Real-Time Feedback for Colonoscopy in a Multicenter Clinical Trial

Wallapak Tavanapong
Iowa State University, tavanapo@iastate.edu

JungHwan Oh
University of North Texas

Gavin Kijkul
Iowa State University, gkijkul@iastate.edu

Jacob Pratt
Iowa State University, jrpratt@iastate.edu

Johnny Wong
Iowa State University

See next page for additional authors

Follow this and additional works at: https://lib.dr.iastate.edu/cs_conf

Part of the Clinical Trials Commons, Computer Sciences Commons, Data Science Commons, and the Medicine and Health Sciences Commons

Recommended Citation
Tavanapong, Wallapak; Oh, JungHwan; Kijkul, Gavin; Pratt, Jacob; Wong, Johnny; and deGroen, Piet C., "Real-Time Feedback for Colonoscopy in a Multicenter Clinical Trial" (2020). Computer Science Conference Presentations, Posters and Proceedings. 50.
https://lib.dr.iastate.edu/cs_conf/50

This Conference Proceeding is brought to you for free and open access by the Computer Science at Iowa State University Digital Repository. It has been accepted for inclusion in Computer Science Conference Presentations, Posters and Proceedings by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Real-Time Feedback for Colonoscopy in a Multicenter Clinical Trial

Abstract
We report the technical challenges, solutions, and lessons learned from deploying real-time feedback systems in three hospitals as part of a multi-center controlled clinical trial to improve quality of colonoscopy. Previous clinical trials were conducted in one center. The technical challenges for our multicenter clinical trial include 1) reducing additional work by the endoscopists to utilize real-time feedback, 2) handling different colonoscopy practices at different hospitals, and 3) training an effective CNN-based classification model with a large variety of patterns of data in day-to-day colonoscopy practice. We report performance of our real-time systems over a period of 20 weeks at each hospital. We conclude that CNN-based classification can achieve very good performance in real-world deployment when trained with high quality data.

Keywords
Multi-center clinical trial, Real-time feedback of colonoscopy quality, Convolution Neural Network (CNN)

Disciplines
Clinical Trials | Computer Sciences | Data Science | Medicine and Health Sciences

Comments

Authors
Wallapak Tavanapong, JungHwan Oh, Gavin Kijkul, Jacob Pratt, Johnny Wong, and Piet C. deGroen
Real-time Feedback for Colonoscopy in a Multi-center Clinical Trial

Wallapak Tavanapong  
Department of Computer Science  
Iowa State University  
Ames, IA 50011, USA  
tavanap@iastate.edu

JungHwan Oh  
Department of Computer Science and Engineering  
University of North Texas  
Denton, TX 76203  
junghwan.oh@unt.edu

Gavin Kijkul  
Department of Computer Science  
Iowa State University  
Ames, IA 50011, USA  
gkijkul@iastate.edu

Johnny Wong  
Department of Computer Science  
Iowa State University  
Ames, IA 50011, USA  
wong@iastate.edu

JungHwan Oh  
Department of Computer Science and Engineering  
University of North Texas  
Denton, TX 76203  
junghwan.oh@unt.edu

Piet C. deGroen  
Division of Gastroenterology  
Hepatology and Nutrition  
University of Minnesota  
Minnesota, MN 55455, USA  
degroen@umn.edu

Jacob Pratt  
Department of Computer Science  
Iowa State University  
Ames, IA 50011, USA  
jpratt@iastate.edu

Abstract — We report the technical challenges, solutions, and lessons learned from deploying real-time feedback systems in three hospitals as part of a multi-center controlled clinical trial to improve quality of colonoscopy. Previous clinical trials were conducted in one center. The technical challenges for our multi-center clinical trial include 1) reducing additional work by the endoscopists to utilize real-time feedback, 2) handling different colonoscopy practices at different hospitals, and 3) training an effective CNN-based classification model with a large variety of patterns of data in day-to-day colonoscopy practice. We report performance of our real-time systems over a period of 20 weeks at each hospital. We conclude that CNN-based classification can achieve very good performance in real-world deployment when trained with high quality data.

Keywords— Multi-center clinical trial, Real-time feedback of colonoscopy quality, Convolution Neural Network (CNN)

I. INTRODUCTION

Colorectal cancer (CRC), despite being a preventable cancer, is still expected to cause about 53,200 deaths in the U.S. in 2020 [1]. Colonoscopy allows for detailed examination of the entire colon and removal of all premalignant lesions during the procedure, the latter typically done during withdrawal of the endoscope after reaching maximum intubation. Colonoscopy is readily available in the U.S. and covered by insurance. Given these facts, why is CRC prevention by colonoscopy lower than expected? Currently, the dominant explanations for the continued incidence of CRC are endoscopist-related factors, such as choice of suboptimal equipment, not removing remaining debris, not reaching the cecum, fast withdrawal, no effort at inspection of areas behind folds and angulations, and an inadequate polyp removal technique. Recent research examining 15 years of data at a private ambulatory surgery center with 80 endoscopists practicing high quality examination of the colon shows a prevention of 67% of CRC compared to

the SEER-18 population [2]. The high-quality examination protocol with an acronym “CLEAR” was used to have the endoscopists (1) Clean the colon, (2) Look Everywhere, and (3) perform complete Abnormality Removal. Since 2003, we have been developing software (Endoscopic Multimedia Information System or EMIS) for automated measurements of quality and feedback during colonoscopy. In a single-center controlled clinical trial in 2012, the quality of colonoscopy performed by ten third-year GI trainees with feedback from EMIS improved significantly [3]. EMIS measured multiple intra-procedure quality metrics (e.g., clear withdrawal time without blurry frames, amount of stool during insertion and withdrawal, “spiral score”, and Boston Bowel Preparation Scale scores), but the provided feedback consisted only of the “spiral score”, or in simple terms how well the endoscopist tried to look everywhere. The objective measurements were then combined into a single automated quality score which was shown by domain experts to correlate with quality of colonoscopy [4]. Since then, there were two reports of real-time feedback systems for colonoscopy in single-center clinical trials [5], [6]. To the best of our knowledge, none have been reported for a multi-center clinical trial. We are nearing the end of the first phase of the deployment of EMIS (version 6) and EMIS-Deep (EMIS extension utilizing CNN) in a multi-center clinical trial at University of Washington Medical Center, University of Minnesota, and Johns Hopkins University. The deployment presented several technical challenges that we had to overcome. Our contributions include 1) presenting the technical challenges encountered and their corresponding solutions in the deployment of real-time measurements and feedback for endoscopy in a multi-center clinical trial, 2) reporting the effectiveness of our real-time systems, EMIS (version 6) and EMIS-Deep, over a period of 20 weeks, and 3) showing that
TABLE I: Existing real-time detection and feedback for colonoscopy

<table>
<thead>
<tr>
<th>Publication</th>
<th>Sub-system</th>
<th>Task</th>
<th>Feature</th>
<th>Classifier</th>
<th>Deep Learning Architectures</th>
<th>Training Data</th>
<th>Validation Data</th>
<th>Test Data</th>
<th>Image Resolution</th>
<th># doctors in a clinical trial</th>
<th># patients in a clinical trial</th>
<th>Average processing frame rate (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wang et al. 2019 [5]</td>
<td>Polyp Detection</td>
<td>Extracted from CNN</td>
<td>ImageNet</td>
<td>SegNet</td>
<td>1,545 images</td>
<td>17,725 images (total)</td>
<td>N/A</td>
<td>600x360</td>
<td>9 endoscopists (2, more than 20,000 procedures)</td>
<td>536 patients (controls)</td>
<td>522 patients (feedback)</td>
<td>25</td>
</tr>
<tr>
<td>He et al. 2019 [6]</td>
<td>Model B</td>
<td>Volumetric Classification</td>
<td>AlexNet features</td>
<td>FC layer</td>
<td>FC layer</td>
<td>NN</td>
<td>3,235 images</td>
<td>912 images</td>
<td>3,409 images</td>
<td>4 endoscopists (5,000-8,000 procedures)</td>
<td>135 patients (controls)</td>
<td>308 patients (feedback)</td>
</tr>
<tr>
<td>He et al. 2019 [6]</td>
<td>Model E</td>
<td>Volumetric Classification</td>
<td>Pre-trained features</td>
<td>FC layer</td>
<td>FC layer</td>
<td>NN</td>
<td>4,006 images</td>
<td>1,118 images</td>
<td>1,280 images</td>
<td>2 FC layer</td>
<td>32x32</td>
<td></td>
</tr>
<tr>
<td>He et al. 2019 [6]</td>
<td>Model BP</td>
<td>BBPS Score Classification</td>
<td>ZFNet features</td>
<td>FC layer</td>
<td>FC layer</td>
<td>NN</td>
<td>6,402 images</td>
<td>916 images</td>
<td>1,000 images</td>
<td>2 FC layer</td>
<td>6x6x6</td>
<td></td>
</tr>
<tr>
<td>He et al. 2019 [6]</td>
<td>Model S</td>
<td>Blurry/Clear Classification</td>
<td>MobileNet features</td>
<td>FC layer</td>
<td>FC layer</td>
<td>NN</td>
<td>1,654 images</td>
<td>92 images</td>
<td>N/A</td>
<td>25x25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Byrne et al. 2018 [11]</td>
<td>Classification of polyp types: Adenoma, Hyperplasic, No polyp</td>
<td>Extracted from CNN</td>
<td>ImageNet</td>
<td>CNN</td>
<td>Segmentation</td>
<td>40,089 images</td>
<td>40 videos total (length and FR N/A)</td>
<td>223 videos total (length and FR N/A)</td>
<td>N/A</td>
<td>No clinical trial</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Elfrink et al. 2018 [10]</td>
<td>Localization of polyps</td>
<td>Fine-tuned features (pre-trained on ImageNet)</td>
<td>CNN</td>
<td>ResNet50 &amp; YOLO</td>
<td>FC layer</td>
<td>6,641 images total (7-field cross validation)</td>
<td>46,277 images</td>
<td>~455,000 images from 11 videos</td>
<td>25x224, 40x400</td>
<td>No clinical trial</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Reiß et al. 2017 [9]</td>
<td>Detection sub-system</td>
<td>Abnormality Detection</td>
<td>Hand-crafted features</td>
<td>YOLO</td>
<td>4,088 PF</td>
<td>4,533 NPF</td>
<td>53,964 PP and 32,313 NPF</td>
<td>25x224, 40x400</td>
<td>No clinical trial</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reiß et al. 2017 [9]</td>
<td>Abnormality localization</td>
<td>Abnormality localization</td>
<td>Hand-crafted features</td>
<td>YOLO</td>
<td>FP detection and polyp localization</td>
<td>None</td>
<td>18,781 images (leave-one-out cross-validation)</td>
<td>131,863 images</td>
<td>920x1080, 656x480, 712x480</td>
<td>No clinical trial</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Wang et al. 2015 [8]</td>
<td>Classification of polyp frames &amp; localization of polyps</td>
<td>Hand-crafted edge localization</td>
<td>Rule-based generalization from SVM classifier</td>
<td>FC layer</td>
<td>Rule-based</td>
<td>None</td>
<td>287,712 images (leave-one-out cross-validation)</td>
<td>1,906,092 images</td>
<td>320x480</td>
<td>No clinical trial</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Basenk et al. 2012 [7]</td>
<td>Detection of the start and end frames of an endoscopic procedure</td>
<td>Hand-crafted features</td>
<td>Rule-based</td>
<td>FC layer</td>
<td>Rule-based</td>
<td>N/A</td>
<td>N/A</td>
<td>~265 million images (564 h of video)</td>
<td>720x480</td>
<td>No clinical trial</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

NN: Neural Network; PF: Polyp Frame; NPF: Non-Polyp Frame; FC layer: Fully connected layer; FR: Frame rate; KNN: K-Nearest Neighbor classification

given representative training data, a CNN-based system can be very effective in real-world deployment.

The rest of this paper is as follows. Section II presents related work on real-time measurement and feedback for colonoscopy. Section III describes the challenges and our solutions. Section IV presents the performance evaluation of EMIS (version 6) and EMIS-Deep over a 20 week period, covering over 2,000 procedures. We provide the conclusion and discussion of the future work in Section V.

II. RELATED WORK

Table I shows a summary of existing works on real-time systems for colonoscopy. Given that several methods are already capable of running in real-time, we omit other methods that were not reported to be capable of running in (near) real-time.

A. Real-time Feedback for Colonoscopy in Single-Center Controlled Clinical Trials

In the aforementioned clinical trials, audio and an extra monitor were used to provide feedback. The results of both trials show a higher adenoma detection rate on procedures with feedback. The system by Wang et al. provided a sound prompt when a polyp appearance was detected [5]. The endoscopist was asked to look at the main monitor until hearing the sound prompt, at which point the endoscopist would look at the second monitor for the detected polyp location displayed in a hollow blue tracing box. The feedback in the trial conducted by Su et al. [6] included audio prompts when continuous blurry or unstable frames were detected. The detected polyp location was displayed on a second monitor. Both systems used deep-learning models. Table I shows the technical differences between the two systems as well as other real-time capable systems for colonoscopy not in a clinical trial. Su et al. utilized five neural-network models, four of which used features extracted from existing pre-trained models as input to shallow fully-connected neural networks [6]. Model B classified cecum and non-cecum images to identify the beginning of the withdrawal phase of a colonoscopy. Model E classified in vitro and vivo frames to identify the end of the withdrawal phase (end of the procedure). Model BP output four probabilities for an image belonging to each Boston Bowel Preparation Scale score (BBPS). Model PD classified whether an image shows a polyp appearance. Finally, model S determined withdrawal stability by classifying blurry and non-blurry frames, as well as computing the similarity between two subsequent frames. The models by Su et al. were trained, validated, and tested on small datasets. Aside from model PD, none of the models’ clinical trial performance was reported.
B. Real-time Capable Polyph Detection Methods for Colonoscopy

This category includes methods reported to being capable of running in (near) real-time but were not mentioned as part of any clinical trial. Earlier works in this category used hand-crafted features while recent studies used deep-learning based methods. These methods were shown to perform well on limited datasets, but performance on full-length colonoscopy procedures to simulate the application in routine colonoscopy screening were not reported.

Stanek et al. proposed real-time detection of the start and end of an endoscopic procedure using hand-crafted features [7]. Polyph-Alert by Wang et al. [8] used edge-cross-section features and a rule-based classifier to detect edges that make the contour of a polyp. It tracked the same polyp appearing in nearby frames and provided a visual marker on the detected polyp edge up to ten times a second. Tests on approximately 18 hours of videos showed that Polyph-Alert returned feedback with a recall of unique polyps of 97.7% and on average about 0.86 min. of false alert per 20 min. procedure video. There are no other publications which report results of test performances on millions of frames as in this work. Riegler et al. proposed an end-to-end multimedia system for automatic disease detection, and visualization in GI tract procedures [9]. Urban et al. trained different CNNs on two different datasets for polyph detection and localization [10]. Polyph localization used a variation of the “You Only Look Once” (YOLO) method, which predicts bounding boxes for multiple polyps within an image. The best performing model for both tasks was pre-trained using ResNet50 that achieved 96.1% accuracy. Byrne et al. proposed a CNN based on the inception network architecture to detect different types of polyps under Narrow Band Imaging [11]. They achieved 94% per frame accuracy in classification of adenoma and hyperplastic polyps. These techniques were tested on different datasets. Hence, we cannot compare their effectiveness directly.

III. EMIS: Real-time feedback systems in a multi-center clinical trial

A. Design goals for real-time feedback in routine practice

Our goals are different from real-time feedback systems used in previously reported single-center clinical trials [5], [6]. Our goals are as follows. 1) Minimum change of the current endoscopy practice to utilize real-time feedback. We do not introduce an extra monitor or additional sound devices for feedback. All feedback is visual on the same monitor without overlapping with the colon mucosa area on the monitor. 2) Ability to handle differences in colonoscopy practice at different hospitals. 3) System switched on and ready at least 12 hours/day during outpatient practice hours and at least 5 days/week in all the sites and recognition of full-length endoscopy procedures performed by any number of participating endoscopists instead of by a fixed set of endoscopists as in existing trials. 4) No need for personnel to manually indicate the start or the end of a procedure.

B. Extension of EMIS to EMIS-Deep

EMIS runs on an off-the-shelf PC equipped with a video capturing card that takes as input a composite signal from an endoscope video processor, configured to always disable any patient- or physician-related information, i.e., endoscope text features are permanently disabled. We achieved the first design goal by using a hardware overlay that overlays the real-time feedback from EMIS—non-black pixels only—with the original signal from the endoscope processor, as developed in the previous trial [3]. Feedback consists of text in the lower right corner of the screen displaying EMIS INSIDE when a procedure is detected, meaning the endoscope is inside the patient, and EMIS ON when the endoscope is outside the patient. In our next phase of deployment, we will also display feedback for spiral score, the number of detected retroflexion frames, colon preparation, and the blurriness of images. Some feedback will only be displayed when the measured quality is below a preset threshold value.

EMIS is built on top of our middleware SAPPHIRE designed for modularity, scalability, configurability, collaboration among multiple developers, and ease of extension [12]. SAPPHIRE comes with several modules for basic tasks, for instance, video capture, reading and writing video files, and displaying frames overlaid with analysis results from other modules on screen. EMIS version 4 has been published in previous work and used in a previous clinical trial. It has the following capabilities: 1) Detection of the begin and end frames of an endoscopy; 2) Classification of informative frames and non-informative frames; 3) Measuring the amount of stool-colored pixels for each frame; 4) Measuring the spiral score; and 5) Calculating various objective measurements of colonoscopy quality.

The last two design goals proved difficult to achieve with EMIS version 4 due to the variety of endoscope use patterns among different hospitals. EMIS version 6 includes a revised automated cropping module for Olympus scopes. After fine-tuning threshold values for detection of an endoscopic procedure, EMIS version 6 has been running well in two hospitals. However, we did not find suitable threshold values for the remaining hospital since there are many more patterns of outside-the-patient images beyond what our hand-crafted features were designed for. To address this issue, we made a decision between 1) developing additional hand-crafted features to upgrade the classifier; or 2) using deep-learning for feature representation and classification. We chose the latter approach given fast development time and ease of adaptability. We had some reservations related to this choice because existing CNN-based medical analyses were tested only on limited datasets (i.e., fewer than 23K images as shown in Table I). In contrast, this clinical trial requires the processing of millions of images.

To address this challenge, we developed EMIS-Deep, an integration framework that enables shared memory communication between EMIS and deep-learning modules. Fig. 1 shows the overview of EMIS-Deep. The framework also provides the flexibility to switch to different deep-learning toolkits without long development time. The disadvantage is the cost of copying data into and from the shared memory. We minimize this cost as much as possible. EMIS-Deep is currently used for detecting the frame numbers of the start and the end of an endoscopic procedure. We can add other types of image
classification as needed. We implemented two new modules in C/C++ as dynamic linked libraries (DLL). Each module runs in its own thread. One module runs inside the EMIS process and the other module runs inside a CNN process for CNN-based classification (Fig. 1). In our current implementation, the CNN process runs an executable file generated from PyInstaller that wraps our Python program calling Google TensorFlow for image classification using CNN.

The new module in the EMIS process initializes the shared memory fields for communication with the CNN process. For every frame routed through the middleware from the video capture module, the new module checks the shared memory field whether the CNN process requests any frame. If so, the module crops and resizes the current frame and copies the resized frame data to the shared memory data field. It also updates the shared memory fields for availability of the frame data as well as the frame ID. The frame ID is unique for each frame and it is assigned by the video capture module which increments the frame ID by one for each frame it sends out through the middleware. The thread monitors the shared memory fields for the start frame ID (SID), the first inside-the-patient frame of a procedure, and the end frame ID (EID), the first outside-the-patient frame, respectively. Finally, the thread also monitors whether the CNN process is still active and forks the new CNN process if it becomes inactive to ensure the robustness of the program.

The CNN process initializes the CNN architecture, modeled after VGG16 [13], and loads the pre-trained model that classifies whether a frame is “inside-the-patient” (i.e., frame shows intestinal mucosa) or “outside-the-patient” (i.e., frame shows non-intestinal mucosa such as endoscopy room). It then loads the interface module implemented as a C DLL with functions that are callable from Python, and begins a control loop. The control loop runs for the duration of the process. In the control loop, for every frame, it calls the Python-C interface through the interface module to update the shared memory field to request a frame from EMIS and waits with time-out for the frame data, which is then input to the trained CNN model. The control loop also periodically saves values of important state variables to the shared memory so that it can resume from those states when restarted by the EMIS process.

Ideally, if the CNN always predicts the label of each frame correctly, we can simply rely on the CNN predicted results to find the values of SID and EID as outlined in Fig. 2(a). That is, if the previously and the newly predicted labels are the same, no further computation is needed (Line 1). However, if they are different, we potentially find the boundary frame separating inside- and outside-the-patient frames. We save the boundary frame ID in the bndFrameID variable and call the detectBoundary function to set the value of the global variable videoOutsidePatient. This global variable is initialized to true since the camera is initially outside the patient. The variable is set to false in Line 8 if its current value is true and a sufficient number of inside-the-patient frames (i.e., count >= enterThres) are seen. The value of videoOutsidePatient is set to true in Line 15 if its current value is false (i.e., the camera was inside the patient) and a sufficient number of outside-the-patient frames are seen (i.e., count <= exitThres value). For a perfect CNN, the values of exitThres and enterThres can simply be one; however, as classification is often not perfect, we use higher threshold values.

C. Dealing with Incorrect CNN Prediction Results

The correctness of CNN classification depends on whether the data used for training are representative of the data seen in deployment. If this is not the case, many inside-the-patient frames may be missing, a single procedure may be cut into multiple video files, or many outside-the-patient frames may be considered as part of the procedure. We optimize CNN classification by 1) using our error correction function outlined in Fig. 2(b) before using the corrected value as input to Fig. 2(a).
and 2) development of a high quality, representative training dataset.

C.1 CNN Error Correction: Fig. 2(b) shows the error correction function based on an observation that inside-the-patient frames have a significant amount of red color compared to blue color which outside frames do not have. The smooth function in Fig. 2(b) implements a linear smoother to output an average of 15 past values including the current value. We apply the function to the mean normalized red value (meanRed), mean normalized blue value (meanBlue), and CNNPr—the probability of the current frame belonging to the inside-the-patient class determined by CNN. In Line 3, the output of NormContDist, a normal continuous distribution function with the mean μ and the standard deviation std, is used to correct the frame label in Lines 4-9. The label is passed to the sliding filter function that adds the label to the sliding window. Only when the input label agrees with the two previous labels stored in the sliding window is the label output; otherwise, it is suppressed, and the previous output’s label is output.

C.2 Training Datasets: CNN performance depends greatly on the quality of the training dataset. Initially, we thought that implementation of inside- and outside-the-patient frame classification using CNN would be easy, until we encountered large variations in appearance of outside-the-patient frames in routine colonoscopy screening. Gradually adding new misclassified frames encountered in the training data and retraining the model created a significant class imbalance in the training dataset, resulting in poor performance. Synthetic data augmentation that generates minor variations on the dataset would not work well due to major differences in the patterns. To overcome all these challenges, we created a new, balanced training dataset that included various patterns seen in the sample population data for model training and validation. This resulted in the best performing model. Fig. 3 shows the variety of patterns in the training dataset that resulted in the best performance in our studies. In this figure, one image represents a pattern—a group of images in the same cluster in the feature space. There are 20 patterns (red color) for the inside-the-patient class and 28 patterns for the outside-the-patient class although the total number of images in each class is the same. A larger image in Fig. 3 represents a cluster with more images.

The distances among the images reflect how far they are in the summary feature space with two dimensions, which are the first two components from Principal Component Analysis applied to the features extracted from the image using the original pre-trained VGG16 model. Using the model trained on this dataset, the CNN accuracy on the validation dataset of 30,496 images is 98.8%. The validation dataset is class balanced and does not overlap with the training dataset.

Does this training dataset represent the population data? Currently, there are no quantitative measures designed to answer this question. Such a measure would be useful so that a representative training dataset can be found semi-automatically without time-consuming manual effort. Currently, one must examine the classification performance during the deployment and retrain the model until the desired performance in real-world deployment is reached.

TABLE II. CNN architecture configuration; the size-in and size-out are described by rows × cols × #nodes. The kernel configuration for the convolutional layers (conv1-5) is specified as rows × cols × #nodes × #filters, stride. For the max pooling layers (pool1-4), we list rows × cols × #nodes, stride. Relu activation function and Adam optimizer were used.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Size-in</th>
<th>Size-out</th>
<th>Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>64 × 64 × 3</td>
<td>64 × 64 × 16</td>
<td>3 × 3 × 3 × 16,1</td>
</tr>
<tr>
<td>Pool1</td>
<td>64 × 64 × 16</td>
<td>32 × 32 × 16</td>
<td>2 × 2 × 16,2</td>
</tr>
<tr>
<td>Conv2</td>
<td>32 × 32 × 16</td>
<td>32 × 32 × 16</td>
<td>3 × 3 × 16 × 32,1</td>
</tr>
<tr>
<td>Pool2</td>
<td>32 × 32 × 32</td>
<td>16 × 16 × 32</td>
<td>2 × 2 × 32,2</td>
</tr>
<tr>
<td>Conv3</td>
<td>16 × 16 × 32</td>
<td>16 × 16 × 64</td>
<td>3 × 3 × 32 × 64,1</td>
</tr>
<tr>
<td>Pool3</td>
<td>16 × 16 × 64</td>
<td>8 × 8 × 64</td>
<td>2 × 2 × 64,2</td>
</tr>
<tr>
<td>Conv4</td>
<td>8 × 8 × 64</td>
<td>8 × 8 × 128</td>
<td>3 × 3 × 64 × 128,1</td>
</tr>
<tr>
<td>Pool4</td>
<td>8 × 8 × 128</td>
<td>4 × 4 × 128</td>
<td>2 × 2 × 128,2</td>
</tr>
<tr>
<td>Conv5</td>
<td>4 × 4 × 128</td>
<td>1 × 1 × 256</td>
<td>4 × 4 × 128 × 256,1</td>
</tr>
<tr>
<td>FC</td>
<td>256</td>
<td>256</td>
<td></td>
</tr>
</tbody>
</table>

IV. PERFORMANCE EVALUATION

As described in Section III, all the collected videos do not contain any information about patients nor endoscopists. We received the formal approvals for the first phase of the deployment from all institutions to capture de-identifiable procedures, one video file per procedure, for evaluation and calibration of software parameters. Our CNN architecture is a smaller version of VGG16 with the architecture configuration listed in Table II.

Table III shows the performance in five rooms of the three hospitals. The software was setup to run from 6:00am to midnight daily. However, the Olympus endoscope video processor in each room was turned on at different times, typically from 7am-6pm during weekdays and was turned off after the last procedure of the day. The composite video resolution was 720 × 480 and the frame rate was 29.97 fps. In one room, our system analyzed over 11 hours’ worth of video (about 1.18M frames) per day 5 days a week for more than 20 weeks. There may not be procedures every day and the endoscope processor may not be turned on as early as 6am. Nevertheless, to the best of our knowledge, this is the largest reported test of real-time systems in endoscopy and of CNN-based systems during routine colonoscopy. Our goal was to ensure that our software is sufficiently accurate and reliable in day-to-day routine endoscopy practice before we use it for real-time feedback of colonoscopy quality. This is to avoid irrelevant
factors (e.g., significant amount of false predictions) from affecting the measured quality metrics. Rooms W1 and W2 denote two endoscopy rooms from the same hospital; M1 denotes one room in the second hospital. E denotes one room in a third hospital that is the only place using EMIS-Deep. E-2020 denotes that the system used the CNN model trained on the latest, most representative training dataset with 48 patterns discussed in Section III.C.2 and summarized in Fig. 3, the error correction algorithm outlined in Fig. 2(b), and the parameter values indicated in Table IV. In contrast, E-2019 denotes the system used prior CNN models without the error correction algorithm and without the best representative training dataset. The number of problem cases reported in Table III includes cases where procedures were split into multiple video files, video files of cases with more than 1 minute of outside-the-patient frames, cases missing a significant amount of procedure content, and multiple cases being incorrectly combined into one video file. Recall measures the number of correctly detected procedures (non-problem cases) to the number of actual procedures. Table III shows high recall values (0.95-0.98) and very small failed ratios (2-4%) for W1, W2, and E-2020 (EMIS-Deep). The results for E-2019 are not as good, as the CNN of E-2019 was trained on a worse quality training dataset. M1 used EMIS, just as W1 and W2, but had different parameter values. Most failed cases are videos containing non-procedure images.

We will utilize the dependability measures to develop effective semi-automated solutions to collect training datasets to train effective CNN models. The measures should be independent from deep-learning models used and should be comparable among different training datasets. Second, we show that EMIS-Deep, our loosely integrated framework, using shared memory works well for integrating EMIS and CNN code using TensorFlow. EMIS-Deep enables flexible integration of existing code and new code using state-of-the-art deep-learning toolkits. The framework is easily extensible to include deep-learning based detection of other objects of interest.

**CONFLICT OF INTEREST AND ACKNOWLEDGMENTS**

Tavanapong, Wong, and Oh have equity interest and management roles in EndoMetric Corp. Dr. de Groen serves on the Scientific Advisory Board of EndoMetric Corp. We thank Jaber Salem for his help in collection of performance data.

**REFERENCES**


