ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/329565281

[Regular Paper] EP-CapsNet: Extending Capsule Network with Inception Module for Electrophoresis Binary Classification

Conference Paper · October 2018

DOI: 10.1109/BIBE.2018.00071

CITATIONS
0

READS
110

4 authors, including:

Murtaza Ashraf
Korea Advanced Institute of Science and Technology
4 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Mun Y. Yi
Korea Advanced Institute of Science and Technology
81 PUBLICATIONS **3,909** CITATIONS
SEE PROFILE

All content following this page was uploaded by Murtaza Ashraf on 16 April 2019.

The user has requested enhancement of the downloaded file.

EP-CapsNet: Extending Capsule Network with Inception Module for Electrophoresis Binary Classification

Elizabeth Tobing, Ashraf Murtaza, Keejun Han, Mun Y. Yi
Graduate School of Knowledge Service Engineering
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, South Korea
{*elizabeth, murtaza, keejun.han, munyi*}*@kaist.ac.kr*

***Abstract*—Electrophoresis (EP) test separates protein components based on their density. Patterns exhibited by this test mostly show very close approximation, making it difficult to examine test results within a short amount of time as it has many variations of patterns and requires a significant amount of knowledge to discern them accurately. To help clinical examiners save time and produce consistent results, a new deep-learning model optimized for EP graphic images was developed. Extending recent work on capsule network, which is a state-of-the-art deep learning model, this study was carried out to develop a best-performing model in classifying abnormal and normal electrophoresis patterns. Instead of extracting features from the image, we used the whole slide image as an input to the classifier. This study used 39,484 electrophoresis 2D graph images and utilized capsule network as the foundation of the deep learning architecture to learn the images without data augmentation. The formulated models were compared for a multitude of performance metrics including accuracy, sensitivity, and specificity. Overall, the study results show that our proposed architecture EP-CapsNet, which combines capsule network with Google's inception module, is the best performing model, outperforming the baseline and alternative models in almost all comparisons.**

***Keywords*-Electrophoresis, Medical Image, Classification, Deep Learning, Neural Networks, Capsule Network.**

I. Introduction

Electrophoresis is an inexpensive and widely used laboratory technique to examine globulins or specific serum proteins [1]. It is also used to identify serum protein disorders based on the globulin composition. Based on the composition of each serum, the clinical abnormality can be detected, which can be a sign of various disorders such as monoclonal gammopathy, oligoclonal band, and nephrotic syndrome [1].

Electrophoresis may be defined as the separation of charged particles under the influence of electric field [2]. Each fraction in the specimen can be separated by making use of electrophoresis scanner where the electrical charge that is applied causes fractions to move towards the respective electrode, leaving density information. Then, the density can be mapped into graphical data and numerical value. In general, both the numerical and graphical data are used by clinical practitioners to examine and conclude the findings of electrophoresis. However, this work process is very tedious and in need of many resources to do the whole process. In between the sub-processes, there is also some manual processes such as cutting the additional region of the graphical image, which requires precision to have the uniform size of the resulting image exacted. The process of electrophoresis has various graph interpretations, requiring a significant amount of knowledge to examine all kinds of cases and minimize misconclusions [1]. The exploitation of deep learning can contribute to minimizing the manual processes, lessening the processing time, and reducing the number of labors, allowing doctors to have a less cognitive burden in analyzing the results and more time to re-examine abnormal results to decide subsequent treatments.

Deep learning has been phenomenal these days due to its compatibility for implementation in various fields. Many researchers are adopting deep learning in various areas including biomedical imaging. One of the well known deep learning architectures for image processing is Convolutional Neural Network (CNN). CNN can be implemented for image processing in the medical field, which includes classification, detection, and segmentation of pathological images. CNN can detect features without any positional or orientation information called ”translational invariance” as it can detect the translated features, because of the pooling layer, which is one of the essential ideas of CNN [3]. Translational invariance could be achieved by using image augmentation so the network can learn from a different point of views. CNN also requires an enormous amount of data to train a model for good prediction. However, in practice, it is often not feasible to acquire a large data set, especially in the bioinformatics field.

Sabour et al. [4] published a new deep learning architecture called Capsule Network (CapsNet). The architecture introduced a new representative output called the pose vector, which means the output of the layer is not scalar but a vector, which encodes more positional and information about a feature. Another critical point is that the pooling layer is not employed in this architecture. The capability

2471-7819/18/$31.00 ©2018 IEEE
DOI 10.1109/BIBE.2018.00071

of capsule network to achieve decent classification results using the MNIST dataset without using pooling layer and data augmentation gave more reasons to implement this architecture.

There are only a limited number of studies about the binary classification of electrophoresis. In particular, there has not been any study that exploited the whole image of electrophoresis, as opposed to cutting the image into fractions, on deep learning. Also, no study has applied the capsule network to electrophoresis. This study extends the capsule network by exploring the possibility of combining it with Google's inception module [5]. The proposed model is called EP-CapsNet, and it is being compared with state of the art deep learning designs, the Inception-V3 [5] and the original CapsNet [4] architectures.

This study tests the performance of EP-CapsNet architecture in the context of electrophoresis graphical data classification. The EP-CapsNet was compared with the original capsule network and the Inception-V3 architecture in similar hyper-parameters settings. In addition to those two image input models, a multilayer perceptron (MLP) and two other popular machine learning techniques, support vector machine (SVM) and random forest, were employed as comparison conditions for the assessment of the performance of the proposed architecture. The comparisons were made by using multiple metrics - sensitivity, specificity, balanced classification rate, and Matthew's Correlation Coefficient as well as accuracy, to examine their performances from multiple perspectives.

II. Methods

A. Capsule Network

Although CNN has been a robust method to achieve good performance, the network has some deficiencies such as the needs of data augmentation to recognize the same features in different spatial locations and the loss of information as it uses the pooling layer, which only returns one value as the representative of the specified region. Different from CNN, Capsule Network (CapsNet) utilizes a capsule as a group of neurons, which learn and recognize the visual entity as vector output while the vector length represents the probability of the presence of the entity [4].

The structure of CapsNet consists of an input layer, a convolution layer, a primary capsule, a routing capsule, and a decoder layer. After the input layer, a convolution layer with 9 x 9 kernel size, the stride of one, 256 filters, and no padding parameters setting were used for the first layer. Afterward, the input is taken into the primary capsule, which contains a convolution layer of 9 x 9 kernel size, the stride of two, 256 filters, and no padding settings. Therefore, the output is reshaped into 32 filters, and each has eight dimensions, thereby having a total of 32 x 8 x 6 x 6 = 1,152 capsules. At the end of primary capsule layer, a squash function is used for the activation of the output vector, producing an output vector v_j as shown in 1. Squash function ensures that the length of the vector to be an estimated probability that has a value between 0 and 1. The prediction vector later was used in the routing-by-agreement mechanism, which sends the output from the primary capsule to the appropriates routing capsules of a higher-level entity. The iteration of routing refines the agreement between two capsules. The final output of the routing capsule is a squashed form of its output.

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|} \tag{1}$$

The output from the digit capsule was used to compute the margin loss function, L_k. L_k in 2 is a sum of losses of all digit capsule [4]. The equation is composed of $m+$ as the minimum probability of true prediction, $m-$ as the minimum probability of false prediction, lambda λ as the hyper-parameter for this equation, and square hinged loss.

$$L_k = T_k \max(0, m^+ - \|\mathbf{v}_k\|)^2 + \lambda(1-T_k)\max(0, \|\mathbf{v}_k\| - m^-)^2 \tag{2}$$

B. EP-CapsNet

The primary objective of this research is to utilize CapsNet and adjust the network for electrophoresis images in order to achieve optimum performance. As mentioned previously, the original CapsNet was trained on MNIST dataset and attained state-of-the-art performance. Moreover, it also overcame the limitation of CNN in classifying the images based on detected features by replacing pooling layer with routing-by-agreement and the scalar output of a layer with vector output, which is called capsule. Hence, the vital intuition behind using CapsNet as the base architecture to build an architecture for electrophoresis is that it can accommodate multiple proteins classification and observe their relevance with each other by making use of capsules instead of using general CNN.

Further, EP-CapsNet utilizes the inception layer to enhance the performance of the network. Hence, this study incorporates the inception module into the capsule network to improve the ability to learn features by making use of filters of different sizes in a single layer. Inception modules have been proven to enhance the network ability because of the multiple learned features from various convolutions, allowing effective multi-level feature extraction in different dimensions in parallel [5]. The inception module, despite the number of layers inside the module, is still efficient because of the dimension reduction by 1 x 1 convolution layer. For these reasons, the inception module was employed to explore the possibility of improving network performance for electrophoresis binary classification. Only one layer of inception module was added in this experiment to see the improvement, and it was added to the base CapsNet architecture. Incorporating the inception layer into the network may seem to make some parameters of the network more

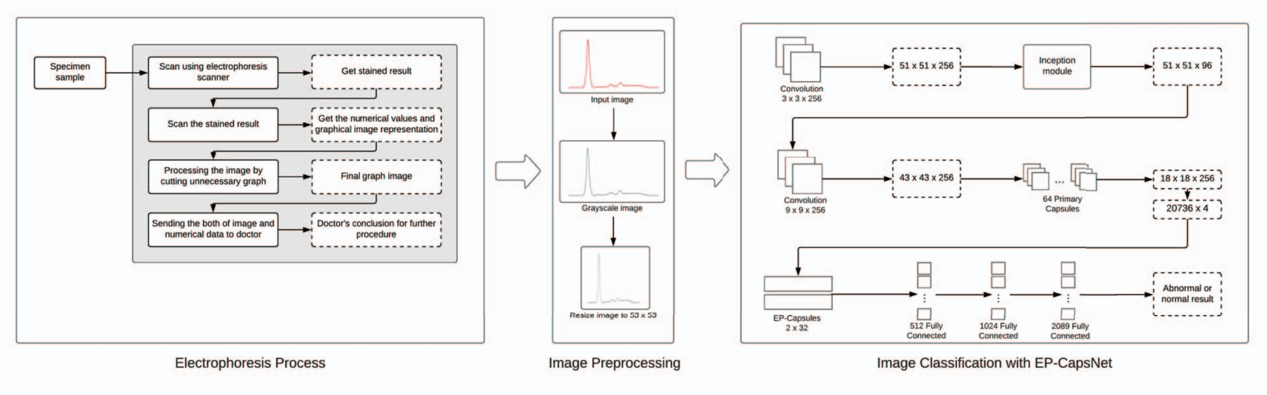

Figure 1: The process of generating electrophoresis image by clinical examiners, the preprocessing image, and the classification to detect the abnormality of electrophoresis graph, which is done by EP-CapsNet, the proposed model.

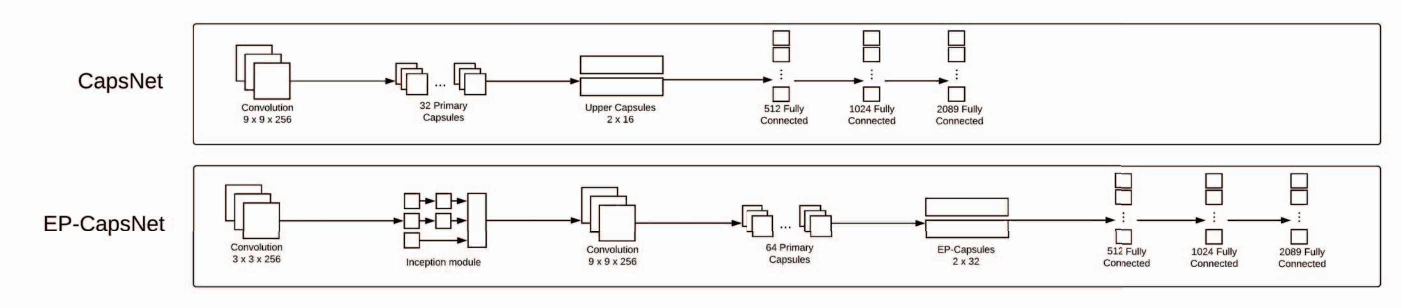

Figure 2: The comparison of original CapsNet and EP-CapsNet where EP-CapsNet utilized inception module to enrich the data input to primary capsule, which is improved the general performance classification result.

massive than the original architecture. However, the number of parameter of EP-CapsNet was proved smaller than original CapsNet, 116,441,764 and 144,679,172, showing the efficiency of the network. The addition of the inception module compressed the network's parameter. As a result, the number was reduced in EP-CapsNet architecture.

The inception module is described in Figure 2. The output from the previous layer will be an input to the inception module. The first 1 x 1 convolutions simultaneously process the input. The first 1 x 1 convolution output will send the output to the 3 x 3 convolution layer, while the second one will send the output to two 3 x 3 convolution layers. Later, the output from third 1 x 1 convolution, one 3 x 3 convolution, and two 3 x 3 convolutions will be concatenated as an output of the inception module.

EP-CapsNet had the input layer to receive the 53 x 53 input images. The second layer of the architecture is the 3 x 3 convolution layer with 256 filters, stride 1, and no padding setting. This layer produced 51 x 51 x 256 outputs. The next layer was the proposed inception layer. Different from the original inception layer from Inception-V3 architecture, the pooling layer was removed for this architecture as it has the opposite work with the capsule based architecture. The output of inception would have the same width and height as the input, but a different number of filters as the output was the concatenation of multiple layers. After the inception layer, another convolution layer was used to extract more features in broader dimensions, so 9 x 9 convolution layer with 256 filters, no padding setting, and stride one were applied. The next layer was the primary capsule. In the primary capsule layer, another 9 x 9 with 256 filters was used to extract smoother features. Instead of using stride 1, this layer incorporated stride 2 to reduce the dimensions of the output. Subsequently, 256 filters from the input were divided into 64 filters, where each capsule filter has four filters, so the output of this layer is 28,224 capsules with four filters. The primary capsule layer sends the output to the upper capsule, which is called EP Capsule in this architecture. The EP Capsules had 32 x 2 dimensions based on the routing algorithm from the original CapsNet. This layer would give the final class probability of the input. Although the classification result can be achieved from the EP Capsules layer, three fully connected layers are used to reconstruct the input. The differences between input and the reconstructed image were used as the regulation of the loss function in Equation 2. This regulation is significantly essential to keep the model from over-fitting.

Table I: Electrophoresis data statistics.

Dataset	Data Type	Normal	Abnormal
00307	CK isoenzyme serum	2,602 (68%)	1,203 (32%)
00309	Protein EP serum	5,945 (30%)	13,886 (70%)
00312	LDH isoenzyme serum	704 (29%)	1,690 (71%)
00323	Protein EP random urine	4,722 (34%)	9,182 (66%)

III. Experiment

A. Dataset

This study used datasets provided by Seegene, a Medical Foundation, which is a non-profit organization set up to examine patients' data for various disease symptoms. There were four types of datasets, which are 00307, 00309, 00312, and 00323. 00307 is a CK (Creatine Kinase) isoenzyme serum data; 00309 is a serum protein electrophoresis data; 00312 is a graph data for LDH isoenzyme serum; Finally, 00323 is a random urine protein electrophoresis data. All of the data were serum based patterns except the 00323 data, which is based on random urine. Each data consists of albumin, alpha-1, alpha-2, beta, and gamma fractions except for 00307 and 00312 datasets. The datasets are described in Table I.

B. Data Preprocessing

This study used 60% of the data for training and the rest for validation and testing. The data were divided by applying the specified percentage to each of the classes. Collected images have RGB colors with 670 x 296 dimensions for all of the images. Instead of the RGB color system, the images were converted into gray-scale images for reduced complexity. For the deep-learning classifiers, the data were further resized into 53 x 53 pixels and used for training, validation, and testing. For vector input classifiers, the two kinds of image sizes were used, the original (670 x 296) and the resized ones. Every image was transformed into a vector of numbers by extracting the average of every non-white pixel in a specific y axis, which ranges from 0 to 1, at each point of x.

C. Experiment Phrase

The proposed architecture was built using the Tensorflow library and Python. It was run on NVIDIA GPU Tesla-V100-SXM2 16GB. The experiment was run for approximately two to three hours for each run and was run five times to take into account the randomization in the network and GPU. Several experiments were run for each of the four datasets separately; 00307, 00309, 00312, and 00323. All of the models used the same data for training, validation, and testing. For all models, the data were divided into 3 distinct groups of data, where each data only belongs to one group; 60% training data, 20% validation data, and 20% testing data. Training data were used to build the model, where validation data were used to validate the model performance. At the end of the experiment, the performance of testing data was run and the result was used in this paper.

For the original capsule network and the proposed architecture, EP-CapsNet, the hyper-parameters of the network were set to 0.0001 learning rate, 64 batch size, two routing iterations, 100 epochs, $m+$=0.9, $m-$=0.1, λ=0.5, and α=0.0005. Different from the original capsule network, this study chose to lessen the over-fitting problem by reducing the number of routing iterations. Adam optimizer [6] was used as a back-propagation algorithm for the network, and sigmoid function was used to calculate the final probability for each candidate class. ReLU was used as the activation function in the hidden layers.

Another baseline was state-of-the-art deep learning architectures including Inception-V3 [5]. Inception-V3 was used as a baseline model due to its popularity in the biomedical imaging field and its compatibility in adopting the architecture to a different dataset. For this study, the Inception-V3 with transfer learning was used, allowing the model to handle the weight from ImageNet training. The number of epochs used for training was 50,000 epochs, which was chosen based on its saturated point of training electrophoresis image classification and ran for five times to get an average result.

For comparison purposes, vector input models were also employed. The vector input models were operationalized by two popular machine learning techniques, SVM [7] and random forest [8]. These models served as the baselines for this experiment. For the SVM, C-support SVM was used by the scikit-learn library, which was set with gamma 1.2 and 0 as a random seed. For the random forest model, it used the Gini criterion to split the node on the tree, with the depth of 20, and ten trees as the estimators. Also, the basic neural network architecture, multilayer perceptron (MLP) [9], was also used as a baseline. The MLP architecture had three layers, which were 53 neurons of the input layer, 106 neurons of the hidden layer, and two neurons of the output layer. This architecture used 0.0001 learning rate, Adam optimizer, and 1,000 epochs for training the model.

IV. Result and Discussion

Accuracy, sensitivity, specificity, BCR, and MCC were used to assess the performance of the classification models. Based on the accuracy measurement on Table II, the proposed model achieved the highest accuracy out of all the datasets, which showed the ability of the model to correctly differentiate between normal and abnormal cases even though the proportion for both classes were different. Among the vector input models, random forest performed the best with 670 lengths of vector and showed higher accuracy than Inception-V3 in almost all datasets. The usage of original size data rather than reduced data in machine learning methods did not always lead to performance improvement. The original CapsNet demonstrated how well it could work for electrophoresis data, and EP-CapsNet exhibited the best and most improved accuracy performance.

Another measurement called BCR (Balanced Classification Rate) was used to measure the performance of each model by considering the balance between sensitivity (true positive rate) and specificity (true negative rate). Different from accuracy that tries to calculate the accurate prediction result from all data in a dataset, BCR estimates the accurate prediction based on each class of positive cases and negative cases and takes the average of true positive rate and true negative rate as the final accuracy rate. Out of all datasets, the best BCR was achieved by EP-CapsNet, the proposed model. However, the original CapsNet did not surpass the BCR rate of the Inception-V3 in all datasets (Table II), which is different from the accuracy measurement. It was also the same as the input vector models, which produced a lower BCR rate than Inception-V3. The result seems to show how well Inception-V3 tries to make a balanced classification between classes. The work of pooling layer in the architecture tries to achieve higher generalization of the result, indicating that the model can get the highest accuracy if the data is balanced. However, the EP-CapsNet delivered the highest BCR by improving the CapsNet architecture with the inception module.

MCC (Matthews Correlation Coefficient) was also used as the final measurement to measure the correlation between the prediction and actual classes. Instead of giving the percentage of accurate prediction, this measurement gives the coefficient, that shows whether the model gives perfect prediction (1), random prediction (0), or perfect incorrect prediction (-1). As shown in Table II, EP-CapsNet still showed the best performance out of all models in all datasets, which demonstrated how the correlation of the prediction and the actual class was closer to a perfect classification than other models. Different from EP-CapsNet, the original CapsNet achieved higher MCC score almost in all datasets except for 00307 dataset, which had the smallest data size and as a result lowered the learning ability of the model. However, the MCC for CapsNet was still higher than the other vector input models, demonstrating the richness of the images in classifying the electrophoresis data.

Based on sensitivity (Table II), the results were different from the overall performance. If the dataset has the highest proportion of normal class, e.g., 00307 dataset, the sensitivity scores were lower than other datasets for all models. The worst sensitivity could be seen in the MLP model for 00307 dataset, which all vector input models performed worse than the image input models, except for the original CapsNet, which was lower than the random forest method. However, the vector input models had similar or higher sensitivity scores than Inception-V3. It demonstrated how the models learned the training data as the datasets except for 00307 have more abnormal than normal data. In the end, EP-CapsNet showed an improved performance from the original CapsNet, demonstrating the highest sensitivity scores out of all datasets except for 00307, which had highest sensitivity score on Inception-V3 model. As expected from its data proportion, the 00307 dataset had overall higher specificity scores in all models than other datasets (Table II). The vector input classifiers had more severe performance differences across the data types, showing their vulnerability to the proportion of the data. In contrast, the image input models (i.e., deep-learning based models) showed more stable performance across the data types.

Overall, as shown in Table II, out of 20 repeated comparisons involving 4 different data types (i.e., 00307, 00309, 00312, 00323) and 5 different metrics (i.e., accuracy, BCR, MCC, sensitivity, and specificity), EP-CapsNet showed the best performance scores in all comparisons except three (sensitivity-00307, specificity-00307, specificity-00323). For accuracy, BCR, and MCC, EP-CapsNet was the best performing model without exception.

V. Conclusion and Future Research

In this study, a novel architecture called EP-CapsNet, which was an extension of CapsNet combined with the inception module was proposed to enhance the classification performance on graphical electrophoresis images. EP-CapsNet extended the state-of-art capsule network and adjusted it for electrophoresis image classification. Furthermore, the new architecture incorporated the inception module into the network to capture various features in different dimensions in parallel. The inception module was used to obtain a combination of features from different convolutions within a single layer before feeding them into the next layer. The results of our study showed clearly superior performances of the proposed model, relative to the original capsule network as well as the original inception network. Even with dimension reduction to 53 x 53 image size, the proposed model still showed a competitive, and mostly outperforming, results compared to the machine learning methods coupled with the original image size (670 x 296). In the future research, an ensemble method of different models might be utilized to combine the strengths of those alternative models. This approach will be particularly useful when the ensemble model wants to develop a strategy in combining the decisions of multiple models in a way that either sensitivity or specificity alone can be maximized. Also, As a follow-up of this experiment, the future research should take into account the second work of Hinton in 2017, which is called ”Matrix Capsules with EM Routing” [10] by utilizing the expectation and maximization routing. Their study was a continuation of the first CapsNet paper, which was published to improve the flaws of the prior routing-by-agreement mechanism. Instead of using vector representation, they used the matrix to represent the pose

Table II: Electrophoresis image classification based on accuracy, BCR, MCC, sensitivity, and specificity measurements.

Method	Input	00307	00309	00312	00323
Accuracy					
SVM	53-vector	80.65%	84.03%	80.88%	86.65%
SVM	670-vector	80.56%	85.04%	78.17%	87.23%
Random Forest	53-vector	83.99%	89.57%	77.32%	86.23%
Random Forest	670-vector	82.94%	90.39%	81.46%	87.58%
MLP	53-vector	82.17%	89.62%	80.78%	87.96%
MLP	670-vector	77.57%	87.59%	76.88%	86.58%
Inception-V3	53 x 53 image	81.00%	85.15%	79.02%	87.53%
CapsNet	53 x 53 image	85.28%	93.14%	88.53%	90.47%
EP-CapsNet	53 x 53 image	90.42%	95.87%	91.41%	93.60%
BCR					
SVM	53-vector	64.56%	79.08%	70.58%	89.23%
SVM	670-vector	67.01%	80.64%	63.43%	87.68%
Random Forest	53-vector	76.84%	87.36%	63.59%	86.32%
Random Forest	670-vector	75.79%	86.65%	67.65%	87.68%
MLP	53-vector	70.92%	86.88%	73.68%	88.56%
MLP	670-vector	58.21%	83.46%	76.39%	89.02%
Inception-V3	53 x 53 image	80.71%	86.68%	77.86%	89.03%
CapsNet	53 x 53 image	75.92%	92.15%	81.76%	91.83%
EP-CapsNet	53 x 53 image	85.36%	94.51%	87.49%	93.37%
MCC					
SVM	53-vector	0.421	0.556	0.424	0.717
SVM	670-vector	0.426	0.585	0.301	0.705
Random Forest	53-vector	0.562	0.712	0.292	0.681
Random Forest	670-vector	0.535	0.722	0.402	0.707
MLP	53-vector	0.487	0.710	0.461	0.722
MLP	670-vector	0.290	0.649	0.462	0.713
Inception-V3	53 x 53 image	0.594	0.687	0.542	0.749
CapsNet	53 x 53 image	0.588	0.809	0.654	0.783
EP-CapsNet	53 x 53 image	0.741	0.880	0.748	0.842
Sensitivity					
SVM	53-vector	31.83%	87.85%	88.87%	83.68%
SVM	670-vector	39.45%	88.44%	89.60%	86.74%
Random Forest	53-vector	62.28%	91.28%	87.96%	86.24%
Random Forest	670-vector	61.25%	93.28%	92.15%	87.79%
MLP	53-vector	48.01%	91.75%	86.25%	87.29%
MLP	670-vector	18.77%	90.80%	77.26%	83.90%
Inception-V3	53 x 53 image	79.84%	82.95%	81.21%	84.30%
CapsNet	53 x 53 image	56.64%	93.90%	93.73%	88.97%
EP-CapsNet	53 x 53 image	74.93%	96.92%	94.41%	93.84%
Specificity					
SVM	53-vector	97.29%	70.31%	52.29%	94.96%
SVM	670-vector	94.58%	72.84%	37.25%	86.39%
Random Forest	53-vector	91.39%	83.44%	39.22%	86.39%
Random Forest	670-vector	90.33%	80.02%	43.13%	87.00%
MLP	53-vector	93.83%	82.02%	60.84%	89.83%
MLP	670-vector	97.66%	76.11%	75.52%	94.14%
Inception-V3	53 x 53 image	81.58%	90.42%	74.52%	93.75%
CapsNet	53 x 53 image	95.20%	90.39%	69.78%	94.69%
EP-CapsNet	53 x 53 image	95.79%	92.09%	80.58%	92.90%

of the feature. Multi-class classification of EP can also be reconsidered as the continuation of this work. Notwithstanding these future improvements possible, the current study shows that the proposed EP-CapsNet is a promising choice for electrophoresis image classification.

References

[1] S. R. Vavricka, E. Burri, C. Beglinger, L. Degen, and M. Manz, "Serum protein electrophoresis: an underused but very useful test," *Digestion*, vol. 79, no. 4, pp. 203–210, 2009.

[2] M. A. Jenkins, "Serum and urine electrophoresis for detection and identification of monoclonal proteins," *The Clinical Biochemist Reviews*, vol. 30, no. 3, p. 119, 2009.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[4] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, pp. 3859–3869, 2017.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[6] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[7] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[8] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] M. W. Gardner and S. Dorling, “Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences,” *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[10] G. Hinton, N. Frosst, and S. Sabour, “Matrix capsules with em routing,” 2018.

View publication stats