Accurate Real-time Polyp Detection in Videos from Concatenation of Latent Features Extracted from Consecutive Frames

Hemin Ali Qadir[1], Younghak Shin[2], Jacob Bergsland[1], Ilangko Balasingham[1,3]

[1]*Intervention Centre, Oslo University Hospital, Oslo, Norway*
[2]*Department of Computer Engineering, Mokpo National University, Mokpo, South Korea*
[3]*Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway*

***Abstract*—An efficient deep learning model that can be implemented in real-time for polyp detection is crucial to reducing polyp miss-rate during screening procedures. Convolutional neural networks (CNNs) are vulnerable to small changes in the input image. A CNN-based model may miss the same polyp appearing in a series of consecutive frames and produce unsubtle detection output due to changes in camera pose, lighting condition, light reflection, etc. In this study, we attempt to tackle this problem by integrating temporal information among neighboring frames. We propose an efficient feature concatenation method for a CNN-based encoder-decoder model without adding complexity to the model. The proposed method incorporates extracted feature maps of previous frames to detect polyps in the current frame. The experimental results demonstrate that the proposed method of feature concatenation improves the overall performance of automatic polyp detection in videos. The following results are obtained on a public video dataset: sensitivity 90.94%, precision 90.53%, and specificity 92.46%.**

***Index Terms*—Artificial Intelligence, Deep Learning, Convolutional Neural Network (CNN), Polyp Detection, Colonoscopy**

I. Introduction

Colorectal cancer ranks third in terms of worldwide incidence, but second in terms of mortality for both genders [1]. Most cases of colorectal cancer originate from abnormal growths of glandular tissue in the inner lining of the colon and rectum. These abnormal tissue growths are known as polyps which are benign in the early stage. Untreated polyps might become malignant and potentially life-threatening cancer [2]. Colonoscopy is the gold standard method for colon screening and allows the detection and removal of polyps during the procedure. It has been reported that polyp miss rate can be as high as 22%-28% depending on the experience of the endoscopists [3].

During the last few years, deep learning (DL), a type of machine learning and artificial intelligence (AI), has proven to be the most successful computational method for automatic polyp detection and reduction of polyp miss rate. In particular, convolutional neural networks (CNN), a special form of DL applied for image analysis, have shown outstanding performance in automatic polyp detection and segmentation in colonoscopy images and videos [4]–[8]. Several studies, however, demonstrated that DL-based networks including CNNs are vulnerable to perturbations and noise [9]–[14]. Jiawei Su et al. [14] showed that CNNs can be easily fooled by small attacks e.g. by adding relatively small perturbations (one pixel) to the input image. In colonoscopy video analysis, CNNs might be fooled by the specular light reflections and small changes in polyp (other elements) structures appearance. This means that CNNs can easily miss the same polyp presenting in a sequence of consecutive frames and produce unstable detection output contaminated with a high number of false positives and false negatives.

In this paper, we propose a novel method to address the above-mentioned problem. The proposed algorithm concatenates the features maps of previous frames with the current frame at the bottleneck layer (latent space) of an encoder-decoder based CNN architecture without adding too much complexity. The hypothesis is that neighboring frames are closely related to each other, and thus their extracted CNN features should be closely similar and contain complementary information. We choose to integrate the proposed method into a two-dimensional (2D) CNN-based encoder-decoder network because its elegant architecture facilitates the concatenation of features extracted from a series of consecutive frames by the encoder part in the latent space. Furthermore, most of the 2D CNN-based encoder-decoder networks are designed from fully convolutional neural networks (F-CNN) predicting outputs in a single shot feed-forward manner which makes them eligible for real-time implementation. Like [6], we enforce the decoder part to predict 2D Gaussian shapes for polyp regions presented in the current input frames from the concatenated features. We demonstrate that the proposed method is efficient to increase polyp detection capability by increasing the number of true positives and reducing the number of false positives.

II. Materials and Methods

A. Methodology

Fig. 1 presents our proposed method to detect polyps in a one-shot manner in videos. The method is developed based on a 2D CNN encoder-decoder network. The 2D encoder-decoder networks are originally developed for single image analysis but do not incorporate temporal information among neighboring frames when they are applied for video analysis.

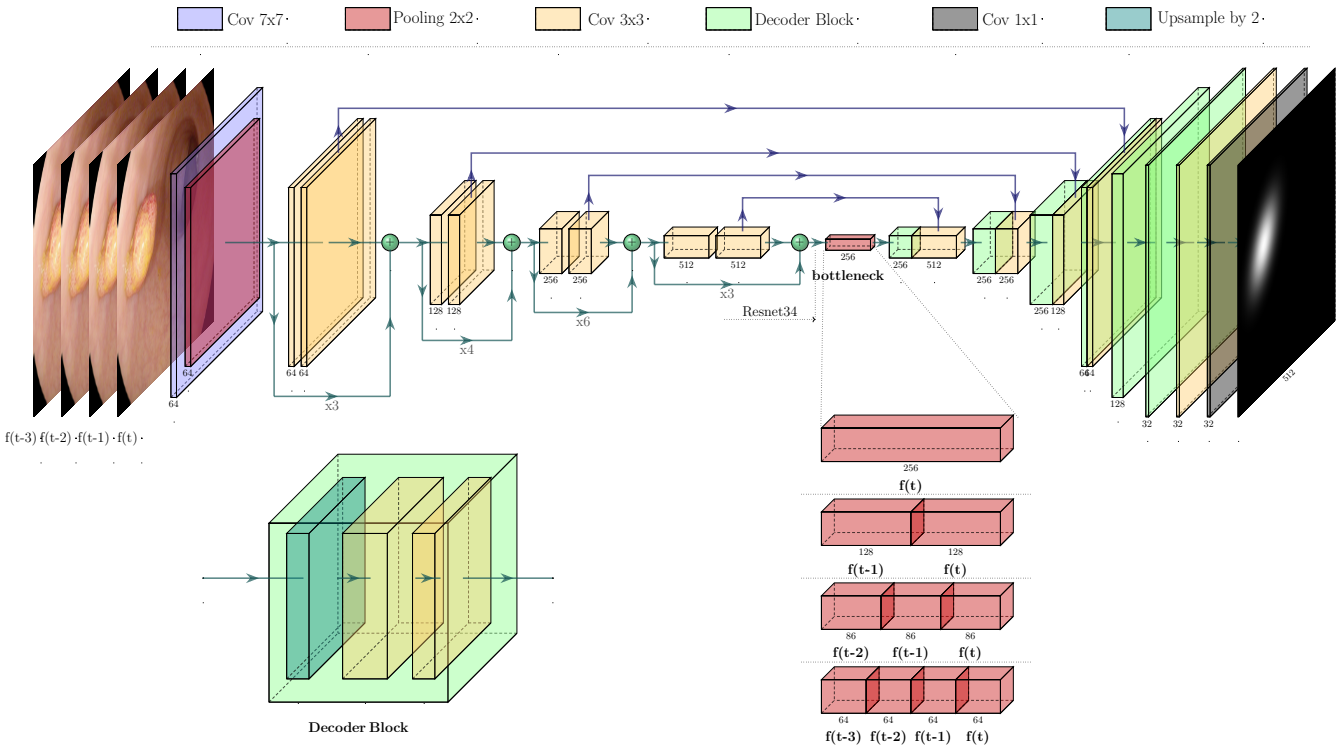


Fig. 1. The proposed model which consists of two paths: an encoder extracting features from the input frames, and a decoder interpreting the extracted features to predict the final detection output.

This property together with the CNN vulnerability make the 2D encoder-decoder networks produce unstable output predictions for consecutive frames contaminated with a lot of false detection outputs. In this section, we provide a detailed description of our proposed method to make a CNN-based encoder-decoder network a more stable and reliable polyp detector suitable for video analysis. We incorporate temporal information from previous frames to analyze the current frame. We concatenate extracted features from N previous frames $f_{xy}(t-n), n \in [1, 2, ..N]$ with the extracted features of the current frame $f_{xy}(t)$ in the bottleneck layer (latent space).

1) Network architecture: In this work, we adapt AlbuNet34 proposed by Shvets et al. [15] for our polyp detection model as shown in Fig. 1. AlbuNet34 is a UNet-like architecture [16] consisting of two paths: the contracting path and expanding path. The contracting path (encoder part) takes in an input image frame $f_{xy}(t)$ and progressively extracts abstract features. The expansive path (decoder part) interprets the extracted features and enables precise localization. AlbuNet34 uses ResNet34 [17] pre-trained on Imagenet dataset [18] for the encoder part. We choose AlbuNet34 because it combines the advantages of both UNet-like architectures [16] and residual learning [17]. In addition, AlbuNet34 is designed from fully convolutional neural networks (F-CNN) predicting outputs in a single shot feed-forward manner which makes them eligible for real-time implementation.

The first block of the encoder is a kernel of size 7×7 with stride 2 followed by a max-pooling layer with stride 2. The rest blocks of the encoder consist of repetitive residual blocks. In every residual block, the first convolution operation is applied with stride 2 to provide downsampling, while the rest convolution operations are applied with stride 1. We apply a 2×2 max-pooling operation on the final output feature maps of the final residual block of ResNet34. The result of this max-pooling operation is stored in the bottleneck layer (the latent space). We add this bottleneck layer to facilitate the incorporation of temporal information from consecutive frames, which will be discussed in Section II-A2.

The decoder part consists of several decoder blocks, each block is concatenated with the corresponding encoder block. In every decoder block, an upsampling operation is applied to upsample the feature maps by 2, followed by two padded convolution operations of a kernel of size 3×3 with stride 1 followed by a rectified linear unit (ReLU). The first block of the decoder starts with interpreting the abstract features stored in the bottleneck layer. To generate the final output, we apply a 1×1 convolution operation followed by the $tanh$ activation function. Our detection model generates an output image which has the same resolution as the input image frame $f_{xy}(t)$, with the predicted 2D Gaussian shapes $\hat{Y}_{xy}(t)$ discussed in Section II-A3.

2) Concatenation of consecutive features: Qadir et al. [7] showed that the same CNN-based detector can miss the same polyp appearing in the neighboring frames due to changes in the light conditions, appearances, inherent noises,

blurriness, etc. This is due to the vulnerability of CNN to small perturbations. We solve this problem by concatenating the feature maps of a series of consecutive frames extracted by the encoder part. Neighboring frames are closely related to each other and thus their extracted feature maps should be closely similar and contain complementary information. The elegant structure of UNet-like architectures facilitates this feature concatenation in the bottleneck layer. This way we can incorporate temporal information among neighboring frames into the detection model.

We store the feature maps from N previous frames $f_{xy}(t-n), n \in [1, 2, ..N]$ and concatenate them with the feature maps extracted from the current frame $f_{xy}(t)$ in the bottleneck layer. We set the number of activation maps in the bottleneck layer to be 256 maps. We equally divide the bottleneck layer into N slots of $256/N$ activation maps based on the number of previous frames involved. For instance, when only one previous frame $f_{xy}(t-1)$ is incorporated, 128 feature maps are extracted from the current frame $f_{xy}(t)$ and 128 feature maps from $f_{xy}(t-1)$. However, when the result of this division is a floating number we round it up to an approximate number. For instance, when we consider two previous frames $f_{xy}(t-n), n \in [1, 2]$, the result of 256/3 is 85.33, thus we use 86 maps for each frame, resulting in 258 feature maps in the bottleneck layer.

This concatenation of feature maps helps combine complementary information from a series of previous frames $f_{xy}(t-n), n \in [1, 2, ..N]$ with the current frame $f_{xy}(t)$, reducing the effect of a small perturbation and/or change that may pop up in the current frame $f_{xy}(t)$ and fool the CNN-based detector. Therefore, this concatenation strategy helps the CNN-based encoder-decoder network improve its accuracy and produce more stable detection outputs for a series of neighboring frames.

3) 2D Gaussian shapes for polyp regions: Qadir et al. [6] demonstrated that a CNN-based encoder-decoder network is more efficient in detecting polyps when it is trained on 2D Gaussian shapes as the ground-truth masks instated of using binary masks. We follow the same procedure proposed in [6] to train our detection model to predict a 2D Gaussian shape, $\hat{Y}_{xy}(t) \in [0, 1]^{W \times H \times 1}$, for a polyp region in an input RGB image frame at time t, $f_{xy}(t) \in [R]^{W \times H \times 3}$, where W is the width and H is the height of both $f_{xy}(t)$ and $\hat{Y}_{xy}(t)$.

We transform the provided binary ground-truth masks, $X_{xy}(t) \in \{0, 1\}^{W \times H \times 1}$, to 2D Gaussian ground-truth masks, $Y_{xy}(t) \in [0, 1]^{W \times H \times 1}$, as described in [6]. The 2D Gaussian ground-truth masks are meant to reduce the impact of the outer edges during training and force the network to learn the surface patterns of different polyps more efficiently.

Using 2D Gaussian shapes can also help to realize polyp detection in real-time. We can generate all detected bounding boxes directly from the predicted 2D Gaussian shapes without the need for computing non-maximum suppression (NMS) to eliminate overlapping bounding boxes [19]. At the inference time, we use the strength of the predicated 2D Gaussian shapes as the confidence values of the detected bounding boxes and calculate the two size-adaptive standard deviations (σ_x and σ_y) for the size of the detected bounding boxes.

B. Datasets

In this study, we used four publicly available datasets of still images and videos:

a) CVC-ColonDB [20]: This is a dataset of 300 still images extracted from 15 colonoscopy videos, each with a unique polyp (15 unique polyps in total). The images have a resolution of 574x500 pixels.

b) CVC-ClinicDB [21]: This is a dataset of 612 still images extracted from 29 colonoscopy videos, each with at least a polyp. There exists 31 unique polyps presented 646 times in the 612 images with a pixel resolution of 384x288.

c) ASU-Mayo Clinic [22]: This is a dataset of 38 colonoscopy videos. 20 videos are assigned as a training set while the other 18 videos are assigned as a testing set. Because of the copyright license, we could not get access to the 18 testing videos. The 20 training videos consist of 10 positive videos with a total of 5402 frames (3846 polyp frames) and 10 negative videos with a total of 13500 frames.

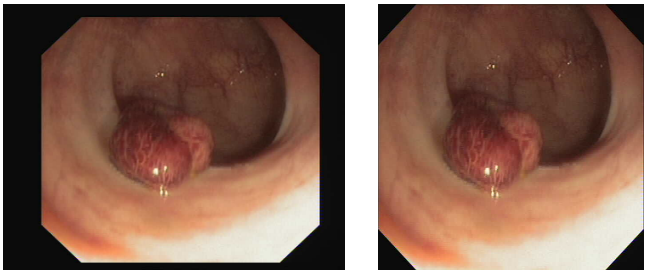
d) CVC-ClinicVideoDB [8]: This is a dataset of 18 colonoscopy videos, each with a different polyp. This dataset comprises 11954 frames (10025 polyp frames). The resolution of the frames is 268x576 pixels.

C. Training Details

During training, a video dataset is needed in order to capture the temporal patterns among neighboring frames. In a video, the neighboring frames look closely similar. If there are not enough diverse frames in the training videos, our detection model may easily get overfitted on the training frames. To train our detection model and avoid this phenomenon, we use the two datasets of still images namely CVC-ColonDB and CVC-ClinicDB alongside the dataset of videos namely CVC-ClinicVideoDB. We build our final training dataset by mixing the frames of the videos and the still images. Whenever a still image is encountered during training, we count it as its previous frames as well.

As mentioned before, the encoder part uses ResNet34 initialized with ImageNet pre-trained weights. In contrast, we randomly initialize the network parameters of the decoder part. To clean the final training dataset, we apply several simple pre-processing methods to the input images (see Fig. 2):

1) Image cropping: to remove the canvas around the informative part of the images.
2) Image resizing: by changing the image resolution to 512×512 because the pre-trained ResNet34 accepts this image resolution.
3) Image normalization: by converting the pixel values from [0, 255] to [0, 1], subtracting them from the mean, and dividing them by standard deviation both pre-calculated from the ImageNet dataset.



(a) The original image (b) Cropped and resized image

Fig. 2. An example showing image cropping and resizing to 512×512. a)

To add father image-level diversity through depth and scale, we apply several image augmentation methods on the fly e.g. rotation, vertical, horizontal flips, random zoom-in (up to 25%), and zoom-out (up to 50%), and color augmentations in HSV space. To keep the balance between large and small polyps and avoid biasing, we apply less zoom-in compared to zoom-out because the training dataset contains more large polyps than small ones.

We randomly split the training dataset into training (85%) and validation (15%) subsets. We use Adam optimizer with a batch size of 10 and a learning rate of 1×10^{-4} to train the model for 20 epochs. Following the recommendations given in [15], we change the learning rate to 1×10^{-5} to train the model up to 60 epochs. We use the validation subset to choose the learning rate decay strategy and the number of epochs.

Finally, we use the mean squared error (squared L2 norm) between each element in the input ground-truth image frame at time t, $Y_{xy}(t)$, and the output image frame at time t, $\hat{Y}_{xy}(t)$ with the predicted 2D Gaussian shapes.

$$L2\,loss = \frac{1}{M}\sum_{i}^{M}[Y_{xy}(t)_i - \hat{Y}_{xy}(t)_i]^2, \tag{1}$$

where M is the batch size. We choose L2 norm loss function because it can significantly (quadratically) penalize large errors. This property of L2 norm makes it favorable especially for the prediction of 2D Gaussian shapes which are normally distributed around a mean value.

D. Evaluation Metrics

The output of the proposed method is a set of bounding boxes around the suspected regions in the input frame. The detected bounding boxes are either true or false alarms. To quantitatively evaluate the performance of the proposed method, we calculate sensitivity (recall) and precision using well-known medical parameters:

a) true positive (TP): a true detected bounding box around a positive region in the input frame,
b) false positive (FP): a false detected bounding box around a negative region,
c) true negative (TN): a true detection output for a negative frame in which no bounding box is detected.
d) false negative (FN): false detection output where a polyp is missed in a positive frame.

Sensitivity measures the ratio of TPs to the total number of polyps in the test set,

$$Sensitivity = TP/(TP + FN) \times 100, \tag{2}$$

while precision measures the ratio of TPs to the total number of detected bounding boxes including FPs,

$$Precision = TP/(TP + FP) \times 100, \tag{3}$$

specificity measures the ratio of actual negative frames that are correctly classified,

$$Specificity = TN/(TN + FP) \times 100. \tag{4}$$

III. Results and Discussion

To quantitatively evaluate our proposed method, we used the ASU-Mayo clinic dataset, more specifically the 20 videos that were originally assigned for training purposes by the authors. The 10 positive videos were used to compute the performance of the proposed method in terms of sensitivity and precision. In contrast, the 10 negative videos were used for the evaluation of specificity. We present our results in curves to facilitate the visualization of the performance evaluation when information from previous frames is incorporated with the current frame.

A. Results on positive videos

Fig. 3 shows sensitivity and precision measurement of the proposed method for all four scenarios. When the current frame is examined alone, AlbuNet34 can provide high sensitivity (91.27%) but struggles to offer the same level of performance for precision (67.21%) due to the generation of a substantial number of FPs.

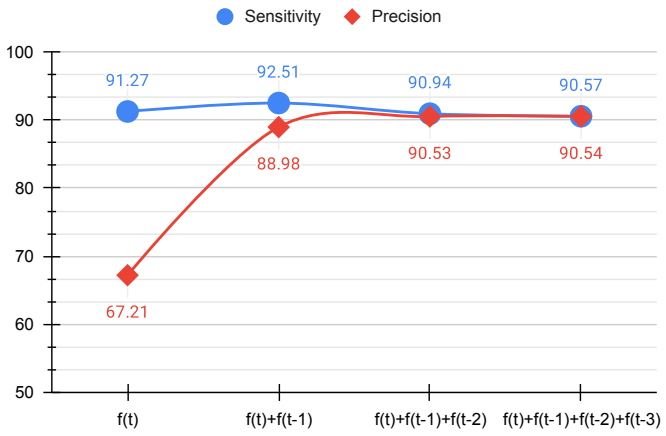


Fig. 3. Numeral results showing the performance improvement of the proposed method in terms of sensitivity and precision

When extracted features from the first previous frame are concatenated with the extracted features of the current frame, the model enjoyed 1.5% sensitivity increase while the increase in precision is as high as 21.4%. This result indicates that integrating information from one previous frame can benefit the model to increase both measures by slightly

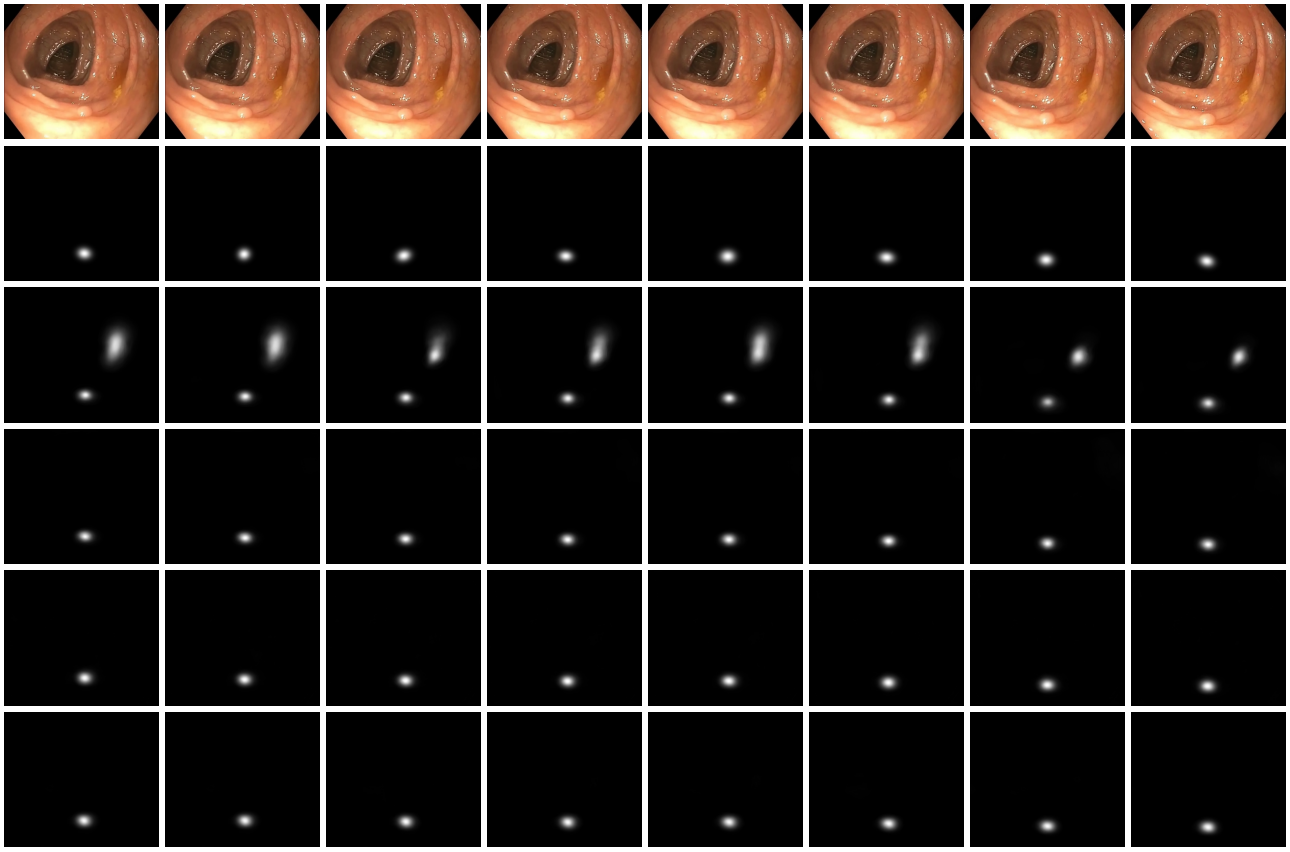


Fig. 4. An example showing detection outputs of a series of consecutive frames. Each row shows the following:- 1st) input image frames, 2nd) corresponding ground-truth image frames. 3rd) detection output when only the current frame is used, 4th, 5th, and 6th) detection outputs when 1, 2, and 3 previous frames are included in the detection, respectively.

increasing the number of TPs and eliminating a large number of FPs. However, when information from the second and third previous frames is incorporated, model sensitivity degraded while precision improved with a little margin. The resulting degradation in the sensitivity can be due to the relative position of polyps in the current frame w.r.t the farther frames. This occurs when the colonsocopy scope moves fast in the colon leading to dramatic changes in the scene among the consecutive frames, specifically in the farther frames.

Fig. 4 shows detection outputs in a series of consecutive frames from all four scenarios. As it can be seen, the model obtained much more reliable results from the concatenated features compared to features coming from a single frame. The model generates a large number of FPs when a single frame is under investigation alone.

B. Results on negative videos

Fig. 5 shows the specificity performance of the model when it was applied to the 10 negative videos. Similar results were observed, i.e., the model's specificity increases when temporal information among neighboring frames is embedded into the model.

When only the current frame is involved in the detection process, the model specificity is low due to a large number of FPs. When the extracted features of the first previous frame are concatenated with the extracted features of the current

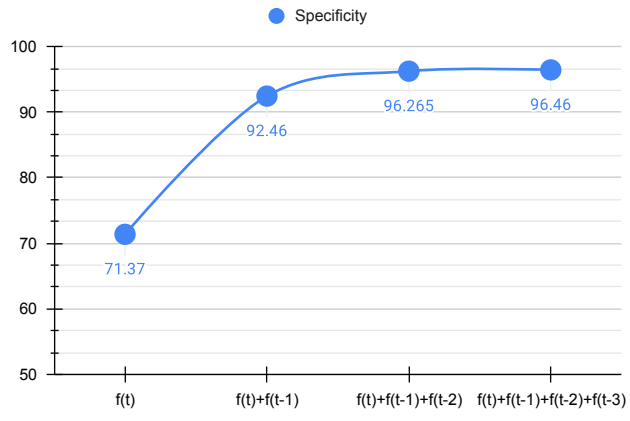
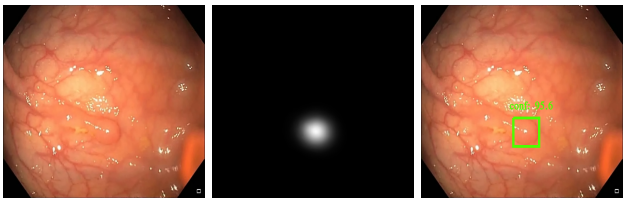


Fig. 5. Numerical results showing the performance improvement of the proposed method in terms of specificity

frame, the model specificity is raised by 20.09% which means that a lot of FPs are eliminated. The model continued to increase its specificity when the extracted features of the second and third frames are integrated into the detection process. These results indicate that temporal information is essential for CNNs to potentially reduce the number of FPs and overcome their vulnerability to small changes.

We measured the time required by the model to process

a single frame. We ran the model on the NVIDIA GeForce RTX 3090 GPU, and we observed the speed of the model which was around 11 ∓ 1 msec per frame. It is worth mentioning that the model runs at the same speed in all four scenarios. This is because we equally split the bottleneck layer based on how many previous frames are involved in the process. This way, we avoid increasing the number of activation maps in the bottleneck layer and thus avoid extra mathematical operations. Fig. 6 presents the final detection output of the proposed method. The predicted 2D Gaussian shapes are projected to bounding boxes and confidence of the detection.



(a) (b) (c)

Fig. 6. Final detection output showing the input image (a) and the predicted 2D Gaussian shape (b) projected as a bounding box and confidence value on the input image (c).

IV. Conclusions

In this paper, we presented a novel algorithm to tackle deep learning vulnerability to small changes appearing in neighboring frames. We proposed an efficient method to concatenate extracted CNN features of previous frames with the extracted CNN features of the current frame. We integrated the proposed method into a 2D CNN-based encoder-decoder model because its elegant architecture facilitates this concatenation of feature maps at the bottleneck layer, the latent space, without adding complexity to the model. The obtained results demonstrated that temporal information is essential to improve the overall performance of polyp detection for the analysis of videos. The proposed model was successful to increase the number of true positives and reduce the number of false positives.

Acknowledgment

This work was supported partially by the EU project called 5G Health Aquaculture and Transport Validation Trials (5G- HEART) funded by the H2020:ICT framework program under grant agreement number 857034.

References

[1] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 71, no. 3, pp. 209–249, 2021. 1

[2] M. Gschwantler, S. Kriwanek, E. Langner, B. Göritzer, C. Schrutka-Kölbl, E. Brownstone, H. Feichtinger, and W. Weiss, "High-grade dysplasia and invasive carcinoma in colorectal adenomas: a multivariate analysis of the impact of adenoma and patient characteristics," *European journal of gastroenterology & hepatology*, vol. 14, no. 2, pp. 183–188, 2002. 1

[3] A. Leufkens, M. Van Oijen, F. Vleggaar, and P. Siersema, "Factors influencing the miss rate of polyps in a back-to-back colonoscopy study," *Endoscopy*, vol. 44, no. 05, pp. 470–475, 2012. 1

[4] D. Wang, S. Chen, Q. Chen, Y. Cao, B. Liu, X. Liu, and X. Sun, "Afp-mask: Anchor-free polyp instance segmentation in colonoscopy," *IEEE Journal of Biomedical and Health Informatics*, 2022. 1

[5] K. Yang, S. Chang, Z. Tian, C. Gao, Y. Du, X. Zhang, K. Liu, J. Meng, and L. Xue, "Automatic polyp detection and segmentation using shuffle efficient channel attention network," *Alexandria Engineering Journal*, vol. 61, no. 1, pp. 917–926, 2022. 1

[6] H. A. Qadir, Y. Shin, J. Solhusvik, J. Bergsland, L. Aabakken, and I. Balasingham, "Toward real-time polyp detection using fully cnns for 2d gaussian shapes prediction," *Medical Image Analysis*, vol. 68, p. 101897, 2021. 1, 3

[7] H. A. Qadir, I. Balasingham, J. Solhusvik, J. Bergsland, L. Aabakken, and Y. Shin, "Improving automatic polyp detection using cnn by exploiting temporal dependency in colonoscopy video," *IEEE journal of biomedical and health informatics*, vol. 24, no. 1, pp. 180–193, 2019. 1, 2

[8] Q. Angermann, J. Bernal, C. Sánchez-Montes, M. Hammami, G. Fernández-Esparrach, X. Dray, O. Romain, F. J. Sánchez, and A. Histace, "Towards real-time polyp detection in colonoscopy videos: Adapting still frame-based methodologies for video sequences analysis," in *Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures*. Springer, 2017, pp. 29–41. 1, 3

[9] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," pp. 86–94, July 2017. 1

[10] N. Narodytska and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1310–1318. 1

[11] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436. 1

[12] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017, pp. 506–519. 1

[13] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582. 1

[14] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2019. 1

[15] A. A. Shvets, V. I. Iglovikov, A. Rakhlin, and A. A. Kalinin, "Angiodysplasia detection and localization using deep convolutional neural networks," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2018, pp. 612–617. 2, 4

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. 2

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 2

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255. 2

[19] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019. 3

[20] J. Bernal, J. Sánchez, and F. Vilarino, "Towards automatic polyp detection with a polyp appearance model," *Pattern Recognition*, vol. 45, no. 9, pp. 3166–3182, 2012. 3

[21] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, and F. Vilariño, "Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Computerized Medical Imaging and Graphics*, vol. 43, pp. 99–111, 2015. 3

[22] N. Tajbakhsh, S. R. Gurudu, and J. Liang, "Automated polyp detection in colonoscopy videos using shape and context information," *IEEE transactions on medical imaging*, vol. 35, no. 2, pp. 630–644, 2015. 3