Automatic Detection of Intestinal Content to Evaluate Visibility in Capsule Endoscopy

1st Reinier Noorda
iTEAM Research Institute
Universitat Politècnica de València
Valencia, Spain
reinoo@upv.es

2nd Andrea Nevárez
Unidad de Endoscopia Digestiva, Servicio de Medicina Digestiva
Hospital Universitari i Politècnic La Fe
Digestive Endoscopy Research Group, IIS La FE
Valencia, Spain
nevarezja78@gmail.com

3rd Adrián Colomer
Instituto de Investigación e Innovación en Bioingeniería, I3B
Universitat Politècnica de València
Valencia, Spain
adcogra@i3b.upv.es

4th Valery Naranjo
Instituto de Investigación e Innovación en Bioingeniería, I3B
Universitat Politècnica de València
Valencia, Spain
vnaranjo@i3b.upv.es

5th Vicente Pons Beltrán
Unidad de Endoscopia Digestiva, Servicio de Medicina Digestiva
Hospital Universitari i Politècnic La Fe
Digestive Endoscopy Research Group, IIS La FE
Valencia, Spain
pons_vicbel@gva.es

***Abstract*—In capsule endoscopy (CE), preparation of the small bowel before the procedure is believed to increase visibility of the mucosa for analysis. However, there is no consensus on the best method of preparation, while comparison is difficult due to the absence of an objective automated evaluation method.**

The method presented here aims to fill this gap by automatically detecting regions in frames of CE videos where the mucosa is covered by bile, bubbles and remainders of food. We implemented two different machine learning techniques for supervised classification of patches: one based on hand-crafted feature extraction and Support Vector Machine classification and the other based on fine-tuning different convolutional neural network (CNN) architectures, concretely VGG-16 and VGG-19.

Using a data set of approximately 40,000 image patches obtained from 35 different patients, our best model achieved an average detection accuracy of 95.15% on our test patches, which is similar to significantly more complex detection methods used for similar purposes. We then estimate the probabilities at a pixel level by interpolating the patch probabilities and extract statistics from these, both on per-frame and per-video basis, intended for comparison of different videos.

***Index Terms*—image processing, machine learning, support vector machines, local binary patterns, capsule endoscopy, small bowel preparation, convolutional neural networks**

I. Introduction

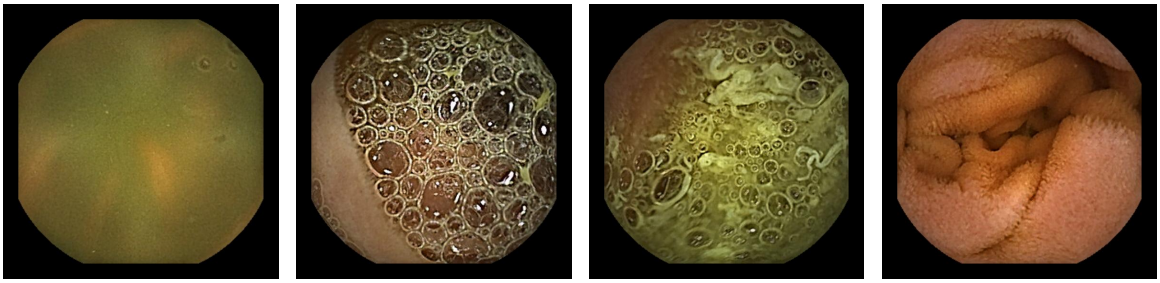
The main advantage of capsule endoscopy (CE) to traditional endoscopy is that it allows visualisation of the middle part of the intestine. However, there is a relatively frequent presence of intestinal content [1], such as bile, bubbles and remainders of food. It is generally believed that adequate preparation of the patient can help to reduce the amount of intestinal content and thus increase visibility. Different preparation methods exist, with different tolerance levels for the patient, but there is no consensus on which is the most adequate due to conflicting results in comparative studies.

The conflicting results are partly due to significant differences between evaluation methods. Some studies have made use of human evaluation methods [1], which clearly have a subjective nature, while other studies have employed different computerised methods [2], which mainly focus on the timeline with the dominant colours per frame displayed in the software that comes with the PillCam® capsule. Therefore, there is a need for an objective automated evaluation method that can be used as a standard among different clinical centres.

In this work, we aim to meet this need through the development of two different methods for automatic detection of bile, bubbles or food debris, and estimation of the degree of mucosa visibility from the detection results. Fig. 1 shows examples of the different cases of interest for our detection method. This paper is structured as follows. First, we discuss relevant work that has already been done in this field in Section II. In Section III we present our method, while the obtained results are presented in Section IV. Finally, we present our conclusions and lines of future work in Section V.

This work was funded by the European Union's H2020: MSCA: ITN program for the "Wireless In-body Environment Communication – WiBEC" project under the grant agreement no. 675353.

Authors' version of the paper published in the proceedings of IEEE 13th International Symposium on Medical Information and Communication Technology (ISMICT 2019). DOI: 10.1109/ISMICT.2019.8743878



(a) (b) (c) (d)

Fig. 1. Different situations of interest in our work. (a) Bile. (b) Bubbles in clear liquid. (c) Bile with bubbles and food debris. (d) Uncovered mucosa.

II. Background

The topic of uninformative frame regions due to intestinal content in CE has been investigated in previous work. The first work done on this, to the best of our knowledge, is the work by Vilariño et al from 2006. [3] It is limited only to the detection of entire frames that contain bubbles, through an unsupervised learning method based on a threshold applied to the response of an image to a bank of Gabor filters.

A multi-stage method detecting both bubbles and bile was first proposed by Bashar et al. [4] and later modified by Sun et al. [5]. It detects uninformative frames in two stages, where each stage is focused on a certain type of content. Both methods use local colour histograms for the detection of liquid, but where Bashar et al. make use of Laguerre Gauss circular harmonic function filters for the detection of bubbles, Sun et al. use local quantised histograms of classic colour local binary patterns (CLBP). Sun et al. also chose for a k-nearest neighbour classifier instead of the support vector machine (SVM) classifier used by Bashar et al.

Other methods have attempted to detect both types of intestinal content in a single stage. Namely, Khun et al. [6] used colour wavelet decomposition in combination with an SVM classifier for this purpose. Seguí et al. [7] instead extracted a 64-bin colour histogram, which they concatenated with the number of points from speeded up robust features (SURF). They separately used SVM and neural network (NN) classification, obtaining higher accuracy for SVM in general.

In all of the methods above, the part of supervised learning was limited to entire frames, while in some of them unsupervised segmentation algorithms are applied to detected frames to segment the area with intestinal content. The work done by Haji-Maghsoudi et al. [8] attempts to directly classify regions instead of frames. They use a complex, multi-step approach and make use of two NN classifiers. The first classifier is applied to whole images, using morphological operations and fuzzy k-means in the feature extraction. The result of the classification of an image is then used to define the parameters of a segmentation algorithm that is applied to it, after which it is divided into non-overlapping regions of 32 x 32 pixels. From these regions they extract statistical features and then classify those through a second NN classifier obtaining the final classification results.

Table I gives an overview of the described methods and their reported evaluation measures and values. Due to the absence of an extensive public database, we also included the number of images and unique patients in the different data sets used in each method.

III. Method

We implemented and compared two separate methods for classification of patches with intestinal content with the purpose of automatically detecting such areas in CE images. The first is based on hand-crafted feature extraction and training a classifier on those features, while the other is an end-to-end learning technique based on convolutional neural networks (CNNs). For fair comparison of both methods, we employed exactly the same data separation for our training and test data in both methods. We performed the detection at a pixel level by bilinearly interpolating probabilities between the centres of the patches containing the pixel, using the probabilities estimated by our model for the corresponding patches as the values at those centres.

A. Data Separation

As we required our data to be regional, the data collection consisted of two parts. First, medical specialists extracted the part of 35 different CE videos that corresponds to only the small intestine, with a resolution of 576x576 pixels. We then extracted frames at regular intervals of one minute leading to 563 frame images, from which we extracted square regions, hereafter referred to as patches, of predefined size. The patches were extracted with a possible overlap of half the dimensions both in width and in height. Specialists manually selected and labelled several patches per image where the mucosa was either entirely visisble, hereafter simply referred to as *clean*, or fully covered by intestinal content, hereafter referred to as *dirty*.

To evaluate how well our models generalise to new input samples, we performed external 5-fold cross validation. We ensured that patches we tested on did not originate from the same images as patches used in the training phase by partitioning our frame images in 5 equally sized sub-sets, taking a different sub-set as test set for each fold and the remaining subsets as the corresponding training data. This way, we ended up with the data shown in Table II.

TABLE I

SUMMARY OF RELEVANT METHODS USED SO FAR, WITH THE TYPE OF INTESTINAL CONTENT THEY DETECT (WHERE B REFERS TO BUBBLES AND L REFERS TO LIQUID), THE EVALUATION MEASURE WITH THE OBTAINED VALUE, THE DETECTION TARGET AND THE NUMBER OF FRAMES IN THE DATA SET, WITH THE NUMBER OF DIFFERENT PATIENTS IN PARENTHESES.

Year	Author	IC Type	Measure	Value	Target	Data Set
2006	Vilariño et al. [3]	B	accuracy	95.5%	frames	-
2008	Bashar et al. [4]	B	accuracy	99.82%	frames	14841 (3)
2008	Bashar et al. [4]	B	recall	97.48%	frames	11354 (3)
2009	Khun et al. [6]	LB	accuracy	94.10%	frames	200 (1)
2012	Sun et al. [5]	L	accuracy	99.31%	frames	2300 (3)
2012	Sun et al. [5]	B	accuracy	97.54%	frames	1700 (3)
2012	Seguí et al. [7]	LB	accuracy	91.6%	frames	100000 (50)
2014	Haji-Maghsoudi et al. [8]	LB	accuracy	98.15%	regions	2400 (60)

B. Method 1: Feature Extraction and SVM (FE-SVM)

In our first method we defined a hand-crafted feature descriptor, using both colour and texture information, and trained a classifier on those features. We later refer to this method as FE-SVM. Specifically, our feature vector was the concatenation of uniform rotation-invariant local binary patterns (LBP^{riu2}) [9] on the R, G and B colour channels and the mean of the a and b colour channels of the patch image in the CIE *L*a*b** (CIELAB) colour space [10]. We then trained an SVM classifier on the extracted feature data to construct a model capable of distinguishing between both classes. In SVM classification, we attempt to find an optimal solution to the problem

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i((w)^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{1}$$

The function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ is the kernel function, which in our case is the is the radial basis function (RBF), given by $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$ with $\gamma > 0$. For the optimisation of the parameters C in Eq. (1) and γ in the kernel function, we performed an extensive grid search using internal 4-fold cross validation to prevent over-fitting, thus training 4 times for each possible combination of parameters, each time leaving out a different partition of the data for validation. First, we performed a coarse grid search attempting all combinations of γ and C with γ ranging from 2^{-15} to 2^3 and C ranging from 2^{-5} to 2^{15}, with the exponent increasing in steps of 2 for each parameter. We then performed a fine grid search in which each of the two parameters takes values between 2^{k-1} and 2^{k+1}, with k being the exponent with highest average validation accuracy for that parameter among the 4 folds in the coarse grid search. Here we increased k in steps of 0.25 instead and we chose the value for which we obtain highest average validation accuracy among the 4 folds.

In the same way, we simultaneously optimised the parameters of LBP^{riu2}, concretely the number of neighbours (8 or 16) and the radius (ranging from 1 to 10), along with the patch size of s x s pixels ($s = 64$, $s = 32$ and $s = 16$), attempting each different combination of both these parameters and the SVM parameters. As for the patch size, the patches were annotated in the greatest size $s = 64$ in order to programmatically extract a smaller region around the centre of each patch as our data for the smaller patch sizes, as all patches were either entirely clean or entirely dirty. Fig. 2c shows an example of all patches we can extract from an image this way when $s = 64$.

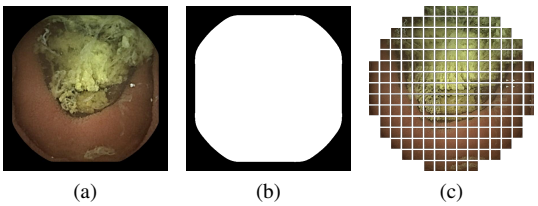
C. Method 2: Convolutional Neural Networks

Our second method is based on deep learning through convolutional neural networks (CNNs). A CNN is a type of artificial neural network specifically designed for image data [11], which automatically automatically extracts features from raw image data as opposed to hand-crafted feature extraction. It extracts these features in *convolutional layers*, i.e. layers that convolve the image through a bank of filters of pre-defined size, where we define the filter values as its weights. For the training procedure, the input images are divided into a training set and a validation set. The images in the training set are then divided into batches of a chosen size and each time a batch has been fed through the network and the error has been measured, the weights of the network are updated according to a chosen optimisation algorithm. Once an *epoch* has been

TABLE II

THE DATA SETS OF IMAGE PATCHES USED FOR TRAINING AND EVALUATION IN THE 5-FOLD CROSS VALIDATION PROCEDURE IN BOTH OF OUR METHODS.

	Training		Test	
	Clean	Dirty	Clean	Dirty
Fold 1	17378	16974	5395	4541
Fold 2	18109	16358	4555	5249
Fold 3	17888	16329	4809	5283
Fold 4	17819	16508	4888	5077
Fold 5	17084	17522	5733	3911



(a) (b) (c)

Fig. 2. (a) An example of an image from our data set. (b) The mask we apply to this image. (c) All patches we could extract from the area of interest using our method with patch size $s = 64$.

completed, i.e. the network has been trained in this way on all batches a single time, the performance is measured on the validation set using a chosen metric. This training procedure is usually repeated until the performance on the validation data no longer improves or until a maximum number of epochs has been completed.

The above training procedure requires a significant amount of labeled input data and computational resources. Regardless, CNNs have been applied successfully in different computer vision domains in recent years, largely made possible by the concept of *transfer learning*: taking the base of existing CNN models that were trained on vast amounts of labelled image data and reusing this in a new model [12]. As the higher layers are more relevant to the specific task, we can define new classification layers for our task and either only train those, using the base taken from the existing model as a feature extractor, or go deeper and also retrain the higher layers of the base, keeping the weights of the lowest layers fixed. We will do the latter, which is also referred to as *fine-tuning*.

In our case, we separately fine-tuned two VGGNet architectures [13], namely VGG-16 and VGG-19. We chose for these architectures since we found the VGGNet architectures to be most successful in previous work on CE images [14]. Additionally, they are the most widely used purely sequential architectures and have consistent filter sizes in the convolutional layers. For these reasons they are intuitive architectures to comprehend and to fine-tune. Namely, VGGNet was the first architecture to consistently use 3x3 kernel sizes in the convolutional layers, replacing larger kernels employed in AlexNet. By concatenating multiple convolutional layers with this kernel size, the network becomes deeper and more complex features can be represented at a lower cost.

VGG-16 and VGG-19 both consist of 5 sequential blocks, which on its turn consist of sequential convolutional layers followed by a max pooling layer. The architectures differ in the number of convolutional layers they contain as visualised in Fig. 3. Both architectures originally have three fully connected layers and a soft-max layer for classification, leading to a total of 16 and 19 *weight layers* respectively. However, for both architectures we removed those layers and replaced them by two fully connected layers, one of size 4096 and the other of size 2048, followed by a soft-max layer. In this way, we ended up with the final architectures shown in Fig. 3. We also defined a drop-out of 0.5 and 0.25 respectively for the two fully connected layers to avoid over-fitting, which is a common problem of deeper CNN architectures.

For training our models, we made use of the Keras framework [15] in Python with TensorFlow [16] as its backend. We retrained the weights of our modified VGG-16 and VGG-19 architectures from the lowest layer of the highest convolutional block in both cases. We partitioned each of our training data sets into a training set and a validation set, ensuring a 20% of the data in the validation set. As per the image size used to pre-train the networks, we had to rescale our patch images from 64x64 to 224x224. We set all the other parameters explained above empirically. We used the accuracy as our performance metric and trained the networks for a maximum of 100 epochs or until validation accuracy no longer improved during 15 epochs or more. We set the batch size to 4 due to memory limitations and the optimisation algorithm to Stochastic Gradient Descent (SGD). For SGD we used the binary cross-entropy loss function and established a momentum of 0.98 and a learning rate of 0.0001 without decay. To increase the robustness of our models, we also used data augmentation with a rate of 0.02.

IV. Results

For method 1, we first optimised the parameters of our feature extraction during the cross-validation procedure as explained in Section III-B. The optimal parameters we found in this way were: 16 neighbours p and a radius r of 1 for LBP, and 64 for s in the patch size $s \times s$. For method 2 we used the optimal value of 64 for s in the patch size $s \times s$ found in method 1. We did not need to optimise any feature extraction

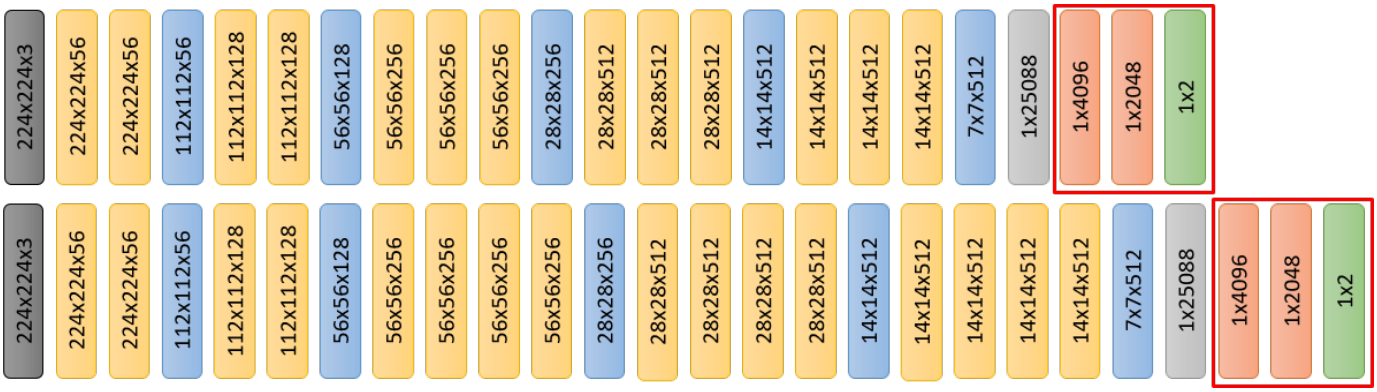


Fig. 3. An overview of the VGG-architectures we implemented, with our modified VGG-16 at the top and our modified VGG-19 at the bottom, where each rectangle represents a layer of the network. The layers surrounded by the red rectangles are the layers we defined by which we replaced the fully connected layers of the original VGG-16 and VGG-19 architectures. The text corresponds to the output sizes. The far left layers are the input layers, yellow are convolutional layers, blue are max pooling layers, light gray are flattening layers, red are fully connected layers and green are the softmax classification layers.

parameters for this method, as the neural networks learns the features by itself.

The results obtained per fold in our 5-fold cross-validation and the average results over the folds, both for FE-SVM with the optimal parameters and for the different CNN architectures, are given in Table III. We used macro-averaging to obtain the average results. Visualising the probabilities as a heat map using the VGG-19 model with the highest average accuracy, we obtained images as shown in Fig. 4.

V. Conclusions and Future Work

We presented and compared two different methods to automatically detect uninformative areas in CE videos: one based on hand-crafted feature extraction and SVM classification and the other based on CNNs. While previous work has applied computationally expensive, multi-stage approaches, in this work we showed that similarly high accuracy can be achieved with the decision of a single model without any pre-processing or post-processing of the images, obtaining an accuracy of 95.15%.

We trained our models on an extensive database of more than 40000 annotated regions from frames from videos of 35 patients, ensuring a high variability of the data, and ensured a fair evaluation of the performance of our classifiers on new patches by using a test set of patches that originated from frames that were not used for training. For the VGG-19 and VGG-16 models from our CNN-based method we obtained similarly high results, which significantly outperformed our hand-crafted feature extraction method.

Instead of employing segmentation methods as in previous research, we developed a method that uses the overlap between patches and the patch probabilities given by the model to estimate probabilities at a pixel level. We can then use the estimated probabilities to determine the percentage of covered mucosa in each frame of a video and thus calculate a percentage over the whole video. In a clinical setting, these results can be used to automatically and objectively evaluate how different methods of small bowel preparation for CE affect the visibility of the mucosa.

In cases where our method fails, it appears to do so mainly on image areas containing slightly turbid liquid that blurring the mucosa, while obscuring it sufficiently to complicate the diagnosis of a pathology. We also observed how it can fail on more densely present bile, particularly when it appears to have a colour similar to the intestinal mucosa (e.g. due to the illumination). Examples of such cases are shown in the bottom row of Fig. 4.

In future work, in order to further improve results, we may incorporate more of the aforementioned images into our training data to train a model that is more invariant to illumination changes and is stricter with turbid liquid. It could also be interesting to investigate combining the strengths of the different models we trained, e.g. by creating a classifier ensemble, and to study further inclusion of pathological areas, accounting for their low frequency. More importantly, in order to meet the objective of developing a standard for this purpose, we need to clinically evaluate our method by comparing sequential frame classification of our algorithm in new videos with manual classification by multiple medical specialists.

TABLE III
Results obtained in terms of accuracy, sensitivity and specificity.

Method	Fold	Accuracy	Sensitivity	Specificity
FE + SVM	Fold 1	85.32%	89.43%	80.42%
	Fold 2	87.62%	86.94%	88.21%
	Fold 3	86.84%	89.83%	84.12%
	Fold 4	87.83%	93.64%	82.23%
	Fold 5	88.75%	89.60%	87.50%
	Average	**87.32%**	**90.73%**	**83.71%**
VGG-16	Fold 1	93.49%	92.59%	94.64%
	Fold 2	95.04%	93.50%	96.45%
	Fold 3	94.10%	90.83%	97.53%
	Fold 4	93.71%	89.46%	98.73%
	Fold 5	95.32%	94.98%	95.86%
	Average	**94.33%**	**92.27%**	**96.61%**
VGG-19	Fold 1	95.15%	94.59%	95.84%
	Fold 2	95.92%	95.43%	96.35%
	Fold 3	94.33%	91.00%	97.83%
	Fold 4	94.62%	91.75%	97.77%
	Fold 5	95.74%	96.60%	94.48%
	Average	**95.15%**	**93.87%**	**96.45%**

Acknowledgement

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

References

[1] V. Pons Beltrán, B. González Suárez, C. González Asanza, E. Pérez-Cuadrado, S. Fernández Diez, I. Fernández-Urién *et al.*, "Evaluation of different bowel preparations for small bowel capsule endoscopy: A prospective, randomized, controlled study," *Digestive Diseases and Sciences*, vol. 56, no. 10, pp. 2900–2905, Oct 2011. [Online]. Available: https://doi.org/10.1007/s10620-011-1693-z

[2] A. Klein, M. Gizbar, M. J. Bourke, and G. Ahlenstiel, "Validated computed cleansing score for video capsule endoscopy," *Digestive Endoscopy*, vol. 28, no. 5, pp. 564–569, 2016, jGES-DEN-2015-5443.R2. [Online]. Available: http://dx.doi.org/10.1111/den.12599

[3] F. Vilarino, P. Spyridonos, O. Pujol, J. Vitria, and P. Radeva, "Automatic detection of intestinal juices in wireless capsule video endoscopy," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4, 2006, pp. 719–722.

[4] M. K. Bashar, K. Mori, Y. Suenaga, T. Kitasaka, and Y. Mekada, "Detecting informative frames from wireless capsule endoscopic video using color and texture features," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008*, D. Metaxas, L. Axel, G. Fichtinger, and G. Székely, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 603–610.

[5] Z. Sun, B. Li, R. Zhou, H. Zheng, and M. Q. H. Meng, "Removal of non-informative frames for wireless capsule endoscopy video segmentation," in *2012 IEEE International Conference on Automation and Logistics*, Aug 2012, pp. 294–299.

[6] P. C. Khun, Z. Zhuo, L. Z. Yang, L. Liyuan, and L. Jiang, "Feature selection and classification for wireless capsule endoscopic frames," in *2009 International Conference on Biomedical and Pharmaceutical Engineering*, Dec 2009, pp. 1–6.

[7] S. Segui, M. Drozdzal, F. Vilarino, C. Malagelada, F. Azpiroz, P. Radeva, and J. Vitria, "Categorization and segmentation of intestinal content frames for wireless capsule endoscopy," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1341–1352, 2012.

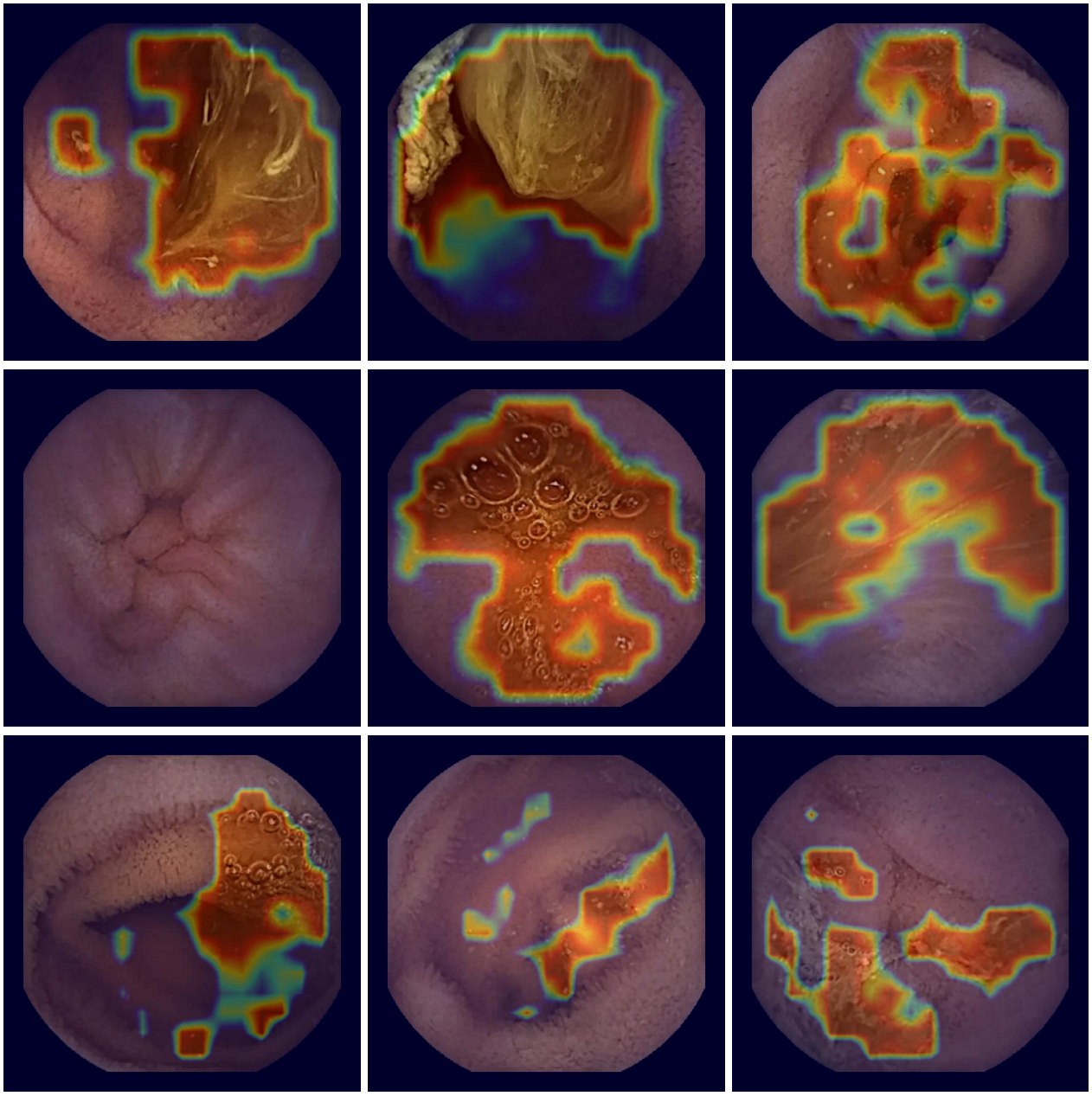


Fig. 4. Visual results from our region-based method, where red areas indicate intestinal content detected by our method, while blue areas were classified as clean mucosa.

[8] O. H. Maghsoudi, A. Talebpour, H. Soltanian-Zadeh, M. Alizadeh, and H. A. Soleimani, "Informative and uninformative regions detection in wce frames," *Journal of Advanced Computing*, vol. 3, no. 1, pp. 12–34, 2014.

[9] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, Jul 2002.

[10] K. McLaren, "Xiii—the development of the cie 1976 (l* a* b*) uniform colour space and colour-difference formula," *Journal of the Society of Dyers and Colourists*, vol. 92, no. 9, pp. 338–341, 1976.

[11] CS231n. Cs231n convolutional neural networks for visual recognition. [Online]. Available: http://cs231n.github.io/convolutional-networks/

[12] TensorFlow. (2018) How to retrain an image classifier for new categories. [Online]. Available: https://www.tensorflow.org/hub/tutorials/image_retraining

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] P. Li, Z. Li, F. Gao, L. Wan, and J. Yu, "Convolutional neural networks for intestinal hemorrhage detection in wireless capsule endoscopy images," in *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1518–1523.

[15] F. Chollet *et al.*, "Keras," 2015, software available from keras.io. [Online]. Available: https://keras.io

[16] M. A. et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/