Comprehensible Reasoning and Automated Reporting of Medical Examinations Based on Deep Learning Analysis

Steven Alexander Hicks
Simula Research Laboratory, Norway
University of Oslo, Norway

Konstantin Pogorelov
Simula Research Laboratory, Norway
University of Oslo, Norway

Thomas de Lange
Oslo University Hospital, Norway
University of Oslo, Norway

Mathias Lux
University of Klagenfurt, Austria

Mattis Jeppsson
ForzaSys AS, Norway

Kristin Ranheim Randel
Cancer Registry of Norway

Sigrun Eskeland
Department of Medical Research
Bærum Hospital
Vestre Viken Hospital Trust, Norway

Pål Halvorsen
Simula Metropolitan Center for
Digital Engineering, Norway
University of Oslo, Norway

Michael Riegler
Simula Metropolitan Center for
Digital Engineering, Norway
University of Oslo, Norway

ABSTRACT

In the future, medical doctors will to an increasing degree be assisted by deep learning neural networks for disease detection during examinations of patients. In order to make qualified decisions, the black box of deep learning must be opened to increase the understanding of the reasoning behind the decision of the machine learning system. Furthermore, preparing reports after the examinations is a significant part of a doctors work-day, but if we already have a system dissecting the neural network for understanding, the same tool can be used for automatic report generation. In this demo, we describe a system that analyses medical videos from the gastrointestinal tract. Our system dissects the Tensorflow-based neural network to provide insights into the analysis and uses the resulting classification and rationale behind the classification to automatically generate an examination report for the patient's medical journal.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning*; *Video summarization*; • **Applied computing** → **Health informatics**;

KEYWORDS

Deep learning, medical documentation, interpretable neural networks, automatic disease detection

ACM Reference format:
Steven Alexander Hicks, Konstantin Pogorelov, Thomas de Lange, Mathias Lux, Mattis Jeppsson, Kristin Ranheim Randel, Sigrun Eskeland, Pål Halvorsen, and Michael Riegler. 2018. Comprehensible Reasoning and Automated Reporting of Medical Examinations Based on Deep Learning Analysis. In *Proceedings of 9th ACM Multimedia Systems Conference, Amsterdam, Netherlands, June 12–15, 2018 (MMSys'18),* 4 pages.
https://doi.org/10.1145/3204949.3208113

Contact author's address: Michael Riegler, Simula Research Laboratory, Oslo, Norway, email: michael@simula.no .

MMSys'18, June 12–15, 2018, Amsterdam, Netherlands

ACM ISBN 978-1-4503-5192-8/18/06…$15.00
https://doi.org/10.1145/3204949.3208113

1 INTRODUCTION

Machine learning has shown much potential in becoming an important asset to medical doctors performing disease detection during patient examinations. As a result of this, we may see a decrease in diagnostic errors (in the form of missed disease), increase in number of patients, and further improve the quality of medical care. Additionally, a significant part of a medical professional's time is spent preparing reports after the performed procedures. Multimedia research can significantly support this phase by collecting patient and examination data and providing automatically generated summaries conveying key information of the performed procedures, e.g., video frames with detected objects. An automatically generated report is also useful for training medical experts: through multimedia enriched reports, medical doctors in training can learn based on real data according to case-based teaching and problem-based learning strategies. Thus, multimedia summarization for automated report generation is a much needed feature [20], but it is still in its infancy. One major obstacle is that it is not always comprehensible or reproducible why an automatic detection system marks a finding, i.e., the machine learning system is a black box. In the field of medicine, among others, this is not acceptable as medical doctors often need the underlying rationale behind a decision besides the decision from the system itself. To the best of our knowledge, this is yet an unexplored area of research.

To both improve the "black box"-understanding and assist the examination reporting, we research automated multimedia summarization methods with a semantic nature exploiting domain ontologies. Based on the detection system, the video backend may be used for easy retrieval and reuse of data for automatic report generation, comparisons, teaching and research. As a case study, we use live colonoscopy. This is the gold standard examination of the large bowel, commonly performed for clinical and screening purposes. It allows inspection of the bowel mucosa, essential for the diagnosis of abnormalities such as inflammation, colorectal

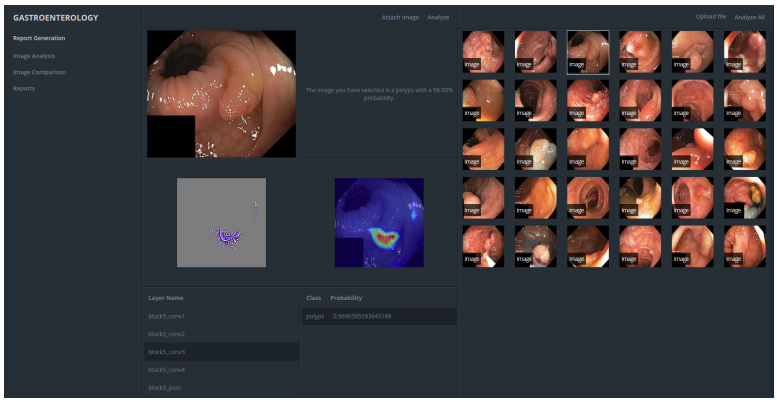


Figure 1: Report and feedback interface, where you may browse through the different neural network layers.

cancer and its precursors (polyps). We have previously developed a live detection system [21, 22]. Under a colonoscopy, the system analyses the captured video frames and gives visual feedback to the doctor if something abnormal is detected [23]. In this paper, we demonstrate how this system can be extended to colonoscopy documentation. After the colonoscopy, an overview (Figure 1) is given where the doctors can make changes or corrections, and add additional information. This can then be stored for later purposes or used in a written endoscopy report. Further, it can be practical to store high quality images of the most important parts [5], i.e., our reporting system also recommends images (frames) to be included in the report and dissects the neural network to give a reasoning why the image is selected.

2 MEDICAL AUTOMATIC REPORTING

Deep learning has greatly improved automatic methods for speech to text conversion, object recognition and image captioning [13]. Structured reporting for colonoscopy procedures, however, is beyond transforming explicit information from one media to another. It also involves finding relevant pieces from multiple modalities and putting them together into a readable report, that supports and argues the diagnosis of a medical expert. In the field of medicine, written reporting of medical procedures is an essential, but cumbersome, part of the physicians' daily work, and the quality and completeness of the reports are critical to the patients care and well being. A more automated reporting system based on automatic video analysis would be extremely helpful for medical experts and contribute to a standardization of the medical report and the implementation of the *Minimal Standard Terminology* (MST) recommended by the World Endoscopy Organization (WEO). Also, the European Society of Gastrointestinal Endoscopy lists the standardization of medical reporting in endoscopic procedures as a requirement [4].

In mission-critical domains, such as the medicine, data driven methods can be questionable if the results are not reproducible or comprehensible by experts within their field. With deep learning in particular, the results of automatic recognition are extremely helpful, but we are still not able to fully understand the rationale of every decision of a network. Theoretical approaches to explain the decisions of a deep neural network have been discussed [14, 29], but it is important to address the problem of understanding and trust among non-technical users, i.e., medical experts and doctors. More visual approaches that present layers using tools such as heat maps or visual representations (texture, heat maps, etc.) [24, 31] come closer to what users can grasp without detailed technical knowledge. All in all, the goal is a tool that generates a structured and readable report composed of text and images from a medical procedure. Moreover, the tool has to this understandable and reproducible for non-technical users to ensure the trust of doctors and patients involved. A recent approach [11] investigates the possibility of creating reports from x-ray images employing neural image captioning methods [30]. A network is trained from a dataset of images along with the reports. Closest to our approach is the work described in [32], where microscope images are fed through a neural network to generate reports and retrieve relevant images of symptoms in addition to an attention map to support the rationale of the networks decision. Both approaches focus on images already classified as relevant by being part of a diagnostic process, whereas the second paper adds the dimension of the rationale of the generated report.

Medical doctors indicated that generating automatic text is not the most important feature for them. More importantly, they need to understand the decisions of the algorithms in an easy and intuitive way, and at the same time, receive support for generating high quality, structured reports [20].

3 ARCHITECTURE AND IMPLEMENTATION

The objectives of our system is to increases classification understanding and reduce the time spent on administrative tasks related to a colonoscopy, e.g., documentation by written reports. The system reports abnormalities commonly found in the gastrointestinal (GI) tract, such as polyps and esophagitis, based on analysis of frame data taken from a video stream. The frames with detected abnormalities are presented to the user for further analysis, with the most prominent images (highest probability of abnormality detected) suggested as attachments to be included in the written report. It is important that the process of disease detection is transparent, i.e., by being able to comprehend why the system concluded as it did and on what basis the diagnosis was set. This is a key component in building trust among the medical professionals who rely on the system to make qualified medical decisions. In addition to building truest among our expected users, it allows us to detect weaknesses in the tool itself and the dataset used to train it. Insight

into the analysis process is done using various visualization techniques to generate intermediate representations of an image as it moves through a neural network, specifically as it moves through the convolutional layers of a convolutional neural network (CNN). This gives us a peek into the decision process of the neural network, showing what regions in the image correspond to a given prediction. This process will be discussed further in section 3.1.

The system is accessed through a React [6] based web application, backed up by a RESTful server API written in Python (using the micro-framework Flask [8]). Image analysis is done using a CNN, specifically a standard VGG-19 model [26] trained on the Kvasir v2 dataset [18] and is implemented using the Keras deep learning framework [12] with a Tensorflow backend [2]. Figure 2 shows the typical case in how we imagine this tool being used for visualizing the analysis process of passing images/frames through the CNN and report generation based on the results from the analysis.

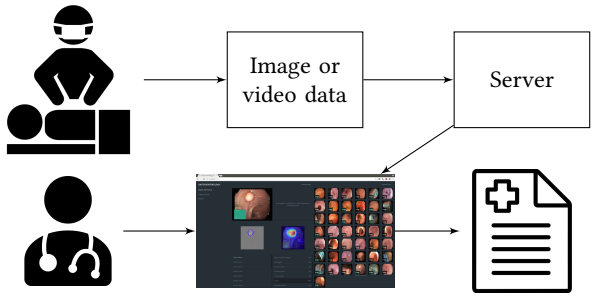


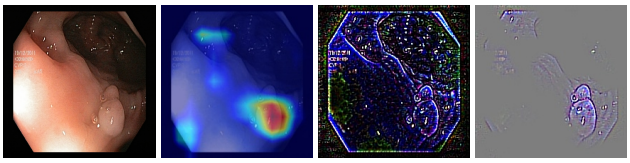
Figure 2: The expected work flow of the reporting system. Images and videos are collected and analyzed during the examination. After the examination, a modifiable report draft is presented to the medical expert in order to produce a final report including text, representative images and video clips.

3.1 Image/Frame Visualization

The visualizations process works on images and videos, with videos being split into frames and processed individually in the same way as a single image. The neural network image representations are generated using a guided grad-cam approach [25], which combines the pixel-level detail of guided back-propagation saliency maps [33] with the class discriminative properties of class activation maps (CAMs) [27]. The result is a high quality image with class discrimination on a pixel level. Each image representation is done with respect to a target class and layer, making it possible to look back through the network and see what less abstract features were picked up by the network.

Figure 3 shows the original image (Figure 3a) and three additional presentations generated by the tool. Figure 3b (grad-CAM) shows the the class-specific regions of the image with respect to a target class at a given layer of the network. Figure 3c (guided back-propagation saliency map) shows a pixel-level representation of what the network sees at a given layer. Figure 3d is a combination of the first two representations, combining the pixel-level detail of the saliency map with the class discriminative features of the grad-CAM.

We start the visualization process by selecting a target layer and class we wish to visualize for a given image. We calculate the gradient of the target layer using the loss for the target class



(a) Original Image **(b)** Grad-CAM **(c)** Guided Backprop **(d)** Guided Grad-CAM

Figure 3: Image representations used to explain decisions.

in regards to the input image. The gradients are globally average pooled and multiplied with the output of the target layer. The result is passed through a relu function before it is re-sized back to the dimensions of the original image. Finally, we squash the values between 0 and 1, and apply a red-blue heatmap filter.

To generate the guided back-propagation saliency map, we start by replacing the activations of our original network with a modified relu activation. During back-propagation, a traditional relu activation would let all gradients whose inputs where larger than 0 pass. We change relu by adding the additional rule of discarding all gradients that are below 0, thereby only back-propagating the positive influence on the activations. With this modified network we calculate the gradients of the target layer with respect to the input image which gives us the saliency map.

Once the grad-CAM and saliency map have been computed, we simply multiply them together to get the guided grad-CAM visualization. This together with the grad-CAM is used in our tool.

3.2 Report Generation

The current state of report generation provides basic functionalities such as changing text and adding additional images. The system presents a preview of the printed report to the users, with direct modification available by clicking and editing the various reports. Images from the analysis can be manually or automatically added or removed. An example report can be found in Figure 4 with text and pre-selected images that the user can change.

4 SETUP AND USAGE

The system is built using the micro-framework Flask [8], which includes a built-in development sever, making it easy to start a local instance of the system. Note that this is not meant to be deployed to a production environment. For a production environment, we can either deploy the application using a popular web server such as Nginx [15] or Apache HTTP Server [3], or by using the pre-built Docker Image available through Docker hub [9]. Setting up the system using the git repository requires several steps of pre-configuration before we can launch the local development server. This includes setting up a Python 3.6 run-time environment and installing the necessary Python dependencies, installing OpenCV [28] with FFmpeg [7] support, configuring Tensorflow and Keras, and optionally (but highly recommended) configuring cuDNN [17] and CUDA [16] for GPU support. With the environment setup, we can launch a local development server by running `app.py` using Python 3. A more detailed setup and configuration guide can be viewed at the application's Github repository [10], but for an easier setup, we recommend using the pre-built Docker image available at Docker hub[9]. The image includes a pre-configured Python environment with the necessary dependencies, CUDA 8 and cuDNN

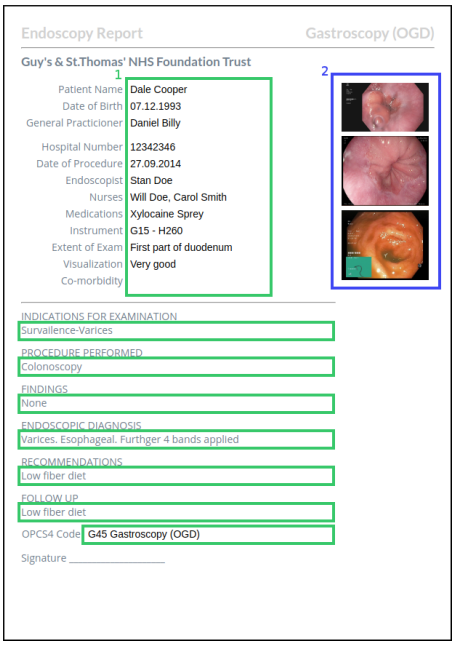


Figure 4: An example of an automatic generated report. The green area marked (1) shows the editable text fields. The blue area (2) shows the images chosen for the report. Report based on sample taken from Wrestling the Octopus [1].

6 for Nvidia GPU support, and hosted using Nginx. Once the tool is up and running, it can be accessed through a web browser using the configured host IP and port.

5 DEMO

In the proposed demo, the participants will be able to see how the system works in real time. In particular, video(s) with disease will be available, and the participants may run it through the system. After the deep learning neural network has analysed the video frames, a screen as shown in Figure 1 will be displayed showing the results of the analysis for each layer. From the list of images selected by the system, the user can select one and see how it has been processed through the network by showing images of the intermediate representations and saliency maps (or heatmaps). Finally, a report can be automatically generated from this interface including both text and images (frames). This report can also be modified after it is created.

REFERENCES

[1] 2015. The Crohnoid Blog. (2015). http://www.wrestlingtheoctopus.com/the-a-to-z-of-my-crohns/

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and others. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[3] Apache. 2018. Apache HTTP Server Project. (2018). https://httpd.apache.org/

[4] Michael Bretthauer, Lars Aabakken, Evelien Dekker, Michal F Kaminski, Thomas Rösch, Rolf Hultcrantz, Stepan Suchanek, Rodrigo Jover, Ernst J Kuipers, Raf Bisschops, and others. 2016. Reporting systems in gastrointestinal endoscopy: Requirements and standards facilitating quality improvement: European Society of Gastrointestinal Endoscopy position statement. *United European gastroenterology journal* 4, 2 (2016), 172–176.

[5] Thomas de Lange, Stig Larsen, and Lars Aabakken. 2005. Image documentation of endoscopic findings in ulcerative colitis: photographs or video clips? *Gastrointestinal Endoscopy* 61, 6 (2005), 715–720.

[6] Facebook. 2018. React. (2018). https://reactjs.org/

[7] FFmpeg. 2018. FFmpeg. (2018). https://www.ffmpeg.org/

[8] Flask. 2018. Flask. (2018). http://flask.pocoo.org/

[9] Steven Hicks. 2018. Demo Docker Repositoryma. (2018). https://hub.docker.com/r/stevenah/mmsys-demo/

[10] Steven Hicks. 2018. Demo Github Repository. (2018). https://github.com/Stevenah/mmsys-demo

[11] Baoyu Jing, Pengtao Xie, and Eric Xing. 2017. On the Automatic Generation of Medical Imaging Reports. *arXiv preprint arXiv:1711.08195* (2017).

[12] Keras. 2018. Keras: The Python Deep Learning library. (2018). https://keras.io/

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[14] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2017. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing* (2017).

[15] NGINX. 2018. NGINX. (2018). https://nginx.org/en/

[16] Nvidia. 2018. Nvidia CUDA Toolkit. (2018). https://developer.nvidia.com/cuda-toolkit

[17] Nvidia. 2018. Nvidia CuDNN. (2018). https://developer.nvidia.com/cudnn

[18] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. 2017. KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection. In *Proc. of MMSYS*. 164–169. https://doi.org/10.1145/3083187.3083212

[19] Konstantin Pogorelov, Michael Riegler, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Carsten Griwodz, Peter Thelin Schmidt, and Pål Halvorsen. 2017. Efficient disease detection in gastrointestinal videos – global features versus neural networks. *Multimedia Tools and Applications* 76, 21 (01 Nov 2017), 22493–22525. https://doi.org/10.1007/s11042-017-4989-y

[20] Michael Riegler, Mathias Lux, Carsten Griwodz, Concetto Spampinato, Thomas de Lange, Sigrun L Eskeland, Konstantin Pogorelov, Wallapak Tavanapong, Peter T Schmidt, Cathal Gurrin, and others. 2016. Multimedia and Medicine: Teammates for Better Disease Detection and Survival. In *Proc. of ACM MM*.

[21] Michael Riegler, Konstantin Pogorelov, Pål Halvorsen, Thomas de Lange, Carsten Griwodz, Peter Thelin Schmidt, Sigrun L. Eskeland, and Dag Johansen. 2016. EIR - Efficient Computer Aided Diagnosis Framework for Gastrointestinal Endoscopies. In *Proc. of CBMI*.

[22] Michael Riegler, Konstantin Pogorelov, Jonas Markussen, Mathias Lux, Håkon Kvale Stensland, Thomas de Lange, Carsten Griwodz, Pål Halvorsen, Dag Johansen, Peter T Schmidt, and Sigrun L. Eskeland. 2016. Computer Aided Disease Detection System for Gastrointestinal Examinations. In *Proc. of MMSys*.

[23] Michael Riegler, Konstantin Pogorelov, Jonas Markussen, Mathias Lux, Håkon Kvale Stensland, Thomas de Lange, Carsten Griwodz, Pål Halvorsen, Dag Johansen, Peter T. Schmidt, and Sigrun L. Eskeland. 2016. Computer Aided Disease Detection System for Gastrointestinal Examinations. In *Proc. of MMSYS*. 29:1–29:4. https://doi.org/10.1145/2910017.2910629

[24] Christin Seifert, Aisha Aamir, Aparna Balagopalan, Dhruv Jain, Abhinav Sharma, Sebastian Grottel, and Stefan Gumhold. 2017. Visualizations of Deep Neural Networks in Computer Vision: A Survey. In *Transparent Data Mining for Big and Small Data*. Springer, 123–144.

[25] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2016. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. *CoRR* abs/1610.02391 (2016). arXiv:1610.02391 http://arxiv.org/abs/1610.02391

[26] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014). arXiv:1409.1556 http://arxiv.org/abs/1409.1556

[27] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. 2014. Striving for Simplicity: The All Convolutional Net. *CoRR* abs/1412.6806 (2014). arXiv:1412.6806 http://arxiv.org/abs/1412.6806

[28] OpenCV team. 2018. Open Source Computer Vision Library (OpenCV). (2018). https://opencv.org/

[29] Rene Vidal, Joan Bruna, Raja Giryes, and Stefano Soatto. 2017. Mathematics of Deep Learning. *arXiv preprint arXiv:1712.04741* (2017).

[30] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. of ML*. 2048–2057.

[31] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* (2015).

[32] Zizhao Zhang, Yuanpu Xie, Fuyong Xing, Mason McGough, and Lin Yang. 2017. Mdnet: A semantically and visually interpretable medical image diagnosis network. In *Proc. of IEEE CVPR*. 6428–6436.

[33] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Learning Deep Features for Discriminative Localization. *CoRR* abs/1512.04150 (2015). arXiv:1512.04150 http://arxiv.org/abs/1512.04150