

Evaluation of Protein Sub-cellular Localization Using High-throughput Micrographs

Wangjie Zheng^a, Xinyu Jia^a, Yixiang Qu^a, Yunzhe Jiang^{*a}, Zhizhuo Zhang^a

^a Department of Bioinformatics and Biostatistics, School of Life Sciences and Biotechnology,

Shanghai Jiao Tong University, Shanghai, China

jiangyunzhe@sjtu.edu.cn^{*} (indicated by asterisk which e-mail for primary communication)

ABSTRACT

Protein sub-cellular localization plays an important role in systems biology. The visualization technology of gene expression in living cells have emerged dramatically. With the help of deep learning, it becomes possible to handle with astronomical high-throughput micrographs, classifying proteins into different sub-cellular localization. Here, we introduced a deep learning model based on ResNet-18 classifying proteins into twelve categories with the accuracy of 87.66%. Moving forward, we successfully calculated the size of sub-cells based solely on the red channel.

KEYWORDS

Protein Sub-cellular Localization; Green Fluorescent Protein (GFP); Convolutional Neural Network (CNN); Image Processing

INTRODUCTION

It is known to all that the regulation of most biological processes are associated with changes in protein abundance and localization. Determination of protein sub-cellular localization facilitates understanding post-translational modifications, protein stability and protein-protein interactions¹.

Previous large-scale analyses of protein localization in *S. cerevisiae* have depended on transposon-mediated random epitope tagging and plasmid-based over-expression of epitope-tagged proteins², which can interrupt important localization signals, leading to abnormal sub-cellular localization. To circumvent these potential problems, green fluorescent protein (GFP) is tagged at the carboxy terminal end by inserting the coding sequence in-frame immediately preceding the stop codon of each ORF³ as Figure 1 displays. It is proved that with this strategy, wild-type levels and patterns of protein expression are minimally perturbed. Furthermore, because GFP fluorescence does not require external cofactors,

GFP signal can be monitored in living cells without disrupting cellular integrity. To efficiently quantify the abundance and localization of the about 4,100 visible fusion proteins in the ORF-GFP collection in yeast, cytosolic red fluorescent protein (RFP), a marker of cell boundaries, is introduced into the ORF-GFP array by synthetic genetic array (SGA) technology.

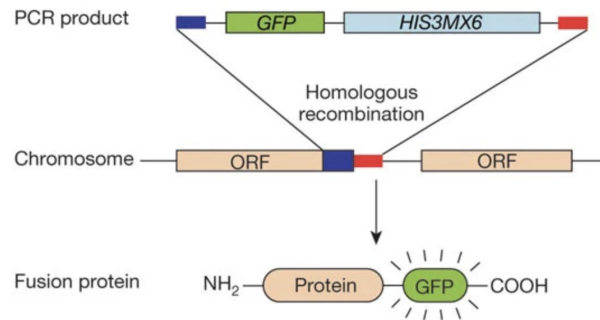


Figure 1. Strategy of Constructing GFP-tagged proteins

Huh *et al.*³ classified these proteins, representing 75% of the yeast proteome, into 22 distinct sub-cellular localization categories using wide-field microscopy and manual inspection, and provided localization information for 70% of previously unlocalized proteins. Benanti *et al.*⁴ developed a proteomic screening method that used high-throughput quantitative microscopy of a two-color system to comprehensively screen for ubiquitin-ligase substrates. Mazumder *et al.*⁵ assayed the transcriptional response of the ribonucleotide reductase (RNR) subunit genes by imaging single transcripts with fluorescence in situ hybridization (FISH) and subsequently combined this technique with immunofluorescence detection of Rnr proteins to simultaneously investigate their translational responses in the same individual cells as a function of the cell cycle. However, all these methods relied on manual assessment of localization, which is non-quantitative and suboptimal since assignments made by different individuals can be subject to biases and are not powerful enough to deal with astronomical imaging data.

With the rapid development of machine learning and conforming booming imaging data, it is imperative to apply automated approaches to analyze protein localization in yeast. Unsupervised algorithm based on clustering was proposed to construct and compare quantitative signatures of protein sub-cellular localization patterns⁶. Moving forward, in terms of supervised learning, Chong *et al.*¹ developed an ensemble of binary classifiers based on support vector machine (SVM) to generate localization data from single-cell measurements and constructed maps of about 3,000 proteins connected to sixteen localization classes on a genome scale.

Compared with machine learning, there is no denying that convolutional neural network

(CNN) are expert on computer vision and pattern recognition. Much of the success of deep neural networks has been accredited to additional layers because it is these layers that progressively learn more complex features. Deeper neural networks are more difficult to train. ResNet presents a residual learning framework to ease the training of networks that are substantially deeper than those used previously⁷. Typical ResNet models are implemented with double- or triple- layer skips that contain non-linearities (ReLU) and batch normalization in between.

Here, we introduced a model based on a pre-trained ResNet-18 to classify protein into twelve sub-cellular localization categories in a systematic and quantitative fashion on a genome scale. It is proved that accuracy of our model increased dramatically compared with those of machine learning models¹. Moving forward, we also calculated sizes of different cells in pixel based on the red channel of high-throughput micrographs.

METHODS AND PROCEDURES

Image classification

The labeled datasets we used in this project contain 65000 images for training, 12500 images for testing and 12500 images for validation. In order to reach a satisfied accuracy towards images to be classified, we choose to use Convolutional Neural Network as classifier. Before fully optimizing the model, we construct two simple models with limited number of convolutional layers, as shown in Figure 2.

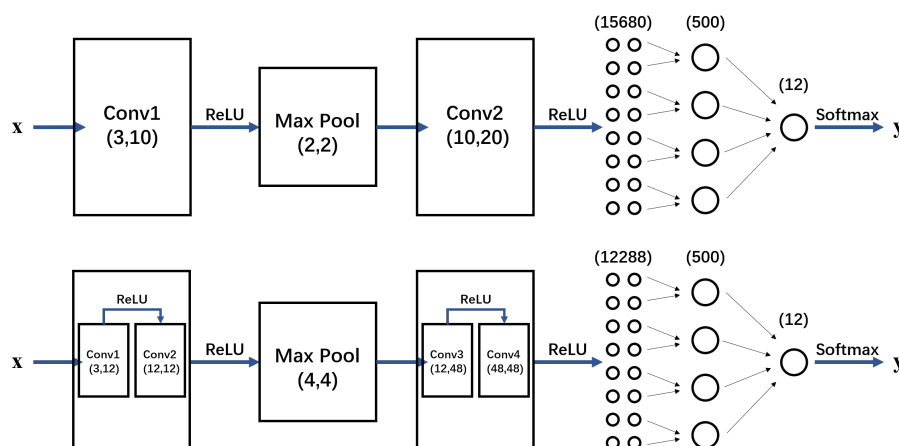


Figure 2. Two *SimpleNet* CNN networks we use at the beginning

However, optimizing models with simple structures couldn't achieve our goal of a higher accuracy. Therefore, we moved on to the idea of optimizing a complex network.

In 2012, Krizhevsky *et al*⁸. rolled out the red carpet for the Deep Convolutional Neural Network. This was the first time this architecture was more successful than traditional, hand-crafted feature learning on the ImageNet. Their DCNN, named AlexNet,

contained 8 neural network layers, 5 convolutional and 3 fully-connected. This laid the foundational for the traditional CNN, a convolutional layer followed by an activation function followed by a max pooling operation. Much of the success of Deep Neural Networks has been accredited to these additional layers. The intuition behind their function is that these layers progressively learn more complex features. And our Simplenet models are also based on this idea. Despite the popular meme shared in AI communities from the Inception movie stating that "We need to go Deeper", He *et al*⁷. empirically show that there is a maximum threshold for depth with the traditional CNN model, shown as Figure 3.

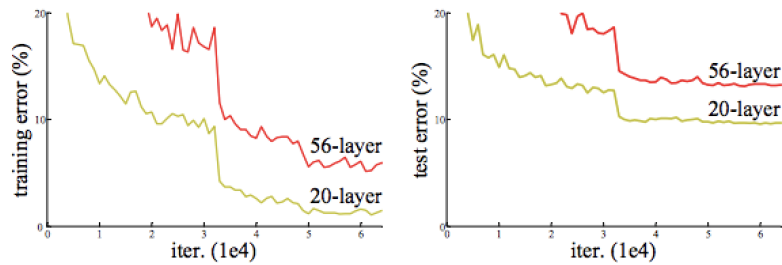


Figure 3. The deeper network has higher training error, and thus test error

The failure of the 56-layer CNN could be blamed on the optimization function, initialization of the network, or the famous vanishing/exploding gradient problem. Amongst the many theories explaining why Deeper Networks fail to perform better than their Shallow counterparts, it is sometimes better to look for empirical results for explanation and work backwards from there. The problem of training very deep networks has been alleviated with the introduction of a new neural network layer - **The Residual Block**⁷.

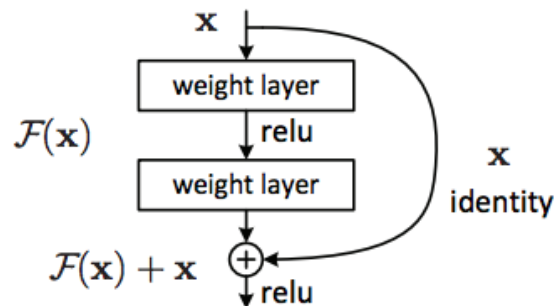


Figure 4. Residual learning: a building block

Figure 4 shows the most important thing to learn from this article. For developers looking to quickly implement this and test it out, the most important modification to understand is the *Skip Connection*, identity mapping. This identity mapping does not have any parameters and is just there to add the output from the previous layer to the layer ahead. However, sometimes x and $F(x)$ will not have the same dimension. The identity mapping

is multiplied by a linear projection W to expand the channels of shortcut to match the residual. This allows for the input x and $F(x)$ to be combined as input to the next layer, which is shown as follows,

$$y = \mathcal{F}(x, \{W_i\}) + W_s x$$

The Skip Connections between layers add the outputs from previous layers to the outputs of stacked layers. This results in the ability to train much deeper networks than what was previously possible. Therefore, we decided to construct a ResNet neural network based on the high-performance ResNet. As Figure 5 shows, there are several ResNet structures that are widely used. We choose to use simple but effective 18-layer ResNet to build the skeleton of our model.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 5. Different ResNet models

In order to make the model better fit our data and increase the accuracy for the test data, we modified the 18-layer ResNet and added two convolutional layers before conventional ResNet-18, which proved to have great performance in our dataset, not only the 3-channel input, but also 2-channel and 1-channel input.

Considering that all of these images in our dataset are with B channels equaling to zero, 3-channel (RGB) input is the same as 2-channel input. But to make our result more convincing, we applied the same model on RG channels input and G channel input.

Calculation of sub-cell size

Since RFP is a marker of cell boundaries, we wanted to explore whether the size of the cell could be estimated solely based on the area covered by RFP. We compared sizes of sub-cells (in pixel) in red channel with the real sizes (in pixel) to determine whether it is feasible that the RFP marked boundaries of cells. In terms of determining the boundary contour of the sub-cells in the image, we chose to use OpenCV package in Python, including functions `cv2.threshold()`, `cv2.findContours()` and `cv2.drawContours()`.

The function of `cv2.threshold()` is to binarize the image. If pixel value is greater than a threshold value, it is assigned one value (may be white), else it is assigned another value (may be black). Binarization of image is to set the gray value of the pixels on the image to 0 or 255, which makes the whole image show a clear black and white effect. In digital image processing, the binary image plays an important role. Binarization of image can significantly reduces the amount of data in the image, thereby highlighting the outline of the target. If an image has a uniform gray value, and is on a uniform background with other levels of gray value, the binarization can reach a comparative segmentation effect. If the difference between the image and the background is not in the gray value, difference features can be converted into a gray difference, and then use the threshold selection to segment the image. By dynamically adjusting the threshold to realize the binarization of the image, the specific result of the segmented image can be dynamically observed.

The function of `cv2.findContours()` is to automatically identify the contours of objects aligned with chromatic aberration and mark them by setting parameters and thresholds. The figure is not necessarily the exact figure used to recognize the outline, and in fact, it can be any figure. In the study, we got the outline of images of the red channel, and finally drew the outline on the original images so that it can be a more effective fashion.

There is another problem in identifying contours of images. The resolution of the boundaries in all images were heterogeneous so that there could not be a constant threshold. To handle with problem, we found that the resolution of the outline could be increased by adjusting contrast. `ImageEnhance.Contrast()` can adjust image contrast. An enhancement factor of 0.0 gives a solid grey image, and a factor of 1.0 gives the original image. Gamma correction is a nonlinear operation used to encode and decode luminance or tristimulus values in image identification. Gamma correction is defined by the following power-law expression,

$$V_{\text{out}} = AV_{\text{in}}^{\gamma}$$

It was suggested that the relative resolution of the outline was increased to a certain extent by increasing the contrast of the picture. When the outline was identified, size of sub-cells can be calculated by `contourArea` function in OpenCV package.

RESULTS

Image classification

At the beginning of our work, we introduced two *SimpleNet* models. After a training process with 30 epoches, the two models gave out less satisfied accuracy results, 71.7% and 75.8% respectively. These results prompted us to turn to complex networks - ResNet. After several adjustments on model's parameters, the image classification result for ResNet was really promising, which reached the accuracy of nearly 90%.

In order to compare the difference between 2-channel input and 1-channel input, we trained the same model with different number of channels input. The results of training for each epoch are shown in Figure 6.

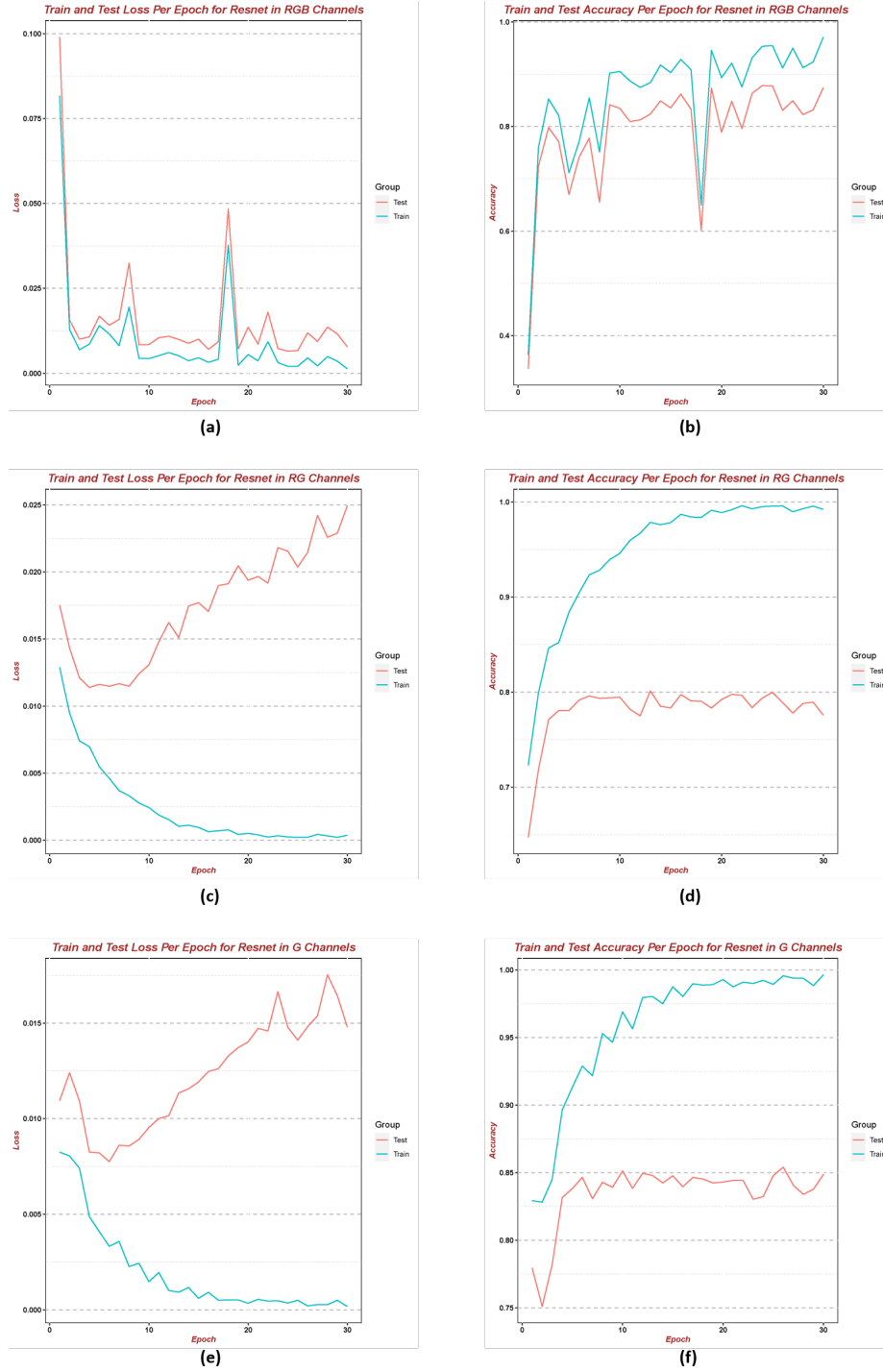


Figure 6. The result comparison between different input channels.

(a) and (b) are Loss and Accuracy of RGB input.

(c) and (d) are Loss and Accuracy of RG input.

(e) and (f) are Loss and Accuracy of G input.

It seems clear that when providing the original RGB channels, both training and testing losses were decreasing and their accuracy increased with the same trend. The rather small difference between training and testing losses indicated the very good fit of the model. But, as for the other two situations, while the training loss decreased and accuracy increased continually, their testing losses kept increasing and their accuracy stayed unchanged after a small number of epoches. This abnormality indicated the early overfit of the model, which brought up a possibility that a large amount of information was ignored with the loss of several channels.

After training, we used validation dataset to evaluate the reliability of our model, the accuracy is shown in Table 1.

Table 1. The validation results of three different input channels

Input channels	RGB	RG	G
Accuracy	87.66%	78.82%	82.74%

Calculation of sub-cell size

From the images randomly selected from the training set, whatever class of sub-cell is, the cell contour image after extracting the red channel was generally the same as that of the original image. Figure 7 displays the comparison of two kinds of images. The first row displays raw images and the second row displays images extracted the red channel. Therefore, we believed that it was feasible to estimate the size of the cells through the area covered by the RFP.

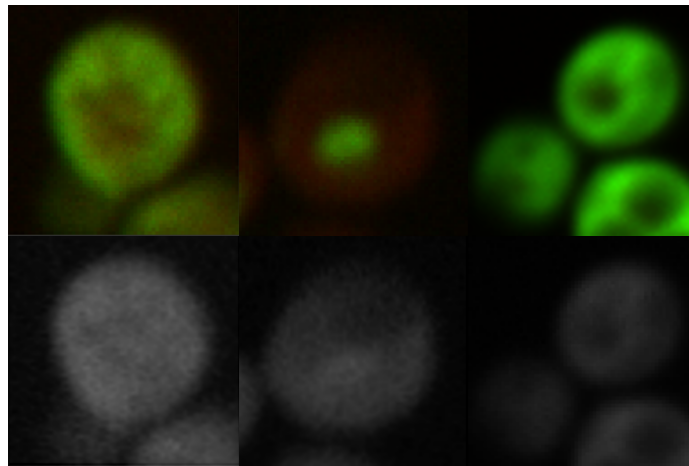


Figure 7. Comparison between images extracted red channel and the original images

In order to identify outline of sub-cell, a threshold should be set as a parameter of functions. When the threshold was set to 127 as default, the so-called outline picture was merely clear. We thought that it was because that red boundary was not obvious enough.

We gradually reduced the threshold from 50 to 20, hoping to find a optimal value. Figure 8 displays the contour identified by the function at different threshold.

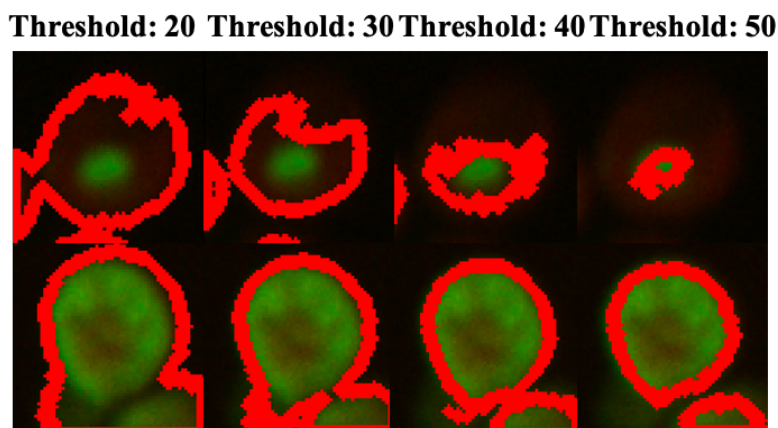


Figure 8. Different threshold set to identify the outline of sub-cells

It was interesting that though in a small range of threshold (from 20 to 50), the outline of the first image changed dramatically (the first row). However, the outline of the second image seemed insensitive to the change of threshold. As for the first image, the smaller the threshold was, the more accurate the image was outlined. As for the second image, the larger the threshold was, the more accurate the area was outlined. It was suggested that the blurriness of the outline of different images was distinct, so that the boundary threshold could not be a constant. We concluded that if only the red channel was used for contour drawing, the threshold must vary from images.

We sought adjusting contrast of images to clearly identify outlines. In order to determine at what level contrast of images should be reset, Figure 9 displays the relationship between the area and the threshold at a specific contrast value.

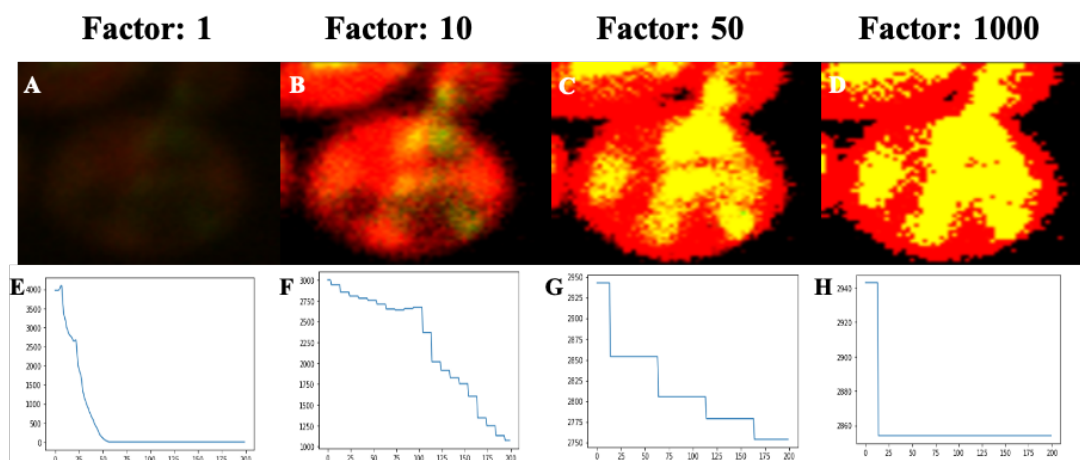


Figure 9. Relationship between the area and the threshold at a specific factor of contrast value (above)

It was noticeable that after enhancing the contrast, the area could be calculated under different thresholds (E and F in Figure 9). However, we were unable to determine the exact threshold yet. Ideally, the area should be independent of the threshold. We argued that the reason of inconsistent change of the curve in Figure 9-F lay in the model detected the a sharp change between the targeted and the background. The higher the contrast value we set, the larger difference in color contrast was likely to be. Compared with Figure 9-F, 9-G and Figure 9-H, it is clear that the more the contrast is enhanced, the less the area change under different thresholds.

For each test image, we calculated the area of covered by the red channel based on the enhanced contrast (1000 times). The outline boundary of the sub-cell was conspicuous, and the area converged at some thresholds.

DISCUSSION

Comparing to the previous research¹ that our datasets come from, our model shows a tremendous progress on the accuracy of classifying the images with a nearly 20% improvement. It shows the powerful ability of Convolutional Neural Network in image classification.

The reason why we choose ResNet-18 as the skeleton of our model was mainly its effectiveness. However, its simplicity can also implicates the limitation of performance. The performance of the model might be higher if we try other ResNet models such as ResNet-50 or even, if we have enough time and computing resources, ResNet-152.

Considering the blue channel is of no use in the model, the RGB-input model could be seen as an example of binary-channel model. Therefore, the experiment showed that the test accuracy of the single-channel is smaller than that of binary-channel. The reason for this is about the design of the datasets, we suggest, that the red channel gives the information of cell boundary and the G channel gives the information of product mark. If we use only G channel to train our model, the loss of the information about the relative position between cell and product will influence the outcome of the model. But with the price of lower accuracy, the advantage of single-channel input is a much faster training step. In short, while giving up some information, a faster training process is as compensation.

Though by enhancing the contrast of images, we were able to identify outline easily, there were some exceptions. As Figure 10 displays, there is no clear sub-cell boundary. When we tried to get outline by enhancing contrast, the model is likely to use the slight color difference of the background to extract an outline boundary deliberately so that the enhanced image (right above) was totally messed up. As Figure 11 displays, there is slight signal of the red channel, which was proved by the enhanced image (right above),

so that the calculation based on the red channel was not accurate.

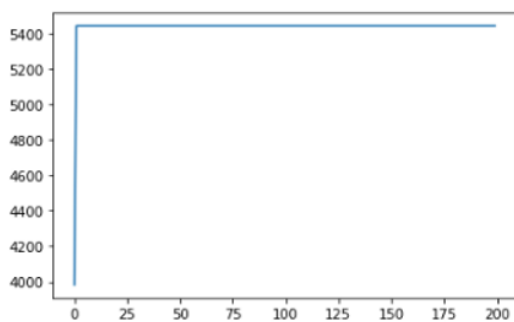
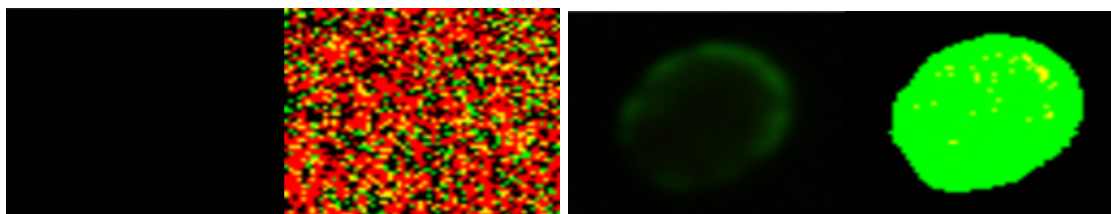


Figure 10. No sub-cell boundary

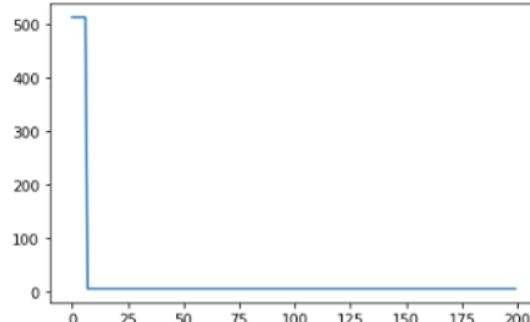


Figure 11. No signal in red channel

To sum up, the primary reason for the above two exception is that the algorithm cannot detect signal of the red channel. Thus, we cannot calculate the size of sub-cells of which RFP is not expressed. We devised our model to make it more robust so that it can determine whether the size of the input image can be calculated or not. Noticeably, if sizes of sub-cells cannot be calculated based on the red channel, we can calculate them based on the green channel with the same method.

CONCLUSIONS

1. From our results, we can conclude that ResNet-18 can be used to develop high-performance determination and classification of protein sub-cellular localization.
2. It is clear that ResNet-18 masters not only ternary-channel images but also single-channel and binary-channel images, and our modified ResNet model perform very well in single-channel and binary-channel images, though there are slight problems of over-fitting for the single-channel images.
3. Training speed could be faster by giving up some information (such as channels), but lower accuracy and possible early overfit may be the price.
4. With the aid of contrast enhancement and image processing algorithms, we are able to identify the contour of sub-cells, and thus calculate the size of sub-cell based on the red channel (RFP).

DATA AND CODE AVAILABILITY

Please see our [GitHub](#) for the results, data and code.

ACKNOWLEDGEMENTS

This work was supported by the Center for HPC at Shanghai Jiao Tong University. We thanked Distinguished Professor Hui Lv and Assistant Researcher Xiaolei Wang from School of Life Sciences and Biotechnology, Shanghai Jiao Tong University for their useful instructions and comments which have greatly improved the manuscript. We were also grateful to T.A. Yan Kong, who accompanied us this semester and replied our questions on the lectures as soon as possible.

REFERENCES

1. Yolanda T Chong, Judice LY Koh, Helena Friesen, Supipi Kaluarachchi Duffy, Michael J Cox, Alan Moses, Jason Moffat, Charles Boone, and Brenda J Andrews. Yeast proteome dynamics from single cell imaging and automated analysis. *Cell*, 161(6):1413–1424, 2015.
2. Petra Ross-Macdonald, Paulo SR Coelho, Terry Roemer, Seema Agarwal, Anuj Kumar, Ronald Jansen, Kei-Hoi Cheung, Amy Sheehan, Dawn Symoniatis, Lara Uman-sky, et al. Large-scale analysis of the yeast genome by transposon tagging and gene disruption. *Nature*, 402(6760):413–418, 1999.
3. Won-Ki Huh, James V Falvo, Luke C Gerke, Adam S Carroll, Russell W Howson, Jonathan S Weissman, and Erin K O’Shea. Global analysis of protein localization in budding yeast. *Nature*, 425(6959):686–691, 2003.
4. Jennifer A Benanti, Stephanie K Cheung, Mariska C Brady, and David P Toczyski. A proteomic screen reveals scf grr1 targets that regulate the glycolytic–gluconeogenic switch. *Nature cell biology*, 9(10):1184–1191, 2007.
5. Aprotim Mazumder, Katja Tummler, Mark Bathe, and Leona D Samson. Single-cell analysis of ribonucleotide reductase transcriptional and translational response to dna damage. *Molecular and cellular biology*, 33(3):635–642, 2013.
6. Lit-Hsin Loo, Danai Laksameethanasan, and Yi-Ling Tung. Quantitative protein localization signatures reveal an association between spatial and functional divergences of proteins. *PLoS computational biology*, 10(3), 2014.
7. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
8. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.