Incorporating prior knowledge of protein localization in a neural network for protein location prediction.

Iwan Hoogenboom

Supervisor: Marcel Reinders

June 28, 2019

Abstract

Determining protein subcellular location is important for understanding cellular functions and biological processes of underlying diseases. High throughput fluorescence images can be used in combination with convolutional neural networks to predict this location. In this work we propose a hierarchical model which uses prior knowledge of proteins to divide the samples in general groups before predicting the subcellular location. Results show mixed results with significant improvements for some labels and a decline in results for others.

1 Introduction

Knowing the subcellular location of proteins within human cells can help in determining and understanding protein functions. This helps researchers in the understanding of complex diseases like cancer and Alzheimer's [1]. With the help of the human cell atlas [2] a large amount of data has become available. This data consists of microscopy images of human cells where the proteins and some reference locations are highlighted with the use of immunofluorescence. An example of such an image is shown in Figure 1. The classical method to examine these images is tedious and time-consuming. There are other methods to predict protein localization, these methods usually work with amino acid sequences instead of microscopy images. To the best of our knowledge there currently is no sufficiently good automatic system available to predict protein localization based on microscopy images.

The research of the use of Convolutional Neural Networks (CNNs) to predict protein location in cells is a new and ongoing research subject. There are a few examples of CNNs being used to predict the protein location in human cells [3, 4], a few more for predicting in yeast cells [5, 6, 7, 8] and recently there has been a kaggle competition [9] to spark interest and increase the amount of attention focused on this issue. So far all the research has been into improving a single network to fit the main challenges of the data. These are underrepresented classes, weakly annotated data and the variance of protein locations. These challenges will be discussed in depth in a later section.

The microscopy images of the human cell atlas have up to 33 different labelled locations where the protein could localize to. These locations are correlated, they have a very clear property which can be used: the locations can be grouped in three general locations: Cytoplasm, Secretory and Nucleus (Figure 1). This grouping is based on the general areas in a human cell according to the protein atlas [10]. These locations are a very simple but useful way of grouping the data and the correlation between locations is a important form of prior knowledge.

Nucleus	Cytoplasm	Secretory
Nuclear membrane	Actin filaments	Endoplasmic reticulum
Nucleoli	Focal adhesion sites	Golgi apparatus
Nucleoli fibrillar center	Centrosome	Cell junctions
Nuclear bodies	Microtubule organizing center	Plasma membrane
Nuclear speckles	Aggresome	Secreted proteins
Nucleoplasm	Cytoplasmic bodies	Endosomes
	Cytosol	Lipid droplets
	Rods & rings	Lysosomes
	Intermediate filaments	Peroxisomes
	Cleavage furrow *	Vesicles*
	Cytokinetic bridge	
	Microtubule ends	
	Microtubules	
	Midbody *	
	Midbody ring *	
	Mitotic spindle	
	Mitochondria	

Table 1: All the labels from the cell atlas, grouped in the three larger locations: Nucleus, Cytoplasm and Secretory. The locations with a star (*) are not in the kaggle dataset.

To the best of our knowledge no one has tried to incorporate the prior knowledge of protein localization in combination with the immunofluorescence images. The use of prior knowledge in Neural Networks is an active research area and it has been shown that it sometimes can be effective at improving the results [11]. An advantage of using prior knowledge is that it can reduce the data dependency of CNNs. The prior knowledge will help in detecting patterns where there is not enough data for the network to find these by itself.

The aim of this research is to investigate whether the predicted protein location can be improved with the use of prior knowledge. The way this will go is by using the grouping to train different CNNs for the sub classes. An advantage of using specialized networks is that these networks can focus on the small difference between similar classes. A large network that has to predict for all classes will be worse in distinguishing small differences between similar classes. Specialized networks can train specifically on the differences between the similar classes and will therefore be better at distinguishing them. An illustration of this idea can be seen in figure 3

In this paper we propose a model that uses a Hierarchical CNN structure to incorporate prior knowledge. The results are mixed with some classes performing better and other performing worse.

2 Methodology

2.1 General Idea

The approach to incorporating prior knowledge will be by grouping the protein based on labels that are related. The grouping can be seen in Table 1. First a network (stage 1)

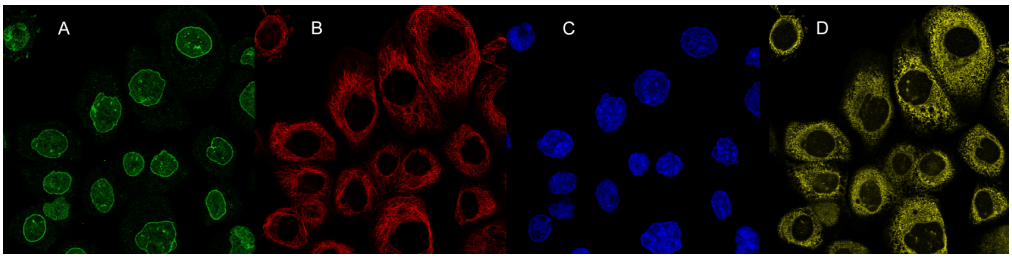

Figure 1: A sample from The Cell Atlas with the channels shown side-by-side. In green(A): target protein; in red(B): microtubules; in blue(C): nucleus; In yellow(D): endoplasmic reticulum.

will have to decide in which of the general locations (Nucleus, Cytoplasm or Secretory) the protein of interest is localized. After this a different network (stage 2) that is trained on only samples from the general location will be used to classify the more accurate subcellular location. In the case where the the stage 1 network predicts multiple locations the labels from the stage 2 networks will be added together to create a multi label prediction (Figure 3.

2.2 Data

The data used are the immunofluorescence images from the Cell Atlas [10]. The specific samples were acquired from the kaggle competition page [9]. Every sample consists of 4 different images as can be seen in Figure 1. There is one image highlighting the protein of interest (A). The other three are reference images highlighting the micro-tubules (B), nucleus (C) and endoplasmic reticulum (D). In total, there are 31.072 samples, each one consisting of the four different images, and a list of labels that belong the the sample. We do not know the name of the protein that is being marked, so this can not be utilized in the prediction.

The kaggle dataset does not have all the labels that are in the human cell atlas. They only have 28 labels instead of the complete 33. The data set itself still has the same difficulties as the data from the Cell Atlas, these challenges are: underrepresented classes, weakly annotated data and the variance of protein locations.

Underrepresented classes: The locations and thus the labels are not evenly represented in the data. A plot of the location distribution can be seen in Figure 2A. This class imbalance is problematic because the network will have many samples of the majority classes to train on, it will however not have enough samples to train at classifying the minority classes. This will cause the model to become biased towards the majority classes. To combat this class imbalance, the samples which do not occur at least 1% are filtered. This means removing seven (Rods & Rings, Microtubule ends, Lysosomes, Endosomes, eroxisomes, Lipid droplets and Mitotic spindle) least occurring classes from the data set.

Weakly annotated data: The weak annotation in this setting is the fact that an image contains multiple cells, and possibly has multiple labels. This is a challenge because the subcellular localization of the protein in the cells in the same sample image might vary. This means that some labels might only apply to a few cells and not all of them. This makes it harder for the network to detect the correct patterns and classify the images correctly. As

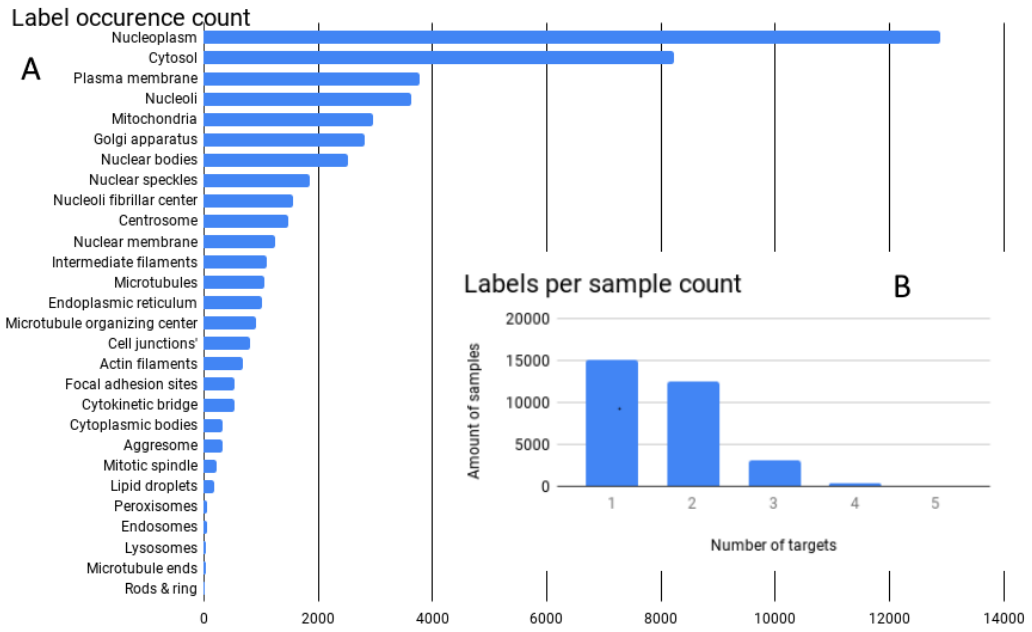

Figure 2: **A.** Label distribution in the kaggle data set. It shows a large class imbalance. From 12885 samples for Nucleoplasm to 11 for Rods & Rings. **B.** Amount of labels for each sample. More than half the samples have multiple labels.

a consequence of this, it is not suitable to apply some data augmentation methods such as multiple cropping from input image since the cropped part might not include all the labels of full image.

Varience in protein location: Proteins do not only localize to only one subcellular location. Stadler et al. found that 60% of the proteins they were investigating localized to multiple locations [12]. Because the proteins can move around and the images are single snapshot of the cell it is possible for the same cell contains protein in multiple locations. Therefore it is important to do a *multi label* prediction for the subcellular localization. Figure 2B shows the number of labels for each sample in the kaggle dataset. It is clear that only half of the proteins localize to only one location.

During the preprocessing phase the pixel values of the images were normalized to values between 0 and 1, then the pictures were scaled down to a size of 256x256. Out of the data set the labels with less than 1% representation were removed. This removed 7 labels to result in 21 labels total. This resulted in a total of 30.560 samples, which are randomly split into test, training and validation sets. 60% of the data is used for training, 20% for validation and 20% for testing.

2.3 Models

To be able to compare the performance of the proposed model, there needs to be a baseline to compare with. The main focus of this research is to improve the classification by using

the prior knowledge. To be able to compare the influence of the prior knowledge, it means that both the baseline and the proposed model need to be similar, with the only difference being the added prior knowledge.

The baseline model uses the ResNet18 [13] model adapted for multi label classification. ResNet is chosen because it is allows to train deep networks with great success [14]. Another advantage of using ResNet is that the model trains relatively fast. This input layer was adapted to fit the 4 layer input images. The loss function used was MultiLabelSoftMarginLoss in combination with a multi label binary output format. This loss function creates a criterion that optimizes a multi-label one-versus-all loss based on max-entropy, between input x and target y.

$$l(x, y) = -\sum_{n}^{i}(y_i log(\frac{exp(x_i)}{1 + exp(x_i)}) + (1 - y_i) log(\frac{1}{1 + exp(x_i)} \tag{1}$$

To generate predictions the output of the network was fed into a sigmoid function. A sigmoid function maps the input x to a value between 0 and 1. The final predictions where then made by applying a threshold of 0.5 to the output of the sigmoid: Every label with a value above 0.5 was classified as present and every label below was classified as not present.

$$S(x) = \frac{e^x}{e^x + 1} \tag{2}$$

The proposed model (Figure 3) consists of a multiple ResNet models. First the model (Stage 1) will distinguish between the three general labels: Nucleus, Cytoplasm and Secretory. Important to note is that because the classification problem is multi label, it has to be possible that a sample will classify into multiple of these three classes. After the general class is determined the sample image is fed into (at least) one of the three specialized models (Stage 2) corresponding to the correct class. These stage 2 models are again ResNet18 models but they are trained on the subset of samples where at least one of the labels fits within the general class. For example the stage 2 Nucleus network. This network will train on all the samples from the training set that contain at least one label (e.g. Nucleoli fibrillar center) that belongs to the general class of nucleus. The output of the stage 2 models is added together to give the final multi label prediction for the sample.

For the stage 1 network there are two different approaches. Approach one where the general locations are translated into 7 different outputs, one class for each combination of large locations (Nucleus, Cytoplasm, Secretory, Nucleus & Cytoplasm, Nucleus & Secretory, Cytoplasm & Secretory, Nucleus & Cytoplasm & Secretory). This network does a single label prediction, so it can be trained with the cross entropy loss function.

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{3}$$

Where:

- M is the number of classes.
- y is a binary indicator (0 or 1) if class label c is the correct classification for sample o
- p is the predicted probability sample o is of class c

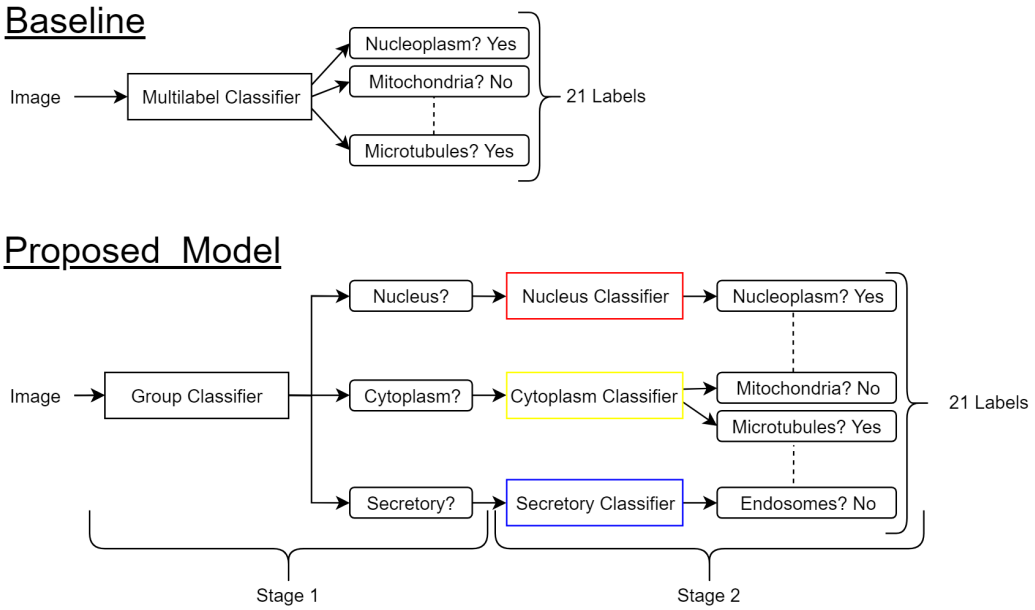

Figure 3: The baseline model and the proposed model. The baseline consists of one ResNet18 with a sigmoid function after the last fully connected layer. The proposed model consists of multiple ResNet18 classifiers.

The second approach is to have the stage 1 network do multi label classification, this network again uses the MultiLabelSoftMarginLoss and the sigmoid function as described above.

The stage 2 network will have to do multi label prediction, so it will use MultiLabelSoftMarginLoss as loss function. The stage 2 network has two approaches as well; Approach one is to train these networks on the specific samples that belong to that network. Approach two does the same but also includes an additional label 'other'. This 'other' label has the purpose to allow for corrections when the stage 1 network makes a mistake. To be able to train the 'other' class, new samples were added to the training set. These samples were randomly chosen out of all the samples that did not contain any of the current network's labels. For example when training the stage 2 Nucleus network, a sample with labels Cytoplasmic bodies and Cytosol would be added as a sample with label 'other'. For every stage 2 network the amount of 'other' samples was determined by taking the average of the number of samples.

To increase traing speed, the stage 2 networks are trained with the use of transfer learning. Transfer learning is a technique used to speed up training. Instead of training the stage 2 networks from scratch, they started with the weights of the stage 1 network. The last fully connected layer was adapted to suit the required amount of labels.

2.4 Training

In the training phase, early stopping was implemented. Every epoch the validation loss was checked. If it was the lowest loss the model weights would be stored. If the validation loss

was higher than a previous value a counter started. If the loss did not get to a new low in 6 epochs the model's weights were reset back to the values that gave the lowest validation loss. Then learning rate was reduced by a factor 10. If then in the next 6 epochs the validation loss did not go lower than the current best the training was terminated and the model with the lowest validation loss was stored.

2.5 Evaluation Metrics

Comparing the results of the baseline and the proposed model will be in recall, precision and F1 score. Recall is a measure to calculate the fraction of a class that the network is able to identify. Precision calculates the fraction of predictions that is correctly identified. The F1 score combines these two values to give an overall insight in performance of the network

$$Recall = \frac{TruePositive}{TruePositive + FalsePositive} \tag{4}$$

$$Precision = \frac{TruePositive}{TruePositive + FalseNegative} \tag{5}$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{6}$$

The measures will be calculated per protein. An important thing to note is that because of the multi label prediction the accuracy measure is not sufficient. The network has 28 possible classes to choose from but the sample only belong to at most 4 labels at the same time. This means that when the network predicts no classes at all for every sample it will have an accuracy of at least: $(28 - 4)/28 = 0.857$. This value might seem good but the network itself is useless when it does not predict anything. Another problem with accuracy is the large class imbalance. This causes a very high accuracy when the network is only good at predicting the most occurring classes. This is why the recall, precision and F1 are more important in measuring the performance.

There are flaws with using the F1 measure [15]. One of the flaws of F1 score is that it is biased towards a class majority. If the network only predicts based on the frequency of labels occurring, frequent occurring labels will have a higher chance of getting a high F1 score. This is certainly a problem with Nucleoplasm making up roughly 25% of the samples, while some other labels barely make up 1%. To more accurately compare the results of this imbalanced dataset we have to take the prediction advantage (PA) into account. To get a more insightful metric El-Yaniv [16] proposes a new formula:

$$PA = 1 - \frac{1 - F1}{1 - p} \tag{7}$$

Where:

- $F1$ is the F1 score
- p the frequency of the label occurring

This new PA score corrects for class imbalance by giving a value between 0 and 1. Where a value of 0 stands for the classifier is as good as guessing with class frequencies and a value of 1 is a perfect classifier. A value of 0.35 can then be interpreted as the classifier is 35% on the way between random with known class frequencies and a perfect classifier.

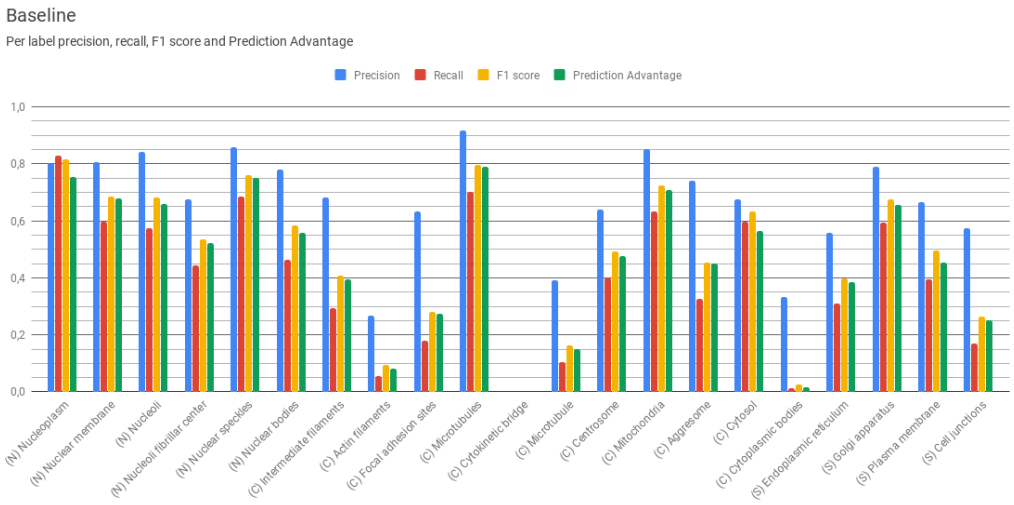

Figure 4: Precision (blue), recall (red), F1 score (yellow) and PA (green) for every protein predicted with the baseline method. Labels with a (N) belong the the Nucleus group, (C) the Cytoplasm group and (S) the Secretory

3 Results

The baseline had an average precision of: $0,6422$, average recall of: $0,3984$ and an average F1 score of: $0,4749$. The per protein results are visible in Figure 4. As can be seen there are a some labels (Cytokinetic bridge, Cytoplasmic bodies, Actin filaments) that the baseline has a lot of trouble with. It also shows that the classifier is actually predicting the labels since only the three most occurring labels (Nucleoplasm, Cytosol and Plasma membrane) have a reasonable drop in PA score vs F1 score.

The results from the stage 1 classifier, here there are two different models: The single label network and the multi label network. To be able to compare the networks and also give better insight in the results from the single label prediction, the output from the single label is converted back to a multi label format. This is done by converting both the prediction and label back to the original values. For example if the label Nucleus & Secretory gets predicted, this gets converted back to the multi label binary format of Nucleus: yes, Cytoplasm: no, Secretory: yes. The same happens to the labels. This way the results of both stage 1 networks are in a multi label format and can be compared. The single label and multi label results are indicated with SL and ML respectively. As can be seen in Figure 5 the multi label network performs roughly the same on Nucleus and Cytoplasm, but perform significantly better on classifying the Secretory. Because of clear superiority the final predictions will be made with only the multi label stage 1 network.

For the stage 2 networks there were two options: Option one was the network with the special label 'other' that gives the networks an option to correct a mistake by the stage 1 network, option two was the network that did not have this 'other' label. The results of the stage 2 networks are shown in Figure 6. For every stage 2 network the results of the network with 'other' class, the network without the 'other' class and the results that the baseline

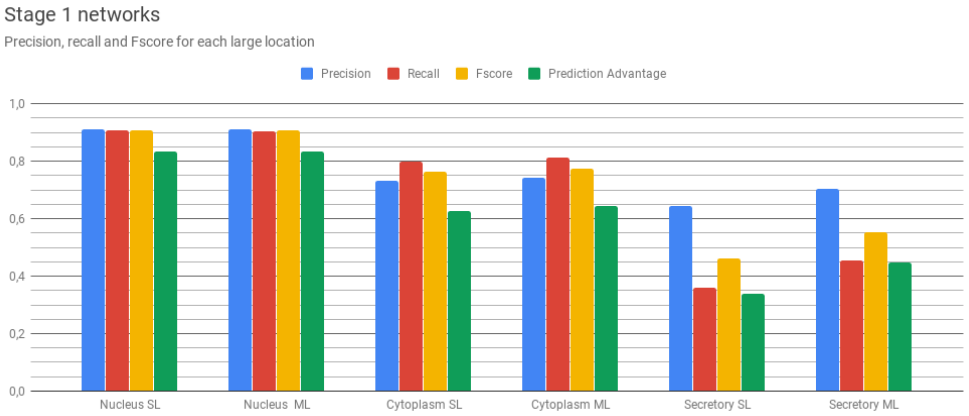

Figure 5: Precision, recall and F1 score for the three large location predicted with the singlelabel (SL) network, transformed back to the location specific scores, and the multilabel (ML) network.

predicted for these classes are shown. The stage 2 networks without the 'other' category scores better on almost every label compared to the baseline. Some classes (Actin filaments, Focal adhesions sites) show significant improvements of a PA score from 0.08 to 0.45 and 0.27 to 0.53. respectively. The secretory also has significant improvements compared to the baseline. The stage 2 network with the 'other' category do not perform as good. Only the Cytoplasm network manages to score similar to the baseline model. The Nucleus and Secretory model make significantly worse predictions than the baseline.

The results of the proposed model are shown i7. The results show that the proposed model improves significantly on some area's such as Actin filaments, but it loses a lot on the classes in the Secretory. What is visible is that the classes that did poorly in the stage 2 scores not always perform that much worse in the overall prediction. For example the Cytosol PA score from the network without the 'other' label is 0.64, where in the network with the 'other' label the PA score for Cytosol is 0.38. However in the proposed model both networks score very similar for Cytosol: 0.56 and 0.58 respectively. The opposite is also shown, Nucleoli fibrillar and Nuclear speckles scored very poorly in the 'other' network in stage 2 and this is clearly visible in the results when combining stage 1 and 2.

4 Discussion

In stage 1 network there is a clear winner. The multi label prediction performs equal or better than the single label. The reason for this could be that the Secretory class, which is already in a minority compared to the Nucleus and Cytoplasm, gets split into multiple labels and because of that becomes even harder to identify correctly.

In the stage 2 networks it is clear that the networks without the 'other' label classify better in most cases. Specially in the Nucleus and the Secretory the 'other' samples confuse the network, resulting in very low scores. When training these two networks the loss did not converge even when lowering the learning rate.

The results of the proposed model clearly show a difficulty of this hierarchical network

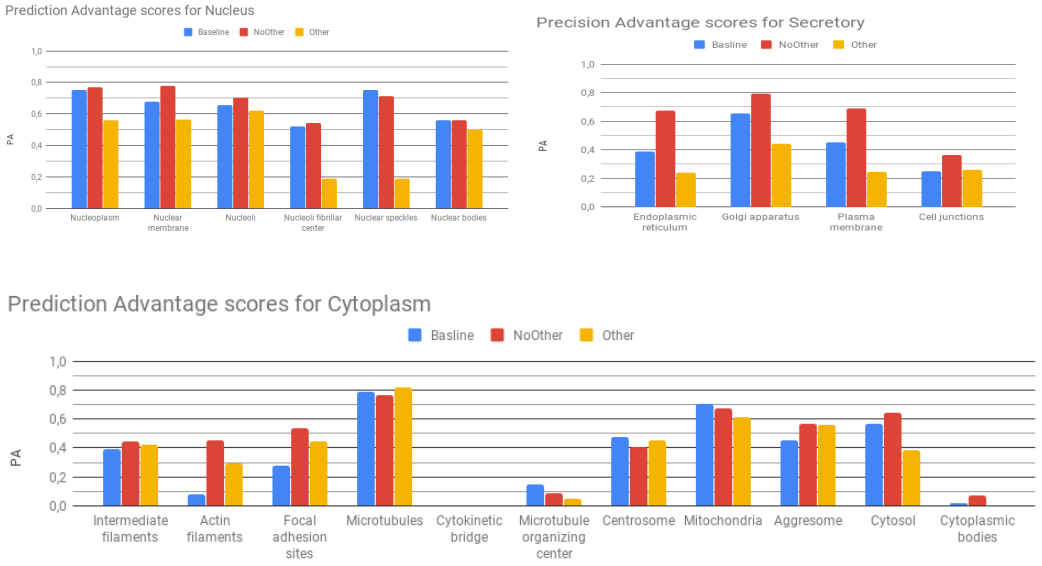

Figure 6: Results of the stage 2 networks. Prediction Advantage score is shown. Baseline results are shown in blue, the results of the network without the 'other' label are shown in red, the results of the network with the 'other' label are shown in yellow.

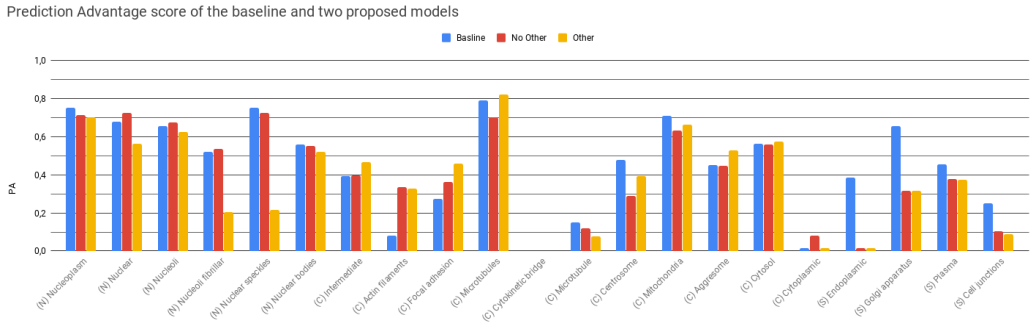

Figure 7: PA score for: the Baseline (blue), the proposed model with stage 2 models trained with out the 'other' label, the proposed model with stage 2 models trained with an 'other' label (yellow)

approach: both the stage 1 and stage 2 need to be good in order to improve the results. The results of the proposed model for the classes in the Secretory are explained by the low score of the stage 1 model for Secretory. Since the scores between the 'other' and 'No other' model in stage 2 are very different, but almost the same when using the proposed model, the bottleneck is the stage 1 classifier. The other way around is also the case: The stage 1 classifier is good at classifying samples that belong the the Nucleus. However the results from the proposed model with the 'other' label show that for the Nucleoli fibrillar and Nuclear speckles the stage 2 model is the bottleneck. Those classes did not as good in stage two and it is visible in the results. When both models are good it does show that significant improvement of the baseline is possible (Actin filaments and Focal adhesion).

A suggestion for further research would be to compare this ResNet18 setup vs a deeper network for example: ResNet50. Since the proposed model has twice the amount of layers as the baseline it would be interesting to compare it with a model which has roughly equal amount of layers. Another things that could be interesting to look into is to group the Cytoplasm and Secretory into one category. This way the first stage classification becomes a fairly simple problem, this can help in making the first stage become sufficiently good that it might be possible to leave out the 'other' category in stage 2. Since the results already showed that the 'other' category does make it more difficult for the stage 2 networks.

To conclude, the current proposed model shows mixed results. Some locations perform significantly better with the proposed model, while others perform worse.

References

[1] Mien-Chie Hung and Wolfgang Link. Protein localization in disease and therapy. *Journal of Cell Science*, 124(20):3381–3392, oct 2011.

[2] Peter J. Thul, Lovisa Akesson, Mikaela Wiking, Diana Mahdessian, Aikaterini Geladaki, Hammou Ait Blal, Tove Alm, Anna Asplund, Lars Björk, Lisa M. Breckels, Anna Bäckström, Frida Danielsson, Linn Fagerberg, Jenny Fall, Laurent Gatto, Christian Gnann, Sophia Hober, Martin Hjelmare, Fredric Johansson, Sunjae Lee, Cecilia Lindskog, Jan Mulder, Claire M. Mulvey, Peter Nilsson, Per Oksvold, Johan Rockberg, Rutger Schutten, Jochen M. Schwenk, Asa Sivertsson, Evelina Sjöstedt, Marie Skogs, Charlotte Stadler, Devin P. Sullivan, Hanna Tegel, Casper Winsnes, Cheng Zhang, Martin Zwahlen, Adil Mardinoglu, Fredrik Pontén, Kalle Von Feilitzen, Kathryn S. Lilley, Mathias Uhlén, and Emma Lundberg. A subcellular map of the human proteome. *Science*, 356(6340), 2017.

[3] Elisabeth Rumetshofer, Markus Hofmarcher, Clemens Röhrl, Sepp Hochreiter, and Günter Klambauer. Human-level Protein Localization with Convolutional Neural Networks. In *ICLR 2019 Conference Blind Submission*, pages 1–14, 2019.

[4] Yijun Tian. TUNet: Incorporating segmentation maps to improve classification. *arXiv e-prints*, jan 2019.

[5] Oren Z Kraus, Ben T Grys, Jimmy Ba, Yolanda Chong, Brendan J Frey, Charles Boone, and Brenda J Andrews. Automated analysis of high-content microscopy data with deep learning. *Molecular Systems Biology*, 13(4):924, apr 2017.

[6] Mengli Xiao, Xiaotong Shen, and Wei Pan. Application of deep convolutional neural networks in classification of protein subcellular localization with microscopy images. *Genetic Epidemiology*, 43(3):330–341, apr 2019.

[7] Tanel Pärnamaa and Leopold Parts. Accurate Classification of Protein Subcellular Localization from High-Throughput Microscopy Images Using Deep Learning. *G3: Genes|Genomes|Genetics*, 7(5):1385–1392, may 2017.

[8] Oren Z. Kraus, Jimmy Lei Ba, and Brendan J. Frey. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52–i59, jun 2016.

[9] Kaggle. Human Protein Atlas Image Classification, 2018.

[10] ProteinAtlas. The Cell Atlas.

[11] Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. Combining knowledge with deep convolutional neural networks for short text classification. In *IJCAI International Joint Conference on Artificial Intelligence*, 2017.

[12] Charlotte Stadler, Elton Rexhepaj, Vasanth R Singan, Robert F Murphy, Rainer Pepperkok, Mathias Uhlén, Jeremy C Simpson, and Emma Lundberg. Immunofluorescence and fluorescent-protein tagging show high correlation for protein localization in mammalian cells. *Nature Methods*, 10(4):315–323, apr 2013.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. Technical report, Microsoft Research, 2016.

[15] David M W Powers. What the F-Â-measure doesn't measure... Features, Flaws, Fallacies and Fixes. Technical report, Beijing University of Technology.

[16] Ran El-Yaniv, Yonatan Geifman, and Yair Wiener. The Prediction Advantage: A Universally Meaningful Performance Measure for Classification and Regression. Technical report, Technion Israel Institute of Technology, 2017.