Deep Learning: Protein Cells Classifications using Resnet-50 Model

Hany Elnashar[1]
Beni-Suef university ,

Islam Abd El Azim[2]
MUST university,

Abstract:- **High-Dimensional data generated from microscopy images for many single cells gives big chance to do data analysis. and as there are problem and very impotent in automatically detecting the cellular compartments. it's a simple task for an experienced human, but so difficult to be automated on a machines. Now in this paper 50- layer neural network Resnet on huge amount of data from microscopy images of cells, achieving classification per cell localization with accuracy of 95%, and per protein accuracy of 99% on images. This paper confirms and setting that low level network features correspond to basic image characteristics, and deeper layers' separate classes. Using Resnet 50 as feature extractor, then train resnet 50 standerd classifiers to assign proteins with unknown compartments with using small number of training examples. Results are accurate subcellular localizations, and give how CNN Resnet 50 as a deep learning model more effective and useful to be used with high dimensional data from microscopy images.**

1.   INTRODUCTION:

A Protein is a one of main nutrients elements in human Body structure, Proteins are often classified by their structure, and analyzing its behavior will present roles and its effectiveness in Human body. data collected about cells from medical images gives more annotated data clarification about these main elements. Cells Protein contents will help to make it possible to gain deep insights about how to manipulates and rebuild cells to avoid diseases its distortion effects in human bodies. Large microscopic images are now available in form of datasets for individual proteins gives large volume of data, Using Machine-learning to automate the noting process and content clarification of this data is challenging topics. Recently, Deep learning is a method applied on medical images gives excellent results using transfer-learning from models trained on different tasks using real Cells images help in Extracting features as a key functional units in the human cell readability and correct localization of each protein in human cells. As per that Transfer learning is based on the improvement of learning in a new task through the transfer of knowledge from a other task that has already been learned [1]. This research based on Kaggle datasets of proteins mapping and Human protein atlas Image Classificationsi, which is an initiative based in Sweden that is aimed at mapping proteins in all human cells, tissues, and organs. This data from the Human Protein Atlas database is freely accessible to scientists all around the world that allows them to explore the cellular makeup of the human body[2]. This the single-cell image classification challenge will help scientists to characterize single-cell heterogeneity in our large collection of images by generating more accurate annotations of the subcellular localizations for thousands of human proteins in individual cells[3]. Using

this will allow accurately modeling of the human cell and provide new open-access cellular data to the scientific, which may accelerate the understanding of how human cells functions and how diseases develop[4]. In this paper we would like to generate accurately protein localizations prediction that represented as integer labels based on the images. This work starts from computer vision point of view using biological image expertise knowledge's, and then use deep learning methodology to predict this localization for each protein. Using Transfer Learning with ResNet50 for image classification methodology. Results in this paper will present that we could achieve decently accurate prediction of protein localization across various cell types. Where data in form of weak image-level labels. This paper has structure started by introduction as part 1, then related and previous works part two, starting with data and features in part three showing our data deeply start from visualizing data to segmentation, part four methods and experiments models with learning rat's behaviors, and part five will show the conclusion and result discussions.

2.   RELATED WORKS:

Cells diseases identification is critical topic that has many studies, and motivated by the need of current world situation, viruses use cells as first line to attack during attacking human bodies to know cells characteristic and its response during attacks some desirable elements to take into account should be. In last decade, several works have done and proposed some nondestructive techniques to overcome those cells elements and its characteristic's. Support vector machines, K-NN classifiers with decision trees are numerous methods used for this study. Neural networks in form of Deep convolutional neural network is used for protein localization in yeasts [5]. Deep convolutional neural network have seen with high accuracy prediction with cells few features types [6]. a frequent pattern tree (FPT) approach to generate a minimum set of rules (mFPT) for predicting protein localization[7]. Without forget that in [8] creation of an open source atlas with information on the subcellular location of every human protein. Then [9] is starts to use A deep-learning framework for protein subcellular and suborganellar localization prediction with residue-level interpretation

3.   DATA AND FEATURES:
3.1.   Datasets
Data is a set of full size images mixed of 1728x1728, 2048x2048 and 3072x3072 PNG files segmented as two types train and test. The training image-level labels are provided for each sample in training data file. Each file consists of four different filter on subcellular protein

patterns represented. File formats should be in forms of shape [filename]_[filter color].png for the PNG files. With Colors are red, blue, yellow, and green where RED for microtubule channels, BULE for nuclei channels, YELLOW for Endoplasmic Reticulum (ER) channels, and GEEN for protein of interest. Datasets contains 19 different labels that have 18 labels for specific locations and label for negative and unspecific signal. The dataset is confocal microscopy with acquired in a highly standardized way using one imaging modality. However, the dataset comprises 17 different cell types of different morphology, which affect the protein patterns for each type of organelles. These image samples are categorized by four filters saved as individual files, the protein of interest (green) and others three cellular landmarks: nucleus, microtubules, and endoplasmic arranged blue, red, and yellow. The green filter should hence be used to predict the label, and the other filters are used as references. Each training image contains a number of cells that have been labeled as described above. But The labels for training are image level labels. And the prediction task is to look at images of the same type and predict the labels of each individual cell within those images. As the training labels are a collective label for all the cells in an image, it means that each labeled pattern can be seen in the image but not necessarily that each cell within the image expresses the pattern. Which mean that its weak image level labels? Now data files represented as follow train images (in .tif), test image (in .png), train.csv, test set file name for predictions. To start with this data set it have raw, with full size images each channel is approximately 8MB. With 82495 images * 4 channels, everything amounts to around 2.6TB of data. Now That in this paper using just 17 cell lines in the training set and test set. Which reflects with data size as 72k images * 4 channels, instead of 82.5k images in formats of jpg files which less that tif files but with much smaller in size so it will lose some information but it enough for current model.

3.1.1. Training data:

Training data from the training csv file. We get image id and label column, which contains the classes corresponding to each image. Looking at this data, it appears to be a multi-label classification challenge.

Table 1: training data sets

	ID	Label
0	5c27f04c-bb99-11e8-b2b9-ac1f6b6435d0	8|5|0
1	5fb643ee-bb99-11e8-b2b9-ac1f6b6435d0	14|0
2	60b57878-bb99-11e8-b2b9-ac1f6b6435d0	6|1
3	5c1a898e-bb99-11e8-b2b9-ac1f6b6435d0	16|10
4	5b931256-bb99-11e8-b2b9-ac1f6b6435d0	14|0

Realizing it is actually not a multi-label classification, but instead it segmentation task, for each image, we are being asked to segment each single cell contained in the image, identify the class of that cell, and give results as a string containing The class of each cell, Confidence of the class prediction, and Segmentation mask for each cell. In other words, the fact that the training data is image-level classes and our task is to predict cell masks and their corresponding classes. Taking in consideration some cells can be associated with multiple classes in that case it need to predict separate detections for each class using the same mask. As table one and its description before that shown there are 4 images in the train folder for each image ID, corresponding to red, blue, green and yellow channels. It has more than one unique label in the dataset start from 0 to 18 as 19 unique labels. With imbalanced dataset, and labels 11 and 18 are going to be especially problematic levels. Data The distribution of labels as shown in figure 1 and classified with The descriptions in [10]. so the class 18 which found with very few examples is actually the everything else category it can influence labeling strategy.
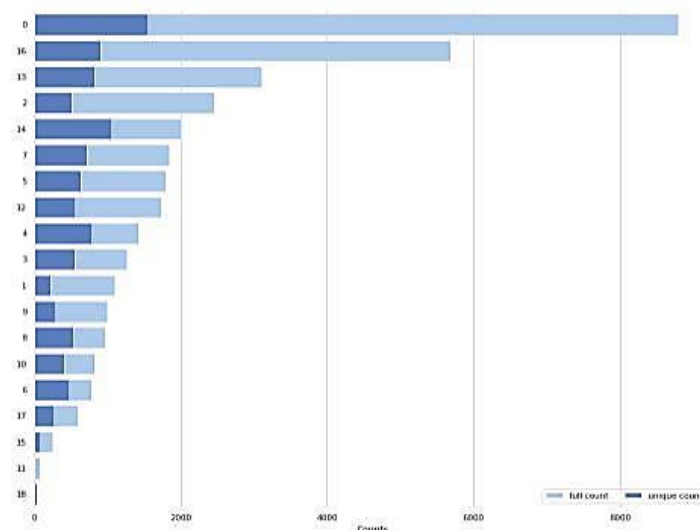

Figure 1: Imbalanced dataset and labels distributions

Number of unique combinations of classes in the train set are 432 as shown in figure 2 and figure 3 show How often do individual classes co-occur in the train set Single class occurrences are on the diagonal
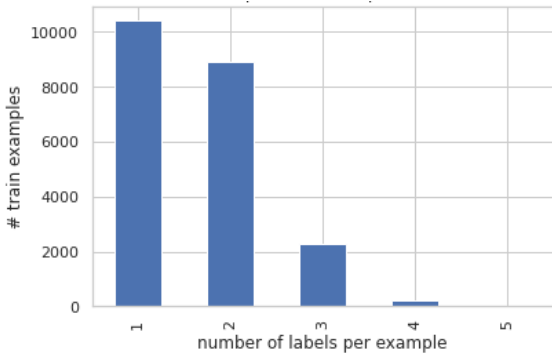
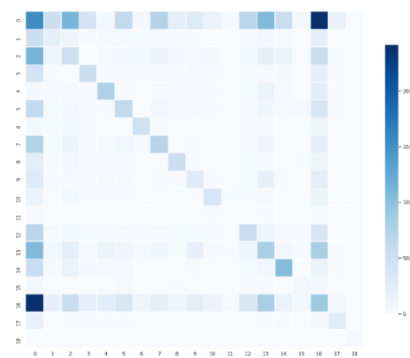Figure 2a: number of labels per example disruptions

Figure 2b: individual classes co-occur in the train set

So from the co-occurrence plot that the classes 0 (nucleoplasm) and 16 (cytosol) tend to co-occur most

frequently with other classes, and especially with each other. These are also the most frequent classes.

3.1.2. Images

All images have 4 channels started from Red (Microtubules), Green (Protein of interest), Blue (Nucleus), to Yellow (Endoplasmic reticulum) full described in Human Anatomy 4th [11]. figure 3 show an example of each channel below

Endoplasmic reticulum   Nucleus   Microtubules   Protein of interest
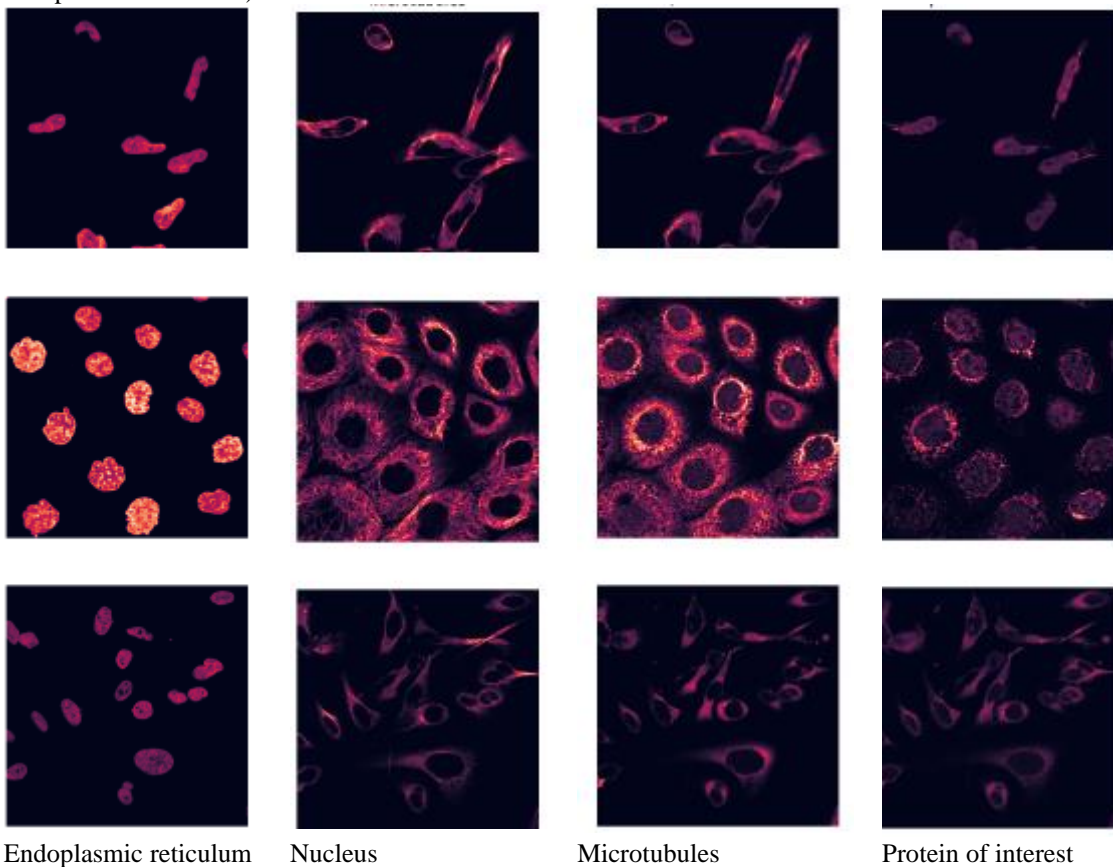
Figure 3: images 4 channels

To display them in a single image as Combining Channels and visualize the red, green and blue let convert them to limited with the visual representation of classes to RGB channels only it will be as in figure 4. to visualize an image for each class with only images representing single class for 18 type in this data set.
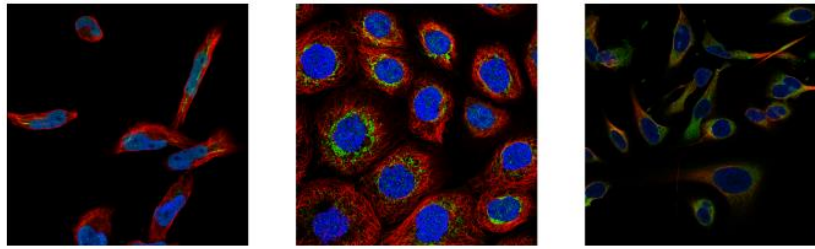
Figure 4: RGB Representations

4. METHODS:
4.1.1. Convolutional neural network

CNN or A convolutional neural network, is a deep learning neural network for processing structured data types such as images. are used in computer vision and now become as the state of the art for visual applications as image classification, and have also found success in many other application as natural language processing. For image processing CNN can run without needs any preprocessing[12]. Figure 5 show model structure and building blocks used to gain more realization about CNN features that aid prediction start by exploring the characteristics of learned weight and the output of neuron. By select images that triggers individual neuron as maximum and minimum values, with deep look for each group of neurons starts from data point of view, Data nearest to The first layers of neurons, and small-scale image characteristics. First four neurons selected as first layer and activated by image patches containing edges. corners and lines with second layer activation of neurons, moving to more complex shapes with third and fourth layers. And combinations of low-level features represented by Neurons in deeper layers.
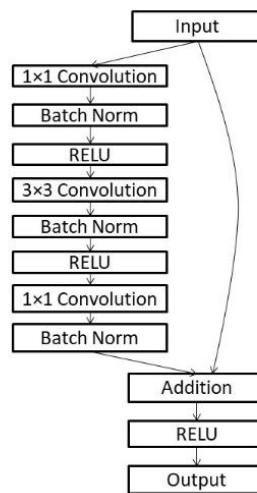

Figure 5: show a typical residual block of ResNet (50 layers), where each layer consists of Convolutional (conv), Rectified Linear Unit (RELU) and Batch Normalization (Batch Norm) layers. The shortcut connections perform identity mapping and their outputs are added to the outputs of the stacked layers. The three convolutional layers are 1×1, 3×3 and 1×1. The 1×1 layers are responsible for reducing the dimensions and then increasing (restoring) the dimensions. After constructing the residual block, very deep networks are built by stacking residual blocks [13]

This makes CNNs highly have efficient for image reading and image processing since a feature may have located anywhere in the image. Where each layer get its input from previous layer, hierarchically and progressively features become more complex[14]. Rasnet50 as a tools for visualize high-dimensional data in two dimensions as CNN model architecture[15], using randomly with 1000 sampled images, and added colored information in compartment reflections. substantially The classes overlap will with lower layer outputs, while deeper layers that make use of fully connected network structure increasingly separate the localizations, and nearby points seems like to the same class.

To identify which neuron outputs are correlated to the CellProfiler features and class membership needed to calculate the strongest Pearson correlation coefficient to a CellProfiler feature as calculated in[16], And the largest mutual information with a class label for each unit output.

4.1.2. Experiments model:

ResNet-50 Is 50 layers deep as CNN, trained on more than a more than million images from databases[13]. 3-layer deep for each ResNet-50 block. the first layer in the ResNet-50 architecture is convolutional, which is followed by a pooling layer or MaxPooling2D in implementation, is followed by 4 convolutional blocks containing 3, 4, 6 and 3 convolutional

layers. And with global average pooling layer called as GlobalAveragePooling2D. The output of this layer is flattened and fed to the final fully connected layer. Now by going to create a new FullyConvolutionalResnet50 function as the baseline for further receptive field calculation, and use available training dataset to train model. stratify based on combination of labels in dataset. The unique combinations will be put into train. Which will give that There are 559 images with unique label combinations out of 21806. Then using green filter, as the green filter should be used to predict the label, and the other filters are used as references. Data have imbalanced performance as shown in figure 1, which mean that the time to data Augmentation by Introducing new synthetic images based on new generation via rotation and mirroring [17], as data represented in part 3.1 of data description now will start to work with images have labels

for training as image level labels while the prediction task is to predict is cell level labels. So this research challenge has both needs to segment the cells in the images and predict the labels of those segmented cells with mask MAY have more than one class which inforce which predict separate detections for each class using the same mask. All cells have their feature (shape, size, and distribution of proteins) and all of them have all the four channels, and signals from the markers (blue, yellow, red). Note that each of the image-level labels can be present in all or in just a fraction of the individual cells in the image, and that some individual cells may also have additional labels, as there are 4 cells and 3 have green staining and the image-level labels are Mitochondria, and Nucleoplasm. So they can be seen as shown in table 2:

Table2: The labels organelles/structures and in which the proteins are located

Cells	Notations
Cell 1:	Green looks to be in the Mitochondria. Therefore, the cell level is **Mitochondria**.
Cell 2:	- Green looks to be in the Nucleoplasm. Therefore, the cell level is **Nucleoplasm**.
Cell 3:	- Green looks to be in the Nucleoplasm and Mitochondria. Therefore, the cell level label is **Nucleoplasm** and **Mitochondria**
Cell 4:	- No green or green is not present in any organelle. Therefore, the cell level label is **Negative**

Table3 : Resent50 training Approach

1.	Identify slide-level images containing only one label
2.	Segment slide-level images (get RLEs for all cells in all applicable slide-level images)
3.	Crop RGBY image around each cell
4.	Pad each RGBY tile to be square
5.	Resize each RGBY tile to be (256px by 256px)
6.	Filter the images based on certain additional factors to obtain a better training dataset
7.	Separate the channels and store as separate datasets
8.	Update the dataset (greatly increase the number of negative class examples)
9.	Train a model to classify these tile-level images accurately

Table4: model PARAMETERS values

N_EPOCHS	10
LR_START	0.005
LR_MAX	0.0011
LR_MIN	0.0005
LR_RAMPUP_EPOCHS	3
LR_SUSTAIN_EPOCHS	2
LR_EXP_DECAY	0.75

the images in the hidden test set are 16-bit let start with train Rasnet50 with dataset by approach in table 3. Then finding class WEIGHTING based on class COUNTS which give one for all except class 11, then start to define model as per

DATASET PARAMETERS as shown in table 4 to Using an LR ramp up because fine-tuning a pre-trained model and Starting with a high LR would break the pre-trained weights Which give learning rate as shown in figure 6:
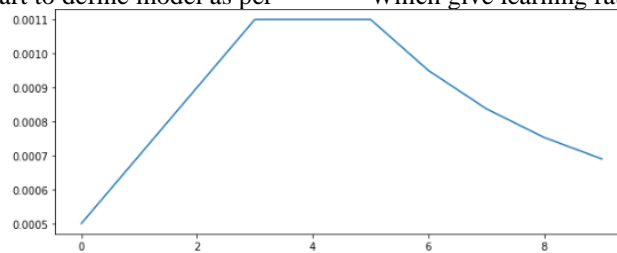

Figure 6: Custom Learning rates

4.1.3. CREATE THE TRAINING AND VALIDATION DATASETS

Now start to create two category of needed data from

produces can be readable and interpreted as an image
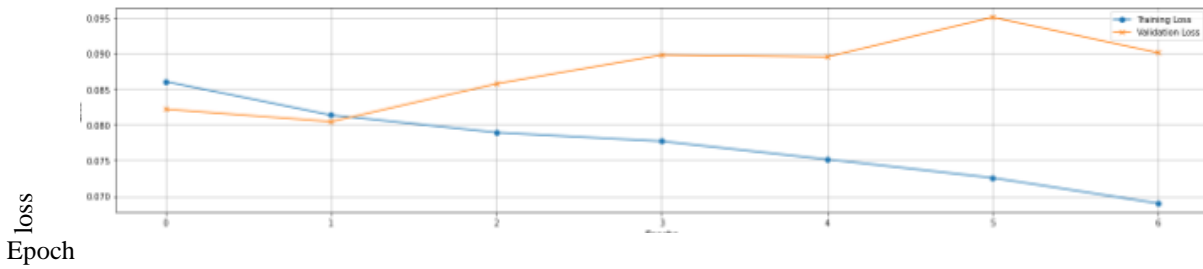

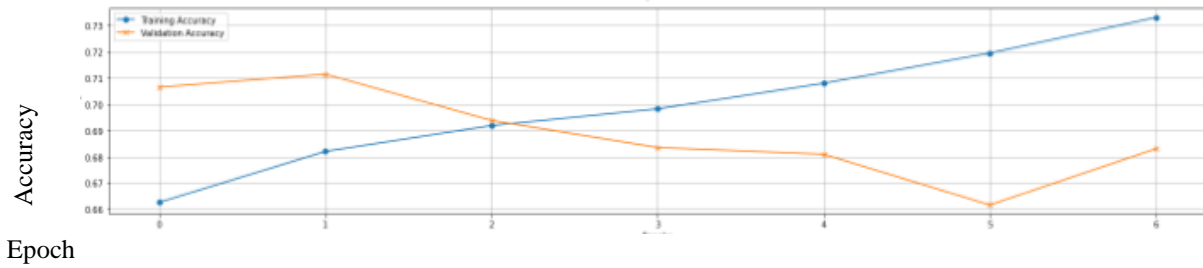Figure 7.a: Losses of training and validations

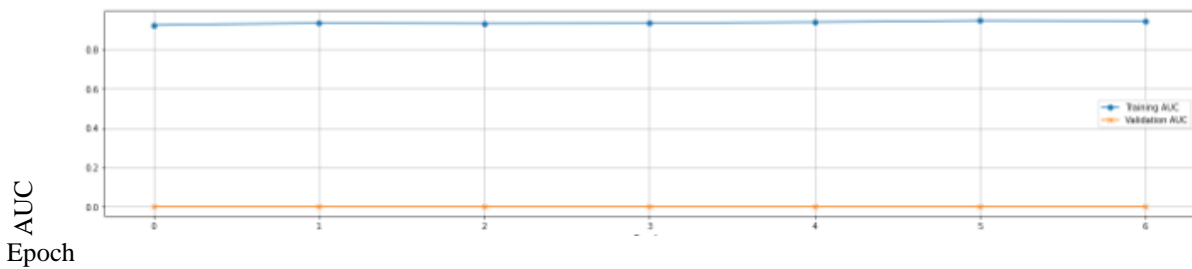Figure 7.b: Accuracy of training and validations

Figure 7.c: AUC of training and validations

datasets as per after reduction and redistribution it arranged as 37472 in full dataset, 34662 for train dataset, and 2810 for validation dataset. Then loading the model backbone for training and fit the model with defined Epoch rate, by visualize training as shown in figure 7 showing that for each Epoch where the training and validation losses, figure 7.b show accuracy for each type of data, and figure 7.c show The area under the ROC-curve with Epochs.

Conclusion:

We have demonstrated that CNN Resnet model, 50-layer convolutional neural network, can achieve classification accuracy of 91% for individual cells over 18 subcellular localizations, and 100%for proteins when entire cell populations of at least moderate size are considered. Far from being a black box. the internal outputs that Resnet

Resnet can be reused for other image analysis experiments with the same marker proteins and magnification, or trained further for specific applications and can be applied for both classifying previously unseen compartments and inferring mixtures of localization patterns. The usual classification implementations do not always provide models that are easy to reuse. Deep neural networks have proved their value in extracting information from large-scale image data [22]. It

characteristic. s. The trained network functions as an extractor of features to successfully distinguish previously unseen classes. Nucleus and nucleolus are patches of similar size; when the characteristic crescent shape of the nucleolus is not showing, it is also difficult to distinguish from the nuclear marker. Overall, the single cell accuracy of 91% is approaching the protein compartment assignment performance of previous reports. The success of Resnet deep neural networks in image analysis relies on architectures that encapsulate a hierarchy of increasingly abstract features relevant for classification, and plentiful training data to learn the model parameters. While first applications used a smaller number of layers [18] and mostly operated on precalculated features [19], pixel level analyses gave good results [20], especially using the latest training methods [21].

would be unreasonable to believe that the same will not be true for high-throughput microscopy. Adaptation of the technology will depend on the ease with which it is deployed and shared between researchers; to this end, we have made our trained network freely available. The utility of these approaches will increase with accumulation of publicly shared data, and we expect deep neural networks to prove

themselves a powerful class of models for biological image and data analysis.

REFERENCES

[1]   M. Zhang, Y. Zhou, J. Zhao, Y. Man, B. Liu, and R. Yao, "A survey of semi- and weakly supervised semantic segmentation of images," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4259–4288, 2020, doi: 10.1007/s10462-019-09792-7.

[2]   K. Institutet, "PRESS RELEASE A 20-year journey with the Human Protein Atlas," pp. 19–21, 2020.

[3]   A. Fleming and B. Chain, "Introduction to the Human Protein Atlas Function of blood proteins A Century of Advances in Immunology reflected in Nobel Prizes awarded for discoveries involving blood cells and proteins," 2018.

[4]   S. Golwalla, M. Nadkar, A. Golwalla, and S. Golwalla, "Infectious Diseases and Infections," *Golwalla's Med. Students*, pp. 693–693, 2017, doi: 10.5005/jp/books/13059_11.

[5]   Curtis KM *et al.*, "Enhanced Reader.pdf," *Nature*, vol. 388. pp. 539–547, 1997.

[6]   M. Buda, "A systematic study of the class imbalance problem in convolutional neural networks SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION A systematic study of the class imbalance problem in convolutional neural networks," p. 49, 2017.

[7]   J. Wang, C. Li, E. Wang, and X. Wang, "An FPT approach for predicting protein localization from yeast genomic data," *PLoS One*, vol. 6, no. 1, 2011, doi: 10.1371/journal.pone.0014449.

[8]   P. J. Thul *et al.*, "A subcellular mapThul, P. J., Åkesson, L., Wiking, M., Mahdessian, D., Geladaki, A., Ait Blal, H., … Lundberg, E. (2017). A subcellular map of the human proteome - Supplemental material. Science, 356(6340), eaal3321. https://doi.org/10.1126/science.aal332," *Science (80-.).*, vol. 356, no. 6340, p. eaal3321, 2017, doi: 10.1126/science.aal3321.

[9]   Y. Jiang *et al.*, "MULocDeep: A deep-learning framework for protein subcellular and suborganellar localization prediction with residue-level interpretation," 2020, doi: 10.21203/rs.3.rs-40744/v1.

[10] P. Biotechnology, "aminoxyTMT Mass Tag Labeling Reagents," vol. 0747, no. 815.

[11] P. D. Sugiyono, "済無No Title No Title," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2016, doi:

[12] H. Alaeddine and M. Jihene, "A CONVblock for Convolutional Neural Networks," no. February, pp. 100–113, 2020, doi: 10.4018/978-1-7998-5071-7.ch004.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[15] R. Vishwakarma, "CNN Model & Tuning for Global Road Damage Detection," no. 2013, 2020.

[16] P. Tanel and P. Leopold, "Accurate Classification of Protein Subcellular Localization from High-Throughput Microscopy Images Using Deep Learning," *G3 Genes, Genomes, Genet.*, vol. 7, no. May, pp. 1385–1392, 2017, doi: 10.1534/g3.116.033654/-/DC1.

[17] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[18] J. Gebert *et al.*, "Microsatellite instability in colorectal cancer is associated with local lymphocyte infiltration and low frequency of distant metastases," pp. 1746–1753, 2005, doi: 10.1038/sj.bjc.6602534.

[19] C. Conrad and D. W. Gerlich, "Automated microscopy for high-content RNAi screening," vol. 188, no. 4, pp. 453–461, 2010, doi: 10.1083/jcb.200910105.

[20] Æ. Chris *et al.*, "The quality of life of children with attention deficit / hyperactivity disorder : A The quality of life of children with attention deficit / hyperactivity disorder : a systematic review," no. June 2014, 2009, doi: 10.1007/s00787-009-0046-3.

[21] J. M. Moriuchi, A. Klin, D. Ph, W. Jones, and D. Ph, "Mechanisms of Diminished Attention to Eyes in Autism," no. January, 2017, doi: 10.1176/appi.ajp.2016.15091222.

[22] H. Tang, B. Wang, and X. Chen, "Deep learning techniques for automatic butterfly segmentation in ecological images," *Comput. Electron. Agric.*, vol. 178, no. May, p. 105739, 2020, doi: 10.1016/j.compag.2020.105739.