RMIT UNIVERSITY

Classify Images of Colon Cancer

Team KV: Vu Anh Tu – s3685121

Kim Sung Jin – s3695340

Course Coordinator: Dr. Duy Dang-Pham

Course Code: COSC2753

Submission Date: 24/05/2021

1. Introduction

With the abundance of data available on the internet, neural networks and deep learning have surpassed traditional algorithms in solving multiple machine learning problems. More recently, deep learning has shown prospective results for image classification tasks in medical fields. Specifically, cell detection and classification tasks which require domain expertise in biology can now be solved using deep learning, paving the way for deeper understanding of cancer [1]. In this report, we present our experiment design and various approaches used to investigate the cancer detection and cell classification tasks.

2. Data Exploration

	Total Instances	Fibroblast	Inflammatory	Epithelial	Miscellaneous	Normal	Cancerous
Main dataset							
Train	6332	1208	1627	2610	887	3722	2610
Validation	1584	302	407	653	222	931	653
Test	1980	378	509	816	277	1164	816
Extra dataset							
Train	8307	N/A	N/A	N/A	N/A	5915	2392
Validation	2077	N/A	N/A	N/A	N/A	1479	598

Table 1: Data splitting information

The "CRCHistoPhenotypes" dataset from the original paper [1] contains 100 RGB images of size 500 x 500 pixels. For this problem, each image has been split into multiple 27 x 27 images, each of which contains the cell nuclei in the middle. We are given two datasets: main and extra. The main dataset contains 9896 images taken from the first 60 patients. Each image is labelled with its cell type and whether the cell is cancerous or not (isCancerous column). The extra dataset contains 10384 images taken from the next 39 patients. Each image is only labelled for the “isCancerous” column. The cell types are grouped into four categories: Epithelial, Inflammatory, Fibroblast and Miscellaneous (which contains all remaining cell types).

In the main dataset, we observe that epithelial takes up slightly more portion compared to other cell types. On the other hand, the proportion between cancerous and normal cells is relatively balanced.

However, we detect that all of the cancerous cells in the main dataset falls into the epithelial category. In addition, the extra dataset also has a relatively imbalanced proportion between cancerous and normal cells. These could pose potential challenges for our cancer detection task.

3. Experiment Design

3.1. Data Preprocessing

We need to convert raw images to numpy arrays for our model to understand and process. We will use three different array formats for different types of models as described below.

Dimensions	Description
27 x 27 x 3	This is the default format when converted from RGB images. Each pixel is represented by three numbers measuring its degree of red, green and blue. This input is fed directly into our convolutional neural network (CNN) models.
27 x 27 x 1	Some of our models were evaluated with gray scale input. Each pixel is represented by a number showing how light/dark it is. This format can reduce computation time but may potentially decrease the model’s performance due to loss of information.
1 x 729	The traditional algorithms do not understand three-dimensional array input. Therefore, we flatten the gray scale array above to feed into these models. We do not flatten the RGB due to computational time and possible overfitting.

Table 2: Input types

3.2. Data Splitting

As described in Table 1, the main dataset is split into train, validation and test sets. First, the test set is reserved from 20% of the main dataset. Next, the remaining data is split into the train and validation sets with an 80/20 ratio, respectively. The train and validation sets are used to build and fine-tune the models, while the test set is used to evaluate how the models perform on an unseen dataset in real life. For consistency, all models of the same task will be benchmarked on the same test set. For cancer detection task, the splits are stratified based on "isCancerous" column to ensure all classes are present in each set. Similarly, the splits for cell type classification are stratified based on the "cellType" column.

The extra dataset is split into the train and validation sets with the 80/20 ratio. The split is stratified based on "isCancerous" column because that is the only target variable.

To ensure the reproducibility of the project, we set random seeds for Python, Numpy and Tensorflow packages to value 123.

3.3. Evaluation Metrics

As cancer detection and cell type classification are both classification tasks, we evaluate our models based on accuracy, precision, recall and f1 score on the test set. We also take training time into consideration to ensure the model is scalable on larger datasets.

3.4. Approaches and Analysis

Approach	Description	Advantages	Disadvantages
Logistic Regression [2]	A classification algorithm which learns the linear relationship of data and uses the sigmoid function to classify data into discrete categories.	• Easy to implement and interpret. • Can be extended to multiple classes. • Efficient to train.	• Easy to overfit with large number of features. • Cannot solve non-linear problems. • Cannot obtain complex relationships of features.
K-Nearest Neighbors [3]	Classifies data based on the intuition that similar data points will be close to one another in space.	• Fast training time. • Easy to interpret.	• The prediction stage can be slow if the data is large. • Accuracy depends highly on data quality. • Requires lots of memory.
Random Forest [4]	An ensemble method which builds multiple decision trees from a dataset. The majority vote from the trees is used to classify a sample.	• Less prone to overfitting. • Robust to outliers. • No need to normalize data.	• Requires lots of computational power and resources. • Hard to interpret. • Cannot detect high level features of an image.

XGBoost [5]	Similar to random forest but each tree is built sequentially and learns from the previous tree.	• Less prone to overfitting. • Easy to interpret.	• Not scalable for large dataset • Sensitive to outliers.
Multilayer Perceptron [6]	A simple feedforward neural network with an input layer, hidden layer and output layer.	• Can be applied for non-linear problems. • Scalable with large input. • Quick prediction time.	• Slow computational time
CNN [7]	A deep neural network which works well with spatial data, especially images.	• Can capture spatial and temporal dependencies of an image. • Can detect high level features in an image.	• Architecture design is challenging. • Long training time. • Might not outperform traditional algorithms with limited data.

Table 3: Approaches and analysis

4. Model Evaluation

4.1. Baseline Models

Since CNN is more suitable for image classification task and performs significantly better than other models, we will only discuss our CNN models in this section. For results of other algorithms, please refer back to our notebook. For consistency, all models are trained in 50 epochs, and if there is no improvement within 10 epochs, the training will stop to avoid overfitting. The success metrics are evaluated on the test set.

	Training Time (s)	**Accuracy (%)**	**P (%)**	**R (%)**	**F1 (%)**
Cancer detection					
VGG	70	89.55	89.55	89.55	89.51
Customized model	38	89.14	89.78	89.14	89.26
SC - CNN	49	89.24	89.71	89.24	89.34
RCCNet	258	**91.26**	**91.37**	**91.26**	**91.29**
Cell classification					
VGG	60	72.88	80.30	72.88	75.02
Customized model	24	74.55	74.64	74.55	74.29
Softmax CNN	64	72.37	**81.89**	72.37	76.21
RCCNet	159	**76.87**	76.35	**76.87**	**76.26**

Table 4: Baseline models

First, we build a model based on the famous VGG architecture as a starting point [8]. The architecture consists of three blocks with two convolutional and one pooling layer for each block. Next, we attempt to build our own customized three-block model with a convolutional and a pooling layer in each block. Two models return very similar results. We have also considered other famous architectures such as ResNet [9], AlexNet [10] and Inception [11]. Nonetheless, these architectures require input of very high resolution images, which means we either have to scale up our images (which means losing information) or adjust the deep layers inside to suit our problem (which is even more problematic). To be efficient, we have researched models that use the exact dataset like ours and try to improve from those. The original work of this dataset [1] proposes the SC – CNN and Softmax CNN models to solve the cancer detection and cell classification tasks, respectively. The models, however, do not produce better results compared to the VGG and our customized model.

Another research proposes the RCCNet to solve the cell classification task [12]. This model outperforms the previous ones on both detection and classfication tasks.

The RCCNet uses Adam optimizer [13] with initial learning rate of $6\ x\ 10^{-5}$iteratively decreased by a factor of $\sqrt[2]{0.1}$ if there is no improvement of validation loss during the training. The images also go through the augmentation process that contributes to the better performance of RCCNet. Each image is distorted and zoomed at random angle with range from 0 to 0.2. The image is then flipped randomly along the x and y axis. The architecture of RCCNet is described in the figure below, where the input size has been adjusted to 27x27x3 to suit our image size, and the FC3 layer is set to 1x2 for the cancer detection task.

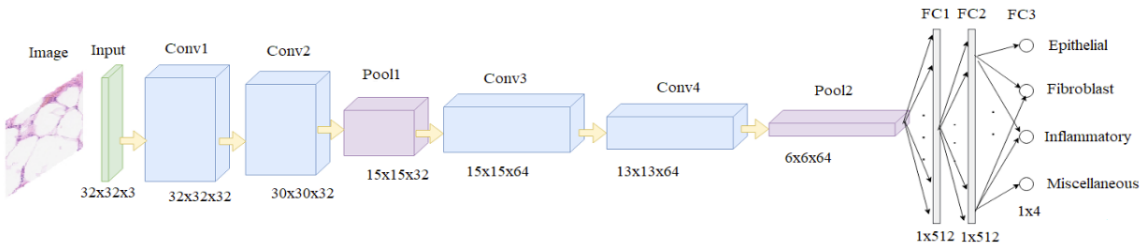

Figure 1: RCCNet architecture

4.2. Model Improvement

4.2.1. Cancer Detection

Model	Training Time(s)	Accuracy (%)	P (%)	R (%)	F1 (%)
RCCNet (main)	258	**91.26**	**91.37**	**91.26**	**92.19**
RCCNet (main + extra)	521	90.40	91.11	90.40	90.52

Table 5: RCCNet trained on main vs RCCNet trained on main and extra data

We combine the train set from the main data and the extra data to train our new model, since neural networks generally perform better with more data. We pick the RCCNet architecture because it is the best performing model so far. Surprisingly, the model trained on the combined data performs slightly worse than the original one. We suspect that the extra data contains lots of noise, which reduces the model's performance on the test set because this test set is split from the main data.

4.2.2. Cell Type Classification

Model	Training Time (s)	Accuracy (%)	P (%)	R (%)	F1 (%)
RCCNet	159	76.87	76.35	76.87	76.26
RCCNet + pseudo labelling	399	**80.40**	**80.33**	**80.40**	**80.32**

Table 6: RCCNet with and without pseudo labelling technique

To improve the cell type classification model, we apply a semi-supervised learning method called pseudo labelling [14] on both the main and extra datasets. The steps are as follows:

- Train a classification model on the main dataset (called teacher model).
- Use the teacher model to predicts cell type for the extra dataset.
- Combine the main dataset and the predicted labels on extra dataset to train a new classification model (called student model).

As expected, our student model with RCCNet architecture achieves a much better result on the test set, even though training time takes twice as long due to the extra dataset.

4.2.3. Parameter Tuning

Run	Batch size	Optimizer	Training Time (s)	Accuracy (%)	P (%)	R (%)	F1 (%)
1	64	Adam	258	91.26	91.37	91.26	**92.19**
2	64	SGD	171	89.09	89.63	89.09	89.19
3	64	Nadam	181	91.21	91.30	91.21	91.24
4	32	Adam	169	**91.52**	**91.62**	**91.52**	91.54
5	32	SGD	152	89.95	90.15	89.95	90.00
6	32	Nadam	114	91.16	91.24	91.16	91.18

Table 7: Cancer detection model tuning

Run	Batch size	Optimizer	Training Time (s)	Accuracy (%)	P (%)	R (%)	F1 (%)
1	64	Adam	399	80.40	80.33	80.40	80.32
2	64	SGD	331	76.46	78.54	76.46	77.27
3	64	Nadam	358	79.44	79.87	79.44	79.56
4	32	Adam	372	79.60	79.70	79.60	79.54
5	32	SGD	385	77.98	78.31	77.98	77.99
6	32	Nadam	420	**80.81**	**81.36**	**80.81**	**80.99**

Table 8: Cell type classification model tuning

Finally, we try to improve our models by tweaking the batch size and optimizer method. For cancer detection, we tune the original RCCNet model trained on the main dataset. For cell classification, we tune the RCCNet with the pseudo labelling technique discussed previously.

The results show that batch size does not significantly affect the model's performance. However, stochastic gradient descent (SGD) optimizer shows slightly worse results compared to the other two. In addition, neural networks have randomness in their results so a slight improvement in the numbers does not guarantee better performance. In a nutshell, our tuning process does not really show any significant boost in the performance for both models.

5. Ultimate Judgement

Based on the empirical results shown previously, we have picked out the best model for each task as follows:

- Cancer detection: RCCNet model trained on the main dataset.
- Cell classification: RCCNet model with pseudo labelling technique applied.

6. Conclusion

This report has investigated five different CNN architectures for the provided tasks, including both our customized models and models from previous research. We have picked out the best architecture from empirical analysis and successfully improved its performance on the cell type classification task using the extra dataset. For future work, we could try more in-depth tuning methods by tweaking the parameters inside the layers such as padding, stride and number of filters to improve the performance of our models.

REFERENCES

[1] K. Sirinukunwattana, S. E. A. Raza, Y. Tsang, D. R. J. Snead, I. A. Cree and N. M. Rajpoot, "Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images," in *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1196-1206, May 2016, doi: 10.1109/TMI.2016.2525803.

[2] A. RanjanRout, "Advantages and Disadvantages of Logistic Regression - GeeksforGeeks", *GeeksforGeeks*, 2020. [Online]. Available: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/. [Accessed: 23- May- 2021].

[3] M. Chatterjee, "The Introduction of KNN Algorithm | What is KNN Algorithm?", *GreatLearning Blog: Free Resources what Matters to shape your Career!*, 2020. [Online]. Available: https://www.mygreatlearning.com/blog/knn-algorithm-introduction/.

[4] Great Learning Team, "Random Forest Algorithm- An Overview | Understanding Random Forest", *GreatLearning Blog: Free Resources what Matters to shape your Career!*, 2020. [Online]. Available: https://www.mygreatlearning.com/blog/random-forest-algorithm/.

[5] "Boosting - Overview, Forms, Pros and Cons, Option Trees", *Corporate Finance Institute*, 2020. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/other/boosting/.

[6] "IBM Docs", *Ibm.com*, 2020. [Online]. Available: https://www.ibm.com/docs/en/spss-statistics/24.0.0?topic=option-multilayer-perceptron.

[7] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way", *Medium*, 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *arXiv.org*, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556.

[9]V. Feng, "An Overview of ResNet and its Variants", *Medium*, 2017. [Online]. Available: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035.

[10]J. Wei, "AlexNet: The Architecture that Challenged CNNs", *Medium*, 2019. [Online]. Available: https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951.

[11]F. Shaikh, "Inception Network | Implementation Of GoogleNet In Keras", *Analytics Vidhya*, 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/10/understanding-inception-network-from-scratch/#:~:text=The%20paper%20proposes%20a%20new,which%20is%2027%20layers%20deep.&text=1%C3%971%20Convolutional%20layer%20before%20applying%20another%20layer%2C%20which,mainly%20used%20for%20dimensionality%20reduction.

[12] S. H. Shabbeer Basha, S. Ghosh, K. Kishan Babu, S. Ram Dubey, V. Pulabaigari and S. Mukherjee, "RCCNet: An Efficient Convolutional Neural Network for Histological Routine Colon Cancer Nuclei Classification," 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2018, pp. 1222-1227, doi: 10.1109/ICARCV.2018.8581147.

[13] V. Bushaev, "Adam — latest trends in deep learning optimization.", *Medium*, 2018. [Online]. Available: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c.

[14]H. Zunair, "Improving performance of image classification models using pretraining and a combination of…", *Medium*, 2021. [Online]. Available: https://medium.com/decathlondevelopers/improving-performance-of-image-classification-models-using-pretraining-and-a-combination-of-e271c96808d2.