Using unsupervised machine learning for polyp detection in the GI tract

Mathias Kirkerd

Thesis submitted for the degree of
Master in Informatics: Technical and Scientific Applications
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2018

Using unsupervised machine learning for polyp detection in the GI tract

Mathias Kirkerd

Using unsupervised machine learning for polyp detection in the GI tract

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

Printed: Reprosentralen, University of Oslo

Abstract

Acknowledgements

my cat, if i had one

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Introduction REM

Cancer is, today, the second leading cause of death in the world, only behind cardiovascular diseases.
It is one of the leading causes of mortality worldwide, with approximately 14 million new cases in 2012. It is defined as a disease that has an abnormal cell growth with the potential to spread into other parts of the body.Contrary to normal cells, cancer cells are often invasive, and it will spread if not treated. In contrast to many other diseases cancer does not start from a foreign entity (such as a bacteria or virus), but it is often from a malfunctioning cell that starts dividing rapidly. This can happen when a cell is damaged, by for instance by radiation or other factors that damages the DNA, and the resulting damage causes the cell to uncontrollably divide. Especially in the later part of life everyone has the chance of getting cancer, and in fact everyone does. Our own body is designed to detect and remove cells that are prone to divide uncontrollably. Unfortunately this system is not perfect, and the immune system can in some cases overlook cells that are cancerous.

1.1.2 Statistics on cancer REM

The western (or modern) world has been in a battle against cancer, and despite a lot of new cures/innovations it is still one of the deadliest killers in the world. *The most common types of cancer in males are lung cancer, prostate cancer, colorectal cancer and stomach cancer.***stewart2014world**

1.1.3 colorectal cancer REM

You can get cancer in every major organ, but some types of cancer are more common than others. For instance cancer in the gastrointestinal tract (GI) is one of the more common places to get cancer. This is just behind x, and it has a mortality rate of x in the first y years. We often call this 5 year survival rate for z. This is the standard way to measure the life expectancy of a patient diagnosed with cancer.

1.1.4 polyps REM

The colorectal cancer often starts in polyps. Polyps are, polyps do.

1.1.5 preventative matters and early detection REM

-colonoscopy
-mri
-pillcam
A good way to fight cancer is to detect and remove it early, or some times remove areas with a high chance of getting cancer. We classify cancer in to x stages, and the stage the patient are in often determines the chance you have for survival. In general, the earlier you find the cancer, the more likely it is that the patient will survive. And as mentioned above, the colorectal cancer often starts in these polyps. A crucial stage to prevent cancer lies in the early removal of there polyps. Reports shows x about this

*4 stages maybe? *early detection *survival rate

Because of this the ability to find, and remove colorectal polyps is great for preventing cancer in the GI tract.

colonoscopy/Ontonoscopy In the most common way to look for polyps in the GI tract is to use a medical team, and perform a colonoscopy or Ontonoscopy colonoscopy is preformed with a camera-stick that is inserted in to the GI tract through the patients anus.
Onoskopy is the same procedure, only the camera is inserted orally.

Advantages

- Accuracy: The use of a camera controlled by the doctor gives him/her the opportunity to stop at any anomalies.

- Quick results: Since the doctor is doing the procedure the result is given live.

Disadvantages

- Expensive: The cost of the doctor and the nurses needed is often high, especially on a routine check.
- Invasion of privacy: Getting an Colonoscopy or Onoskopy is a

MRI MRI (Maggnetic stuff) is the act of taking pictures blabla blabla
MRI (Maggnetic stuff) is the act of taking pictures blabla blabla
MRI (Maggnetic stuff) is the act of taking pictures blabla blabla
Advantages

- This is why mri is good
- This is why mri is good

Disadvantages

- This is why mri is bad
- This is why mri is bad

pillcam In the last 3-4 years there have been testing and development on the pillcam project EIR. Machine learning has, through many of the earlier projects, got the detection rate for the polyps up to x%
Advantages

- This is why pillcam is good
- This is why pillcam is good

Disadvantages

- This is why cam is bad
- This is why pillcam is bad

1.1.6 Simulas contribution to the pillcam project REM

Simulas EIR
* CAD ACD (computer aided diagnosis, Automated computer diagnosis)

1.2 Goal / Problem

1.2.1 pillcam project has lots of data, can be used to train an unsupervised network REM

The video sequence from the pillcam can last several hours resulting in thousands of images, combined with colonoscopy images we have over 60 000 unlabelled images at our disposal.

1.2.2 Use Unsupervised learning as a pre-processing tool REM

The act of finding an algorithm that can enhance the training data. Either through removing artifacts or virtually enhancing resolution.

1.2.3 use Unsupervised-NN/GAN for image enhancements so that a NN can train better REM

* Now that we got a lot of tests, why not unsupervised As mentioned, simula research centre has done a lot of testing on the pillcam project.

* We know that we can get some results using a neural network * Can this be done unsupervised? * Can it be done in a fashion that is better than S-ML

REM

1.3 Scope and Limitations

1.3.1 Use Unsupervised NN to find polyps REM

1.3.2 Use Unsupervised NN for pre-processing REM

* Something about earlier research already got far, so the scope is mainly unsupervised deep learning. * (and how to generalise it?) *REMegression

1.4 Research method

1.5 Related work

1.6 Outline

The rest of the thesis is structed as follows:
Chapter 2 - Background
*talk about cancer *talk about machine learning. *how to use ML on the pillcam video? **Chapter 3 - Me doing stuff**
Chapter 4 - Me got and present result
Chapter 5 - Me saying result was good A+

Chapter 2

Background

2.1 Cancer and polyps

2.1.1 What we are looking for REM

Different types of disorders. Polyp is harmless, but if left untreated it can become cancerous Pictures is from the pillcam project, kvasir dataset.

2.1.2 images from pillcam, and what we are looking at/for REM

2.2 Naive Methods REM

Now that we have an idea of what we are looking for we can first turn to some more naive methods for detecting anomalies, and for enhancing the images.
The field of image processing has been researched since

Using some of the classic methods in image processing we can see if

We often describe the method in to two groups of information: First and Second order statistics.
First order: First order statistics does not take in to account the relative positioning of the pixels in the image, and because of this, gives much less information than the second order statistics.
Example of First order statistics is often what information we can get out of a histogram. This can be scewness, variance, and mean value.

Second order: Second order statistics takes in to account the relative positioning of the pixels in the image. We can calculate the GLCM matrix and get a much more detailed view of the image.

2.2.1 GLCM

A GLCM (Grey-level co-occurrence matrix) is a matrix that is used when examining the spatial relationship of pixels in a texture. The calculation of a GLCM gives us how often pairs of pixels with spesific values and a specified spatial relationship occur at a given place in an image.

2.2.1.1 Algorithm

For simplicity we use only greyscale in this example: The algorithm starts by

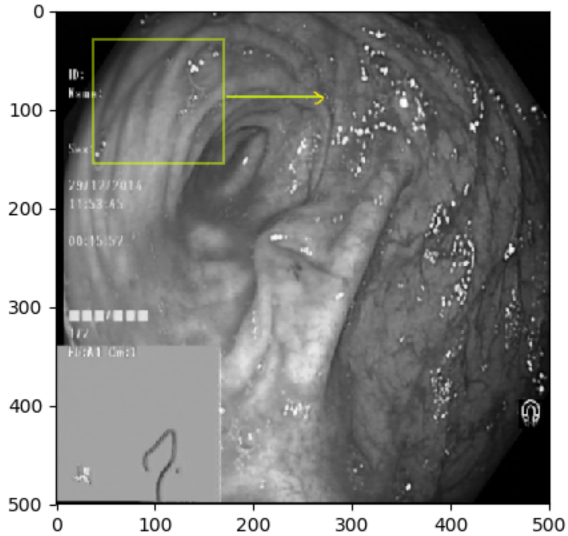

Figure 2.1: GLCM capturing features

running a sliding window over the image, often with a stride, and for each stops calculates the spatial relationship between each pixel specified. The result can be something like this figure where we can read out the most likely neighbouring pixel. The darker colours on in the matrix is indicating that we often have a jump between, for instance pixel-value of 1 and a pixel-value of 4, but no from 1 to 1.
With this information we can get a naive pattern-recogniser.

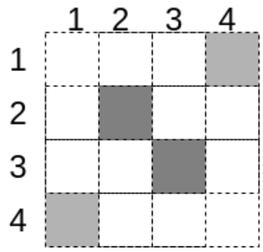

Figure 2.2: GLCM Matrix

2.2.1.2 Other uses

Besides for the pattern recognition we can use the GLCM to get the information on:

- **Contrast** is the difference in luminance or colour in the picture. We would expect low contrast in the "background" and higher contrast around edges and irregular objects.
- **Homogeneity** is how similar a local area is to itself
- **Variance** σ^2 , is directly a measure of "roughness"
- **Mean** value of a GLCM can give us areas with higer or lower pixel values. Good way to find polyps if they are lighter than the tissue around.
- **Entropy**
- **Energy**

2.2.2 Edge detection

Using Edge detection in is another viable way to look for polyps.

2.2.2.1 Algorithm

For each pixel look at the neighbouring pixel, if

$$abs(p_a - p_b) > tresh$$

then mark pixel as an edge pixel.

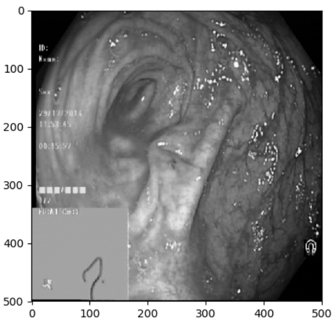

Figure 2.3: Original image

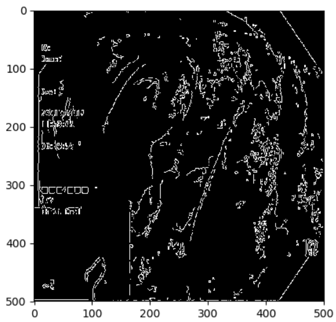

Figure 2.4: Edges of the picture

2.2.3 Hough Transforms

Using for instance Canny edge detection we can get a better view of where the potential border of the polyp/anomaly is. (As shown in

A hough transform can i theory have many/any shape(s), and together with edge detection, we might find some of the polyps this way.

2.3 Machine Learning

Machine learning is a very broad term, but can i short be summarised by:

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with the experience E. **MitchellTomM1997Ml**

Here we have a couple of parameters:
E text about e
T text about t
P text about p

From this we see that the goal of machine learning is to improve some performance P with experience. **might here talk about different tasks ML can do?**

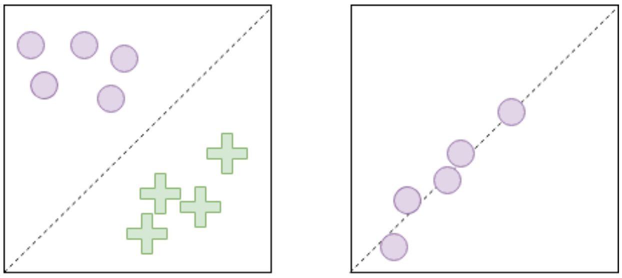

Figure 2.5: Left: Example of binary classification. Right: Example of regression

2.3.1 Supervised & Unsupervised machine learning

We often divide machine learning in to two (diffuse) categories: supervised and unsupervised.

Supervised learning: is the act of training with data that has an answer or a label. The learning algorithm can get supervision while training on the task. An example on a supervised task is to recognise handwritten numbers, or differentiate between dogs and cats. The task is supervised if the images comes with the correct label in the data set. These examples are typical classification examples, where the task is to identify the right group to classify the data to A simpler classification assignment is binary classification, where the target is (often) yes or no. Examples for binary classification is if an email is spam or not, is a car Norwegian or International. In the last example the classification changes from binary to multi-class if you sort the cars on every nationality, and not just Norwegian/non-Norwegian.

Another type of supervised learning is regression. This is the act of prediction given prior data. Examples of regression is everything from prediction of stock prices, to house prices in an area, to

Unsupervised learning: is the act of training without any supervision, on the sense that we do not give the algorithm the answer to the training data set.

Since we do not have categorised data in unsupervised learning, we often Types of unsupervised learning can for instance be clustering, the act of sorting data based on similarity. An example of this can be if you want to sort plants based on species, or you are detecting anomalies in a dataset. Unsupervised learning can be used for PCA or other dimensionaly reduction methods.

A third method to used unsupervised learning is the adversarial route, where you use machine learning to make similar looking data to the original

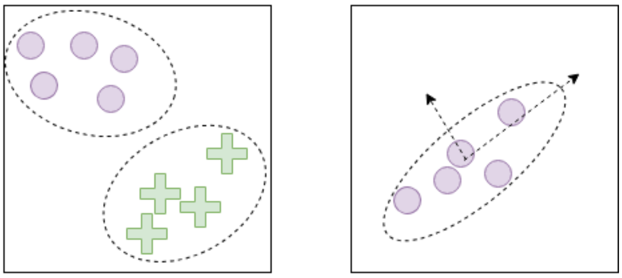

Figure 2.6: Left: Example of binary clustering. Right: Example of principal component analysis

data set.

In the description of supervised vs unsupervised we looked at a specific branch of machine learning: Classification. Classification is, as the name implies, the task of getting data sorted in to groups of similarity.

- subsfication
- r to the pillcam projression
- transcription/translation
- de-noising /finding missing inputs

2.3.2 Types of machine learning

There are a number of different machine learning algorithms.

Machine Learning				
Supervised Learning		Unsupervised Learning		Reinforcement Learning
Classification	Regression	Clustering	Dimensionality reduction	-
Support vector machines K nearest neighbours Neural networks	Linear Regression Decision trees Neural networks	K means clustering Hidden Markov models Neural Networks	PCA	SOMething

Table 2.1: Machine leaning types

K nearest neighbours
Talk about KNN

Linear Regression
How to regress linearly

Support vector machine
SVM and 2 class

Others?
Other important ones to talk about?

Neural networks
NN is future
own chapter

2.4 Neural Networks

2.4.1 How it works

2.4.2 Convolutional neural networks

2.4.3 Advaserial neural networks

2.4.3.1 UCNN?

2.5 The problem at hand

Now that we have the definition of machine learning we focus on the task at hand; finding polyps. In an ideal world we have a Classification problem with only two classes: Non-polyp and polyp.

- SVM
- CNN
- random forests
- knn

2.6 In painting

We have discussed the importance of good input data, and the potential benefits to resource usage and ease of making a good model. So a priority when it comes to image classification is to have data without anomalies and other areas that can be interpreted as a feature for the classifier. In a machine learning perspective, the data is best if it has the same structure, and is In painting is the process of reconstructing lost or deteriorated parts of images and videos.

From prior papers on polyp detection in the GI tract we have clear results that the black corners, and the green squares trigger a big activation when it comes to classifying images. From 's paper, we can see that the activation map on a regular image gives very high result on, in addition to the polyp, the corners and the green sqare.

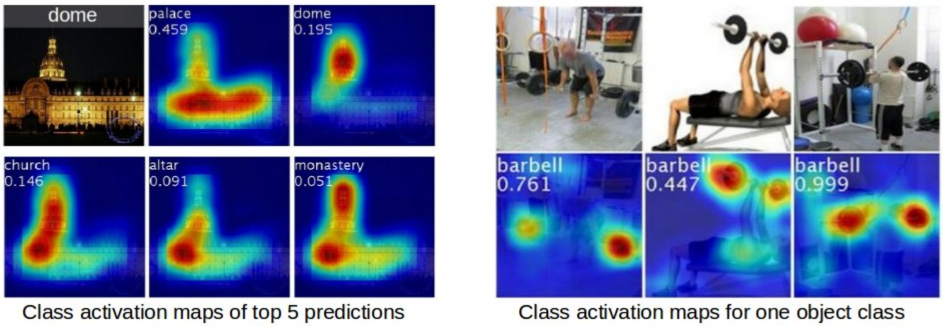

Figure 2.7: Using X's activation map we can see that the edges triggers unwanted activations

In addition to sqares and edges, we also have the problem that parts of the image is over saturated at points where the light from the led is reflected directly back to the camera. Another problem is when the camera captures images that are too close to the wall. Both of these scenarios creates patches where the saturation is maximum. in an ideal scenario the image would have no pixel values at max, and as little frame as possible. We therefor want to make a tool that can help us with this.

2.6.1 Naive methods for In painting

Inpainting is not a new area of research, as it has been around since Because of this there are many naive methods that gives good inpaintings.

2.6.1.1 Textured syntesys based on image inpainting

2.6.1.2 MOARE

2.6.1.3 MOARE

As we can see from this, there are a lot of old methods that can give approximations. We can also conclude that none of these methods are perfect.

Figure 2.8: we have two different types of saturation: the reflected area in the top part of the image, and the right side of the image.

We will therefore look at methods that takes learning in to use.

2.6.2 Using machine learning for inpainting

As discussed earlier, machine learning is using prior experiences to make decisions given the problem at hand. It is also worth mentioning that we do not need labeled data, since we are in a way looking at a global average of every image both with, and without polyps. We are therefor insetiviced to use an Unsupervised approach. Since machine learning is learning from a training set, it is important that the training set contains as little as possible of the features we want to remove.

Because of this the first thing we need to do if we are going to mask out corners and sqares, is to limit the training set to only contain cropped, non-sqare images.

Now that we have better images to train our data with, we need the correct algorithm. We have already seen unsupervised learning aproaches in chapter

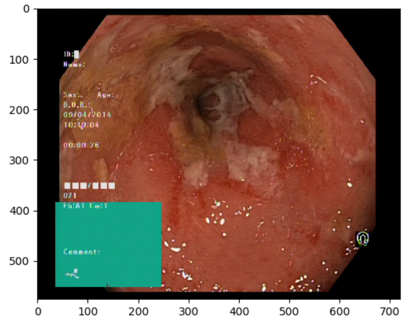

Figure 2.9: Original image with black padding

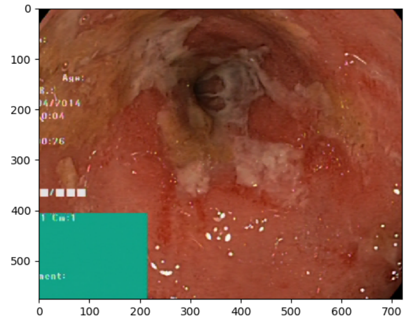

Figure 2.10: Black edges cropped away + 8% zoom

Figure 2.11: Here we have an example on how we would make an image better to train on. This is not representative of the training, since we only use images without the green square under training

2.6.2.1 Algorithm

Text about presenting UML, and stuff.

2.6.2.2 Autoencoder

2.6.2.2.1 This is explaining Autoencoders, put me in the right place As we recall from earlier, an autoencoder is a type of neural network that tries to output a recreation of the output.
We can do this by having an encoder, $h = f(x)$, connected to a decoder, $r = g(h)$. An autoencoder has the job to set $g(f(x)) = x$ over the whole input, but in most cases this is not a practical program. We often gives the autoencoder the restriction that it has to map the input through a latent space that has a smaller dimension than the input dataset.
This is called an undercomplete autoencoder.

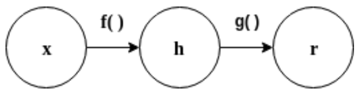

Figure 2.12: The general structure of an autoencoder, mapping **x** through **h** to an output **r**.

As with supervised classifiers we can use gradient decent to optimize the model. This is because we are trying to recreate the input $\mathbf{x}$ from out output $\widetilde{\mathbf{x}}$

This can simply be done by minimizeing the loss function

$$L(\mathbf{x}, g(f(\mathbf{x}))) \tag{2.1}$$

with for instance MSE with gradient decent.

Now we can transfer this to a more relevant example by making an image as input and use convolutions to reduce the dimensionality in the encoder and increase the dimentionality in the encoder.

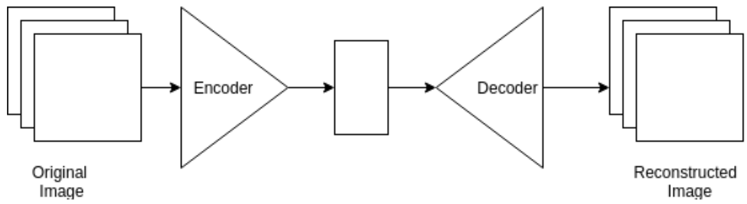

Figure 2.13: Convolutional autoencoder with an RGB image as input, and the reconstructed image as output.

test

As we recall from earlier, an autoencoder is a type of neural network that tries to output a recreation of the output.
We can use this for inpainting by setting the algorithm to train on images with areas cropped away. There are a couple of different way we can train an autoencoder to do this.

Denoising Autoencoder with MSE loss:
The simplest way to train the autoencoder is to first take the trainingset $\mathbb{X}$ and make an augmented copy $\widetilde{\mathbf{x}}^{(i)}$ for every data point in $\mathbf{x}^{(i)}_{\sim\mathbb{X}}$.
Here $\widetilde{\mathbf{x}}$ is a copy of x with random areas masked.

Now we minimize the loss function

$$L(\mathbf{x}, g(f(\widetilde{\mathbf{x}}))) \tag{2.2}$$

over the whole image.

With this approach the autoencoder learns to fill in the blank spots with plausible data, without changing the rest of the image. One problem by this approach is that we do not want the rest of the image to change for obvious reasons, and the alogrithm as it is here has the flaw that it will change all the pixels in the image, at least to a minor degree.

This can be somewhat fixed by only taking the augmented parts, and pasting the directly in to the image. This will leave most of the original image, except for the parts that was cropped randomly.

Denoising Autoencoder with :
If we take what we learned from 2.6.2.2.1, we can make a more optimal autoencoder: Rather than taking a loss like

$$L(\mathbf{x}, g(f(\widetilde{\mathbf{x}}))) \tag{2.3}$$

over the whole image, we can rather just focus on the parts that matters, namely the cropped areas.

If we add the cropped image to the output from the autoencoder to make an image image, we can use this new image to train out loss. For most of the image, the loss will be zero, since the only part that is changed is the cropped area. We can also make a new loss that is more optimal for the task at hand.

$$MSE_{crop} = \frac{1}{n} \begin{cases} (\widetilde{\mathbf{x}} - \mathbf{x})\ if\ \widetilde{\mathbf{x}} \in \mathbf{x}_{crop} \\ 0\ else \end{cases} \tag{2.4}$$

Where $\mathbf{x}_{crop}$ is the area that was cropped away from the original image and n is the number of pixels in that area.
With this modified MSE we are assured that only the pixels in the cropped area is changed with gradient decent, and we save a lot of computation as an added bonus.

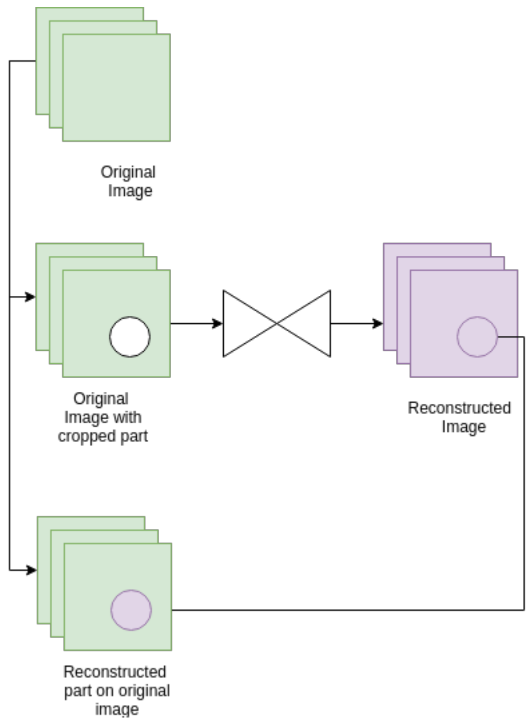

Figure 2.14: Final result of the autoencoder used in the testing

2.6.2.3 Explaining GANs, this will be moved

2.6.2.3.1 This is explaining GANS, put me in the right place Now that we have looked at autoencoders we can take it a step further. generative advaserial models can be used as a generator of new data, and can have som reseblance to autoencoders 2.6.2.2.1, especially variational autoencoders

The difference lies in that advaserial networks is based on game theoretic scenarios in which a generator network is compeating agenst an advasery. The generator produces samples $x = g(z;\theta^{(g)})$, where g is the network given the weights θ. Then the discriminator network predicts if a sample is drawn from the dataset or from the generator. More spessific, it gives a probably given by $d(x;\theta^{(d)})$, and determins if x is from the generator or the data-set. Since we have two networks compeating agenst each other we can look at this as a Zero-sum game with the generators payoff is determined by $v(\theta^{(g)},\theta^{(d)})$, and the discriminators payoff is determined by $-v(\theta^{(g)},\theta^{(d)})$. *$v$ is here a function that is determined by both the sucsess rate of the discriminator and the generator, the most common used is*

$$v(\theta^{(g)},\theta^{(d)}) \; = \; \mathbb{E}_{x\sim p_{data}} \log d(x) + \mathbb{E}_{x\sim p_{model}} \log (1 - d(x)) \tag{2.5}$$

as derived from Goodfellow et al.

Lets look at a gan in detail.

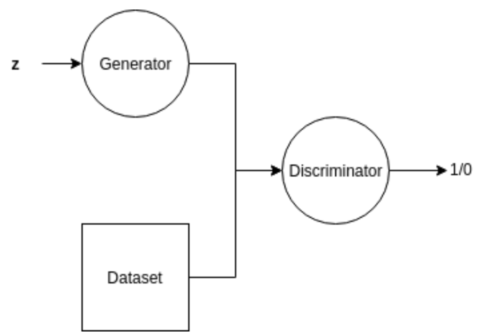

Figure 2.15: The idea behind a GAN. Here the generator saples from a random (Gaussian) distribution and generates samples that the discriminator classifies as real or fake

2.6.2.4 Contextencoder

2.6.2.5 CCgan

2.6.2.6 PixelCNN

Chapter 3

Methods

3.1 Using machine learning for preprocessing

When you work with machine learning a lot of the job is to make the data as clear as possible.
Imagine that you want to do something simple as reading an analogue clock. The s̈traight forwardẅay to do it is to make a convolutional neural network to look at the dials. This will require a much more complex network compared to if you could convert the angle of the dials to degrees and have that as an input to your model.

Figure 3.1: A clock needs a more complex network compared to just the degrees

The trick is often to make the data as refined as possible. Further some of the techniques used is described.

3.1.1 Removing borders

A normal picture from the kvasir dataset, as seen earlier, has more to it than just the picture we are after. As mentioned over the less clutter we can get in our data the better. So one of the first thing we should do is to remove the frame. The biggest reason why this is done is because of the lack of information in the black, uniform pixels. Since the black area differs in every image, the area is

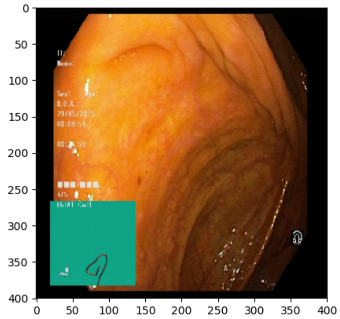

Figure 3.2: Original image with no edges removed

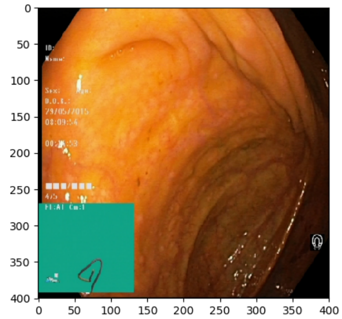

Figure 3.3: Edges of the image removed

saved as a feature. This feature contains no relevant information for the analysis, so it is easier if it is just discarded.

The image is cropped using 2 simple algorithms, first we run a morphological opening with a 5×5 kernel on a black/white copy of the image. This will remove any unwanted artifacts and/or text in the black area. Now that the border is guarantied black, we can do a simple crop where the border stops.

3.1.2 Adjusting brightness and contrast

TODO LATER.

3.1.3 Removing artifacts and saturated spots

In addition to the border areas there are also other areas that bear little to no information. This is the bright spots where the light from the camera-pill is saturating the CCD in the pill. White areas like this can be treated as a feature, and as with the border, this feature contains no relevant information. In the article *Computer-Aided Screening of Capsule Endoscopy Videos* from Zeno Albisser it is proposed a way to remove bright areas by using a horizontal gradient over the bright areas. In the loading of the images a similar treatment of the images is done.

3.1.4 Using a Contextencoder to predict image parts

At this point we have used naive methods to enhance the data. Another big part of the image has not been mentioned this far, and that is the green square many of the images contains.

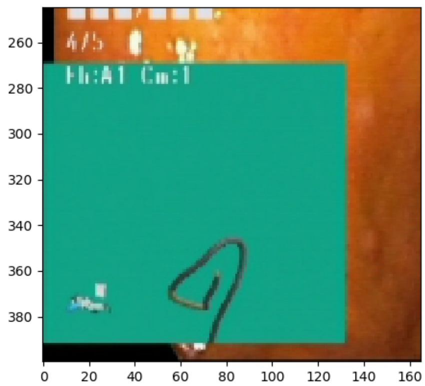

Figure 3.4: A typical example of a green square containing information about where in the GI-tract the image is taken from

A way to remove the square is to continue to use a naive method, perhaps with a horizontal gradient or a similar technique. However we can use a convolutional neural net to try to predict what would be behind the area.

3.1.4.1 Using a GAN

As described earlier in the thesis we can use an adversarial network to generate images. The general Contextencoder has three main parts: Encoder, decoder and a Discriminator.

First an (often random) area is removed from the original image, and is stored as a copy. Then the decoder takes the reminding image and compresses it to a latent space. From there the encoder tries to build a new image that is the same size as the missing peace.

After the peace is made it is sent to the discriminator, together with the original peace. The discriminator takes both the generated image as well as the original image and tries to give a prediction if it believes if the part it got was a generated or original image. Training and evaluation is the same as described in the general description of how an generative adversarial network is working.

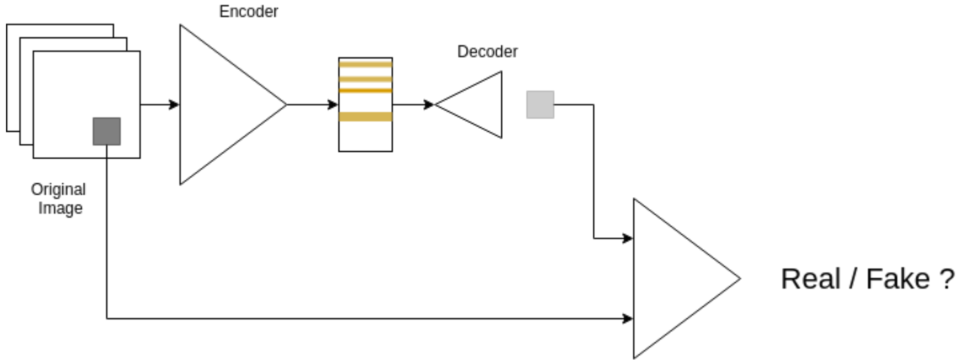

Figure 3.5: A simple Contextencoder

3.1.5 Using a Variational Autoencoder to train the adversarial network

This chapter talks about the use of a Variational autoencoder to train the generator.

3.1.5.1 Setup of the GAN

Using the Contextencoder described we made 3 similar programs for image prediction.

Random masker: The first program is designed to work with any data as long as you train the weights on sufficient data. The random masker trains on images where the area masked is uniformly distributed throughout the images. This approach is highly general, and will give a better masking on nongreen squares, compared to the corner masker.

Corner masker: The corner masker is designed to only mask the bottom left corner where the green square is located, this means that it is better at finding the area behind the green square, but it is much worse at predicting any other part of the image.

Categorical masker: The categorical masker is a mix between the two prior models. The categorical masker divides the image in to 9 different equally placed squares, and only masks those areas. The result is a mix between a general approach and a specific corner approach.

3.1.5.2 Result of the GAN

The result of the trained weights can be found at It is part of a pygame demo, that can take any image and predict the area marked, based on the loaded weights.

Random masker: As we can see from the images, the area is not perfectly masked, but these weights can be used on any part of the image with the same result.

Corner masker: This is the prediction given the weights trained only on the bottom left corner. This result gives a better prediction than the random masker, but can not be used other places in the image

Categorical masker: DONT KLNOW yet

3.2 Making the dataset larger

One of the reasons why machine learning has become a hot topic the last years is the amount of data stored the last century.

Chapter 4

Implementation

Chapter 5

Result and Discussion

Chapter 6

Conclusion

Chapter 7

Future Work