Time series analysis for medical videos

Henrik Løland Gjestang

Thesis submitted for the degree of
Master in Computational Science
(Imaging and Biomedical Computing)
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2019

Time series analysis for medical videos

Henrik Løland Gjestang

Time series analysis for medical videos

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

Printed: Reprosentralen, University of Oslo

Abstract

This is the abstract part. How does it look?

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this project we aim to design and develop a system for analyzing medical videos from a camera pill, as seen in Figure 1.1. The pill is swallowed and records video of the entire digestive system The goal is to be able to detect different irregularities in the patients digestive system, like a colon polyp, Chron's disease, Colorectal cancer, etc. by using video object tracking, object detection, machine learning or other relevant tools.

Neural networks models that we would like to explore further for this purpose are Convolutional neural networks (CNN), Recurrent neural networks (RNN), Capsule neural networks, Long Short-Term memory networks and more.

The main idea is to go beyond image-based methods and also exploit the time factor of the data. The videos we will be using for this is delivered by Bærum Hospital, and is carefully labeled by using tools such as described in the paper **ExpertDriven15**. In this paper **ExpertDriven15** presents a semi-supervised method to gather the annotations in a easy and time saving way **ExpertDriven15**.

Figure 1.1: Illustration of how such a camera pill could look like **PillCamCamera**.

Colorectal cancer (CRC) is the third most common cause of cancer mortality for both men and women **CancerStatistics10**, and it is a condition where early detection is of clear value for the ultimate survival of the patient. As statistics show that 15% of male and female above 50 years are at risk, the procedure is recommended on a regular basis (every 3-5 years) for the population over 50, and from an earlier age for high-risk groups. Colonoscopy is a demanding procedure requiring an significant amount of time by specialized physicians, in addition to the discomfort and risks inherent in the procedure. Traditional methods based on colonoscopy are not cost-effective for population-based screening purposes, so only about 2-3% of the target population is reached at present. The cost of a population screening program is prohibitively expensive. Colonoscopy is the most expensive cancer screening process in the US, with annual costs of $10 billion dollars ($1100 per person). In Norway we have similar costs of around $1000 per person, with a time consumption of about 1 doctor-hour and 2 nurse-hours per examination. By researching an automatic system for a camera pill the aim is to greatly increase the number of patients that can be examined, i.e., making the public health care system more scalable and cost effective, while at the same time reducing the need for intrusive procedures like "bottom-up" examinations like colonoscopy.

Chapter 2

Wireless Capsule Endoscopy

The basic technology behind the modern endoscope was developed in the early 1950s by English physicist Harold Hopkins and his student Narinder Kapany which let light travel through flexible pieces of glass, now known as optical fibers **NewMethod54**.

Before the year 2000 the only option you had to visualize the foodpipe, stomach, duodenum, colon and terminal ileum (see Figure **??** for details) was to use a fiber-optic endoscope, which is a tool with a relatively wide cable that is pushed into the bowel with as much as 50 000 optic fibers (as seen on Figure 2.1). These cables have to carry fiber optic bundles, water pipes, operations channel and control cables. Although these cables can be quite flexible there is a limit for how far they can advance into the small bowel. This method cause pain and discomfort for the patient, and there was a clinical need for an improved methods.

That is why in the year **WirelessCapsule00 WirelessCapsule00** developed a new type of video-telemetry capsule endoscope that was swallowable **WirelessCapsule00**. It could travel through the entire digestive system because it had no external wires, fiber-optic bundles or cables of any sort. The capsule travels by peristalsis[2] through the gastrointestinal tract, which takes from 10 to 48 hours, and transmit images on a regular interval to receivers attached around the outside of the patients stomach for as long as the battery allows, usually in the range 6 to 8 hours. Two example images taken by WCE are presented in Figure 2.2. By triangulating the signal strength and the location of the receivers taped on the body it is possible to roughly estimate the position of the capsule. This is however not very precise and can not tell us the rotation or direction of the capsule. Regardless, that information will not be available for us in this study as we only have access to the images themselves. Therefore we could implement an algorithm to predict which region of the degistive system the image is taken from (see section 3 and 4.0.3).

[1] Image credit: Jacaranda Physics 1 2nd Edition © John Wiley & Sons, Inc.

[2] Peristalsis is a radially symmetrical contraction and relaxation of muscles that propagates in a wave down a tube, in an anterograde direction.

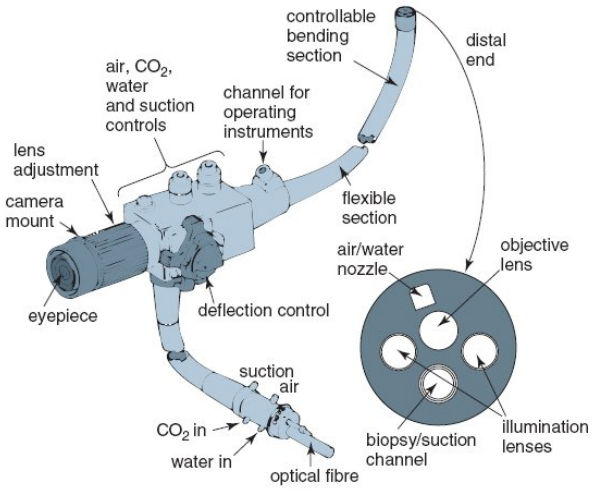
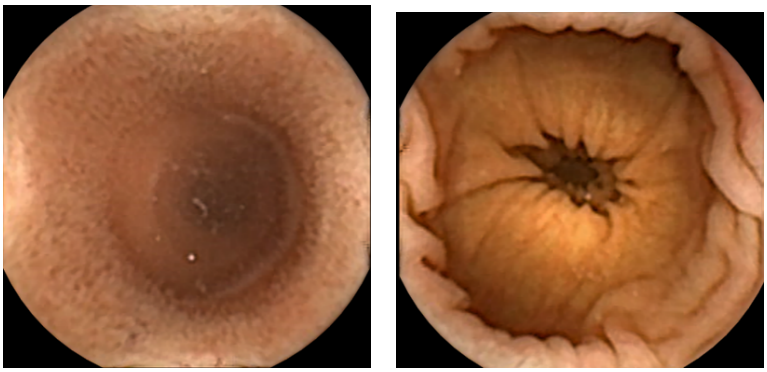


Figure 2.1: Image of a fibre optic endoscope with explanation of different parts of the tool[1].



(a) Small Intestine **(b)** Colon

Figure 2.2: Images taken with WCE[3].

[3] CC BY-SA 3.0 / Attribution to Dr.HH.Krause at English Wikipedia;
https://commons.wikimedia.org/wiki/File:Normales_Colon.PNG
https://commons.wikimedia.org/wiki/File:Dunndarm.PNG

Chapter 3

Neural network models

As apposed to using regular optic-fibre endoscopy, it can be difficult to know the location and orientation of the capsule when it is traveling through the digestive system. In a paper by **ClassifyingDigestive15** it is shown that by using Deep Convolutional Networks (DCNN) it is possible to classify the digestive organs in wireless capsule endoscopy with about 95% classification accuracy on average **ClassifyingDigestive15**. The DCNN-based WCE digestive organ classifiaction system is constructed of three stages of convolution, pooling and two fully-connected layers. This is illustrated in Figure 3 in the paper **ClassifyingDigestive15**. The main steps of this convolutional neural network are described in detail in section 3.0.1.

3.0.1 Convolutional Neural Network

One of the most used neural networks for image classifcation is the Convolutional Neural Network (CNN). The model was first proposed by **ImageNetClassification12** in **ImageNetClassification12** **ImageNetClassification12** where they trained a deep convolutional neural network and used it to classify 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes with top-1 and top-5 error rates of 37.5% and 17.0% which far surpassed all other models at the time. Next we will get into a bit of the details of a CNN.

Convolution layer

The first step in a convolutional neural network is to extract features from the input image. This is done to preserve the relationship between pixels by learning image features using filters, or *kernels*. As a result, the network learn filters that activate when it detects some specific patterns or features.

The convolution of f and g is written as $f * g$, and is defined as the integral of the product of the two functions after one (usually the filter) is reversed and shifted.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \tag{3.1}$$

Non Linearity (ReLU)

Rectified Linear unit function, known as simply ReLU, is an activation function represented by equation (3.2). It sets all negative numbers to zero, by discarding them from the activation map entirely. In this way, ReLU increases the nonlinear properties of the decision function and thus of the overall network without affecting the receptive fields of the convolution layer.

$$ReLU(x) = max(0, x) \tag{3.2}$$

Pooling layer

Pooling layers are applied to reduce the number of parameters when the images are considerably large. Spatial pooling, or merely down sampling, reduces the dimensionality of each image but it keeps the important information. The most used down sampling is max pooling. It extracts the largest element from the rectified feature map and thus reduces computational complexity of the algorithm. In addition average pooling is also frequently used, this method computes the average value of the input map. The input-output model is denoted as:

$$y_i = f(pool(x_i)) \tag{3.3}$$

Fully-connected layer

In a FC-layer every neuron in one layer is connected to every neuron in the previous layer. It is here the high-level reasoning is done. The activation function in the neurons is a *sigmoid* or *tanh* function.

$$f(z) = \frac{1}{1 + exp(-z)} \quad \text{or} \quad f(z) = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{3.4}$$

At the end of FC-layer we have an activation function such as softmax (equation 3.7) to calculate probability of the predicted classes.

Feed Forward

In the feed forward algorithm input image will be processed through all the layers in the neural network. The first layer will be a convolution layer, containing K filters F_i^1, $i = 1, ..., K$, of size $k \times k$ and a bias b^1. The image will be convoluted with each filter, and the bias is added.

$$\hat{z}_i^l = I * \hat{F}_i^l + b^l, \tag{3.5}$$

where $*$ (asterisk) is the convolution operator in equation 3.1. The final output of each convolutional layer l is a^l,

$$\hat{a}_i^l = f(z_i^l), \tag{3.6}$$

where f represents the ReLU activation function. After going through the convolution layer, the next layer could be a pooling layer, which will reduce the spatial dimensionality either by using the max value or the average value. Before getting our final output $\hat{y}$, we need to collect the outputs from all the filters, which will be an input to a fully connected layer. The fully connected

layer use the softmax activation function to classify the input image, much like a neural network would. The softmax function is an accepted standard probability function for a multiclass classifier **NotesBackpropagation16**. The total sum of the probabilities will always add up to 1 when using softmax.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, ..., K. \tag{3.7}$$

To calculate the error of the forward propagation it is common to use cross-entropy error function.

$$C(\hat{y}) = -\sum_{i=1}^{N} t_i log(y_i) \tag{3.8}$$

Back propagation

Starting from the last layer L, we calculate the derivative of the loss function (function 3.8) with regards to the activation function in order to update the weights. Computing the gradient of the loss function yields

$$\frac{\partial C}{\partial y_i} = -\frac{t_i}{y_i} \tag{3.9}$$

We also require the gradient of the output of the final layer y_i with regards to the input z_k^L of the activation function (equation 3.7)

$$\frac{\partial y_i}{\partial z_k^L} = \begin{cases} y_i(1 - y_i), & i = k \\ -y_i y_k, & i \neq k \end{cases} \tag{3.10}$$

Now with regards to z_i^L

$$\begin{aligned} \frac{\partial C}{\partial z_i^L} &= \sum_{k}^{N} \frac{\partial C}{\partial y_k} \frac{\partial y_k}{\partial z_i^L} \\ &= \frac{\partial C}{\partial y_i} \frac{\partial y_i}{\partial z_i^L} - \sum_{k}^{N} \frac{\partial C}{\partial y_k} \frac{\partial y_k}{\partial z_i^L} \\ &= -t_i(1 - y_i) + \sum_{k \neq i} t_k y_i \\ &= y_i - t_i \end{aligned} \tag{3.11}$$

And finally with regards to the weights

$$\frac{\partial C}{\partial w_i j^L} = (y_i - t_i) a_j^{L-1} \tag{3.12}$$

where $\hat{a}_j^{L-1}$ is the vectorized output from the previous layer. From here, we will propagate the error throughout the layers. The error with regards to the input a_i^L to the fully connected layer is:

$$\delta^{L-1} = \frac{\partial C}{\partial a_i^L} = \sum_{i}^{N} (y_i - t_i) w_{ji}^L \tag{3.13}$$

Thus the error is propagated backwards through each layer. If max pooling was used in a pooling layer, the error will only be propagated to the input that had the highest value in the forward pass. The other values will be set to zero. If average pooling was used, the error is averaged in the backwards pass. In equation 3.13 a^l is the output of a convolutional layer l. Since a convolutional layer is always preceded and followed by a activation layer, the input to layer l is $a^{l-1} = \sigma(z^l)$. Now consider the error with regards to z^l.

$$\begin{aligned}
\delta_{ij}^l &= \frac{\partial C}{\partial z_{ij}^l} \\
&= \sum_i^{'} \sum_j^{'} \frac{\partial C}{\partial z_{i'j'}^{l+1}} \frac{\partial z_{i'j'}}{\partial z_{ij}^l} \\
&= \sum_{i'} \sum_{j'} \delta_{i'j'}^{l+1} \frac{\partial (\hat{W}\sigma(z^l) + b^{l+1})}{\partial z_{ij}^l} \\
&= \delta^{l+1} * ROT180(w^{l+1})\sigma'(z^l)
\end{aligned} \tag{3.14}$$

Having found the error, the gradient of the cost function with regards to the weights is

$$\frac{\partial C}{\partial w_{ij}^l} = \delta_{ij}^l * \sigma ROT180(z_{ij}^{l-1}) \tag{3.15}$$

Chapter 4

Other video processing tools

We will go through some other methods not directly related to neural networks but which we think may come in handy for my thesis later on.

4.0.1 Object tracking

Object tracking is one of the harder problem to overcome in computer vision and is key to achieving good results in endoscopic video analysis. Tracking algorithms are developed to determine the movement of the object or objects in each video frame. The algorithm has to take into account the dynamic environment such as differences in lightning, occlusions and scaling changes. Also the absence of any prior knowledge to the object and its position further increase the complexity of the problem. **DeepReinforcement17** proposed an approach for visual tracking in videos that learns to predict the bounding box locations of a target object at every frame in the paper **DeepReinforcement17 DeepReinforcement17**. While other models depends on the capability of a CNN to learn a good feature representation for the target location in the new frame, which means that the model only tracks properly if the target lies in the spatial vicinity of the previous prediction. This is not always the case for WCE videos, where the lens of the camera can suddenly and unpredictably rotate towards the wall of the intestine. This method integrates convolutional network with recurrent network, and builds up a spatial-temporal representation of the video which means that the model is able to predict the target object's location over time.

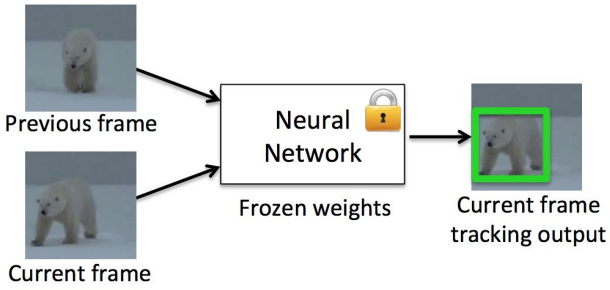


Figure 4.1: Illustration of how object in two frames is tracked with a bounding box[1].

Our hope is that by implementing an object-tracking algorithm we can use it to classify irregularities in the colonoscopy video, and then track that object in the later frames until it disappear out of frame. This will hopefully help with reducing the robustness of the network so that the classifier will not have to check every frame for irregularities.

4.0.2 Segmentation

Image segmention is the process of partitioning a image into multiple segments of pixel, usually each segment describing some feature of the image or an entire object or class of objects. The goal of segmentation is to simplify the image and make it easier to analyze or further process. **UNetConvolutional15** propose a method in the paper **UNetConvolutional15 UNetConvolutional15** for using a network and training strategy that relies on the strong use of data augmentation to use the available labeled samples more effieciently. This network outperform the old method of sliding-window-convolution by a great deal. They extend the "fully convolutional network" **FullyConvolutional15** such that it works with very few training images and yields more precise segmentations. The way this is achieved is to supplement a contracting network by successive layers, where instead of using pooling operators, upsampling operators are used. This means that these successive layers increase the resolution of the output. The high resolution features from the contracting path are combined with the upsampled output to localize objects and with that a convolution layer can then learn to produce more precise output based on this information.

Another important feature in this architecture is that in the upsampling portion of the network there is also large number of feature channels. These channels allow the network to pass on context information to the higher resolution layers.

A common problem in training neural networks are too little labeled training data. This is also the case for us. We require a lot of medical data, and personell with the expertise to correctly label our data are of high demand and they usually have very little time for projects like these. This is why **UNetConvolutional15** use different methods of data augmentation to generate more training data. They apply elastic deformations to the available images, and this allows the network to learn invariance to such deformations without the

[1] https://www.learnopencv.com/goturn-deep-learning-based-object-tracking/

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	**0.9203**	**0.7756**

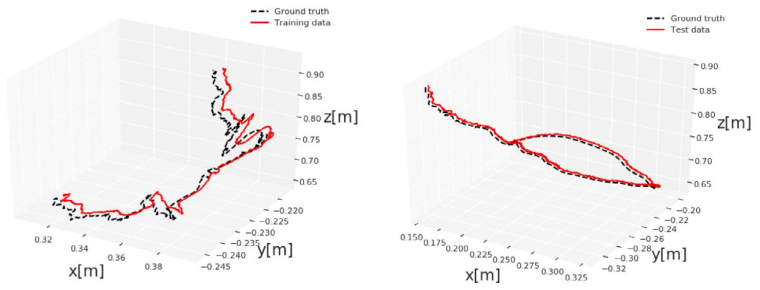
Table 4.1: Segmentation results on the ISBI cell tracking challenge in 2015.

need to see these transformations in the annoted image corpus. Which is particular important in biomedical segmentation since deformation used to be the most common variation in tissue and realistic deformations can be simulated efficiently **UNetConvolutional15**. By doing this **UNetConvolutional15** were able to achieve very good results (Table 4.1).

4.0.3 Mapping

As mentioned in section 2, a concern when processing the images taken with a WCE is not having the spatial data you get when using a normal fiber-optic endoscope. This is why **DeepEndoVO18** has recently made substantial progress in converting passive capsule endoscopes to active capsule robots, enabling more accurate, precise, and intuitive detection of the location and size of the diseased areas by developing reliable real time pose estimation functionality of the capsule with RCNN's[2] **DeepEndoVO18**. See Figure 4.2 for an example.

This architecture uses inception modules for feature extraction and a RNN for sequential modelling of motion dynamics to regress the robot's orientation and position in real time. By taking multiple of RGB Depth images with timestamps it can calculate the 6-DoF pose of the capsule without the need of any extra sensors. For obtaining the depth images **DeepEndoVO18** use the shape from shading (SfS) technique of **ShapeShading94 ShapeShading94**. This model outperforms state-of-the-art models like LSD SLAM and ORB SLAM.



(a) Training data vs ground truth. **(b)** Test data vs ground truth.

Figure 4.2: An example of Deep EndoVO accuracy **DeepEndoVO18**.

[2]Deep recurrent convolutional neural networks