FUSECAPS: INVESTIGATING FEATURE FUSION BASED FRAMEWORK FOR CAPSULE ENDOSCOPY IMAGE CLASSIFICATION

Bidisha Chakraborty
Government College of Engineering and Leather Technology
Kolkata, India
bidisha23102000@gmail.com

Shree Mitra
Indian Institute of Information Technology Guwahati
Guwahati, India
shree.mitra23m@iiitg.ac.in

ABSTRACT

In order to improve model accuracy, generalization, and class imbalance issues, this work offers a strong methodology for classifying endoscopic images. We suggest a hybrid feature extraction method that combines convolutional neural networks (CNNs), multi-layer perceptrons (MLPs), and radiomics. Rich, multi-scale feature extraction is made possible by this combination, which captures both deep and handmade representations. These features are then used by a classification head to classify diseases, producing a model with higher generalization and accuracy. In this framework we have achieved a validation accuracy of 76.2% in the capsule endoscopy video frame classification task.

***K*eywords** Radiomics · Projection Head · Feature Fusion

1 Introduction

Endoscopy is a procedure that is used to have a close look at the organs. This is done either to detect the diseases or observe the cellular patterns. Early detection of diseases is important as it helps to reduce the mortality rate and also improve the development of medicines. Here the subset of Machine Learning that is Deep Learning comes into play. Deep Learning has been implemented widely in the medical field to detect gastrointestinal and liver-related diseases. As a result, many models have been developed to classify capsule endoscopy images. Some of them involve the use of Convolution Neural Networks or Transfer Learning. But to ensure that the classification of the images is to the point, we have introduced the combination of Radiomics and Convolution Neural networks to enrich the feature dataset and lastly use that feature dataset to perform classification.

2 Methods

In general a wide variety of techniques are used to determine the category of diseases. Some of them are Supervised Learning, Transformation Learning, Convolutional Neural Networks and so on. The above defined techniques have been also combined to enrich the feature dataset.

In this we have proposed a methodology that will not only increase the accuracy of the model but will also improve the generalisation of the model and will also handle class imbalance. For extraction of features from the dataset, we have implemented a combination of Radiomics followed by Multi Layer Perceptron and Convolutional Neural Networks. The classification head is used for classification of diseases.

2.1 Radiomics Feature Extraction

Radiomics[1] is a process in which we try to extract quantitative or handcrafted features from the images. This method is particularly useful in the medical field as medical images has wide variety of handcrafted features including shape, texture, density etc and this technique largely helps to derive those features using data characterisation algorithms. The algorithms also makes use of advanced mathematical concepts like Laplacian, Gaussian, Gradient formulas to apply the filters. Radiomics also makes use of advanced statistical methods like Gray Level co-occurrence matrix, Gray Level Run Length matrix, Gray Level Zone Size Matrix etc. But we Radiomics is particularly useful in segmentation where the Region of Interest is masked and the foreground is present but unmasked.

In our case since there was no foreground present, so first we considered the centre of image and converted the pixel values in the central region to 1 while the rest to 0. We applied the Radiomics to extract features from the central part of the image. The results are stored in NPY file and is ultimately concatenated in a Comma Separated file that has features like mean, version etc. Now for the second case we are extracting features from the sides and excluding the central portion of the image. So basically we are creating our own foregrounds as per our needs, masking the Region of Interest and accordingly extracting the features.

After concatenating the results, we combine the two csv files side by side and dropped the unnecessary columns. After that it is passed through the Multi Layer Perceptron.

2.2 Multi Layer Perceptron

Multi-Layer Perceptrons[2] are Artificial Neural Networks that comprise an input layer, one or more hidden layers, and output layers. We used a multi-layer perceptron to preprocess the handcrafted features further.

The MLP maps an input radiomics vector $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$ to a compact embedding $\mathbf{z} \in \mathbb{R}^{d_{\text{embed}}}$ by first applying a linear transformation,

$$\mathbf{h}_1 = \text{ReLU[3]}(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1),$$

where $\mathbf{W}_1 \in \mathbb{R}^{1024 \times d_{\text{in}}}$. Dropout[4] regularization,

$$\mathbf{h}_2 = \text{Dropout[4]}(\mathbf{h}_1, p = 0.5),$$

mitigates overfitting by randomizing feature selection. Another linear transformation,

$$\mathbf{z} = \text{Dropout[4]}(\mathbf{W}_2\mathbf{h}_2 + \mathbf{b}_2, p = 0.5),$$

with $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{embed}} \times 1024}$, completes the embedding. This process reduces dimensionality and enhances the representation of complex, non-linear feature relationships, producing a robust embedding that can be effectively fused with CNN features for multi-modal classification of capsule endoscopy images.

2.3 Convolutional Neural Networks

For image classification tasks or processing tasks we mainly use Convolutional Neural Networks or CNNs[5]. The backbone CNN model of the proposed framework is the DenseNet[6] CNN architecture, which we used to extract features from the dataset's complicated endoscopic pictures. DenseNet's ability to retrieve feature maps from all previous levels is the primary driving force behind its use. Dense blocks, in which the output of every layer inside a block is concatenated with all of the inputs of its preceding layers, are used to accomplish this. In particular,

$$X_l = H_l([X_0; X_1; \ldots; X_{l-1}]) \in \mathbb{R}^{h_b \times w_b \times d_b} \tag{1}$$

where $X_0, X_1, \ldots, X_l \in \mathbb{R}^{h_b \times w_b \times d_b}$ represent the output feature maps from the 0-th to the l-th layers, ";" denotes the concatenation operation, d_b is the feature dimension of the dense block b, and the convolution function $H_l(\cdot)$ consists of a 3 × 3 convolution layer, a ReLU activation function, and batch normalization.

2.4 Projection Head

In the context of image classification, let $x \in \mathcal{X} \subset \mathbb{R}^{H \times W \times C}$ be the input image and $f_\theta(x) = z \in \mathcal{F} \subset \mathbb{R}^{h \times w \times d}$ represent the high-dimensional feature vector obtained from a convolutional neural network (CNN). The dimensionality of z is often large, leading to potential redundancies that can impede the learning of discriminative features. To mitigate this, a projection head $g_\phi(z) = y \in \mathbb{R}^k$ is employed, where $k \ll h \times w \times d$.

The projection head[7] incorporates an Adaptive Average Pooling operation, which reduces spatial dimensions while preserving essential information:

$$z' = \text{GAP}(z) \in \mathbb{R}^d.$$

Following this, a fully connected layer with Batch Normalization and ReLU activation transforms the pooled representation:

$$y = \sigma(\mathbf{W}z' + b) \in \mathbb{R}^k,$$

where σ denotes the activation function. This transformation enhances feature disentanglement and regularizes the representation space, which is crucial for effective generalization.

By constraining the representation to a lower-dimensional space, the projection head not only reduces the risk of overfitting but also helps the classifier focus on relevant features, minimizing the empirical loss:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \ell(\hat{y}_i, y_i).$$

Thus, the introduction of a projection head serves to amplify the discriminative power of the model while ensuring a compact, interpretable representation suitable for classification tasks.

2.5 Integration of CNN Extracted Features and Numerical Radiomics Features

Let $\mathbf{X}_{img}$ be the image input, and $\mathbf{X}_{num}$ be the numerical features derived from radiomics. The model outputs are represented as follows:

1. DenseNet Feature Extraction:

$$\mathbf{F}_{cnn} = \text{DenseNet}(\mathbf{X}_{img})$$

2. MLP Feature Extraction:

$$\mathbf{F}_{mlp} = \text{MLP}(\mathbf{X}_{num})$$

3. Projection Head for Dimensionality Reduction:

$$\mathbf{F}_{proj} = \text{ProjectionHead}(\mathbf{F}_{cnn})$$

4. Feature Concatenation: The combined feature vector is given by:

$$\mathbf{F}_{combined} = \mathbf{F}_{proj} \oplus \mathbf{F}_{mlp}$$

where $\oplus$ denotes the concatenation operation.

5. Classification Output: The final classification output is obtained by passing the combined features through the classifier:

$$\mathbf{y} = \text{ClassificationHead}(\mathbf{F}_{combined})$$

The combined features capture both visual ($\mathbf{F}_{cnn}$) and numerical ($\mathbf{F}_{mlp}$) information, enhancing the representation power:

$$\mathcal{L} = \text{Loss}(\mathbf{y}, \mathbf{y}_{true})$$

By integrating features, the model can learn a more complex decision boundary $f(\mathbf{F}_{combined})$ that separates different classes more effectively. The integration leads to improved generalization on unseen data, represented mathematically as:

$$\mathbb{E}[\mathcal{L}_{test}] < \mathbb{E}[\mathcal{L}_{train}]$$

3 Implementation details

In our implementation, DenseNet is divided into three blocks, each of which has sixteen bottleneck levels in the encoder. When $\theta = 0.5$ is applied, this transition layer shrinks the channel and spatial sizes of the feature maps between each of the DenseNet's two blocks. The growth rate is $k = 24$, and the dropout rate is $p = 0.2$. We have employed an MLP[2] with two linear layers of out-feature shapes 1024 and embedding size, respectively, as a non-linear projection head. This projection head provides an output feature of shape (N, embedding size), where N is the batch size, and is combined with the densenet feature extractor.

We use Adam[8] as our optimizer, with a learning rate of $1e-3$ and a weight decay of $1e-6$. The batch size is set at 64.

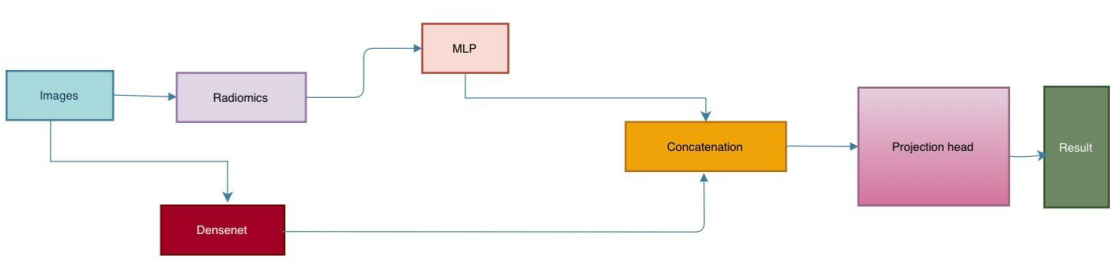

Figure 1: Block diagram of the developed pipeline.

4 Results

The graphs in **Figure 2** depict the training and validation performance over 100 epochs for a model.

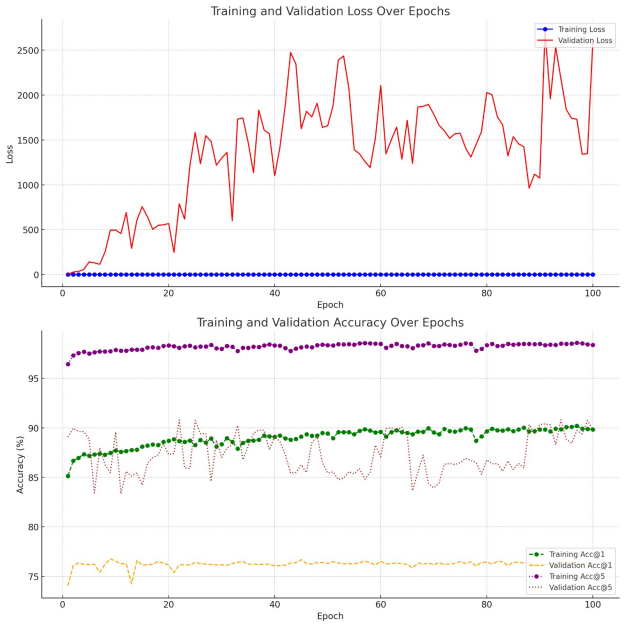

Figure 2: Training and Validation Loss and Accuracy Over Epochs

Overall, the results still indicates some chances of overfitting, where the model tries to generalize the images despite good training performance.

4.1 Achieved results on the validation dataset

In **Figure 3**, the ROC curves and AUC scores for each class demonstrate how our model handles class imbalance. The AUC metric provides a threshold-independent assessment, which is less sensitive to class distribution than metrics like accuracy. Notably, despite class imbalance, certain minority classes (e.g., Class 3 and Class 8, with AUCs of 0.7720 and 0.8117, respectively) show high discriminatory power, indicating effective separation from other classes. Lower AUCs, such as for Class 7 (AUC = 0.3730), highlight areas for improvement. Overall, the per-class AUCs suggest that our approach maintains a balanced performance across both majority and minority classes, thereby partially mitigating the effect of class imbalance.

Here we have calculated the weighted average of the metrics. As we can see our model performs well when compared with other models but we can expect our model to perform better which we can improve in the near future.

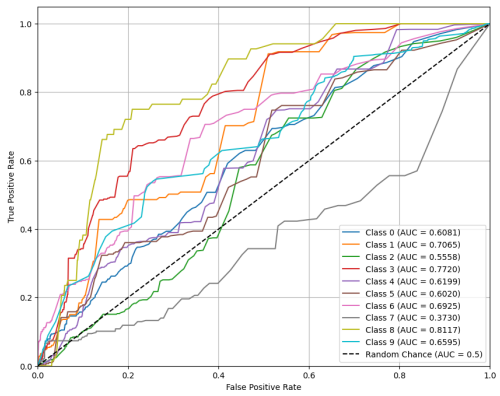

Figure 3: AUC-ROC Curve

Table 1: Performance Comparison of Different Methods

Method	Avg. ACC	Avg. Sensitivity	Avg. F1-score	Avg. Precision
Custom CNN (baseline)	0.460	0.097	0.093	0.100
ResNet50 (baseline)	0.760	0.320	0.373	0.602
SVM (baseline)	0.818	0.408	0.487	0.833
VGG16 (baseline)	0.568	0.543	0.484	0.525
Proposed Method	**0.762**	**0.762**	**0.659**	**0.611**

5 Discussion

As we can see the model is a bit over fitted due to huge class imbalance. In future we can improve the generalization of the model by introducing some more images of the minority classes. We can use GANs for synthetic generation of minority class images.

6 Conclusion

We have proposed a novel methodology in the field of capsule endoscopy video frame images for classification of diseases. We have achieved validation accuracy of 76.3% but this can be further improved using more advanced techniques like Self-Supervised Learning, Synthetic Image Generation using GANs etc.

7 Acknowledgments

As participants in the Capsule Vision 2024 Challenge, we fully comply with the competition's rules as outlined in [9]. Our AI model development is based exclusively on the datasets provided in the official release in [10].

References

[1] Philippe Lambin, Emmanuel Rios-Velazquez, Ralph Leijenaar, Sara Carvalho, Ruud G.P.M. van Stiphout, Patrick Granton, Catharina M.L. Zegers, Robert Gillies, Ronald Boellard, André Dekker, and Hugo J.W.L. Aerts. Radiomics: Extracting more information from medical images using advanced feature analysis. *European Journal of Cancer*, 48(4):441–446, 2012.

[2] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[3] Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018.

[4] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[6] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[9] Palak Handa, Amirreza Mahbod, Florian Schwarzhans, Ramona Woitek, Nidhi Goel, Deepti Chhabra, Shreshtha Jha, Manas Dhir, Deepak Gunjan, Jagadeesh Kakarla, et al. Capsule vision 2024 challenge: Multi-class abnormality classification for video capsule endoscopy. *arXiv preprint arXiv:2408.04940*, 2024.

[10] Palak Handa, Amirreza Mahbod, Florian Schwarzhans, Ramona Woitek, Nidhi Goel, Deepti Chhabra, Shreshtha Jha, Manas Dhir, Deepak Gunjan, Jagadeesh Kakarla, and Balasubramanian Raman. Training and Validation Dataset of Capsule Vision 2024 Challenge. *Fishare*, 7 2024.