DeepGx: Deep Learning Using Gene Expression for Cancer Classification

Joseph M. de Guia [1,2] Madhavi Devaraj[1] Carson K. Leung[2]
[1] *School of Information Technology, Mapua University*
Manila, Philippines
Email: {jmdeguia, mdevaraj}@mapua.edu.ph

[2] *Department of Computer Science, University of Manitoba*
Winnipeg, MB, Canada
Email: kleung@cs.umanitoba.ca

***Abstract*—This paper aims to explore the problems associated in solving the classification of cancer in gene expression data using deep learning model. Our proposed solution for the cancer classification of ribonucleic acid sequencing (RNA-seq) extracted from the Pan-Cancer Atlas is to transform the 1-dimensional (1D) gene expression values into 2-dimensional (2D) images. This solution of embedding the gene expression values into a 2D image considers the overall features of the genes and computes features that are needed in the classification task of the deep learning model by using the convolutional neural network (CNN). When training and testing the 33 cohorts of cancer types in the convolutional neural network, our classification model led to an accuracy of 95.65%. This result is reasonably good when compared with existing works that use multiclass label classification. We also examine the genes based on their significance related to cancer types through the heat map and associate them with biomarkers. Our CNN for the classification task fosters the deep learning framework in the cancer genome analysis and leads to better understanding of complex features in cancer disease.**

Keywords*—*deep learning, machine learning, neural network, convolutional neural network (CNN), gene expression, ribonucleic acid sequencing (RNA-seq), bioinformatics

I. Introduction

In the World Health Organization (WHO) report [1], cancer disease accounts as the second leading cause of death worldwide resulting 9.6 million deaths in 2018. This account was supported by the World Cancer Report 2014 emphasized cancer as a global problem and projected that cancer incidence will increase to 20 million new cases by 2025. The significance of the cancer disease in the population motivates many researchers and scientific networks to explore ways on reduce the risk factors. The burden of the disease can be improved by early detection, screening and with the application of specific technology for the personalized and precision medicine. With the need for cancer genomics studies, *The Cancer Genome Atlas* (*TCGA*) [2] was established in 2005. It is a publicly available Genomic Data Commons Portal containing more than 11,000 primary cancer samples from its 33 cancer types in its repository. The genomic information of cancer disease was based on the molecular characteristics and its correlation to genomic, epigenomic, and transcriptomic across all cancer types were identified according to the PanCancer Atlas database. The comprehensive molecular analysis of TCGA data portal is publicly available for further analysis of its cancer dataset where researchers can discover and contribute to the improvement of PanCancer-omics analysis.

In this paper, we consider all 33 cancer types in TGCA with its *ribonucleic acid sequencing* (*RNA-seq*) gene expression data to determine the candidate signature or biomarker of the gene expression data. The gene expression characteristics were specific for each cancer type, and would not be known for other types in terms of multiclass type of analysis. *Machine learning* is a tool being used to determine the molecular characteristics and biological meaning cancer. Machine learning can handle high-dimensional input and provide accurate outcomes. However, there is a trade-off in the slow training process when dealing with high volume of input data. Similarly, more hand-crafted feature selection, fine-tuning of the parameters are needed in order to get an accurate outcome. Hence, *deep neural network* (*DNN*) or *deep learning* (which is a branch of machine learning) uses neural networks to extract features and make prediction from input data. Deep learning has gained more the attention of the researchers because of its ability to handle large input data, less hand-crafted feature selection and fine-tuning of parameters, faster and accurate prediction, and can be used in genomic data. The number of layers needed to optimize the

ASONAM '19, August 27–30, 2019, Vancouver, BC, Canada

ACM 978-1-4503-6868-1/19/08…$15.00
https://doi.org/10.1145/3341161.3343516

process of learning needs to be considered. In addition, the understanding of how neural networks can be represented in visualized form. In this method, the visualization—in the form of heatmaps—creates an inference according to the input images and helps in the classification task of providing the accurate contribution to the image characteristics. Our *key contributions* of this paper include:

- our proposal of a deep learning model for classifying the cancer using the gene expression;
- our demonstration of the significance and importance of our deep learning model to cancer genes being inferred as a method to classify each of the genes for each cancer type; and
- our approach of embedding the 1-dimensional (1D) preprocessed gene expression data into 2-dimensional (2D) image for the proposed deep learning model using *convolutional neural network* (*CNN*).

We run experiments using Pytorch framework and determine the number of layers and parameters suited for the given input transformed dataset. Then, we check the trained model of the multiclass label classification. The model is subjected to a performance test using 10-fold cross validation. A technique for the visualization method using heatmaps to approach the genes selection and evaluate to better understand if the top genes corresponds to the result being classified as cancer type and associate the classified cancer genes with the biomarker. Through this experiment, we can compare our result from other implementations using the same dataset and classification task, then prove if the method used in deep learning model is suitable for this method and genomic data.

The remainder of the paper is organized as follows. The next section discusses related works and background. Section III presents our methods. Then, Section IV reports and discusses our evaluation results. Conclusions are drawn and future work is suggested in Section V.

II. Related Works

Deep learning was derived from the *artificial neural network* (*ANN*) family. This method has been the current state-of-the-art in many applications [3] such as:

- machine translation,
- natural language processing (NLP),
- object detection,
- question-answer,
- speech recognition,
- visual object recognition, and
- other large data-intensive task related to multiple processing learning representations of data and their abstraction.

There are many implementations of the deep learning models that constantly improving the machine-level performance in disease diagnosis with great accuracy for various genomic applications.

In the genomic application, deep learning has been the extensively exploited in all areas of the omics research [4]–[9]. The deep learning methods in genomic problems are image detection and segmentation in many image classifications tasks (e.g., magnetic resonance imaging (MRI), Ultrasound, X-rays, etc.) to characterize genomic data using CNNs. *Recurrent neural networks* (*RNNs*) are very popular in handling sequence data such as

- deoxyribonucleic acid (DNA) sequences, and
- ribonucleic acid (RNA) sequence,

For gene expression data, a more diverse implementation of different neural network models and features extraction techniques was explored. ANN and discrete cosine transform extract features from the stomach microarray [10]. Deep forest can considered as another approach for using DNN used to classify cancer subtypes [11]. Ahn et al. [12] explored the multi-classification of cancer dataset from databases like

- Gene Expression Omnibus (GEO)[1],
- Genotype-Tissue Expression (Gtex)[2], and
- TCGA[3].

Their evaluation showed that their classification model led to a high training and test/classification accuracy over 13,123 samples. A forest DNN model with supervised feature detector [13] was applied for the sparsity of the gene expression data that solves the problems in feature identification and classification tasks. Lyu and Haque [14] applied deep learning to RNA-seq data for tumor classification. They converted the sequence data to 2D images, and applied CNN to classify the 33 tumor types. They achieved a 95% accuracy when using this technique.

III. Our Deep Learning Model

In this section, we describe our deep learning architecture and classification model, machine configuration, datasets, and the details of the methodology for validation.

A. DNN Architecture and Classification Model

The network architecture of the deep learning network is a convolutional neural network, where the input nodes are fully connected to the three hidden layer and each connection with the parameter weight. There are three fully connected layers to the output layer. The input size is 102, and the output size is 100. The kernel size is 5, 3, and 2, with paddings of 1. The max-pool has kernel size of 2, padding of 0, and stride of 2.

[1] https://www.ncbi.nlm.nih.gov/geo/

[2] https://gtexportal.org/home/

[3] https://portal.gdc.cancer.gov/

```
DataParallel(
      (module): Net(
            (conv1): Conv2d(1, 64, kernel_size=(5, 5), stride=(1, 1), padding=(1, 1))
             (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
             (conv2): Conv2d(64, 128, kernel_size=(5, 5), stride=(1, 1), padding=(1, 1))
             (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
             (conv3): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
             (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
             (drop2D): Dropout2d(p=0.25)
             (vp): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
             (fc1): Linear(in_features=36864, out_features=1024, bias=True)
             (fc2): Linear(in_features=1024, out_features=512, bias=True)
             (fc3): Linear(in_features=512, out_features=33, bias=True)
```

Fig. 1. CNN module definition and design.

TABLE I. CANCER TYPES AND DESCRIPTION WITH ACCURACY

	TGCA-code	**Cancer cohort type**	**Samples**	**Accuracy**		
				CNN [14]	***GA/kNN [17]***	***Our proposed CNN***
0	ACC	Adrenocortical carcinoma	79	0.95	0.97	0.95
1	BLCA	Bladder urothelial carcinoma	408	0.97	0.91	0.97
2	BRCA	Breast invasive carcinoma	1093	0.99	0.99	0.99
3	CESC	Cervical and endocervical cancers	304	0.93	0.94	0.93
4	CHOL	Cholangiocarcinoma	36	0.56	0.73	0.56
5	COAD	Colon adenocarcinoma	457	0.95	0.99	0.95
6	DLBC	Lymphoid neoplasm diffuse large B-cell lymphoma	48	1.00	1.00	1.00
7	ESCA	Esophageal carcinoma	184	0.77	-	0.77
8	GBM	Glioblastoma multiforme	160	0.94	0.99	0.94
9	HNSC	Head and neck squamous cell carcinoma	520	0.98	-	0.98
10	KICH	Kidney chromaphobe	66	0.87	0.96	0.87
11	KIRC	Kidney renal clear cell carcinoma	533	0.95	0.96	0.95
12	KIRP	Kidney renal papillary cell carcinoma	290	0.93	0.92	0.93
13	LAML	Acute myeloid leukemia	179	1.00	1.00	1.00
14	LGG	Brain lower grade glioma	516	0.98	1.00	0.98
15	LIHC	Liver hepatocellular carcinoma	371	0.97	0.98	0.97
16	LUAD	Lung adenocarcinoma	515	0.95	0.96	0.95
17	LUSC	Lung squamous cell carcinoma	501	0.91	0.88	0.91
18	MESO	Mesothelioma	87	0.94	0.90	0.94
19	OV	Ovarian serous cystadenocarcinoma	304	0.99	1.00	0.99
20	PAAD	Pancreatic adenocarcinoma	178	0.97	0.95	0.97
21	PCPG	Pheochromocytoma and paraganglioma	179	1.00	1.00	1.00
22	PRAD	Prostate adenocarcinoma	497	1.00	1.00	1.00
23	READ	Rectum adenocarcinoma	166	0.35	0.00	0.35
24	SARC	Sarcoma	259	0.97	0.96	0.97
25	SKCM	Skin cutaneous melanoma	469	0.98	0.97	0.98
26	TGCT	Testicular germ cell tumors	150	0.99	1.00	0.99
28	THCA	Thyroid carcinoma	501	1.00	0.94	1.00
29	THYM	Thymoma	120	0.99	0.96	0.99
30	UCEC	Uterine corpus endometrial carcinoma	545	0.96	0.62	0.96
31	UCS	Uterine carcinosarcoma	57	0.81	1.00	0.81
32	UVM	Uveal melanoma	80	0.99	1.00	0.99
		Total	**10267**			

The activation functions are max-pool and drop-out. There are three max-pool and three dropout layers:

- If the output is 0, then the input is less than 0.
- If the input is greater than 0, then the output can be computed by Eq. (2).

The maxpool function is given by Eq. (2). The number of epochs is 200. When no improvement in the training and error performance, the process is terminated. The Pytorch CNN model and module definition is presented in Fig. 1. The dropout [nn.Dropout2d [4]] rate is 0.25. The size is 500 for the batch, and epoch is 200 times. The learning rate is 0.0001 with three convolution layers set at:

- 102×102×64 nodes,
- 50×50×128 nodes, and
- 24×24×256 nodes;

[4] https://pytorch.org/docs/stable/nn.html#dropout2d

and max-pool of:

- 50×50×64,
- 24×24×128, and
- 12×12×25.

The optimum uses two gradient history method algorithms for the learning rate:

- momentum GD, and
- RMSProp GD.

Adam uses bias correction and optimizes the batch sizes of the images. The individual genes in the input layer of the DNN architecture were set based on the algorithm to calculate the score of each gene in the input and output layer. This computation of the score of each genes should be done to normalize the expression values of profile S[I,:], replace the values, and leave the other values of the genes in the sample. The following equations are used for the Pytorch framework:

$$f(x) = \max\{x, 0\} \qquad (1)$$

$$output(N_i, C_{out}) = bias(C_{out}) + \text{sum of weighted input} \qquad (2)$$

where sum of weighted input

$$= \sum_{C_{in}-1}^{k=0} weight(C_{out}, k) \times input(N_i, k)$$

$$H_{out} = \frac{\text{denominator}}{stride[0]} + 1 \qquad (3)$$

where denominator

$$= H_{in} + 2 \times padding[0] - dilation[0] \times (kenelSize[0] - 1) - 1$$

This is followed by the fully connected layers with sizes 36864, 1024, and 512. For the batch normalization [nn.BatchNorm2d [5]], the estimated momentum is 0.1, which can be computed by:

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} \times \gamma + \beta \qquad (4)$$

$$\begin{aligned} loss(x, class) \\ &= \log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) \\ &= -x[class] + \log\left(\sum_j \exp(x[j])\right) \end{aligned} \qquad (5)$$

$$m_t = \beta_1 \times v_{t-1} + (1 - \beta_1)(\nabla\omega_t) \qquad (6)$$

$$v_t = \beta_2 \times v_{t-1} + (1 - \beta_2)(\nabla\omega_t)^2 \qquad (7)$$

B. Dataset

The dataset from the TGCA common data portal is presented Table I. These sources are from open repository of GEO and TCGA:

- A list of the 33-tumor classes, and
- samples that contains 10267 tumors from 20,531 genes extracted from the data repository.

C. Training and Testing

The dataset has been preprocessed in order to obtain the input sample need for the training model. The gene sample were normalized and scaled to get the corresponding genes required for the sample without the noise. This technique uses the log transform

$$y = \log_2(x+1) \qquad (8)$$

for the scale reduction of the genes. The expression levels were examined to get the allowed levels for in relation to the features to be selected for the classification task. In order to make it more suitable to the training and testing phases, the genes were filtered and transformed for the 2D embedding of images. This transformation of the gene array was embedded in the 2D scaling it to 102×102 image that will be used for the CNN. It will be more effective to come up with the image for training and testing of the CNN and taking into account the classification task. To make it more meaningful, features are identified and classified according to the learning model and framework.

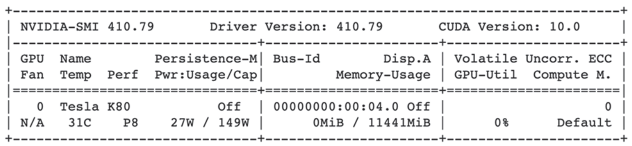
The size of the image 102×102 was obtained from the normalized genes, and then filtering out and ordering the genes related to each other so that the images in the training and testing to make sure of the range [0, 255]. The gene feature selection is to obtain the required features and labels to the index of the chromosome ID. This is then divided into k=10 folds for the sampling for training and test data. The features were transformed into the CNN.

The architecture to train and tests the CNN model designed with three convolution layers and three fully connected layers. The sample size is a small network with three layers. The configuration of the network has the following:

- convolutions 102×102×64, 50×50×128, and 24×24×256;
- max-pool of 50×50×64, 24×24×128, and 12×12×25;
- drop out was set to 25% rate;
- fully connected layers are 36864, 1024, 512, respectively;
- epochs observed for the optimal training of 500 samples was 200 times; and
- training and test were taken into 10-fold cross validation.

[5] https://pytorch.org/docs/stable/nn.html#batchnorm2d

The machine configuration is Pytorch framework with the required libraries running in the testing machine using the Google Cloud Platform (GCP) with the GPU Tesla K80. The local machine, Anaconda, is set up with all the libraries and Pytorch framework and libraries installed with configuration of iCore 7, 16MB memory, NVIDIA GeForce GT 750M 2GB, 500GN storage as system workbench. See Fig. 2 for a snapshot of the GPU specification.



```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.79       Driver Version: 410.79       CUDA Version: 10.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:04.0 Off |                    0 |
| N/A   31C    P8    27W / 149W |      0MiB / 11441MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
```

Fig. 2. GPU specification

IV. Evaluation

A. Heatmap and Validation

The principle of the using a method without a change in the neural network design is possible by using guided backpropagation and gradient-weighted class activation mapping (Grad-CAM) [15]. In this method, the input image (logit of the input images) in the end of the convolutional layer produces a map of the same image with the important salient spot (or section of the image) for predicting the gene in the CNN. In the experiment, we used this approach to interpret the coarse localization map for the calculated activation of the significant genes of the gene expression. Hence, we can visualize through the heatmap and the intensity produced that represents the score of each gene in the classification task.

The validation of the heatmap is based on the how the top genes were related to the specific types of cancer cohort in the TCGA. To prove this, we apply a functional analysis based on the top scoring genes of the heat map and take the top 400 genes of each type in the cohort for the corresponding biomarker relationship in the pathway analysis. Then, we capture only a representative of the first analysis—namely, the top 5 genes—for further investigate if the significant genes are related to the cohort of the cancer type.

B. Evaluation Results

The result of the classification task performance bearing the following evaluation metrics:

- *Precision* (aka positive predictive value):

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{9}$$

where

  - TP is true positive (aka hit), which means that a gene of actual cancer type *T* is correctly predicted and classified as cancer type *T* (e.g., (e.g., an ACC gene is correctly predicted and classified as cancer type ACC for adrenocortical carcinoma); and
  - FP is false positive (aka false alarm or Type-I error), which means that a gene *not* of actual cancer type *T* is incorrectly predicted and misclassified as cancer type *T* (e.g., a non-ACC gene—say, a UVM gene—for uveal melanoma is incorrectly predicted and misclassified as ACC).

In other words, precision measures the fraction of genes that are of actual cancer type *T* among those genes that are predicted and classified as cancer type *T*.

- *Recall* (aka hit rate, sensitivity, or true positive rate):

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{10}$$

where FN is false negative (aka miss or Type-II error), which means that a gene of actual cancer type *T* is *not* predicted or classified as cancer type *T* (e.g., an ACC gene is incorrectly predicted and classified as UVM instead of the correct cancer type ACC).

In other words, recall measures the fraction of genes that are predicted and classified as cancer type *T* among those genes that are actually of cancer type *T*.

- Accuracy:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{11}$$

where TN is true negative (aka correct rejection), which means that a gene *not* of actual cancer type *T* is *not* predicted or classified as cancer type *T* (e.g., a non-ACC gene is correctly predicted and classified not as cancer type ACC).

In other words, accuracy measures the fraction of correctly predicted and classified genes (i.e., a collection of (i) genes of actual cancer type *T* that are correctly predicted and classified as *T* and (ii) genes *not* of actual cancer type *T* that are correctly predicted and classified as a non-*T* cancer type) among all the genes.

- F_1-score (aka harmonic mean of precision and recall):

$$\text{F}_1 = \frac{2\ \text{Precision} \times \text{Recall}}{\text{Precision+Recall}} = \frac{2\text{TP}}{2\text{TP+FP+FN}} \tag{12}$$

In addition to the aforementioned four commonly used evaluation metrics, we also use the following metrics in our evaluation:

- Fallout (aka false positive rate (FPR)):

$$\text{Fallout} = \frac{\text{FP}}{\text{FP+TN}} \tag{13}$$

Fallout measures the fraction of incorrectly predicted and classified genes (i.e., genes *not* of actual cancer type *T* but incorrectly predicted and classified as *T*) among those genes that are actually *not* of cancer type *T*.

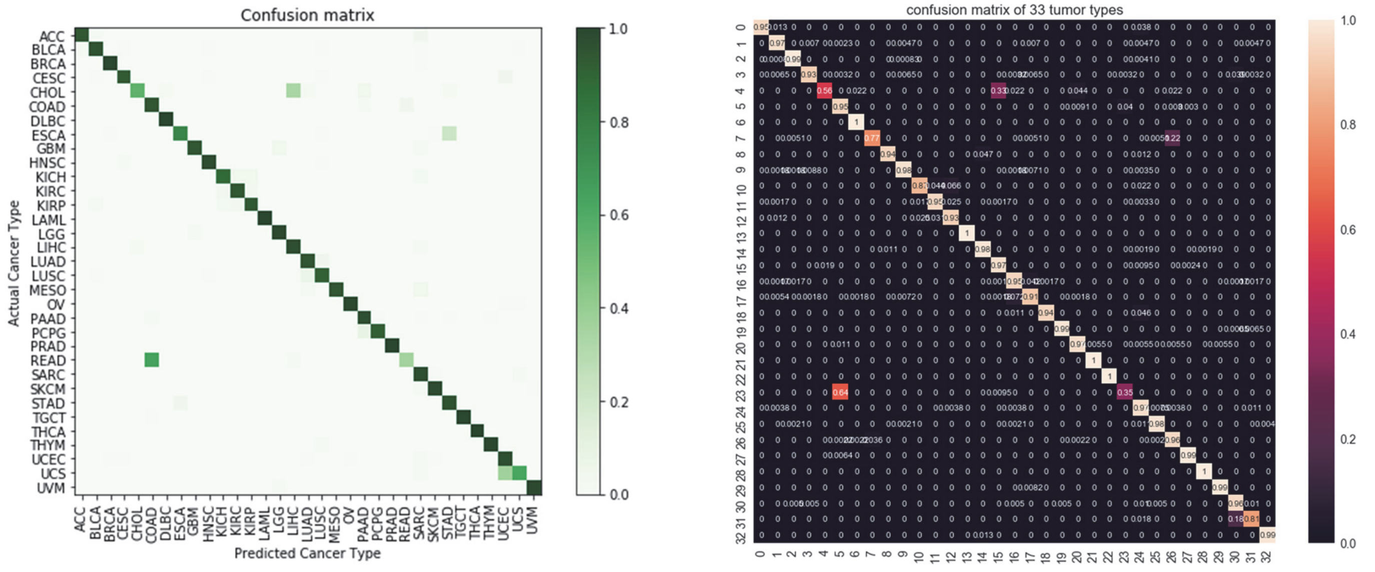


Fig. 3. Confusion matrix (a) classification performance and with (b) annotation of accuracy scores

- False discovery rate (FDR):

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP+TP}} \tag{14}$$

FDR measures the fraction of incorrectly predicted and classified genes (i.e., genes that are *not* of actual cancer type *T* but incorrectly predicted and classified as *T*) among all the genes that are predicted and classified as cancer type *T*.

- False negative rate (FNR, aka miss rate):

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN+TP}} \tag{15}$$

FNR measures the fraction of incorrectly predicted and classified genes (i.e., genes that are of actual cancer type *T* but incorrectly predicted and classified as non-*T*) among all the genes that are actually of cancer type *T*.

- False omission rate (FOR):

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN+TN}} \tag{16}$$

FOR measures the fraction of incorrectly predicted and classified genes (i.e., genes that are of actual cancer type *T* but incorrectly predicted and classified as non-*T*) among all the genes that are predicted and classified as of a non-*T* cancer type.

- Negative predictive value (NPV):

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN+FN}} \tag{17}$$

NPV measures the fraction of correctly predicted and classified genes (i.e., genes that are *not* of actual cancer type *T* and correctly predicted and classified as non-*T*) among all the genes that are predicted and classified as of a non-*T* cancer type.

- Selectivity (aka specificity or true negative rate):

$$\mathrm{Selectivity} = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \tag{18}$$

Selectivity measures the fraction of correctly predicted and classified genes (i.e., genes that are *not* of actual cancer type *T* and correctly predicted and classified as non-*T*) among all the genes that are *not* of actual cancer type *T*.

Using the significant genes in the evaluation that will validate the top genes to consider for the tumor specific and that the genes is corresponding to the type of tumor being classified. When classifying the 33 tumor classes, the confusion matrix in Fig. 3 reveals classes correctly classified and misclassified. Specifically, Fig. 3(a) shows the confusion matrix on relative *classification accuracy*. See Eq. (11). We visualize the confusion matrix using a heatmap. The darker green the cell in this 33×33 matrix, the higher is the accuracy (close to 1.0). Conversely, the lighter green the matrix cell, the lower is the accuracy (close to 0.0).

Similarly, Fig. 3(b) shows the confusion matrix on absolute classification accuracy scores ranging from 0.0 to 1.0 inclusive: [0.0, 1.0]. The lighter red the cell in this 33×33 matrix, the higher is the accuracy (close to 1.0). Conversely, the darker red the matrix cell, the lower is the accuracy (close to 0.0). For instance, the upper left corner having a light red color represents a high accuracy with an annotated accuracy value of 0.95. The lower right corner having a dark red color represents a low accuracy with an annotated accuracy value of 0.99. Cells along the diagonal represent TP (i.e., genes that are correctly predicted as their actual cancer types). Most cells lying off the diagonal are mostly with dark color, which represent low accuracy. They represent genes that are incorrectly predicted from their actual cancer types. Based on these two confusion matrices in Fig. 3, observant readers may notice the following:

- The accuracy is high (mostly above 92.3% accuracy over all 33 cancer types).
- The two less accurate predictions are for cancer types cholangiocarcinoma (CHOL) with accuracy=0.56 and rectum adenocarcinoma (READ) with accuracy=0.35. However, the accuracy of our proposed CNN is consistent with the baseline CNN [14], [16] as shown in Table I.
- Moreover, when ignoring cancer types CHOL and READ, accuracy over the remaining 31 cancer types is mostly above 95.4%.

The last three columns of Table I show the accuracy of (i) existing baseline CNN [14], [16] and (ii) existing genetic algorithm or *k*-nearest neighbors algorithm (GA/kNN) [17] when compared with our proposed CNN:

- As shown in Table I, our proposed CNN led to higher average accuracy than existing baseline CNN.
- The existing baseline CNN, in turn, led to higher average accuracy than existing GA/kNN.

In addition, we evaluated our proposed CNN with existing random forest (RF) and support vector machine (SVM). It was noted that, in other experiments, GA/kNN accuracy was reported [17] to have more than 90% accuracy over the 31 cancer types with less than 10,000 samples. Similarly, the cohort cancer class that were mostly misclassified are only the CHOL (0.56) and READ (0.35). The result of the accuracies reported in the experiment are listed in Table II with the proposed CNN and in comparison to related works [14], [17]. In the same experiment, where we run the SVM (0.945) and Random forest (0.925) with the accuracies for the classification task.

TABLE II. PERFORMANCE RESULT OF THE PROPOSED MODEL COMPARED FROM RELATED WORKS USING THE TGCA GENE EXPRESSION DATASET

Methods	Precision	Recall	Accuracy	F_1 score
GA/kNN [17]	-	-	90.00%	-
RF	92.01%	92.21%	92.50%	91.00%
SVM	94.10%	94.33%	94.50%	93.10%
Baseline CNN [14]	95.54%	95.59%	95.59%	94.43%
Our proposed CNN	95.55%	95.69%	**95.65%**	94.45%

The determination of the significant genes based on score was illustrated in the images after the training and testing by the generated the image maps. The intensity of each image indicates the significance of the genes from the total sample 10,267. The selected genes as implied to be significant in the plot of image based on scores from 400 top genes that is relevant for the biomarker indicator in reference with the other studies under this area [16], [18]–[22]. For the top genes plotted in the heatmap, we observed the difference in the intensities of the images from the 100th to 400th. Comparing to the size of the gene samples we had in the experiment (i.e., 10,381 samples), it can be reasonable that biomarkers can also be in small size. To determine the potential contribution of the top 400 genes considered, a functional analysis using gene functional classification tool[6] can be used to annotate the genes based on its functional similarity. It can be interpreted from the existing 75,000 terms from the 14 available functional annotations. Although this part of the task is to determine the relevance of the top genes identified in the classification task, another task of carefully inspecting the gene list from the functional toll is necessary. The generated functional analysis result needs to be examined with the relevant pathways of the cohort cancer types using parameters such as the *p*-values for the correlation with the significant genes pathways and its identified biomarkers.

V. CONCLUSIONS

In this paper, we presented a deep learning model using CNN and exploited the gene expression and classified cancer types from the several samples of gene expression extracted from TCGA gene data common portal. The CNN designed for the experiment for training and testing was 3-layers with the parameters set, which is shown to be appropriate for the transformed genes as input to the learning model. The classification result shows a comparable accuracy of 95.43% to the related works. The feature or gene selection technique is useful for the discovery of top related genes, and can also be useful to correlate with the biomarker identification through functional analysis.

For ongoing and future work, we plan to further improve the interpretation of “the black box” (i.e., the deep learning algorithm) on how to identify the genetic markers and its subtypes. The resulting improvement is expected to lead to a truthful presentation of the feature representation, a better explanations on the deep learning results, and a more trustworthy predictions.

ACKNOWLEDGMENT

This work is partially supported by the following organizations:

- Mapua University DRIVE institutional research,
- Commission on Higher Education - Canadian Bureau for International Education (CHED-CBIE) Graduate Studies Scholarship in Canada,
- Natural Sciences and Engineering Research Council (NSERC) of Canada, and
- University of Manitoba.

REFERENCES

[1] C. P. Stewart and B. W. Wild, “World cancer report 2014 - WHO - OMS,” IARC Nonserial Publ., p. 630, 2014.

[2] K. A. Hoadley et al., “Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer,” Cell 173(2), pp. 291-304.e6, 2018.

[3] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature 521, pp. 436–444, 2015.

[6] https://david.ncifcrf.gov/gene2gene.jsp

[4] J. Zou, M. Huss, A. Abid, P. Mohammadi, A. Torkamani, and A. Telenti, “A primer on deep learning in genomics,” Nat. Genet. 51(1), pp. 12–18, 2019.

[5] D. D. Miller and E. W. Brown, “Artificial intelligence in medical practice: the question to the answer?” Am. J. Med. 131(2), pp. 129–133, 2018.

[6] T. Yue and H. Wang, “Deep learning for genomics : a concise overview,” pp. 1–40, 2015.

[7] V. F. O. Teixeira, “Deep learning for genomic data analysis,” Integrated Master in Informatics and Computing Engineering (MIEIC) thesis, University of Porto, 2017.

[8] P. Chaudhari and H. Agarwal, “Progressive review towards deep learning techniques,” in ICDECT 2016, Vol. 1, pp. 151–158.

[9] M. Wainberg, D. Merico, A. Delong, and B. J. Frey, “Deep learning in biomedicine,” Nat. Biotechnol. 36(9), pp. 829–838, 2018.

[10] P. Guillen and J. Ebalunode, “Cancer classification based on microarray gene expression data using deep learning,” in CSCI 2016, pp. 1403–1405.

[11] Y. Guo, S. Liu, Z. Li, and X. Shang, “Towards the classification of cancer subtypes by using cascade deep forest model in gene expression data,” in IEEE BIBM 2017, pp. 1664–1669.

[12] T. Ahn, T. Goo, C. Lee, S.Kim, K. Han, S. Park, and T. Park, “Deep learning-based identification of cancer or normal tissue using gene expression data,” in IEEE BIBM 2018, pp. 1748–1752.

[13] Y. Kong and T. Yu, “A deep neural network model using random forest to extract feature representation for gene expression data classification,” Sci. Rep. 8(1), pp. 16477:1–16477:9, 2018.

[14] B. Lyu and A. Haque, “Deep learning based tumor type classification using gene expression data,” in ACM-BCB 2018, pp. 89–96.

[15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: visual explanations from deep networks via gradient-based localization,” in IEEE ICCV 2017, pp. 618-626.

[16] E. Manzanarez-Ozuna, D.-L. Flores, E. Gutiérrez-López, D. Cervantes, and P. Juárez, “Model based on GA and DNN for prediction of mRNA-Smad7 expression regulated by miRNAs in breast cancer,” Theor. Biol. Med. Model. 15(1), pp. 24:1-24:12, 2018.

[17] Y. Li, K. Kang, J. M. Krahn, N. Croutwater, K. Lee, D. M. Umbach, and L. Li, “A comprehensive genomic Pan-cancer classification using The Cancer Genome Atlas gene expression data,” BMC Genomics 18(1), pp. 508:1–508:13, 2017.

[18] S. Qu, X. Yang, X. Li, J. Wang, Y. Gao, R. Shang, W. Sun, K. Dou, and H. Li, “Circular RNA: a new star of noncoding RNAs,” Cancer Lett. 365(2), pp. 141–148, 2015.

[19] Q. Liao, Y. Ding, Z. L. Jiang, X. Wang, C. Zhang, and Q. Zhang, “Multi-task deep convolutional neural network for cancer diagnosis,” Neurocomputing 348, pp. 66-73, 2018.

[20] Y. Guo, X. Shang, and Z. Li, “Identification of cancer subtypes by integrating multiple types of transcriptomics data with deep learning in breast cancer,” Neurocomputing 324, pp. 20–30, 2019.

[21] C. A. Schnabel and M. G. Erlander, “Gene expression-based diagnostics for molecular cancer classification of difficult to diagnose tumors,” Expert Opin. Med. Diagn. 6(5). pp. 407–419, 2012.

[22] Y. Yuan, Y. Shi, C. Li, J. Kim, W. Cai, Z. Han, and D. D. Feng, “DeepGene: an advanced cancer type classifier based on deep learning and somatic point mutations,” BMC Bioinformatics 17(Suppl 17), 476:1–476:14, 2016.