JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 07, ISSUE 01, 2020

Deep CNN Model for Multiclass Classification of Human Protein Atlas Images

Juttu Suresh[1], Hima Bindu Kunchanapalli[1], A. Poornima[1]

[1]*Assistant Professor, Department of Information Technology*

[1]*Malla Reddy Engineering College and Management Sciences, Medchal, Hyderabad, Telangana, India.*

Abstract

Protein, which serves as a service provider for the cell to exert specific function, is ubiquitously distributed in a biological system. After the transcription of genomic information, protein starts to form as the product of RNA translation. Therefore, in order to facilitate medical research in molecular biology, we need to develop complete and accurate models to identify and classify organelles in cells from a large number of images produced by microscopy. But the existing machine learning architectures are failed to provide the highest classification accuracy. Thus, this work majorly focused on implementation of Human Protein Atlas Image Classification using Deep learning Convolutional Neural Networks (DLCNN). The simulation results show that the propose method gives the highest classification accuracy compared to the state of art approaches.

Keywords: Human protein atlas image, Image classification, machine learning, Convolutional neural networks.

1 Introduction

As the executor of genomic information, protein enables cells to transmit and receive information from outside and behave accordingly [1]; then, it is the coordination of cells that maintains the behavior and operation of the biological system. In order to better understand how a cell work as well as the function related to its organelles, it is significant to classify the protein structure of a cell's organelles and find the occurrence of different organelles among cells. In the past, the detection and classification of protein is far from ideal in that people can only classify single pattern in one or few cells [2]. To fully understand the complexity of cell structure and function, a way to categorize mixed patterns for different range of human cells is in need. Currently, due to the advance development of biological imaging, we are able to produce images across cells with different channels of features using high through-put confocal microscopy [3]. Therefore, in order to facilitate medical research in molecular biology, we need to develop complete and accurate models to identify and classify organelles in cells from a large number of images produced by microscopy. For our part, we decided to use convolution neural network as the core to build our model [4].

The rising volume of imaging data obtained in medical research is a considerable database of unexploited medical-relevant information; images are collected in millions annually worldwide. Clinicians struggle under diagnostic strain and monitoring such immense quantities of images [5]. This concept has given rise to a multitude of methods for enhancing and assisting clinicians with effective search functionality. In recent years Protein Classification technique of image is major complicated issue in the image processing area and is a significant research field [6]. Methods of X-ray Protein Classification recently received significant attention. Protein Classification [7] of X-rays is difficult because of complexity. These may also influence by noise, sampling artifacts or spatial aliasing to differentiate or separate the boundaries [8]. On the classification of the medical images with more effective algorithms, it can be implemented in an effective AI model that is developed to predict the various types of diseases with greater accuracy compared to the radiologist in a short

ISSN- 2394-5125 VOL 07, ISSUE 01, 2020

period of time. The AI-enabled machines also help to annotate large-scale images and train more such models to improve and incorporate the automated process of interpreting medical imaging in hospitals and medical centers to aid both doctors and patients get quicker and more accessible treatments and medical care [9].

2. Proposed Method

2.1 Pretrained model

Pretrained model is a model that has been trained on some kind of data. Although it may not perform well on a specific task, it can be a good corner stone for a model that is aimed for a similar task. In our case, we want to classify the pattern of protein, which is a task that can be categorized into general image classification. Therefore, using a model pre-trained on millions of images as a base model will be a better choice rather than training a new model from scratches. However, in our case, there is one drawback for using pretrained model: The input shape is fixed on 3 channels (RGB). But our data includes four channels. Thus, it is a challenge for us to find a solution to merge information from 4 channels into 3.

2.2 Convolution network

Convolution network is used to extract features from 2-D images. The basic principle behind it is to apply filters on the matrix to achieve a certain image processing like blur or edge detection. This process is called hierarchical feature learning, which is similar to how brain identifies the objects. Thanks to the existing libraries, we are not required to produce filters by ourselves but only to decide the number and size of filters. There is a figure below displaying the function of convolution layer. The extracted features will later be used to calculate the weights for each neuron.

2.3 Batch normalization

Generally, what normalization do is to adjust and scale the input from a much larger range to [0,1]. The reason why we need this normalization layer is that by normalizing the inputs, the amount by what the hidden unit values shift around will reduce, which means less time for the model to reach the optimal parameter. Batch normalization can also help the model to increase resistance for overfitting because it has a slight regularization effect.

2.4 Dropout and dense

Dropout and dense layers play a similar role in the neural network, which is to decrease the size of parameters. But the principles behind are quite different. As for dropout layer, it randomly set a fraction of input units to 0 in order to prevent overfitting. The reason why this works is that it helps the model to reduce the dependencies on a certain unit. Compared to dropout layer, what dense layer do is much simpler. It just connects several neurons on one layer to one neuron on the next layer to reduce dimensions. One thing is noticeable in this case is that the final layer before the output should includes an activation function called sigmoid, which is shown below. It is especially useful for models to predict the probability since its output range is [0,1].

2.5 Network Architecture

In order to get a relatively accurate classification, we applied DLCNN to get better prediction. Generally, CNN contains input layer, hidden layer in middle and output layer, just like the Figure 1 shown below.

We have training data, the confocal microscopy coming from 4 channels (R: microtubule, G: protein, B: nucleus, Y: endoplastic reticulum) as our input; then, the hidden layer is the neural network which we will discuss afterwards. The output should be a 28-element vector, with each component reflecting

ISSN- 2394-5125 VOL 07, ISSUE 01, 2020

the probability of existence of a certain protein class. Our input are images, from which we cannot easily extract features by common operations and update weights based on f1 score. Fortunately, there are deep learning libraries at hand, like Tensor-flow and Keras, which help us save the troubles. And I will list several methods we tried during our experiment.

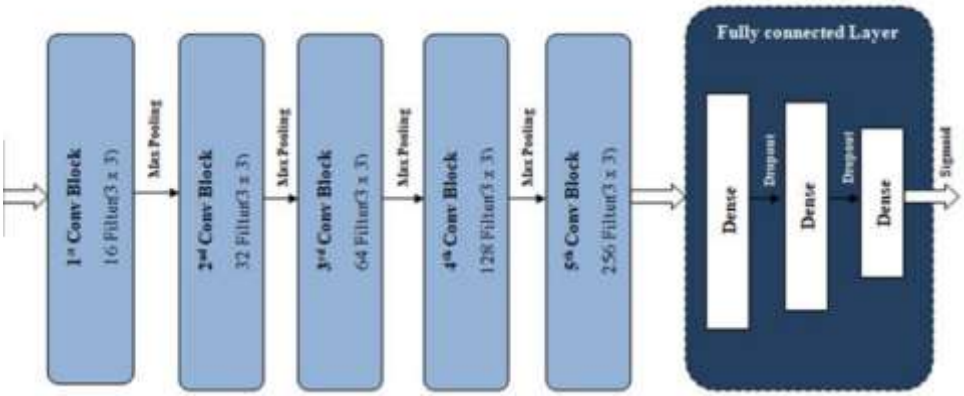

Figure 1: Structure of DLCNN.

The networking structure is light to avoid over fitting of data. The convolution neural network structure is explained as follows: A colored RGB image is an input to the network with size 3X200x200, then the image is resized to 64X64 and then the image is reshaped to 64X64X1. Now we have converted our images to gray scale images. As a first step, we have performed preprocessing during which we have divided age into four categories, and if any of the ages contain missing values we are calculating the mean value and filling the field. Similarly, for Protein preprocessing we have assigned 0 or 1, if a particular image does not have Protein representation then it will be assigned standard deviation values. After performing the preprocessing, the data set is divided into a training set and a test set. Here we have taken 70 percentages of data for training and 30 percentages of data for testing by shuffling the data set. The neural network consists of 5 convolution layers followed by 3 fully connected layers. First, we defined the model for Protein then followed by the Atlas Image Classification model.

Corresponding The 5 convolution neural network layers are defined as follows:

- Convolution layer-1 is taken the input of size 64X64X1 and is defined with 32 filters, 3X3 kernel size, and ReLU is used as the activation function. Then the max-pooling layer takes 3X3 data and converts into 2X2 data and scales down the image and uses an LRN layer.
- Convolution layer-2 takes input as output produced by a max poll layer of size 31X31X32 and is defined with 64 filters, 3X3 kernel size, and ReLU is used as the activation function. Then the max-pooling layer takes 3X3 data and converts into 2X2 data and scales down the image and uses an LRN layer.
- Convolution layer-3 takes input as output produced by a max poll layer of size 14X15X64 and is defined with 128 filters, 3X3 kernel size, and ReLU is used as the activation function. Then the max-pooling layer takes 3X3 data and converts into 2X2 data and scales down the image and uses an LRN layer.
- Convolution layer-4 of size 6X6X128 is given to the dropout function to prevent the neural network from the over fitting problem. The output from the dropout function is given to the flatten layer which performs the reshaping operation producing the number of elements in the tensor.

ISSN- 2394-5125 VOL 07, ISSUE 01, 2020

- Convolution layer-5 is defined with 256 neurons fully connected to the output from the flattening layer of size 4608, ReLU as activation function, and dropout layer. Convolution layer-5 is defined with 128, 64 and 32 neurons fully connected to the output from the dropout layer, ReLU as activation function, and dropout layer

The Neural network model for Protein prediction:

- Dense layer-1 is defined with 16 neurons fully connected to the output from the dropout layer, ReLU as activation function, and dropout layer.
- Dense layer-2 is defined with 8 neurons fully connected to the output from the dropout layer, ReLU as activation function, and dropout layer.
- Dense layer-3 is defined with 1 neuron fully connected to the output from the dropout layer, Sigmoid as activation function as it needs to classify Protein as Atlas Image Classification.

3. Simulation Results

We have built two models based on kernels in our experiments, which are different in their structures. Now, let's analyze these two models and their results.

3.1 Training and validation

Generally, when we train our model with data, we will not use the whole dataset, because it may expose our model to the risk of overfitting. A wiser way would be splitting a fraction of data from original dataset, which is called validation data, and use the rest to train the model. Then we can use the validation dataset to evaluate the model skill. Since the model is independent of validation data, it can serve as an unbiased evaluation criterion for the model. This is typically called a train-test split approach to algorithm evaluation. Now, we have all the methods needed to build a model. The next step will be to train the model with input data sets and produce predictions, which will be shown in the next section.

3.2 CNN with pretrained model

With the help of pretrained model of InceptionResNetV2, we no longer need to construct very complex network structures to extract features. However, it is mentioned before that one limitation of using pretrained model is that it requires inputs only from 3 channels. To settle this conflict, the solution original kernel gives is to compress 4 channels into 3 by:

Channel 1 = R/2+ Y/2

Channel 2 = G/2 +Y/2

Channel 3 = B

RGBY means data from 4 channels respectively. However, since green channel contains most information about the protein location, we decide to adjust the weights of each channel to:

Channel 1 = R/2+ B/2

Channel 2 = Y/2+ B/2

Channel 3 = G

in the hope that the model can better predict the pattern of protein in the context of other organelles. We also improve the learning efficiency of the model by adding a callback, a function that is able to adjust the learning rate of model. In the gradient descent, learning rate defines the step size of learning. It is a good way to adjust learning rate in this process to prevent the model from getting

ISSN- 2394-5125 VOL 07, ISSUE 01, 2020

stuck at a local minimum. What's more, in the original kernel, the learning curve is not very smooth and carries lots of small fluctuations, which is showed in Figure 2.

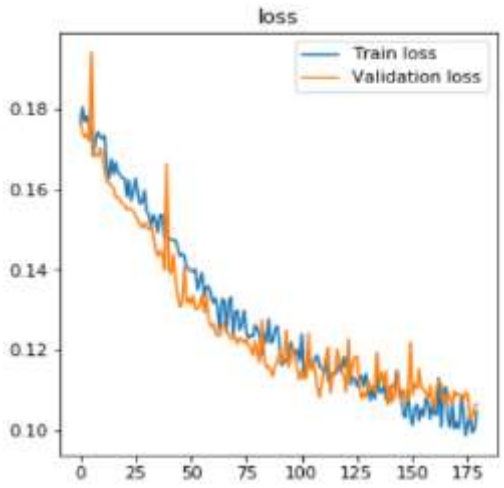

Figure 2: Learning curve of original kernel

It is shown that the parameters have experienced many updates, but the quality of gradient descent is not so well. Therefore, we decide to increase the batch size in order to improve the efficiency of learning. With the assistance of adaptive learning rate, the model can reach the optimum value very quickly as shown in Figure 3.

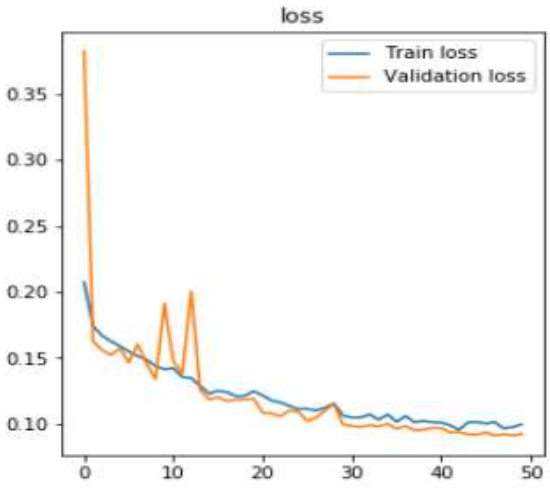

Figure 3: Learning curve of our model

As for prediction, we need a probability threshold to determine whether a certain protein class exists. In original kernel, the threshold is fixed to be 0.2 for all classes. However, the probability mass function for each class in training dataset is not equal, which may lead to the case that the minor class will never appear in our predictions. To fix this, we decide to calculate thresholds for each class using validation data with f1 score as evaluation criterion. By doing these things, our model gets 0.945 on the leader board as the best score, which is an improvement about 28% compared to 0.669 from the original kernel.

4.3 CNN with GAP layer

GAP is shortened from global average pooling, which is used to reduce spatial dimensions of a three-dimensional tensor:

ISSN- 2394-5125 VOL 07, ISSUE 01, 2020

$$h \times w \times d \rightarrow 1 \times 1 \times d$$

The benefits of using global average pooling is that it is able to achieve object localization as well as classification, meaning that the neural network can learn different distribution patterns from training data. This is exactly what this task requires us to do. The original kernel based on this GAP layer as well as other components forms a much more complex network than that of the other kernel. We did not change structure of the kernel a lot, but we found that the final result is slightly overfitted as shown in Figure 4.

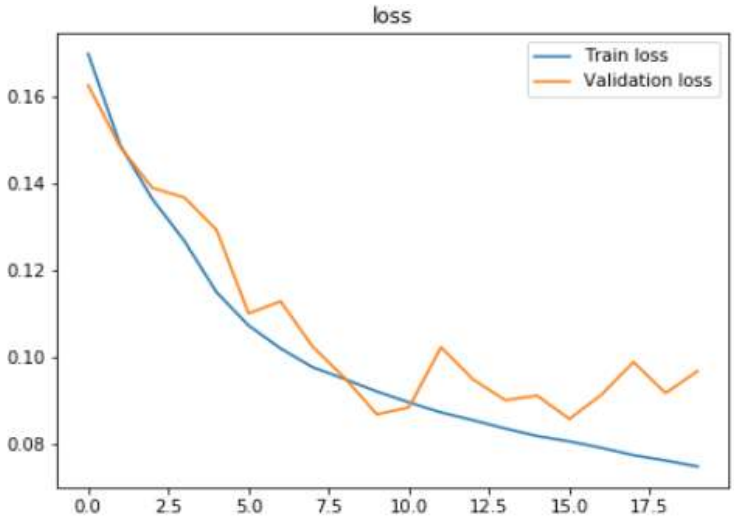

Figure 4: Learning curve of original kernel

So, we decide to increase the validation ratio from 0.1 to 0.2, which means that there is more validation date available to determine the real performance of our model as shown in Figure 5.

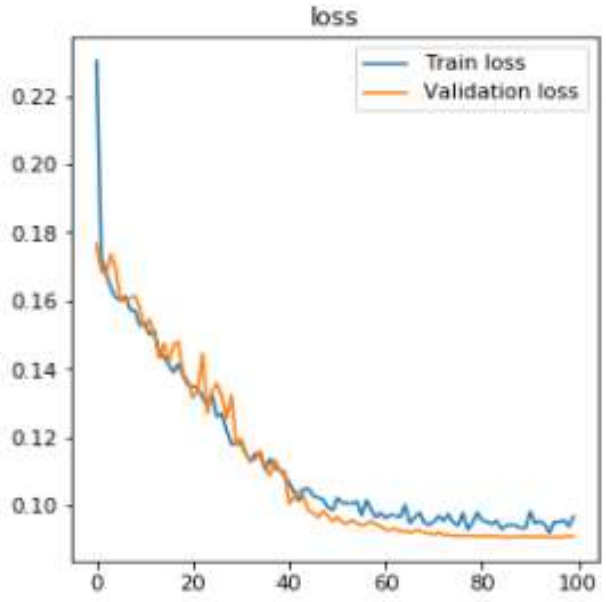

Figure 5: Learning curve of our model

What's more, like what we did on the other kernel, we add a callback to adjust the learning rate and change the number of epochs from 20 to 100; Hopefully we can obtain more information about the learning process and also prevent the model from getting stuck into a local minimum. As for prediction, we found that the original calculator for the threshold of each class may be in troubles

ISSN- 2394-5125 VOL 07, ISSUE 01, 2020

when none of the validation data contains a certain protein class. In this case, the threshold for this protein class will be 0, which is extremely terrible, because this means that every sample in our prediction will also contain this protein class. Therefore, we add a small intercept: 1e-4 to all the thresholds in order to fix this small bug. By doing these things above, our model gets 0.422 on the leader board as the best score, which is an improvement about 10% compared to 0.385 from the original kernel.

Table 1: performance comparison

METHOD	Accuracy	Specificity	Recall	Precision	F1-score
SVM [7]	87	82	92	83	87.26
HMM [9]	91	90	93	89.5	91.47
RNN [8]	97.49	93.6	94.3	95.6	94.4
DLCNN	98.33	98.61	98.93	97.73	97.49

4. Conclusion

Automatic diagnosis of Protein Atlas problem identification and classification is more helpful and it takes less time for diagnosis in addition. This paper presents the implementation of DLCNN based approach for Protein Atlas classification. Pre-processing methods like contrast enhancement, illumination correction and artifact removal techniques are useful for improving the image quality and can obtain more robust generalization ability. Imbalance class problems are solved by data augmentation technique. This increase the scale of the training set and also it reducing the overfitting problem. In the traditional approach, feature selection process is that the time consuming and also selection of relevant features is most vital. The DLCNN deep learning algorithm learns the features automatically and efficiently, specifically in the feature extraction part, DLCNN selects the filters intelligently as compared with manually. DLCNN based approach needs quite big dataset for training, so it can learn the features effectively as compare with the normal way of Protein Atlas classification. In future to increase accuracy of skin disease prediction we proposed Generative Adversarial Networks with Pre-trained models like AlexNet, VGG16, VGG19, Inceptionv3, ResNet and etc are trained on very large dataset with several general images and it are often used with transfer learning or fine tuning.

References

[1]. Ouyang, W. & Zimmer, C. Te imaging tsunami: computational opportunities and challenges. Curr. Opin. Syst. Biol. 4, 105–113 (2017).

[2]. 0. Bojarski, M. et al. End to end learning for self-driving cars. Preprint at https://arxiv.org/abs/1604.07316 (2016).

[3]. . Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. Preprint at https://arxiv.org/abs/1409.1556 (2014).

[4]. Szegedy, C., Vanhoucke, V., Iofe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. in IEEE Conference on Computer Vision and Pattern Recognition 2818–2826 (IEEE, 2016).

[5]. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. in Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015 (eds Navab, N. et al.) 234–241 (Springer, 2015).

ISSN- 2394-5125 VOL 07, ISSUE 01, 2020

[6]. Hestness, J. et al. Deep learning scaling is predictable, empirically. Preprint at https://arxiv.org/abs/1712.00409 (2017)

[7]. . Moen, E. et al. Deep learning for cellular image analysis. Nat. Methods https://doi.org/10.1038/s41592-019-0403-1 (2019).

[8]. Godinez, W. J., Hossain, I., Lazic, S. E., Davies, J. W. & Zhang, X. A multi-scale convolutional neural network for phenotyping high-content cellular images. Bioinforma. Oxf. Engl. 33, 2010–2019 (2017).

[9]. Hofmarcher, M., Rumetshofer, E., Clevert, D.-A., Hochreiter, S. & Klambauer, G. accurate prediction of biological assays with high-throughput microscopy images and convolutional networks. J. Chem. Inf. Model. 59, 1163–1171 (2019).