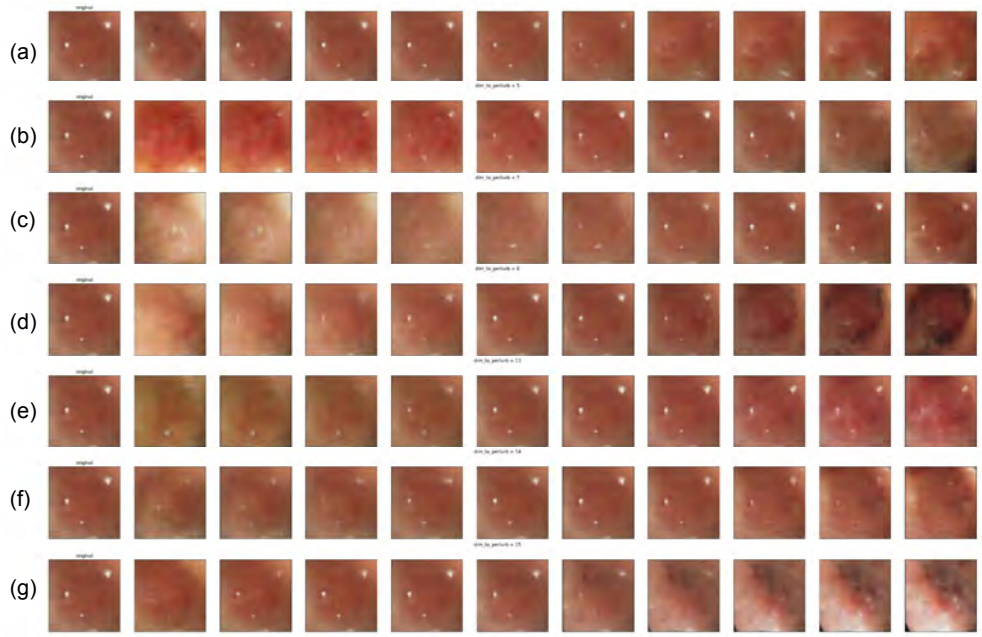


Figure 38: Latent code visualization with training parameter β set to 1.0. From (a) to (g), reconstructed results by perturbing $z_0, z_5, z_7, z_8, z_{13}, z_{14}, z_{15}$ in the latent vector separately. The first column of each row is the same input image.

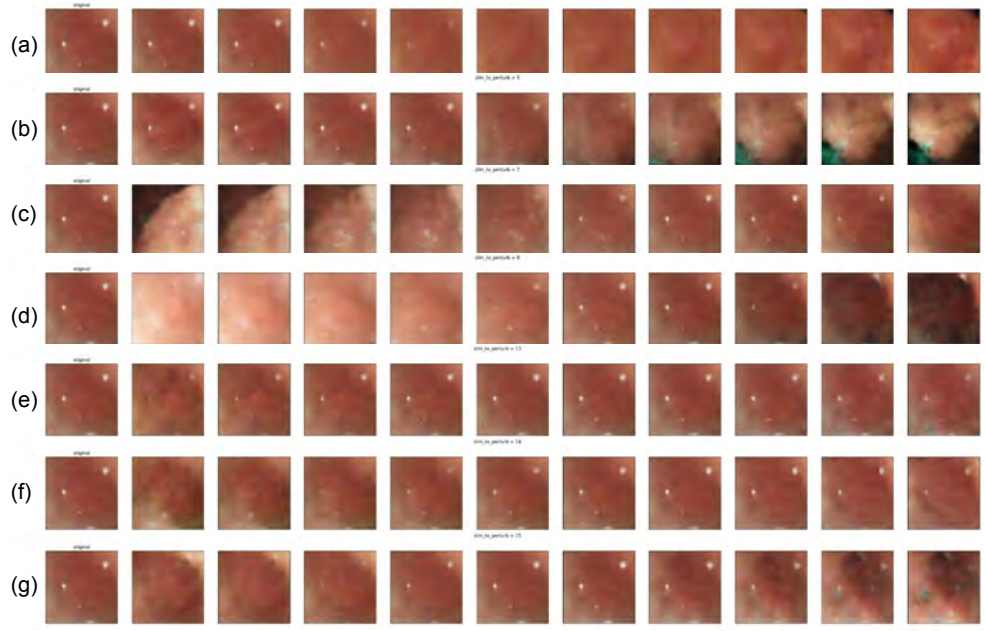


Figure 39: Latent code visualization with training parameter β set to 10. From (a) to (g), reconstructed results by perturbing $z_0, z_5, z_7, z_8, z_{13}, z_{14}, z_{15}$ in the latent vector separately. The first column of each row is the same input image.

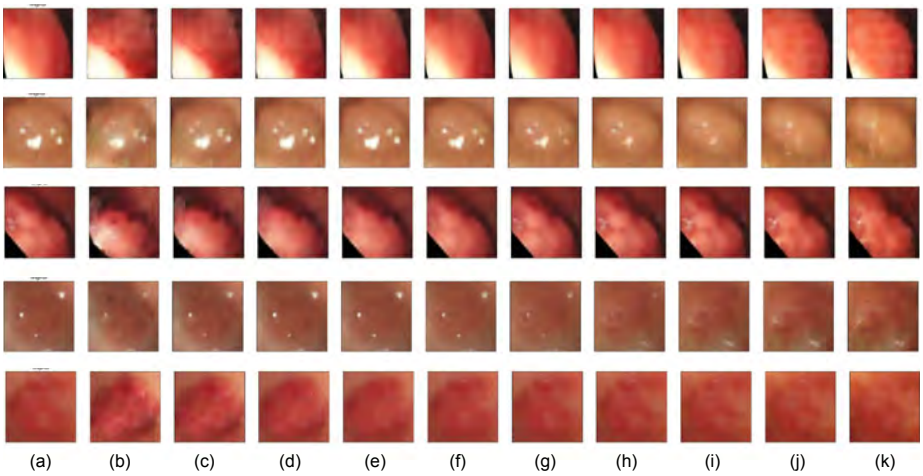


Figure 40: Latent code visualization with training parameter β set to 1.0. Interpolation results by changing z_0. (a), the original reconstruct image; from (b) to (k), the reconstructed image by replacing z_0 with a number equidistant in [-3,3].

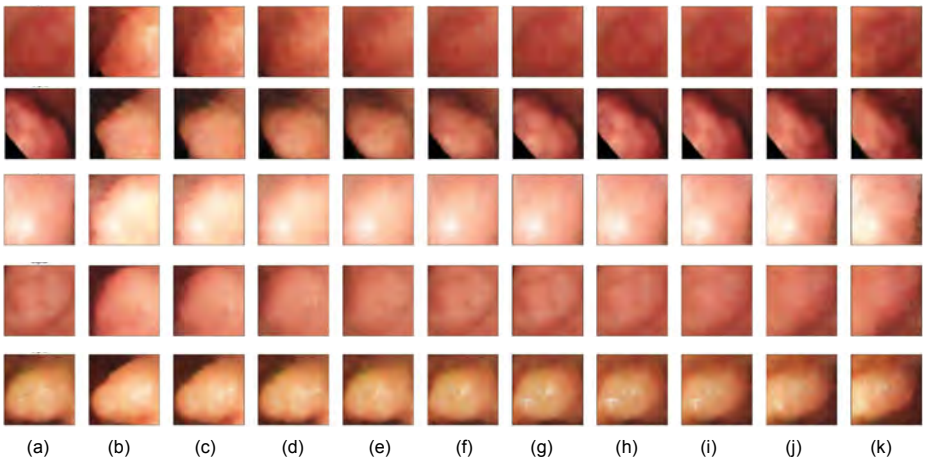


Figure 41: Latent code visualization with training parameter β set to 10.0. (a), the original reconstruct image; from (b) to (k), the reconstructed image by perturbing z_7 in the latent vector.

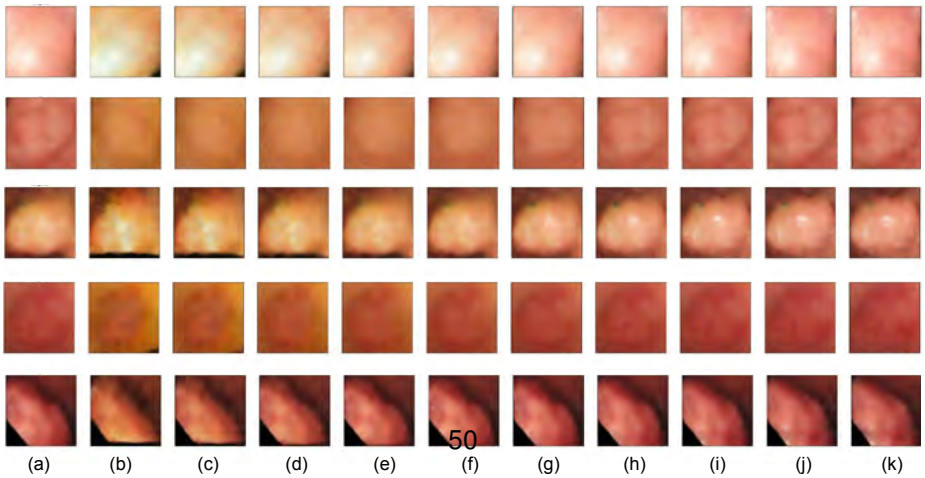


Figure 42: Latent code visualization with training parameter β set to 10.0. Interpolation results on the sixteenth dimension. (a), the original reconstruct image; from (b) to (k), the reconstructed image by perturbing z_{15} in the latent vector. This change reflects the color transformation between different images.

5 Discussions & Conclusion

5.1 Uncertainty Quantification

In Section 4.1.1 we presented an approach to construct an uncertainty profile over handwritten digit predictions (a toy challenge), by sampling from weight distributions that were learned using variational inference. This proved successful, and appropriate distributions of uncertainty were shown on out-of-distribution samples (other digits from the training set).

In Section 4.1.2 we introduced three MC Dropout architectures and qualitatively investigated reporting their uncertainty distributions on the polyp prediction task. As part of this investigation, we proposed a graphical representation of the the uncertainty distribution, readily interpretable by clinicians. In Section 4.1.3 we quantitatively investigated different metrics to try to understand the quality of the uncertainty estimates, i.e. how well the model was representing the uncertainty, and a preliminary framework was proposed to judge the quality of a given model. Finally in Section 4.1.4 we investigated approaches to provide highly interpretable metrics, by classifying the level of undertainty into easy to understand ordinal classes.

We found limited dependence on the magnitude of the dropout, with similar performance on narrow layers as well as broad layers. The uncertainty profiles produced were strongly dependant on the architecture, with the deeper ThreeDrop model exhibiting a markedly different uncertainty profile from the other two architectures investigated. An avenue of future work would be to further tune the architecture used to find a setup that produces the most representative uncertainty profile on the task at hand.

Conclusion

Although initial observations showed some challenges in training such models, MC Dropout based models proved relatively easy to incorporate into existing architecture. For this reason we recommend them as a starting point to further investigation by Odin-Vision researchers. By simply adding dropout layers before Linear layers in computer vision classifiers, a level of uncertainty metric can be prototyped for inclusion in their products. It is hoped that some of the work in classifying and comparing performance of uncertainty distributions between model architectures could be a useful as a starting point for internal evaluation of their models. However, it is anticipated that more systematic investigation would be required, such as training and testing on the same, in-house Odin-Vision dataset.

5.2 Attribution Methods

This workstream explored a variety of methods for interpreting neural network classifications. We focused on gradient and perturbation-based attribution methods, and found that Guided GradCam was the most insightful method, based on discussions with Odin-Vision clinicians. In Section 4.2.1 we presented our initial exploration. Since the Odin-Vision product is a real-time method, we considered computational cost in Section 4.2.2, and the considerable limitations and challenges of using these methods in a clinical setting in Section 4.2.3. These challenges included image quality, i.e. glare (specular reflections), hyperparameter and baseline tuning of algorithms for consistent results.

In Section 4.3 we combined the best performing attribution method, GradCam, with a predicted level of uncertainty, from the first workstream, to increase the trustability of the model predictions and

identify potential failure modes, such as false negative predictions. Lastly, both workstreams were incorporated into a bespoke interactive dashboard. we see this as crucial to provide to clinicians, to easily communicate the output of our models. This is a major take-away from this project (illustrated in Figure 1).

Proposed Future Work

There are other AMs available including Feature Ablation and DeepLift. However, neither worked particularly well within our time frame and will require further investigation. Feature Ablation [71] is a perturbation-based AM. Unlike Occlusion it is not limited to a fixed shape of the occluding window (frequently a square) to ablate the input features. Instead, a segmentation mask can be provided that allows us to finely define the shape or group of input features to be removed in order to calculate their impact on the model's prediction. DeepLift (Deep Learning Important Features) is another AM based on the gradient formulation of [72] and the algorithm of [73]. It backpropagates the activations of every neuron in the network with respect to the inputs. It then calculates its contribution scores by comparing the difference between the activation of each neuron to a reference activation. This method was not successful in our experiments due to an additional requirement of redefining the ReLU modules in the pre-trained model. It produced a runtime error that prevented us from using it. This issue is covered in [74] and is caused by the computational graph not being able to know the ReLU calls of the network if its being used more than once.

Although we had the chance to investigate different choices of hyperparameters for different AMs, we believe that a more exhaustive exploration is needed in order to find their most suitable values. For example, GradientSHAP can be provided with multiple baselines such that the expectation of the gradients will be calculated by randomly picking a random baseline from them, but we have only experimented with one baseline in order to keep it comparable with the other gradient based method.

There are different perturbations such as the difference between the input and the base line on the input that can explored in order to optimise the infidelity metric. We have only used Gaussian noise as the perturbation function for the infidelity metric calculated in our results and used for the integration with the uncertainty work.

In addition to layer attributions the attributions of individual neurons can also be explored. They are able to measure the contribution of each input feature based on the activation of specific neurons. The attributions from neurons could potentially be useful to the ML researchers at Odin-Vision for improving their models. There is also work that look into adding noise (such as Gaussian noise) to the inputs in order to create smoother or sharper versions of the attribution heatmaps. This is covered in detail in the SmoothGrad paper [75]. It is also mentioned in [76–78].

Other attribution methods that we were not able to explore during the project are: Layer-wise Relevance Propagation (LRP) [79]; KernelSHAP [47]; Feature Permutation [80]; GradCam++ [81]; AblationCAM [82]; ScoreCAM [83]; SmoothGradCam++ [84]; DeepDream [85].

Although we have decided to use Captum for our project, there are other libraries that can be considered to obtain similar results. A comprehensive collection of ML/AI interpretability Github projects can be found in [86]. It should also be noted that the infidelity metric implemented in Captum is not the exactly the same as in the original paper [53]. The Captum implementation of infidelity is not robust to the norm of the attribution maps and the attribution explanations are not currently scaled by a factor. This is a known issue and is documented in [87].

Clinical interpretability

In order to assess the ease of interpretation of AMs, we worked closely with clinicians from the Odin-Vision. They suggested that attribution heatmaps are intuitive tools that clinicians do find easy to understand. However, this was not the case for infidelity as a performance metric. The reason for this is that the definition of infidelity is technical and relies on the concept of (mathematical) sensitivity, which conflicts with the clinical use of this term. As a result, we concluded that performance metrics are perhaps only useful to ML specialists at Odin-Vision.

The authors in [88] have explored the use of similar AMs to the ones that we use here in a clinical setting. While some of these methods are good at closing the gap between human and AI-enabled predictions, it should be emphasised that AMs are not foolproof and are subject to some weaknesses. For example, in [89] it has been shown that targeted perturbations of the input images have the potential to trick the AMs, so that the highlighted areas in the heatmaps shift away from the features that are most influential on the model predictions.

Discussion & Conclusion

Some gradient-based methods are dependent on the model's implementation, hence our results could vary depending on the architecture of the NN model and the training data set. This should be taken into consideration when attempting to reproduce our results Odin-Vision. An interesting approach to explore would involve the comparing the results provided by AMs methods for the same inputs when processed by different models.

In [90], a review of the use of CNNs in a Computer-Aided Diagnosis (CAD) system is provided. The authors show promising results in their performance (sensitivity, specificity and accuracy) when compared to humans (experts and non-experts). As the AMs do not require training any model from the ground up and are relatively easy to implement, we believe that they are an interesting tool to add to the existing CAD pipeline.

Overall, we hope the experiments undertaken during this project are an insightful starting point to be able to include attribution methods into their product.

5.3 Representation Learning

We demonstrate that some interesting latent factors can be discovered directly from data using approaches, such as variational autoencoders (VAEs) introduced in Section 3.3.1. We also investigate what information each latent factor is likely to encode using some latent space interpolation techniques as introduced in Section 3.3.2. Although this stream of work is set out to be exploratory, we consider the outcome from this data study is rather encouraging. The results we present in Section 4.4 indicate that these derived features control certain visual features consistent across different polyp images. Hence, they are likely to be useful for downstream tasks, such as classifying cancerous polyps.

We leave the task of classifying cancerous polyps based on these latent factors for future exploration. A key question to answer is how to obtain interpretability when a classifier is designed using these features that are discovered fully from data? Some work [91] proposes to organically integrate the classification task with the representation learning module by constraining the classifier to be linear and, thus, effectively deriving a set of semantically meaningful latent factors.

Another promising direction to consider is to apply the attribution analysis as introduced in Section 3.2 to the discovered latent factors. This way, we can examine which latent factors have an important role in identifying the cancerous polyps. We believe that many interesting questions can be explored following this line of research.

Conclusion

Investigating self-supervised representation learning was a key aim of this project from the outset. The key idea being to move beyond standard optical feature representation, such as NICE. The generative VAE models investigated during this project have provided this proof-of-concept for learning better, automatic representations from the data. However, as we demonstrate, interpreting these latent features is not trivial, and considerable further investigations are required.

We believe to fully interpret the prediction models used at Odin-Vision, further work is needed to understand the model representations, and this project provides a promising and viable direction of research.

Figure 43: Team Odin Vision

6 Team members

Batool Almarzouq is a bioinformatician with a background in Pharmacology and has a PhD in Cancer Biology from the University of Liverpool. She is a core contributor of the *Turing Way* and a member of the R-Ladies Global Team. She advocates for Open Science and its role in improving research reproducibility, currently working as a postdoctoral researcher in KAIMRC. Batool contributed to this project as part of the attribution methods group.

Carmen Cabrera-Arnau is a Research Fellow at the Centre for Advanced Spatial Analysis (CASA), UCL. She joined CASA through the realTRIPS project, which aims to open a new avenue of research in urban mobility analysis using emerging automatic data. Particularly, her work focuses on developing an analytical and modelling framework that addresses variability across spatial-temporal scales and of population groups. Prior to this, she did the MSc in Complex Systems at King's College London, followed by a PhD at UCL's Department of Mathematics.

Emma-Jane Ditter has a background in physics, doing her PhD and a postdoc at Imperial College London. Now a postdoctoral research at the University of Cambridge she is analysing next generation sequencing data for the early detection of cancer. She contributed to this project as part of the uncertainty quantification group.

Paul Duckworth is a postdoctoral research assistant at the Oxford Robotics Institute, and a stipendiary lecturer at Brasenose College, both at the University of Oxford. His research is in machine learning and sequential decision making for robots, usually in challenging environments. He is the PI for the project and first author of this report.

Rob Jewsbury is a PhD student in the Tissue Image Analytics Centre at the University of Warwick. His research focuses on the application of machine learning and computer vision methods in the field of computational pathology for cancer diagnosis and treatment. In this challenge, he lead the work exploring the gradient attribution methods.

Michal Kazmierski is a second-year graduate student in the Department of Medical Biophysics at University of Toronto. His research focuses on using machine learning on large, multi-modal

datasets (including imaging and electronic health records) to predict outcomes of cancer patients and help guide clinical care. Previously, he studied at Imperial College London, graduating with First Class Honours. In this challenge, he contributed to the work on uncertainty quantification methods.

Zhiqi Li is a first-year PhD student in National Center for Computer Animation at Bournemouth University. Her research interests include computer graphics, 3D computer vision, and geometry processing. She contributed to this project in the representation learning tasks.

Shuyu Lin is a PhD student at the Computer Science department, University of Oxford. Lin's research focuses on understanding machine learning techniques - their benefits and limitations - and how to apply them in real world applications. During this project, she worked on the representation learning task and used VAEs to derive latent factors of the polyp images.

Dave Lines is a research scientist at Odin Vision and focuses on the use of machine learning techniques for the detection and diagnosis of early-stage cancer in the gastrointestinal tract. He is the challenge owner for the project.

Timothy Sum Hon Mun is a first year PhD student at the Institute of Cancer Research after having spent time working in the insurance and tech industry. His research focuses on AI in medical imaging for automated response evaluation and prediction of soft-tissue sarcomas which are a group of rare cancers that are highly heterogeneous to systemic treatment using multi-centre multi-modal imaging data.He contributed to this project as part of the attribution methods group.

David Yallup is a postdoc in Bayesian Machine Learning in the Astrophysics group at the University of Cambridge. Transitioning to the ML field after a PhD in Particle Physics at UCL. His research focuses on scaling principal Bayesian techniques to computer vision applications.

Lorenzo Zanisi Lorenzo is a prospective PhD graduate in Astrophysics at the University of Southampton. His research focuses on applying data-driven methods to understand how galaxies evolve in our Universe, and he also has a record of successful projects in healthcare applications of data science. He co-facilitated this project and contributed to it directly in the uncertainty quantification and the representation learning tasks.

Appendix A

Sample code for the architectures described in Reporting Uncertainty on Polyp Image Data in Section 4.1.2.

Listing 1: Additional Architecture for OneDrop model

```
Sequential(
  (0): Dropout(p=0.2, inplace=False)
  (1): Linear(in_features=2048, out_features=2, bias=True)
)
```

Listing 2: Additional architecture for the ThreeDrop model

```
Sequential(
  (0): Dropout(p=0.2, inplace=False)
  (1): Linear(in_features=2048, out_features=2048, bias=True)
  (2): ReLU(inplace=True)
  (3): Dropout(p=0.2, inplace=False)
  (4): Linear(in_features=2048, out_features=2048, bias=True)
  (5): ReLU(inplace=True)
  (6): Dropout(p=0.2, inplace=False)
  (7): Linear(in_features=2048, out_features=2, bias=True)
)
```

Listing 3: Additional architecture for Resnet-VAE model

```
Sequential(
  (0): Dropout(p=0.2, inplace=False)
  (1): Linear(in_features=2048, out_features=16, bias=True)
  (2): ReLU(inplace=True)
  (3): Dropout(p=0.2, inplace=False)
  (4): Linear(in_features=16, out_features=2, bias=True)
)
```

References

[1] Yarin Gal and Zoubin Ghahramani. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. 2016. arXiv: 1506.02142 [stat.ML].

[2] Bowl Cancer UK. *Facts and Figures about Bowel Cancer*. 2019. URL: www.bowelcanceruk.org.uk/about-bowel-cancer/bowel-cancer/.

[3] Marzieh Araghi et al. "Global trends in colorectal cancer mortality: projections to the year 2035". In: *International Journal of Cancer* 144.12 (2019), pp. 2992–3000.

[4] Rebecca L Siegel et al. "Colorectal cancer incidence patterns in the United States, 1974–2013". In: *JNCI: Journal of the National Cancer Institute* 109.8 (2017).

[5] Royal College of Pathologists. "Meeting Pathology Demand. Histopathology Workforce Census". In: *Royal College of Pathologists* (2018).

[6] Ignasi Puig and Tonya Kaltenbach. "Optical diagnosis for colorectal polyps: a useful technique now or in the future?" In: *Gut and liver* 12.4 (2018), p. 385.

[7] Colin J Rees et al. "Narrow band imaging optical diagnosis of small colorectal polyps in routine clinical practice: the Detect Inspect Characterise Resect and Discard 2 (DISCARD 2) study". In: *Gut* 66.5 (2017), pp. 887–895.

[8] Omer F Ahmad et al. "Artificial intelligence and computer-aided diagnosis in colonoscopy: current evidence and future directions". In: *The Lancet Gastroenterology & Hepatology* 4.1 (2019), pp. 71–80. ISSN: 2468-1253. DOI: https://doi.org/10.1016/S2468-1253(18)30282-6. URL: https://www.sciencedirect.com/science/article/pii/S2468125318302826.

[9] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

[10] Anh Nguyen, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 427–436.

[11] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[12] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[13] *7.6. Residual Networks (ResNet) — Dive into Deep Learning 0.16.3 documentation*. URL: https://d2l.ai/chapter_convolutional-modern/resnet.html (visited on 05/10/2021).

[14] *matlab-deep-learning/resnet-101*. original-date: 2019-10-10T12:20:43Z. Mar. 2, 2021. URL: https://github.com/matlab-deep-learning/resnet-101 (visited on 05/10/2021).

[15] Josipa Patrun et al. "Diagnostic Accuracy of NICE Classification System for Optical Recognition of Predictive Morphology of Colorectal Polyps". In: *Gastroenterology Research and Practice* 2018 (2018), p. 7531368. ISSN: 1687-6121. DOI: 10.1155/2018/7531368. URL: https://doi.org/10.1155/2018/7531368.

[16] *Polyp Classification: NICE*. 2019. URL: https://www.endoscopy-campus.com/en/classifications/polyp-classification-nice/.

[17] *NICE: Diagnostic Assessment Report*. 2016. URL: https://www.nice.org.uk/guidance/dg28/documents/diagnostics-assessment-report.

[18] Olympus. *NARROW BAND IMAGING (NBI) A New Wave of Diagnostic Possibilities*. URL: https://www.olympus-europa.com/medical/rmt/media/en/Content/Content-MSD/Images/SCP-Pages/EndoAtlas/E0428859-NBIClinical_brochure_EN.pdf (visited on 05/10/2021).

[19] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. *THE MNIST DATABASE of handwritten digits*. 2008. URL: http://yann.lecun.com/exdb/mnist/ (visited on 05/10/2021).

[20] Quentin Angermann et al. "Towards real-time polyp detection in colonoscopy videos: Adapting still frame-based methodologies for video sequences analysis". In: *Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures*. Springer, 2017, pp. 29–41.

[21] Jorge Bernal et al. "WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps fromphysicians". In: *Computerized Medical Imaging and Graphics* 43 (2015), pp. 99–111.

[22] Hanna Borgli et al. "HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy". In: *Scientific Data* 7.1 (2020), pp. 1–14.

[23] Luisa F Sánchez-Peralta et al. "PICCOLO White-Light and Narrow-Band Imaging Colonoscopic Dataset: A Performance Comparative of Models and Datasets". In: *Applied Sciences* 10.23 (2020), p. 8501.

[24] Pablo Mesejo et al. "Computer-aided classification of gastrointestinal lesions in regular colonoscopy". In: *IEEE transactions on medical imaging* 35.9 (2016), pp. 2051–2063.

[25] Masashi Misawa et al. "Development of a computer-aided detection system for colonoscopy and a publicly accessible large colonoscopy video database (with video)". In: *Gastrointestinal Endoscopy* 93.4 (2021), pp. 960–967.

[26] Hayato Itoh et al. *SUN Colonoscopy Video Database*. 2020. URL: \url{http://amed8k.sundatabase.org/}.

[27] Krushi Patel et al. "A comparative study on polyp classification using convolutional neural networks". In: *PloS one* 15.7 (2020), e0236452.

[28] Liu Renting, Li Zhaorong, and Jia Jiaya. "Image partial blur detection and classification". In: *IEEE CVPR*. 2008, pp. 1–8.

[29] Alexandru Telea. "An Image Inpainting Technique Based on the Fast Marching Method". In: *Journal of Graphics Tools* 9 (Jan. 2004). DOI: 10.1080/10867651.2004.10487596.

[30] M. Bertalmio, A.L. Bertozzi, and G. Sapiro. "Navier-stokes, fluid dynamics, and image and video inpainting". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990497.

[31] S. Damelin and N. S. Hoang. "On Surface Completion and Image Inpainting by Biharmonic Functions: Numerical Aspects". In: *Int. J. Math. Math. Sci.* 2018 (2018).

[32] *Image Reconstruction with Belief Propagation*. 2016.

[33] Ryo Abiko and Masaaki Ikehara. "Single Image Reflection Removal Based on GAN With Gradient Constraint". In: *IEEE Access* 7 (2019), pp. 148790–148799. DOI: 10.1109/ACCESS.2019.2947266.

[34] Chuan Guo et al. *On Calibration of Modern Neural Networks*. 2017. arXiv: 1706.04599 [cs.LG].

[35] Geoffrey E. Hinton and R. Neal. “Bayesian learning for neural networks”. In: 1995.

[36] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[37] Carl Edward Rasmussen. “Gaussian processes in machine learning”. In: *Summer school on machine learning*. Springer. 2003, pp. 63–71.

[38] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*. 2017. arXiv: 1612.01474 [stat.ML].

[39] Yaniv Ovadia et al. *Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift*. 2019. arXiv: 1906.02530 [stat.ML].

[40] C. E. Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[41] Lewis Smith and Yarin Gal. *Understanding Measures of Uncertainty for Adversarial Example Detection*. 2018. arXiv: 1803.08533 [stat.ML].

[42] Narine Kokhlikyan et al. “Captum: A unified and generic model interpretability library for PyTorch”. In: *arXiv:2009.07896 [cs, stat]* (Sept. 2020). arXiv: 2009.07896. URL: http://arxiv.org/abs/2009.07896 (visited on 05/10/2021).

[43] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. “Visualizing the Impact of Feature Attribution Baselines”. In: *Distill* (2020). https://distill.pub/2020/attribution-baselines. DOI: 10.23915/distill.00022.

[44] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. Tech. rep. arXiv: 1312.6034v2. URL: http://code.google.com/p/cuda-convnet/.

[45] Jost Tobias Springenberg et al. *Striving for Simplicity: The All Convolutional Net*. 2015. arXiv: 1412.6806 [cs.LG].

[46] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic Attribution for Deep Networks”. In: *arXiv:1703.01365 [cs]* (June 2017). arXiv: 1703.01365. URL: http://arxiv.org/abs/1703.01365 (visited on 05/10/2021).

[47] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI].

[48] Ramprasaath R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *International Journal of Computer Vision* 128.2 (2019), 336–359. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

[49] PyTorch. *Introduction to Captum — A model interpretability library for PyTorch*. en. Mar. 2020. URL: https://medium.com/pytorch/introduction-to-captum-a-model-interpretability-library-for-pytorch-d236592d8afa (visited on 05/10/2021).

[50] *Algorithm Descriptions · Captum*. en. URL: https://captum.ai/ (visited on 05/10/2021).

[51] Ramprasaath R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”. In: *Int J Comput Vis* 128.2 (Feb. 2020). arXiv: 1610.02391, pp. 336–359. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://arxiv.org/abs/1610.02391 (visited on 05/10/2021).

[52] *Algorithm Comparison Matrix · Captum*. en. URL: https://captum.ai/ (visited on 05/10/2021).

[53] Chih-Kuan Yeh et al. *On the (In)fidelity and Sensitivity of Explanations*. Tech. rep. arXiv: 1901.09392v4. URL: https://github.com/chihkuanyeh/saliency{_}evaluation..

[54] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *arXiv* (2013).

[55] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”. In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2014, pp. 1278–1286.

[56] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Neural Information Processing Systems (NeurIPS)*, pp. 2672–2680.

[57] Laurent Dinh, David Krueger, and Yoshua Bengio. “NICE: Non-linear Independent Components Estimation”. In: *Conference on Robot Learning (CoRL)* (2014).

[58] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. “Density estimation using Real NVP”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017.

[59] Benigno Uria et al. “Neural Autoregressive Distribution Estimation”. In: *J. Mach. Learn. Res.* (2016).

[60] Irina Higgins et al. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *International Conference on Learning Representations (ICLR)*. 2017.

[61] Shuyu Lin et al. “Balancing Reconstruction Quality and Regularisation in ELBO for VAEs”. In: *CoRR* (2019).

[62] Kyle Young et al. “Deep neural network or dermatologist?” In: *arXiv:1908.06612 [cs, eess, stat]* 11797 (2019). arXiv: 1908.06612, pp. 48–55. DOI: 10.1007/978-3-030-33850-3_6. URL: http://arxiv.org/abs/1908.06612 (visited on 05/10/2021).

[63] Huiqi Deng et al. *A Unified Taylor Framework for Revisiting Attribution Methods*. 2021. arXiv: 2008.09695 [stat.ML].

[64] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[65] Bastiaan S. Veeling et al. “Rotation Equivariant CNNs for Digital Pathology”. In: *ArXiv* abs/1806.03962 (2018).

[66] Simon Graham, David Epstein, and Nasir Rajpoot. “Dense Steerable Filter CNNs for Exploiting Rotational Symmetry in Histology Images”. In: *IEEE Transactions on Medical Imaging* 39.12 (2020), pp. 4124–4136. DOI: 10.1109/TMI.2020.3013246.

[67] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. “CNNs on Surfaces Using Rotation-Equivariant Features”. In: *ACM Trans. Graph.* 39.4 (July 2020). ISSN: 0730-0301. DOI: 10.1145/3386569.3392437. URL: https://doi.org/10.1145/3386569.3392437.

[68] *TORCHVISION.MODELS*. https://pytorch.org/vision/stable/models.html. Accessed: 2021-05-14.

[69] *Odin Vision Code: VAE architecture*. https://github.com/alan-turing-institute/DSGApril2021OdinV/blob/main/representation_learning/module_PowerfulDecoder.py. Accessed: 2021-05-14.

[70] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[71] Richard Meyes et al. *Ablation Studies in Artificial Neural Networks*. 2019. arXiv: 1901.08644 [cs.NE].

[72] Marco Ancona et al. "Towards better understanding of gradient-based attribution methods for deep neural networks". In: *arXiv preprint arXiv:1711.06104* (2017).

[73] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. *Learning Important Features Through Propagating Activation Differences*. 2019. arXiv: 1704.02685 [cs.CV].

[74] *DeepLift ReLU Runtime Error*. 2021.

[75] Daniel Smilkov et al. "SmoothGrad: removing noise by adding noise". In: *CoRR* abs/1706.03825 (2017). arXiv: 1706.03825. URL: http://arxiv.org/abs/1706.03825.

[76] Sara Hooker et al. "Evaluating Feature Importance Estimates". In: *CoRR* abs/1806.10758 (2018). arXiv: 1806.10758. URL: http://arxiv.org/abs/1806.10758.

[77] Julius Adebayo et al. "Sanity Checks for Saliency Maps". In: *CoRR* abs/1810.03292 (2018). arXiv: 1810.03292. URL: http://arxiv.org/abs/1810.03292.

[78] Julius Adebayo et al. "Local Explanation Methods for Deep Neural Networks Lack Sensitivity to Parameter Values". In: *CoRR* abs/1810.03307 (2018). arXiv: 1810.03307. URL: http://arxiv.org/abs/1810.03307.

[79] Moritz Böhle et al. "Layer-Wise Relevance Propagation for Explaining Deep Neural Network Decisions in MRI-Based Alzheimer's Disease Classification". In: *Frontiers in Aging Neuroscience* 11 (2019), p. 194. ISSN: 1663-4365. DOI: 10.3389/fnagi.2019.00194. URL: https://www.frontiersin.org/article/10.3389/fnagi.2019.00194.

[80] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. *All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously*. 2019. arXiv: 1801.01489 [stat.ME].

[81] Aditya Chattopadhyay et al. "Grad-CAM++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks". In: *CoRR* abs/1710.11063 (2017). arXiv: 1710.11063. URL: http://arxiv.org/abs/1710.11063.

[82] Saurabh Desai and Harish G. Ramaswamy. "Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization". In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 972–980. DOI: 10.1109/WACV45572.2020.9093360.

[83] Haofan Wang et al. "Score-cam: Improved visual explanations via score-weighted class activation mapping". In: *arXiv preprint arXiv:1910.01279* (2019).

[84] Daniel Omeiza et al. "Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models". In: *arXiv preprint arXiv:1908.01224* (2019).

[85] V. Couteaux et al. “Towards Interpretability of Segmentation Networks by Analyzing DeepDreams”. In: *iMIMIC/ML-CDS@MICCAI*. 2019.

[86] *AI/ML Interpretability Github Projects*. 2021.

[87] *Captum Infidelity Implementation Difference*. 2021.

[88] Sana Tonekaboni et al. *What Clinicians Want: Contextualizing Explainable Machine Learning for Clinical End Use*. 2019. arXiv: 1905.05134 [cs.LG].

[89] Amirata Ghorbani, Abubakar Abid, and James Zou. “Interpretation of neural networks is fragile”. In: *arXiv* Lipton 2016 (2017). ISSN: 23318422.

[90] OF Ahmad et al. “Artificial intelligence and computer-aided diagnosis in colonoscopy: current evidence and future directions”. In: *The Lancet Gastroenterology & Hepatology* 4.1 (2019). © 2018 Elsevier Ltd. All rights reserved.This is an author produced version of an article published in The Lancet Gastroenterology & Hepatology. Uploaded in accordance with the publisher’s self-archiving policy., pp. 71–80. URL: http://eprints.whiterose.ac.uk/158066/.

[91] S. Lin et al. “Learning Semantically Meaningful Embeddings Using Linear Constraints”. In: *CVPR Workshops*. 2019.

turing.ac.uk
@turinginst