Improved Gastrointestinal Screening: Deep Features using Stacked Generalization

Sourodip Ghosh
Department of Electronics Engineering
KIIT University
Bhubaneswar, India
sourodip.ghosh02@gmail.com

K.C. Santosh, *Senior Member, IEEE*
KC's PAMI Research Lab, Department of Computer Science
University of South Dakota
Vermillion, SD 57069, USA
santosh.kc@ieee.org

***Abstract*—Gastric malignancy - one of the five most deadliest types of cancer - exceeds annual cases by a million worldwide since 2017. Automated screening tools may help speed up the screening and clinical procedures. In this paper, we propose a binary classification approach to classify between two different types of gastrointestinal cancer tissues, namely Microsatellite Instable (MSI) and Microsatellite Stable (MSS) through stacked generalization based ensemble Deep Neural Network (DNN). Using a dataset of size $192,315$ images, we achieve an overall accuracy of 94.91% and sensitivity of 95.95%. Our results outperform previous works.**

***Index Terms*—Gastrointestinal Cancer; Microsatellite Instable; Microsatellite Stable; Deep Neural Networks; Stacked Generalization**

I. Introduction

Gastric cancer is the fifth most common malignancy in the world, after cancer affecting lung, breast, colorectum, and the prostate region [1]. Gastric cancer mostly affects older people, age ranged from $60-70$. When medical experts are considered, tumor invasive region selection is often costly and time-consuming. Besides, comprehension of study differs between each clinician to another, and therefore an automated screening procedure may aid the clinician to deconstruct results with a clearer jurisdiction.

Deep learning tools have been aiding researchers and medical practitioners during screening procedures for almost two decades [2]. In 1990, LeCunn et al. [3] reported the first attempt to establish the modern framework of Convolutional Neural Networks (CNN), and later improved it, as published in [4]. Some of these networks optimized and updated weights based on back-propagation [5] algorithms. Currently, deep learning tools like CNNs [6] have been explicitly used to train and validate Computer-Aided Diagnosis CADx tools and predict outputs accordingly. These networks extract and analyze patterns and are tested or validated with external test data. Different types of pre-trained DNNs, such as ResNet [7], DenseNet [8], AlexNet [9], VGG [10], MoblieNet [11], and Xception [12] have won international competitions. Thus, DNNs efficiently reduce cost-function, thereby are highly accurate in prediction in a variety of problems: detection and/or classification.

In due course of time, ensembling of DNNs have proved to optimize accuracy of DNN architectures, thereby making networks more diverse and accurate [13]–[15]. Typical ensemble learners tune the updated weights and follow a majority rule that increases the overall detection and/or classification accuracy [16]. Therefore, the reliability of the CADx systems increase manifold.

Different studies conducted on gastrointestinal cancer images further reveal the importance of screening procedures and use of CADx tools [17], [18]. In [19], the authors present a review of the latest research advances involving detection and classification of gastrointestinal cancer tissues using deep learning tools. Zhang et. al [20] utilized a CNN-based network to classify ulcers, erosion, and gastric polyps. They used fire modules extracted from SqueezeNet framework [21] and an iterative reinforcement process. The mentioned optimization mechanisms followed on $1,331$ images allowed them to achieve a classification accuracy of 88.9%. Lee et al. [22] used a transfer-learning approach and DNNs. They validated their performance using their own dataset of 787 images distributed across normal, cancer, and ulcer cases. They achieved an overall accuracy of 90% across three classes. Billah et al. [23] made use of the color wavelet features together using CNN features, using SVM classifier to extract features. They used video processed endoscopy data to train and validate their method and achieved an overall accuracy of 98.5%. Ribeiro et al. [24] improved performance of CNN architectures for colonic polyp classification, with a specificity of 74.19% and a sensitivity score of 95.16%. Jia et al. [25] used DNNs on $10,000$ wireless capsule endoscopy images for an automated bleeding detection strategy with an F1-score of 0.9955. Park et al. [26] showed greater values when considering invariance to image quality factors, to that of an Eigen model. Byrne et al. [27] used a deeply convoluted neural network (DCNN) on 125 video clips trained their model frame-wise, each containing diminutive polyps in the colorectum, and achieve an accuracy of 94%.

The above observations, and from works from Kather et al. [28] directly motivate towards the problem statement of this paper. We use the dataset (*ref.* Section IV-A) from the same author, Kather et al. [29]. In this paper we propose a stacked generalization method to automatically screen gastrointestinal malignancy and therefore, speed up the screening procedures.

The rest of the paper is segmented as follows. Section II briefly comprehends the different specifications to create the modified individual CNN and DNNs, and training procedures of the individual models. Section III explains the stacked generalization design created for the second phase of training. We stack the individual models and retrain an ensemble network to deconstruct our performance on an external test set of $19,230$ images. Section IV describes the dataset, pre-processing techniques, and results obtained after the models elaborated before are trained and tested with the external test set of the pre-processed data. The section also contains a comparison outline which briefly compares our proposed method with benchmark works conducted, utilizing detection and classification of gastrointestinal cancer tissues. Section V concludes our paper.

II. DNN: Base learners

The images are trained using two DNN architectures, namely VGG16 and DenseNet201. A third custom designed CNN architecture is also used to initially assess these images and to provide preliminary results. In what follows, we discuss these architectures.

A. VGG16

The architecture VGG16 [10] preferred over VGG19 for the same reason that the latter contains more number of layers with similar performance with added trainable parameters, thereby increasing model complexity. The modified VGG16 architecture for our experimentation is created by stacking first and second blocks, containing two convolution layers with kernel size 3×3, followed by a maxpool layer of stride and pooling size of 2×2, for each of the blocks. Furthermore, the next three blocks are created with three convolution layers, each of kernel size 3×3, followed by maxpool layer of size and stride 2×2. Furthermore, we have added a global average pooling layer, a batch normalization layer, and two dropout layers, each of value 0.5, followed by a dense layer. Dropout layers are a simple measure to prevent overfitting, used to promote sparse representation from any given layer. Thus they contribute active regularization by freezing neurons and narrowing the training network. Each of the convolutional layers are combined with a ReLU function, to diminish negative values to be passed through the hidden neurons. Adadelta optimizer [30] was used to update the weights. The governing equations are expressed as follows:

$$B[h^2]_t = \gamma B[h^2]_{t-1} + (1-\gamma)h_t^2, \tag{1}$$

where $B[h^2]_t$ refers to running average at time interval t, depends on γ, known as the momentum term. We select the momentum value as around 0.9. Therefore, the parameter update vector of Adagrad takes place as:

$$\Delta\theta_t = -\frac{\eta}{\sqrt{H_t + \epsilon}} \odot h_t. \tag{2}$$

Following squared gradients, the squared parameters used for defining exponentially decaying average can be expressed as,

$$B[\Delta\theta^2]_t = \gamma B[\Delta\theta^2]_{t-1} + (1-\gamma)\Delta\theta_t^2. \tag{3}$$

Therefore, the Root Mean Square (RMS) for the parameter updates is $RMS[\Delta\theta]_t = \sqrt{B[\Delta\theta^2]_t + \epsilon}$. Replacing the learning rate η with $RMS[\Delta\theta]_t$, the final weight is updated as,

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[h]_t} h_t \text{ and} \tag{4}$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t. \tag{5}$$

Using Adadelta, the default learning rate is not required to be initialized, as it directly gets eliminated from the update rule. The batch size was set to 64, the model was trained with 30 epochs, and the validation accuracy was used to monitor the training instances.

B. DenseNet201

The DenseNet201 [8] is a complex DNN architecture, comprising of 201 layers. The motivation behind using this architecture is that, the high growth rate ($k = 32$) with very narrow layers facilitate memory retention through concatenation layers, each at the end of dense blocks. The dense blocks effectively utilize feature reusability procedure, using feature mapping technique. This is facilitated by using dense blocks. Additionally, the vanishing gradient problem in CNN are solved using dense blocks. The input layer is followed by a convolutional layer, a maxpool layer, then a dense block. The sequence gets repeated, with each convolutional block following a batch normalization layer, each convolutional layer activated by ReLU activation function. We modify this existing DenseNet201 model by adding an extra convolutional layer, a global average pooling layer, and a dropout layer of value 0.5. The batch size was kept to 64, loss function was set to Binary Cross Entropy (BCE). The governing equation for BCE is

$$\text{BCE} = -\frac{1}{2}\sum_{a=1}^{2} x_a \cdot \log \hat{x}_a + (1-x_a) \cdot \log(1-\hat{x}_a), \tag{6}$$

where $\hat{x}_a$ is the a-th scalar value, corresponding to output neurons of the model, and x_a is the target output. Adagrad optimizer was used to update weights. The model was trained for 30 epochs and the validation accuracy was used to assess training curve.

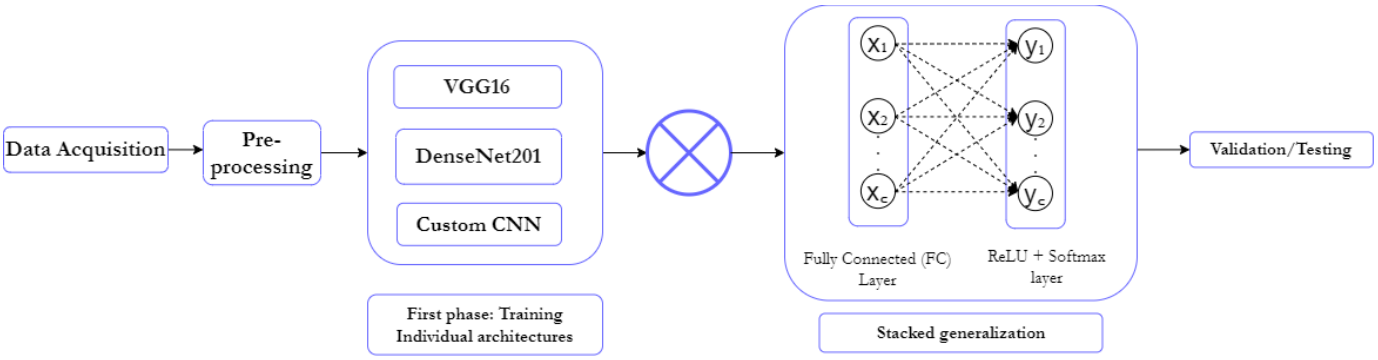

Fig. 1: Mechanism: stacked generalization

C. Convolutional Neural Network (CNN)

The CNN architecture follows a depth-wise convolution function approach. This is created using 3 kernels, each of size $5 \times 5 \times 1$. Each kernel iterates 1 channel of the image, therefore we obtain scalar products of every 25 pixels, with a $8 \times 8 \times 1$ image. Stacking three kernels allows the image to restore 3 channels. So by tuning the depth multiplier argument, we allow more parameters in a single iteration, thus the network trains with multiple numbers of features in a single epoch. We used five convolutional blocks, each containing two separable convolutional 2D layers with a kernel size of 3×3, a batch normalization layer, a maximum pooling layer of pool size 2×2 and a dropout layer of rate 0.2. Each layer of the block is activated by the ReLU activation function so that the negative feedback does not traverse to the next layer. The dropout rate is kept the same for every block, selected by tuning the hyper-parameters and implementing an optimized network. The blocks continue to a fully connected layer, and the output layer contains a dense layer, activated by the softmax activation function.

III. Stacked generalization

Stacked generalization [31] is the implementation of ensemble learning, where the individual base learner architectures are a diverse set of algorithms. The combined model is also referred to as a super learner, or an ensemble DNN architecture. The convex combination of individual base predictors can be expressed as,

$$f_{sl}(y) = \alpha_1 f_1(y) + \alpha_2 f_2(y) + \alpha_3 f_3(y), \qquad (7)$$

where $\alpha \geq 1$ and $\sum_{i=1}^{3} = 1$ for the three base learners used for this particular study, namely VGG16, DenseNet201, and custom designed CNN. The updated weights are used an input to the ensemble framework and the model is then trained for 40 epochs. The stacked generalization architecture consists of an extended fully connected (FC) layer and a softmax layer, activated by ReLU activation function. Adagrad optimizer is used to optimize this architecture.

The training procedure is simulated in Fig. 1. The results achieved by this architecture is compared with the individual DNN base learners, which is described further (*ref.* Section IV-B).

IV. Experiments

The tissue images are trained in two procedures. Firstly, we train the individual models using VGG16, DenseNet201, and a custom designed CNN architecture, as described in Section II. The results obtained from these architectures are further analyzed and fed to the stacked generalization architecture, as described in Section III. Therefore, further training of the images on the ensemble architecture determines performance of the model. For the training procedure, we have utilized an NVIDIA ® P100 graphics processing unit (GPU) system. With accession to 16 GB memory, 1.32 GHz memory clock with performance power of 9.3 TFLOPS, the specifications proved to be ideal for conducting experiments for training approximately $192,315$ images in a considerable amount of computation time, therefore providing more scope towards advanced experimentation.

A. Dataset and pre-processing

To train and validate results, we utilized images from [29], which are publicly available. The dataset contains unique patches from the 'The Cancer Genome Atlas' (TCGA) cohort. The images are derived from Formalin-Fixed Paraffin-Embedded (FFPE) diagnostic slides. FFPEs are generated by fixing a specimen in formaldehyde and then embedding it in a paraffin wax block for cutting.

The dataset contains a total of $192,315$ images, with $75,039$ images in MSI category, and $117,276$ images in MSS/MSIMUT category. The MSS category is termed as "MSIMUT" (microsatellite instable or highly mutated) in the original dataset. For better understanding, the sample images are shown in Fig. 2.

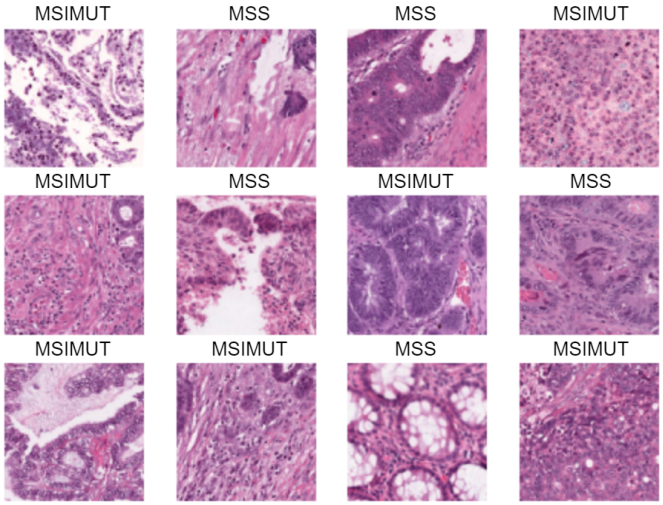

Fig. 2: Sample data for MSI and MSIMUT category

TABLE I: Number of images after data pre-processing

	Training Set	Validation Set	Testing Set
No. of images	153, 849	19, 230	19, 230

The images were initially 224×224 pixels dimensions, with a resolution of 0.5μ/pixel. The dimensions were reduced to 128×128. Color normalization of the images were done individually, following Macenko's method [32]. The histopathological images used were high quality assessed, therefore no further pre-processing of the images were required. The images were divided into training (80%), validation (10%), and an external test set (10%). The total number of images are shown in Table. I.

B. Results analysis

The individual DNN models, as described in Section II are used to train the pre-processed data to acquire initial results. The training of the individual DNNs is comprehended with 30 epochs each. Table III shows the performance of base learners and the stacked generalization architecture. The base learners achieve a benchmark accuracy of 89.71% for DenseNet201 architecture, the lowest for the custom CNN network with an accuracy of 81.91%. The preliminary results denote the inefficiency of using basic convolutional blocks for training and validating the CNN with the histological dataset. However, the dense blocks in DenseNet201 suggest a high amount of feature reusability, thus the best results.

The stacked generalization architecture created using individual base learners show remarkable increase in terms of model performance. A confusion matrix created to gauge performance of the stacked generalization architecture (*ref.* Fig. 3) shows that out of $19,230$ images in the

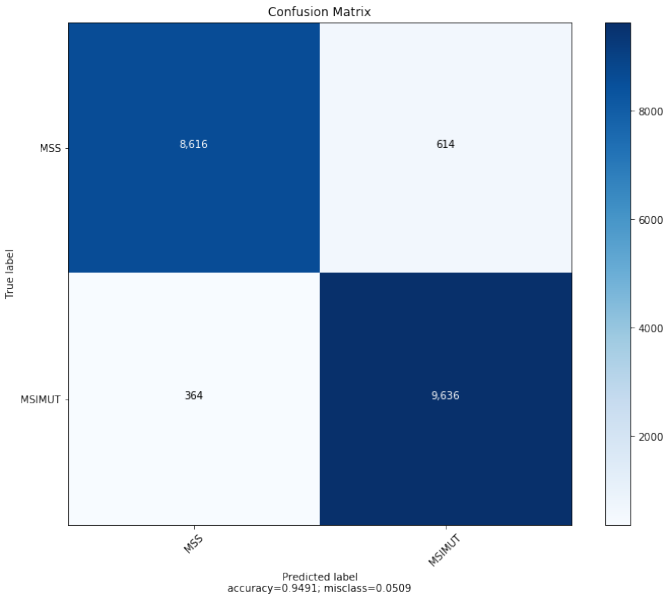

Fig. 3: Confusion matrix evaluated on the external test set for stacked generalization model

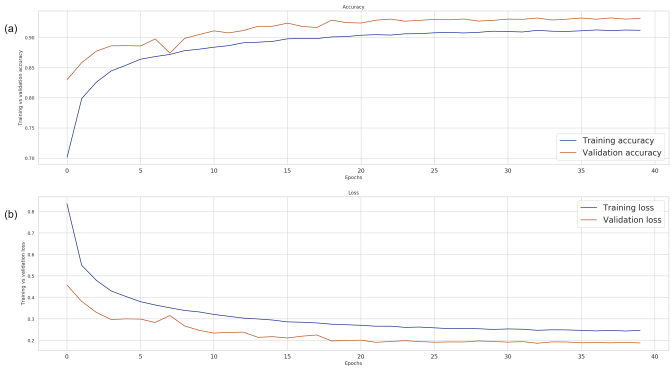

Fig. 4: Performance of the stacked generalization model across 40 epochs, with respect to training versus validation accuracy, and training versus validation loss

test set, the model correctly classifies $18,251$ images, with the total balanced accuracy evaluated at around 94.91%. The sensitivity and precision of the model is calculated at around 95.95% and 93.35%, respectively. The model achieves an AUC score of 0.9821, which has increased when compared to the individual DNNs and CNN. The training mechanism of the stacked generalization architecture has been monitored and simulated, as shown in Fig. 4.

C. Comparison with state-of-the-art works

The stacked generalization model is compared to the mechanism of similar models on similar datasets using histopathological images. Kather et al. [28], the contributor of the dataset used in our article had analyzed the same dataset. They used CNNs and a ResNet18 framework to classify between MSI and MSS. They achieved

TABLE II: Performance evaluation of selected architectures

Metrics Used	VGG16	DenseNet201	Custom CNN	Stacked Generalization
F1-Score	0.8505	0.8851	0.8058	0.9463
Precision (PPV)	0.8251	0.8251	0.7818	0.9335
Specificity (SPE)	0.8470	0.8565	0.8091	0.9401
Sensitivity (SEN)	0.8774	0.9544	0.8313	0.9595
Cohen's kappa (κ) [33]	0.7200	0.7930	0.6370	0.8980
Balanced accuracy (BAC)	0.8607	0.8971	0.8191	0.9491
Area under curve (AUC)	0.9512	0.9553	0.9246	0.9821
Matthews correlation coefficient (MCC)	0.7216	0.7998	0.6379	0.8983

TABLE III: Comparison of the proposed architecture with similar works

Methods (author)	Images	Sensitivity	AUC
CNN, Kather et al. [28]	100,570	–	81%
Bootstrapping, Echle et al. [34]	8,836	95%	92%
Stacked Generalization (Proposed)	192,315	95.95%	98.21%

an AUC score of 0.81 for patient-wise classification using TCGA-STAD test cohorts, along with the other two test datasets with a 95% confidence interval for identifying MSIs (score: 0.61). The analysis, however, can be put forward by using stacked generalization, which is the motive of our work. Echle et al. [34] conducted an assessment to detect MSI using 8,836 colorectal tumor slides. They used a 10-fold bootstrapping cross-validation approach that achieved an overall AUC of 0.63 for the cross-validation cohort, and dMMR/MSI classification procedure achieving an AUC of 0.92. Our work summarizes these findings with more precision and with a larger testing database.

V. Conclusion

We propose a stacked generalization architecture to classify between two different types of gastrointestinal cancer tissues, Microsatellite Instable (MSI) and Microsatellite Stable (MSS). This serves as a proof-of-principle that the proposed method must be investigated and validated further towards automated screening procedures with the aim of utilizing these tools as a complementary detection procedure for clinical screening and standard-of-care treatment.

References

[1] http://globocan.iarc.fr.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in neural information processing systems*, vol. 2, pp. 396–404, 1989.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[6] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[7] 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*, pp. 770–778, 2016.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

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

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[12] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[13] D. Paul, A. Tewari, S. Ghosh, and K. Santosh, "Octx: Ensembled deep learning model to detect retinal disorders," in *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 526–531, IEEE, 2020.

[14] S. Ghosh, A. Bandyopadhyay, S. Sahay, R. Ghosh, I. Kundu, and K. Santosh, "Colorectal histology tumor detection using ensemble deep neural network," *Engineering Applications of Artificial Intelligence*, vol. 100, p. 104202, 2021.

[15] S. Ghosh, M. Majumder, and A. Kudeshia, "Leukox: Leukocyte classification using least entropy combiner (lec) for ensemble learning," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2021.

[16] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15, Springer, 2000.

[17] B. Li and M. Q.-H. Meng, "Computer-aided detection of bleeding regions for capsule endoscopy images," *IEEE Transactions on biomedical engineering*, vol. 56, no. 4, pp. 1032–1039, 2009.

[18] N. Tamai, Y. Saito, T. Sakamoto, T. Nakajima, T. Matsuda, K. Sumiyama, H. Tajiri, R. Koyama, and S. Kido, "Effectiveness of computer-aided diagnosis of colorectal lesions using novel software for magnifying narrow-band imaging: a pilot study," *Endoscopy international open*, vol. 5, no. 8, p. E690, 2017.

[19] W. Du, N. Rao, D. Liu, H. Jiang, C. Luo, Z. Li, T. Gan, and B. Zeng, "Review on the applications of deep learning in the analysis of gastrointestinal endoscopy images," *IEEE Access*, vol. 7, pp. 142053–142069, 2019.

[20] X. Zhang, W. Hu, F. Chen, J. Liu, Y. Yang, L. Wang, H. Duan, and J. Si, "Gastric precancerous diseases classification using cnn with a concise model," *PloS one*, vol. 12, no. 9, p. e0185508, 2017.
[21] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.
[22] J. H. Lee, Y. J. Kim, Y. W. Kim, S. Park, Y.-i. Choi, Y. J. Kim, D. K. Park, K. G. Kim, and J.-W. Chung, "Spotting malignancies from gastric endoscopic images using deep learning," *Surgical endoscopy*, vol. 33, no. 11, pp. 3790–3797, 2019.
[23] M. Billah, S. Waheed, and M. M. Rahman, "An automatic gastrointestinal polyp detection system in video endoscopy using fusion of color wavelet and convolutional neural network features," *International journal of biomedical imaging*, vol. 2017, 2017.
[24] E. Ribeiro, A. Uhl, and M. Häfner, "Colonic polyp classification with convolutional neural networks," in *2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 253–258, IEEE, 2016.
[25] X. Jia and M. Q.-H. Meng, "A deep convolutional neural network for bleeding detection in wireless capsule endoscopy images," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 639–642, IEEE, 2016.
[26] S. Y. Park and D. Sargent, "Colonoscopic polyp detection using convolutional neural networks," in *Medical Imaging 2016: Computer-Aided Diagnosis*, vol. 9785, p. 978528, International Society for Optics and Photonics, 2016.
[27] M. F. Byrne, N. Chapados, F. Soudan, C. Oertel, M. L. Pérez, R. Kelly, N. Iqbal, F. Chandelier, and D. K. Rex, "Real-time differentiation of adenomatous and hyperplastic diminutive colorectal polyps during analysis of unaltered videos of standard colonoscopy using a deep learning model," *Gut*, vol. 68, no. 1, pp. 94–100, 2019.
[28] J. N. Kather, A. T. Pearson, N. Halama, D. Jäger, J. Krause, S. H. Loosen, A. Marx, P. Boor, F. Tacke, U. P. Neumann, *et al.*, "Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer," *Nature medicine*, vol. 25, no. 7, pp. 1054–1056, 2019.
[29] J. N. Kather, "Histological images for msi vs. mss classification in gastrointestinal cancer, ffpe samples," *ZENODO*, 2019.
[30] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.
[31] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
[32] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, and N. E. Thomas, "A method for normalizing histology slides for quantitative analysis," in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1107–1110, IEEE, 2009.
[33] J. Cohen, "Introduces kappa as a way of calculating inter rater agreement between two raters," *Educational and Psychological Measurement*, 1960.
[34] A. Echle, H. I. Grabsch, P. Quirke, P. A. van den Brandt, N. P. West, G. G. Hutchins, L. R. Heij, X. Tan, S. D. Richman, J. Krause, *et al.*, "Clinical-grade detection of microsatellite instability in colorectal tumors by deep learning," *Gastroenterology*, vol. 159, no. 4, pp. 1406–1416, 2020.