A REPORT
ON

Application of GCNs for Renal Cell Carcinoma Gene Expression and WSI Datasets

BY

Ishita Mediratta

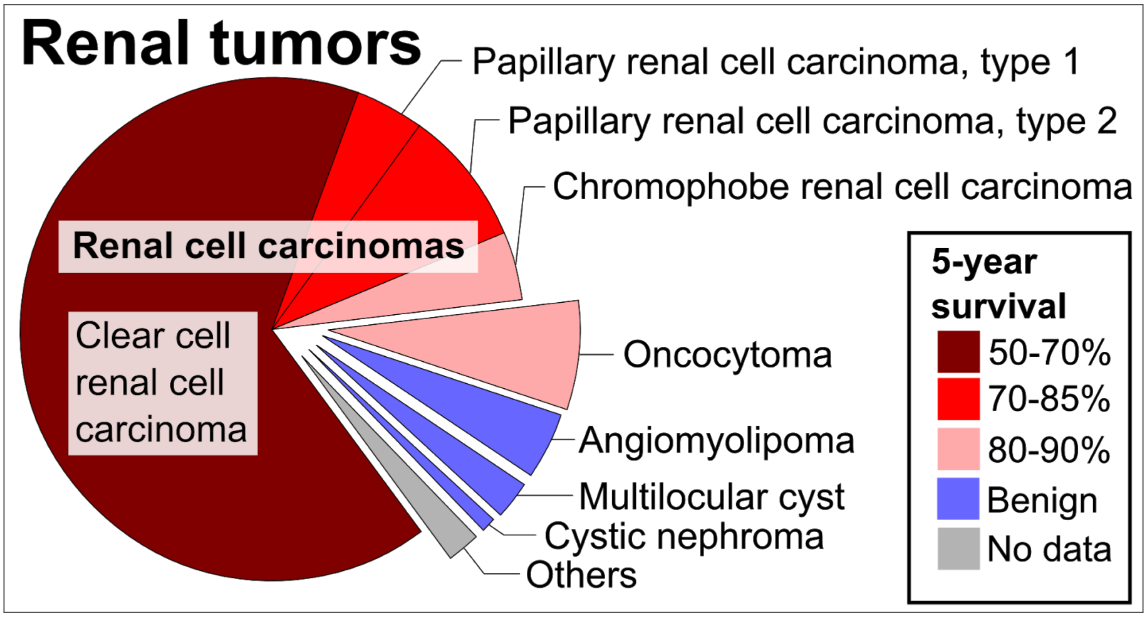
Prepared on completion of the
Internship
AT
Computational Systems Biology Lab, IIIT Hyderabad

TABLE OF CONTENTS

1 INTRODUCTION

Renal Cell Carcino (RCC) is the most common type of kidney cancer in adults, responsible for approximately 90–95% of cases. RCC has over 40 subtypes, however the most common of them are:

1. Clear Cell RCC (KIRC)
2. Papilary Cell RCC (KIRP)
3. Chromophobe RCC (KICH)



Recently, there has been a sudden interest in using Machine Learning techniques for early prognosis as well as clinical decision making. More notably, present-day Deep Learning methods have shown considerable performance, and in many cases, are now being deployed in real-time (Rajpurkar et al., 2017). The ability of Deep learning to deal easily with the large amounts of structural data that are increasingly becoming publicly available, without much feature engineering (which is still an issue with vanilla ML methods) as well its ability to generalize well in unseen, multimodal domains, makes it a suitable algorithmic choice. While Deep ANNs and CNNs have been explored extensively for prediction tasks on multimodal RCC datasets ((Han et al., 2019; Li et al., 2020; Suarez-Ibarrola et al., 2020; Tabibu et al., 2019)), we aim to approach this problem using Graph Convolutional Networks. In the past few years, Graph Neural Networks have shown to be effective in many representation learning tasks (Zhang et al. 2019). The fact that one can encode "structural" information in the dataset, and hence use that as a prior while learning, makes these models quite powerful. Therefore, the aim of this summer internship project is to "**Develop novel multimodal deep learning methods using Graph Convolutional Networks for KIRC/KIRP prediction tasks**". We first start with exploring already available literature, implement on our Gene Expression datasets for two subproblems:

1. Cancer vs Normal Classification for KIRC+KIRP+KICH patients using Gene Expression values

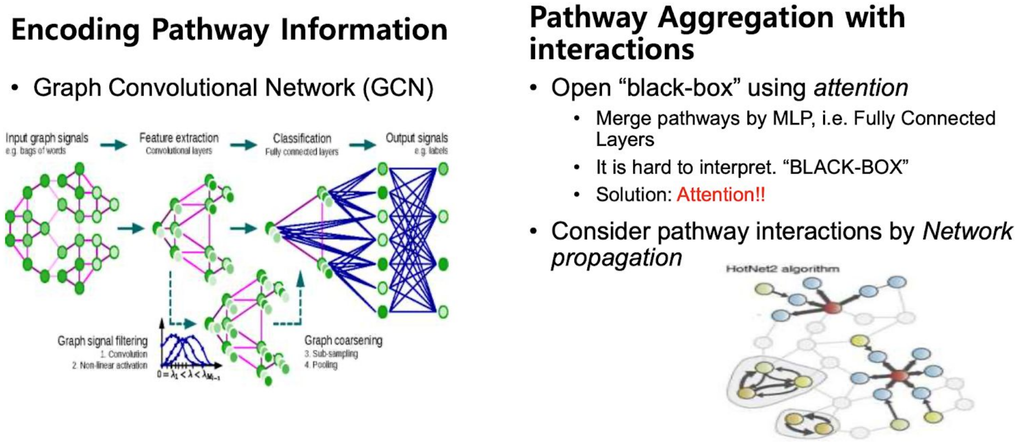
2. Early vs Late Stage Classification for KIRC patients using Gene Expression values

We also extend these methods using ML models as well as a self-supervised Autoencoder-based approach, and these methods perform comparably to SOTA models.

2 Related Works

Paper 1: Cancer subtype classification and modeling by pathway attention and propagation (Lee et al., 2020)

This paper is the foundation for our project, and is currently the SOTA baseline for Subtype Classification problem. For each input training example, it operates on KEGG Pathway Graphs (nodes as genes) such that node features represent the gene expression values and a label associated with each patient, which represents the cancer subtype. The outputs of this model are 1. RCC Subtype Prediction and 2. Importance of each KEGG Pathway in influencing the prediction outcome.



Encoding Pathway Information

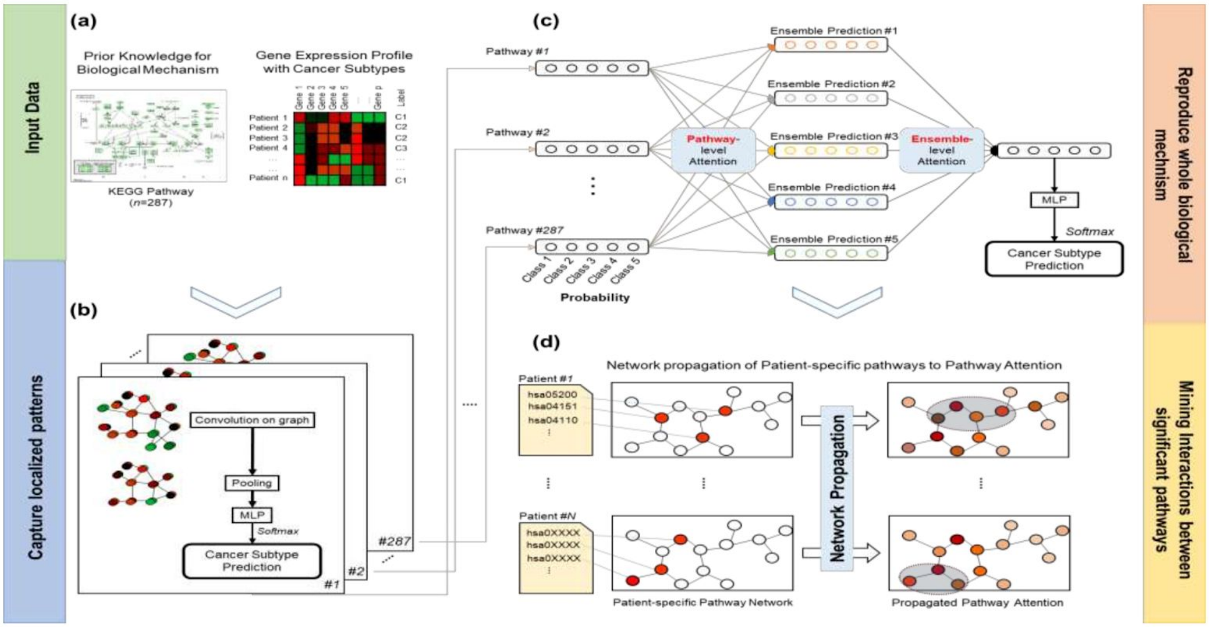
- Graph Convolutional Network (GCN)

Pathway Aggregation with interactions

- Open "black-box" using *attention*
  - Merge pathways by MLP, i.e. Fully Connected Layers
  - It is hard to interpret. "BLACK-BOX"
  - Solution: Attention!!
- Consider pathway interactions by *Network propagation*

The workflow of this paper is divided into 3 parts:

Workflow

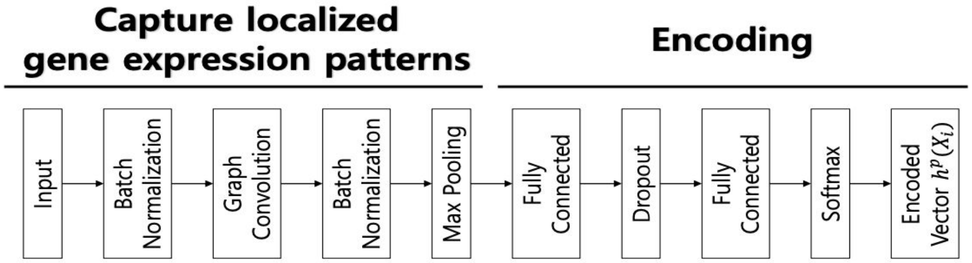


1. *Pathway Collection*

- Collected 114 Pathways from KEGG Database
- Excluded pathways that were related to Drug Development, or had <10 genes (preprocessing)
- Converted the pathways in .xml format into .csv having adjacency list

2. *GCN creation*

Here, the authors have used ChebNet (Defferrard et al., 2016), a graph convolutional network model which calculates Chebyshev polynomials to approximate the convolutional kernel. They apply this convolutional operation using ChebNet, and at the end perform a softmax operation to get an encoding vector.



3. *Attention Modelling*

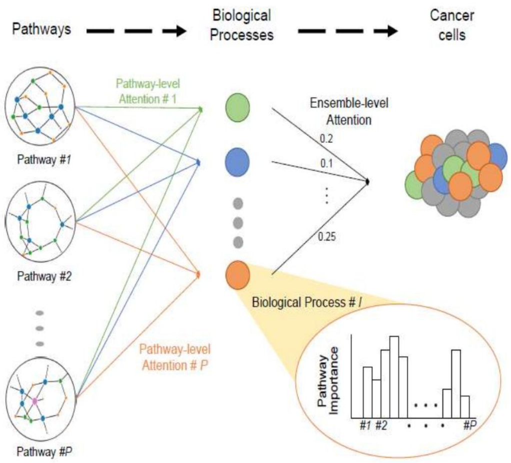
Finally, attention, although a bit different from (Vaswani et al., 2017), is used to find out the contribution of each pathway towards predicting that patient's class. The method is as follows:

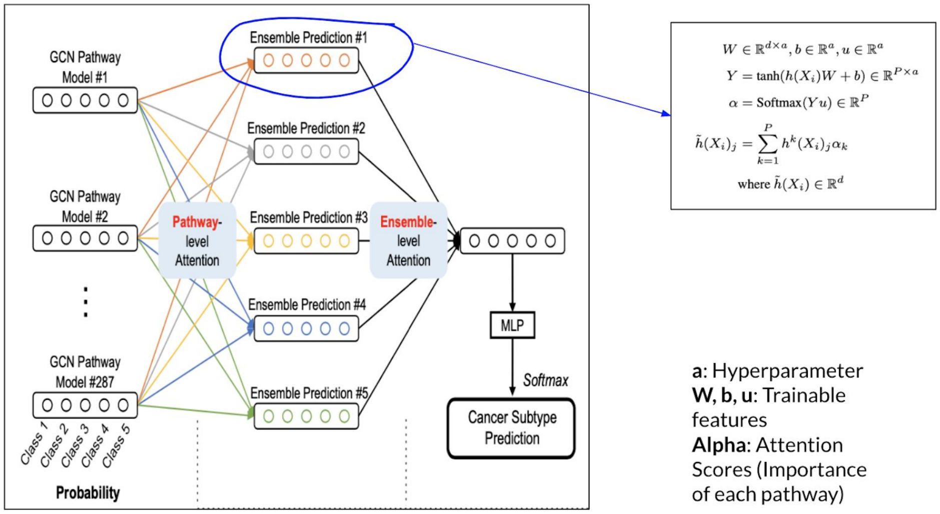
$h(X_i) \in \mathbb{R}^{P \times d}$ ← Outputs from 111 GCN Models for a single patient

$$W \in \mathbb{R}^{d \times a}, b \in \mathbb{R}^{a}, u \in \mathbb{R}^{a}$$
$$Y = \tanh(h(X_i)W + b) \in \mathbb{R}^{P \times a}$$
$$\alpha = \mathrm{Softmax}(Yu) \in \mathbb{R}^{P}$$
$$\tilde{h}(X_i)_j = \sum_{k=1}^{P} h^k(X_i)_j \alpha_k$$
$$\text{where } \tilde{h}(X_i) \in \mathbb{R}^{d}$$

Output of single pathway-attention model

Multi attention based ensemble (MAE)





$$W \in \mathbb{R}^{d \times a}, b \in \mathbb{R}^{a}, u \in \mathbb{R}^{a}$$
$$Y = \tanh(h(X_i)W + b) \in \mathbb{R}^{P \times a}$$
$$\alpha = \mathrm{Softmax}(Yu) \in \mathbb{R}^{P}$$
$$\tilde{h}(X_i)_j = \sum_{k=1}^{P} h^k(X_i)_j \alpha_k$$
$$\text{where } \tilde{h}(X_i) \in \mathbb{R}^{d}$$

a: Hyperparameter
W, b, u: Trainable features
Alpha: Attention Scores (Importance of each pathway)

Our experiment results on Gene Expression Datasets extended on top of this paper

Experiment 1: RCC Subtype Classification using "individual" GCN Models

1. Trained individual models on these 111 pathways using 2-level 10x3 cross-validation, 10 epochs
2. Compared the best performing pathways (having ~95% accuracy) with a Random Forest model trained on the same set of genes

3. The best performing pathway models (gene sets) are:

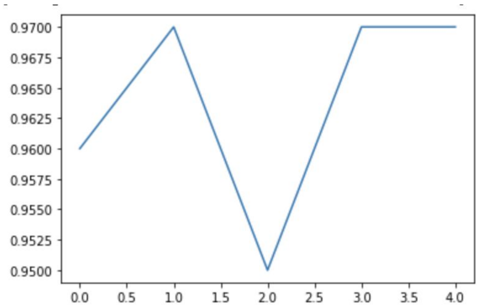
Pathway	Model	F1-weighted Score
hsa04530: Tight junction - Homo sapiens (human)	Random Forest	0.9381
	GCN	**0.9516**
hsa04010: MAPK signaling pathway - Homo sapiens (human)	Random Forest	0.9417
	GCN	**0.9494**
hsa04068: FoxO signaling pathway - Homo sapiens (human)	Random Forest	0.9413
	GCN	**0.9503**

4. The least performing pathways are:

Pathway	F1-weighted Score
Ferroptosis	0.8191618193427251
IL-17 signaling pathway	0.8507
Carbohydrate digestion and absorption	0.8773

Experiment 2: RCC Subtype Classification using attention-based “ensemble” of GCN Models [full implementation of this paper]

1. Trained an ensemble of 11 attention-based models
2. Used 5-fold cross-validation
3. **Mean f1-weighted score: 96.8%**
4. Mean weighted recall: 96.4%
5. Mean weighted precision: 96.4%
6. The graph on right shows the f1-weighted score for 5-folds



The attention scores for best for each subtype are:

PATHWAY	**KIRP (20)**	**KIRC (518)**	**KICH (81)**
Hsa04530	0.09979762	0.096965	0.09397491
Hsa04068	0.09854131	0.094651714	0.096787274
Hsa04010	0.09969057	0.09493622	0.09815481
Hsa05211	0.09534384	0.094353616	0.09505498

<u>Experiment 3: Early/Late Stage RCC (all subtypes) Classification using just one KEGG Pathway:hsa05211 (one GCN model) having 55 genes</u>

1. Used 8-level (train-test) and 10-level (train-val) cross-validation (with stratified sampling to deal with imbalance)
2. In total, 563 Early-stage patients and 293 late-stage patients
3. Used expression values from 55 genes from RCC pathway for classification

Metric	Random Forest	GCN	Random Forest (on all 21k genes)
Mean F1-score	0.6387	0.6547	**0.6835**

Clearly, for Early/Late stage classification problem, a single GCN didn't help. Therefore, we check if this paper's overall attention based modelling helps or not.

<u>Experiment 4: KIRC Early/Late Stage Classification using this paper's model on all 111 KEGG Pathways</u>

F1-score was ~40%

It's unclear as to why it performed badly. One possible hypothesis would be that KEGG graphs might not be good indicator of early/late stage classification, and hence, a different arrangement/set of genes needed to be found.

PAPER 2: CONVOLUTIONAL NEURAL NETWORK APPROACH TO LUNG CANCER CLASSIFICATION INTEGRATING PROTEIN INTERACTION NETWORK AND GENE EXPRESSION PROFILES (MATSUBARA ET AL., 2018)

Implementation Details

INPUT:

1. HINT 4.0 PPI Network
2. Gene expression profile with cancer subtype

OUTPUT:

1. Subtype Classification

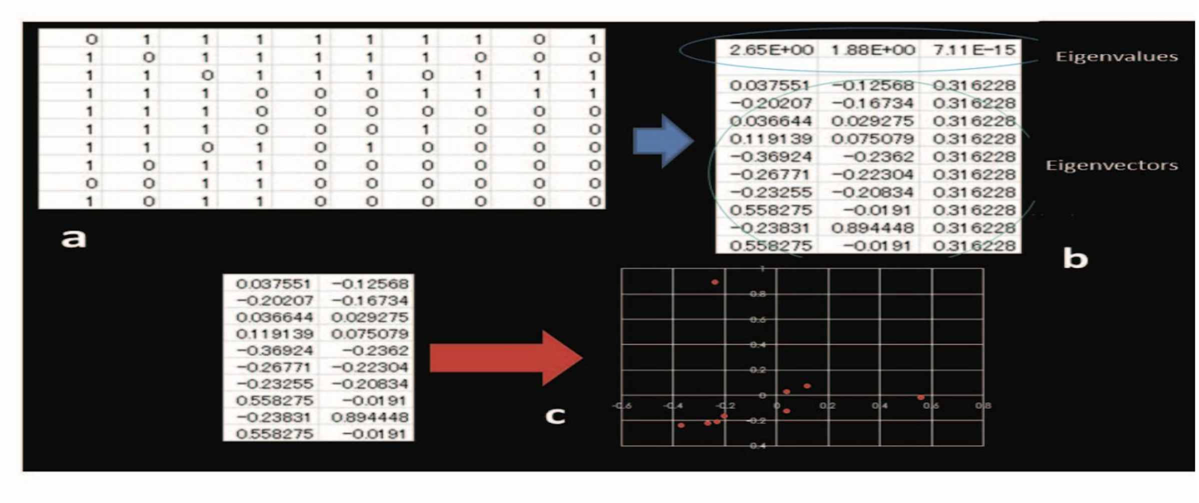
Theory: Uses smallest two (non-zero) eigenvalues, and their corresponding eigenvectors of the Laplacian for creating a 2-D representation of PPI graph.

Why? Because, spectral clustering is a technique with roots in graph theory, where the approach is used to identify communities of nodes in a graph based on the edges connecting them.

- The first nonzero eigenvalue is called the spectral gap. The spectral gap gives us some notion of the density of the graph.
- The second eigenvalue is called the Fiedler value, and the corresponding vector is the Fiedler vector. The Fiedler value approximates the minimum graph cut needed to separate the graph into two connected components.

Method:

1. Applied a spectral clustering to the PPI network and mapped the eigenvectors corresponding to the next smallest eigenvalues
2. Created a 2D-matrix by creating a grid and adding the gene expression value corresponding to that gene at its corresponding eigenvectors' coordinates
3. Get 100x400 matrix for each patient
4. Pass that to CNN model for subtype classification



0	1	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	0	0	0
1	1	0	1	1	1	0	1	1	1
1	1	1	0	0	0	1	1	1	1
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	1	0	0	0
1	1	0	1	0	1	0	0	0	0
1	0	1	1	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0

2.65E+00	1.88E+00	7.11E−15	Eigenvalues
0.037551	−0.12568	0.316228	
−0.20207	−0.16734	0.316228	
0.036644	0.029275	0.316228	
0.119139	0.075079	0.316228	
−0.36924	−0.2362	0.316228	Eigenvectors
−0.26771	−0.22304	0.316228	
−0.23255	−0.20834	0.316228	
0.558275	−0.0191	0.316228	
−0.23831	0.894448	0.316228	
0.558275	−0.0191	0.316228	

0.037551	−0.12568
−0.20207	−0.16734
0.036644	0.029275
0.119139	0.075079
−0.36924	−0.2362
−0.26771	−0.22304
−0.23255	−0.20834
0.558275	−0.0191
−0.23831	0.894448
0.558275	−0.0191

Figure 2: From a network (adjaceny matrix), we compute spectral clustering and (b) select the two eigenvectors corresponding to the second and third smallest eigenvalues. (c) These two eigenvectors are mapped into a 2D space representation.

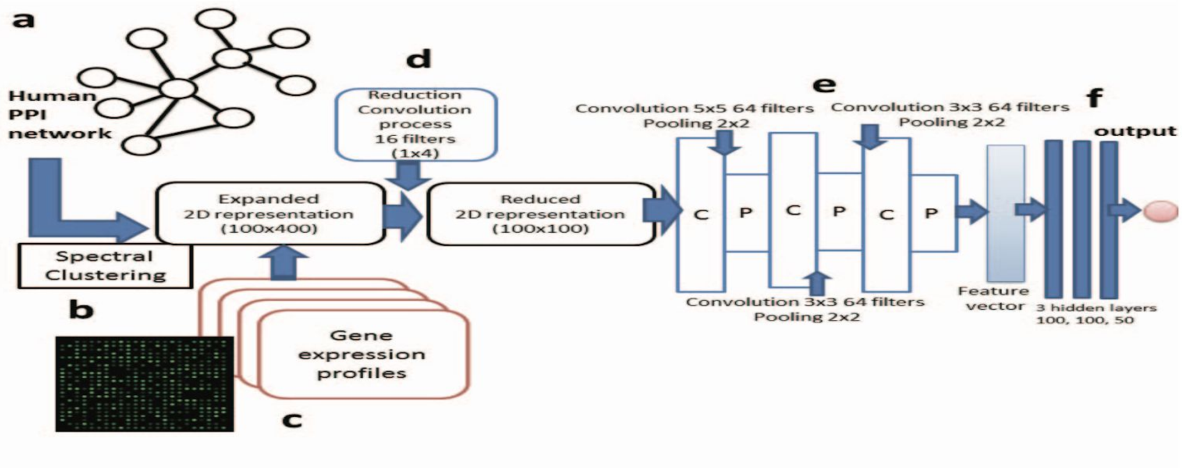
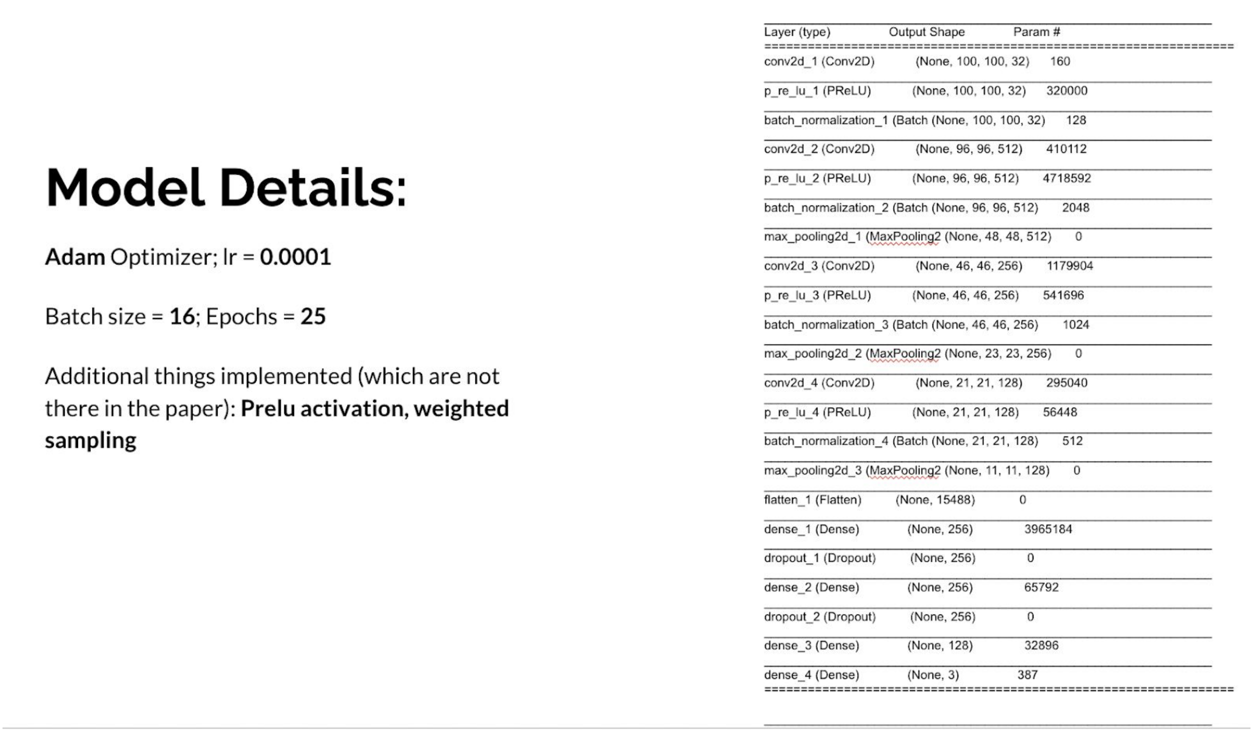


Figure 1 : Overview of the proposed methodology. (a-b) Spectral clustering is performed on human protein inteaction network. The resulting 2D representation is expanded with a resolution of 100x400 cells (see Fig.3). (c-d) Gene expression data is mapped on the 2D space and the dimension is reduced to 100x100 using convolution filters. (e) Three convolutional layers are applied as shown in figure. (f) Three hidden layers with 100,100 and 50 neurons, respectively, are applied. Final prediction score integrates the hidden layer outputs.



Model Details:

Adam Optimizer; lr = **0.0001**

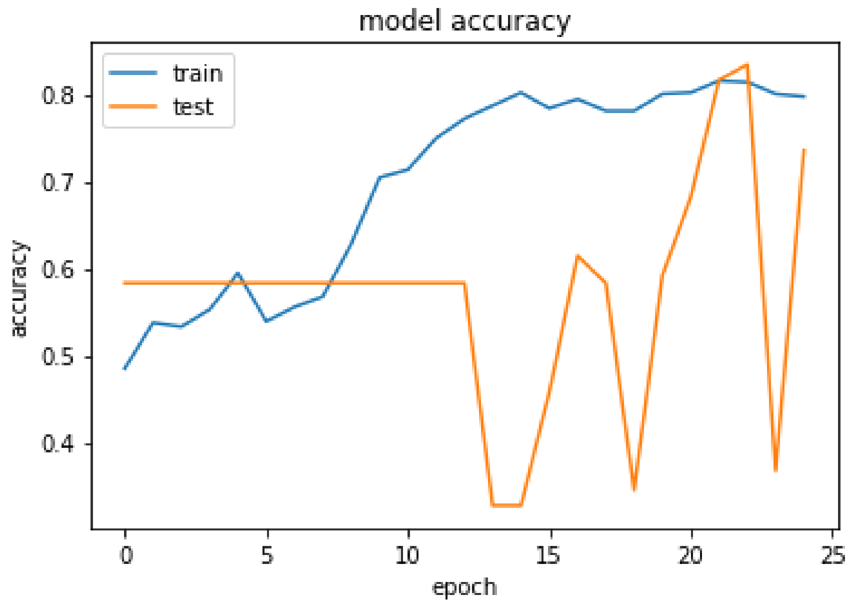
Batch size = **16**; Epochs = **25**

Additional things implemented (which are not there in the paper): **Prelu activation, weighted sampling**

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 100, 100, 32)      160
_________________________________________________________________
p_re_lu_1 (PReLU)            (None, 100, 100, 32)      320000
_________________________________________________________________
batch_normalization_1 (Batch (None, 100, 100, 32)      128
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 96, 96, 512)       410112
_________________________________________________________________
p_re_lu_2 (PReLU)            (None, 96, 96, 512)       4718592
_________________________________________________________________
batch_normalization_2 (Batch (None, 96, 96, 512)       2048
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 48, 48, 512)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 46, 46, 256)       1179904
_________________________________________________________________
p_re_lu_3 (PReLU)            (None, 46, 46, 256)       541696
_________________________________________________________________
batch_normalization_3 (Batch (None, 46, 46, 256)       1024
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 23, 23, 256)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 21, 21, 128)       295040
_________________________________________________________________
p_re_lu_4 (PReLU)            (None, 21, 21, 128)       56448
_________________________________________________________________
batch_normalization_4 (Batch (None, 21, 21, 128)       512
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 11, 11, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 15488)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               3965184
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               65792
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_4 (Dense)              (None, 3)                 387
=================================================================
_________________________________________________________________
```

Experiment 5: KIRC Early/Late Stage Classification using their approach (HINT 4.0 PPI Graph)

	precision	recall	f1-score
0	0.92	0.47	0.62
1	**0.00**	**0.00**	**0.00**



Some other KIRC EARLY/LATE STAGE

EXPERIMENTS ALONG SAME LINES

Experiment 6: Using Coefficient of Variance method for feature selection

Method:

1. Pass whole dataset to sklearn's SelectKBest function to get **top 3000 features/genes**
2. Model is VotingClassifier (made of LR and RF)

Accuracy: 0.7476

Experiment 7: Using lasso regression model to get important features and then passing those onto a Voting Classifier

total input features: 20531

top selected features: 336

Model: VotingClassifier →ExtraTreesClassifier + RandomForestClassifier

Accuracy: 0.7281

Experiment 8: Using Self-supervised Autoencoder based approach

Similar to the approach given here

Accuracy: 0.6213

Experiment 9: Ensemble-models of Machine Learning with Attention

Similar to Experiment 4, but instead of GCN, tried a voting classifier made up of RF+ExtraTrees+Gaussian Naive Bayes

Accuracy: 0.78

Experiment 10: Ensemble-models on individual KEGG pathways' genes as features

Similar to Experiment 3, i.e. take one KEGG Pathway at a time - use its genes as features for training a Voting Classifier made of RF+SVM+Gaussian Naive Bayes+MLP (128-128)

Highest Accuracy: 80.6% and 0.821 AUC using genes from hsa04015 KEGG Pathway

Comparison with SOTA

Currently, the SOTA model is (Li et al., 2020), wherein they used SVM based prediction model using 23 selected genes to get an **accuracy of 81.15% and AUC 0.86**

HOWEVER, they have used a dataset from USCS Xena Browser, having 606 patient samples, whereas we are using a dataset of 515 patients for training our models. Hence, for fair comparison, we applied our smaller dataset onto their model, and it gave an **accuracy of ~77-78%, which is clearly lesser than ours (80.6%).**

But it'd be better to train our model on their larger dataset, and we leave that as a future work.

Working with WSIs

Since GCN on gene expression dataset alone are not enough to get good accuracy on KIRC Early/Late stage classification task, we hypothesize whether including information prior from Whole Slide Images (taken from TCGA portal) will help us in getting some boost.

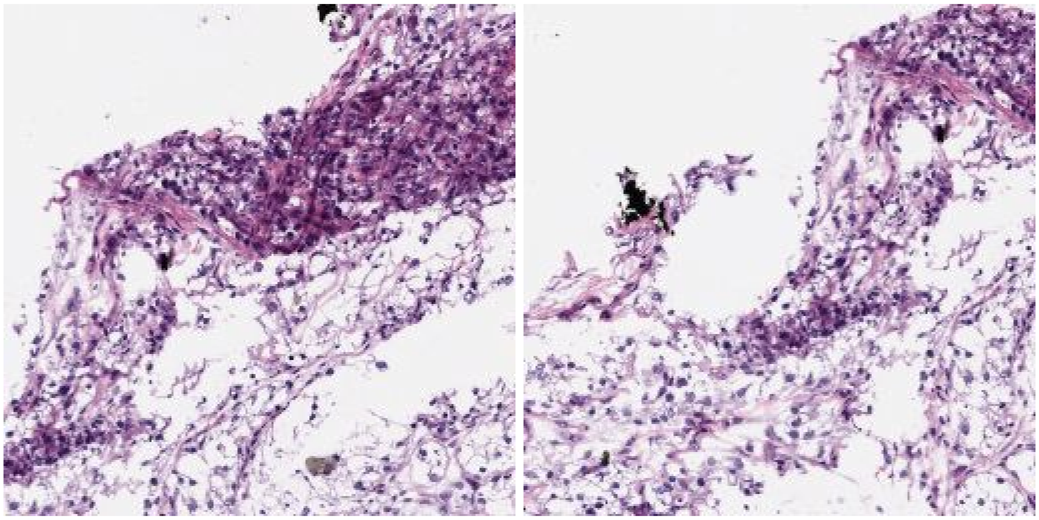
We intend to develop a pipeline wherein:

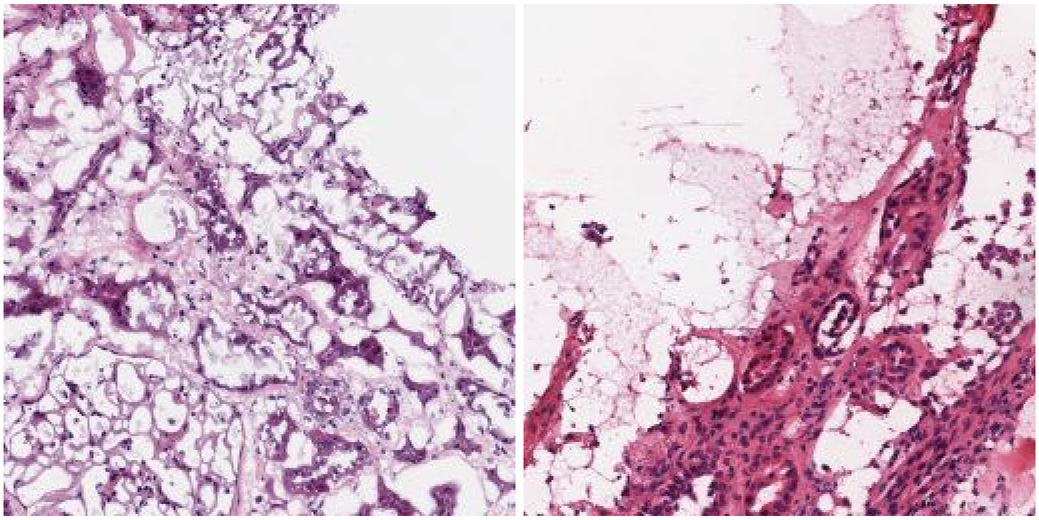
1. Extract patches from WSIs having mean ratio of > 0.8 [DONE]

TODOs:

2. Perform nuclei segmentation on these patches
3. Perform feature extraction on these segmented nuclei and store it in a dataframe, using approaches given in (Rawat et al., 2018; Zhou et al., 2019)

Here are some sample patches extracted at level 1





The metadata for these patches have been saved in a CSV file (patches_coords.csv) in the following format:

<WSI ID>, <patch ID>, <y coordinate of left corner>, <x coordinate of left corner>

Future Work

1. Extract features from WSI patches and merge with GCN pipeline

3 References

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1606.09375

Han, S., Hwang, S. I., & Lee, H. J. (2019). The Classification of Renal Cancer in 3-Phase CT Images Using a Deep Learning Method. *Journal of Digital Imaging*, *32*(4), 638–643.

Lee, S., Lim, S., Lee, T., Sung, I., & Kim, S. (2020). Cancer subtype classification and modeling by pathway attention and propagation. *Bioinformatics* , *36*(12), 3818–3824.

Li, F., Yang, M., Li, Y., Zhang, M., Wang, W., Yuan, D., & Tang, D. (2020). An improved clear cell renal cell carcinoma stage prediction model based on gene sets. *BMC Bioinformatics*, *21*(1), 232.

Matsubara, T., Nacher, J. C., Ochiai, T., Hayashida, M., & Akutsu, T. (2018). Convolutional Neural Network Approach to Lung Cancer Classification Integrating Protein Interaction Network and Gene

Expression Profiles. *2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE)*, 151–154.

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M. P., & Ng, A. Y. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. In *arXiv [cs.CV]*. arXiv. http://arxiv.org/abs/1711.05225

Rawat, R. R., Ruderman, D., Macklin, P., Rimm, D. L., & Agus, D. B. (2018). Correlating nuclear morphometric patterns with estrogen receptor status in breast cancer pathologic specimens. *NPJ Breast Cancer*, *4*, 32.

Suarez-Ibarrola, R., Hein, S., Reis, G., Gratzke, C., & Miernik, A. (2020). Current and future applications of machine and deep learning in urology: a review of the literature on urolithiasis, renal cell carcinoma, and bladder and prostate cancer. *World Journal of Urology*, *38*(10), 2329–2347.

Tabibu, S., Vinod, P. K., & Jawahar, C. V. (2019). Pan-Renal Cell Carcinoma classification and survival prediction from histopathology images using deep learning. *Scientific Reports*, *9*(1), 10509.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *arXiv [cs.CL]*. arXiv. https://arxiv.org/abs/1706.03762

Zhou, Y., Graham, S., Koohbanani, N. A., Shaban, M., Heng, P.-A., & Rajpoot, N. (2019). CGC-Net: Cell Graph Convolutional Network for Grading of Colorectal Cancer Histology Images. In *arXiv [eess.IV]*. arXiv. http://arxiv.org/abs/1909.01068