Deep Convolutional Neural Networks for Breast Cancer Histology Image Analysis[1]

Problem Statement

Breast Cancer is one of the most common causes for cancer deaths in the world. In order to overcome the subjectivity & human error in detection and diagnosis, this paper aims to develop an efficient way to classify the breast cancer cells into **4** classes, viz., normal, benign, in situ carcinoma, and invasive carcinoma. The dataset contains **400 images(2048×1536)** of breast cancer cells stained with hematoxylin and eosin (H&E). This paper utilizes features extracted by standard deep CNNs pre-trained on large datasets like ImageNet, and for classification, gradient boosted trees, **LightGBM**.

Data pre-processing and augmentation

Normalizing the H&E stains on the tissue[2]:

1. Converting RGB color vectors(I) to optical density(OD) space:
   OD = –log10(I), neglect values below the threshold β.
2. Project the data onto a plane created from **SVD**(on OD tuples) directions corresponding to two largest singular values and normalize to unit length.
3. Find the robust extremes(**αth** and **(100–α)th** percentiles) of angle of each point wrt the first SVD direction and convert those values back to the OD space.
4. Find the **99th** percentile(approximate maximum) of the intensity values of pixels and scale the intensity histograms of every image to this pseudo-maximum.

Data pre-processing and augmentation

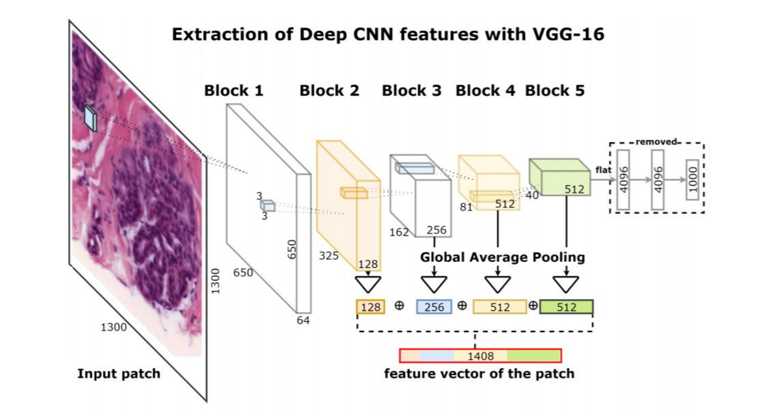
- Furthermore, **50** random augmentations are performed on each image.
- Then, **deconvolution** is utilized to decompose the image in **RGB space to HE colour space**[3]. Also, we multiply HE of every pixel by a random number from range **[0.7,1.3]**
- The images are downscaled to **1024 × 768**. We extract crops of **400×400** and **650×650**. **20** crops representing each image.
- These **20** crops give us **20** descriptors, which are combined by **3-norm pooling**:

$$\mathbf{d}_{pool} = \left(\frac{1}{N} \sum_{i=1}^{N} (\mathbf{d}_i)^p \right)^{\frac{1}{p}}$$

- Hence, the dataset size increases **300x** times i.e., **2** patch sizes x **3** encoders x **50** color/affine augmentations

Feature Extraction

1. We utilize standard pre-trained deep CNNs- **ResNet-50, InceptionV3 and VGG-16.**
2. In order to allow variable input sizes, we remove the fully connected layers of these networks and obtain a single 1D vector via Global Average Pooling on last convolutional layer in case of ResNet-50 and InceptionV3. But in case of VGG-16, we apply it to 4 internal convolutional layers.



LightGBM

LightGBM[4] utilizes GOSS and EFB as follows:

GOSS(Gradient-based One-Side Sampling) aims to assign less importance to data instances with low gradients(well-trained) and discard those. But in order to maintain the distribution of the dataset, it sorts the data instances based on the absolute values of their gradients, selects the top **a*100%**, randomly samples **b*100%** from the remaining ones and amplifies the sampled data with small gradients by a constant **(1−a)/ b** when calculating the information gain.

EFB(Exclusive Feature Bundling) on the other hand, bundles exclusive features into a single feature. It utilizes the analogy with the graph coloring problem,i.e., exclusive bundle of features in our problem corresponds to a set of vertices with the same color. These improvements certainly help us reducing the computational cost and speed up the training process as compared to traditional GBDT(Gradient Boosting Decision Trees).

Training and Results

For training, we use **10-fold cross-validation** across each encoder, scale and crop size combination. Also, train the **LightGBM** at five random seeds, in turn training **10 (number of folds) ×5 (seeds) ×4 (scale and crop) ×3 (CNN encoders) = 600** gradient boosting models. We preprocess the test data just like the training data and predict the class by averaging the probabilities over all augmentations and models.**Bagging and Boosting** help us diversifying the models, hence, overfitting is avoided.

Hence, we offer **87.2%** accuracy for this 4-class classification task.

Modification/Experiment

We employed the methods utilized in this paper on **COVID-19** dataset and the best accuracy obtained was .

Also, in order to compare LightGBM with other classifiers, we employed multiclass classification. The best accuracy obtained was .

Contributions By Individual Members

Aditi Ganesh Joshi(180020010): Research on methodologies used, code execution, Video presentation

Prakash Prasad (170070026): COVID-19 experimentation, code execution

Neha Jahnavi(18D070017): Background study, PPT, code execution

References

1. @article{rakhlin2018deep, title={Deep Convolutional Neural Networks for Breast Cancer Histology Image Analysis},author={Rakhlin, Alexander and Shvets, Alexey and Iglovikov, Vladimir and Kalinin, Alexandr A},journal={arXiv preprint arXiv:1802.00752},year={2018}} https://github.com/alexander-rakhlin/ICIAR2018
2. Marc Macenko, Marc Niethammer, JS Marron, David Borland, John T. Woosley, Xiaojun Guan, Charles Schmitt and Nancy E. Thomas, A method for normalizing histology slides for quantitative analysis, Biomedical Imaging: From Nano to Macro (ISBI'09), 1107–1110, 2009.
3. Arnout C. Ruifrok, Dennis A. Johnston and others, Quantification of histochemical staining by color deconvolution, Analytical and Quantitative Cytology and Histology, 23 (4), 291–299, 2001.
4. Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye and Tie-Yan Liu, LightGBM: A highly efficient gradient boosting decision tree, Advances in Neural Information Processing Systems, 3149–3157, 2017.