HUMAN-LEVEL PROTEIN LOCALIZATION WITH CONVOLUTIONAL NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Localizing a specific protein in a human cell is essential for understanding cellular functions and biological processes of underlying diseases. A promising, low-cost, and time-efficient biotechnology for localizing proteins is high-throughput fluorescence microscopy imaging (HTI). This imaging technique stains the protein of interest in a cell with fluorescent antibodies and subsequently takes a microscopic image. Together with images of other stained proteins or cell organelles and the annotation by the Human Protein Atlas project, these images provide a rich source of information on the protein location which can be utilized by computational methods. It is yet unclear how precise such methods are and whether they can compete with human experts. We here focus on deep learning image analysis methods and, in particular, on Convolutional Neural Networks (CNNs) since they showed overwhelming success across different imaging tasks. We propose a novel CNN architecture "GapNet-PL" that has been designed to tackle the characteristics of HTI data and uses global averages of filters at different abstraction levels. We present the largest comparison of CNN architectures including GapNet-PL for protein localization in HTI images of human cells. GapNet-PL outperforms all other competing methods and reaches close to perfect localization in all 13 tasks with an average AUC of 98% and F1 score of 78%. On a separate test set the performance of GapNet-PL was compared with a human expert. GapNet-PL achieved an accuracy of 91%, significantly (p-value $2e^{-10}$) outperforming the human expert with an accuracy of 61%.

1 INTRODUCTION

Proteins perform their function at specific times and locations within a cell. Understanding in which cell organelles, such as nucleus or mitochondria, a protein is located, is fundamental in understanding biological processes (Thul et al., 2017). The identification of mislocalized proteins may hint at cellular dysfunctions, advancing our knowledge about diseases. Microscopy-based methods assess changes in the sub-cellular localization or the abundance of proteins which helps to reveal their interactions in human cells.

Currently, the Human Protein Atlas project (HPA) is dedicated to annotating the location of all human proteins in a cell using a large battery of biotechnologies and approaches, such as microarrays, confocal microscopy, knowledge from literature, bioinformatics predictions and additional experimental evidence, such as western blots, or small interfering RNA knockdowns (Uhlen et al., 2010). A complementary and highly promising biotechnology that is used to localize proteins, is high-throughput fluorescence microscopy imaging (HTI), which is characterized by low costs and being time efficient (Pepperkok and Ellenberg, 2006). With this imaging technology, a selected protein can be stained with fluorescent antibodies and subsequently a microscopic image of the whole cell is taken. Together with the information of other stainings, such as the Hoechst staining of the cell nucleus, and the actin staining of the cytoskeleton, these images provide a rich source of information on the protein location (Stadler et al., 2013). Human experts are able to utilize those HTI images to localize proteins (Swamidoss et al., 2013). It is yet unclear how precise computational methods are and whether they achieve the performance level of human experts. High performance in localizing proteins is expected from deep learning methods since imaging data together with the annotation by the HPA project are an auspicious source of training data.

Convolutional Neural Networks (CNNs) are state-of-the-art for image analysis in a variety of fields (Krizhevsky et al., 2012), with variants proposed that increase performance significantly, e.g., DenseNet (Huang et al., 2017), ResNet (He et al., 2015), and Fully Convolutional Network (Long et al., 2015). Recently, CNNs have been successfully applied to analyze biological and medical images, in tasks such as detection of melanoma, performing on par with dermatologists (Esteva et al., 2017; Haenssle et al., 2018) or the automatic identification of malaria (Poostchi et al., 2018). Several authors have recently employed CNNs for high-throughput microscopy data (Dürr and Sick, 2016; Kraus et al., 2016; 2017; Pärnamaa and Parts, 2017; Godinez et al., 2017), however certain characteristics of HTI data, such as relatively high resolution, prohibit application of standard models. As a result, the majority of the approaches focus on single-cell crops, i.e. segmented images containing a single cell, and thereby depend on data pre-processing steps. Thus, data pre-processing and cell segmentation algorithms still play a large role in HTI analysis.

With the big leap in performance caused by CNNs, comparisons of computational methods with humans or even human experts have recently appeared in literature. ResNet(He et al., 2015) has reached human level performance in image classification of general images. Esteva et al. (2017) have compared CNNs with expert performance at detecting melanoma in images of skin lesions. In the area of protein localization, Swamidoss et al. (2013) have conducted a comparison with two human experts. They worked with microscopy data on tissue level belonging to four classes. The performance of the machine learning approaches was still below the performance of the human experts. However, this pre-dates the rise in popularity of CNNs and therefore the authors extract features via several different methods and apply support vector machines (SVM) and Linear Discriminant (LDA) classifiers on top of these features. In this work, we aim at a more challenging task for both human experts and computational methods since proteins have to be localized within 13 classes with multiple possible locations per sample.

The ideal approach to protein localization would be a generic, robust and fully automated pipeline that is as accurate or even more accurate than a human expert. Even more desirable would be a platform to which HTI images could be uploaded and automatically annotated. In contrast, the current state-of-the-art in protein localization involves experts adjusting the segmentation algorithm, cell crop extraction and then applying Deep Learning on those with performance being supposedly below human experts. We offer a general approach that uses arbitrarily large input sizes, does not need segmentation but rather works on whole images and performs at the level of human experts or even above. With increased size and variability of training sets we suppose that the architecture could be used for localizing proteins in images from diverse biotechnological devices, cell lines and labs.

In the following we apply current state-of-the-art methods as well as previous purpose-built approaches to high resolution images. We assess and compare their performance on the largest available public dataset of high resolution HTI data in the field of subcellular protein localization in human cells. Furthermore, we compare the performance of a human expert to the best performing network.

2 DATASET

All experiments were conducted on a dataset released for the "Cyto Challenge 2017" by the Congress of the International Society for Advancement of Cytometry (ISAC). The main challenge dataset contains 20,000 samples taken from the Cell Atlas (Thul et al., 2017) which is part of the Human Protein Atlas. The Cell Atlas contains images from antibody-based profiling using immunofluorescence confocal microscopy capturing multiple cells per image. These high resolution images of around 12,000 proteins across more than 20 cell lines are taken from different organs of the human body. Thus, the cell atlas is a rich source of cell images of high diversity with respect to shape, cell type, amount of cells and spatial relation of cells, which we use to perform our experiments on.

In the dataset, every sample consists of four high-resolution images corresponding to the different fluorescent channels. The channels represent three reference channels for different stained subcellular structures (nucleus, endoplasmic reticulum and microtubules) and one channel with the stained protein of interest (see Figure 1 and 2). For each sample, the task is to determine in which of the 13 major organelles the protein of interest appears, where multiple organelles are possible. Thus, each

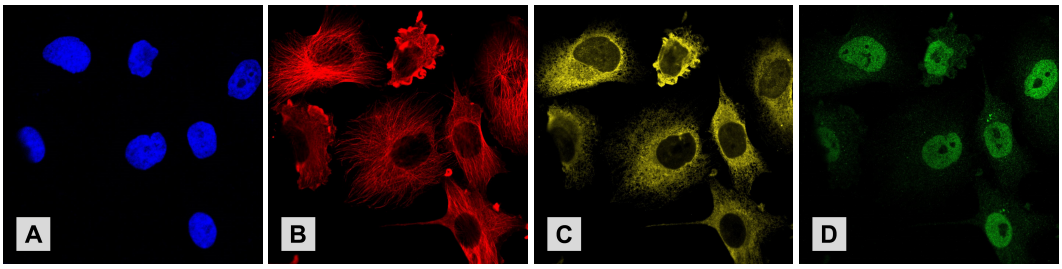

Figure 1: The four fluorescent channels of a sample from the dataset, namely (A) nucleus, (B) microtubules, (C) endoplasmic reticulum, (D) protein.

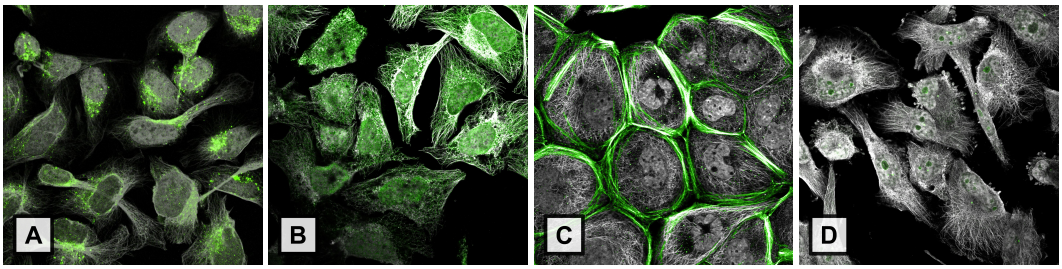

Figure 2: Exemplary samples from the dataset. Overlay of reference channels in gray and protein in green. The protein of interest is located in (A) golgi apparatus and vesicles, (B) cytosol, nucleus and plasma membrane, (C) actin filaments, (D) nucleoli and centrosome.

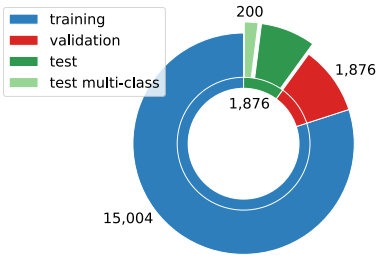

Figure 3: Visualization of the datasets. We used an 8:1:1 split of dataset into training (15,004 samples), validation (1,876 samples) and test (1,876 samples) sets. An additional test set was defined for the comparison with a human expert, which is a subset of 200 samples from the test set with single-class labels.

sample is associated with a binary label vector indicating the presence or absence of the protein in each of these 13 organelles. Most importantly, these labels have not been derived from the given microscopy images, but from other biotechnologies such as microarrays or from literature. In approximately 50% of the samples, the protein appears in more than one organelle and the distribution of proteins across organelles is slightly unbalanced (see Appendix Figure 6).

The dataset was provided as a collection of high resolution TIFF-files for every channel. Samples with at least one corrupted channel were removed which reduced the dataset from 20,000 to 18,756 samples. Due to differences in the capturing process channel-images are encoded differently, e.g., in different color channels and with varying dimensions (1728x1728, 2048x2048, 3072x3072) and were converted to grayscale if necessary. In a subsequent step these were normalized to zero mean and unit variance. To keep the quality of the original images and not lose finegrained details from interpolation, samples were not resized but kept at their original resolution. The pre-processed samples were then randomly split into a training (80%), a validation (10%) and a test (10%) set. For the comparison with a human expert a subset of 200 examples, in which the protein appears in only one organelle ("multi-class" rather than "multi-task"), was chosen from the test set (see Figure 3).

3 METHODS

GapNet-PL We propose an architecture designed specifically to process high-throughput microscopy images. Our architecture, which we call GapNet-PL, can deal with high resolution images

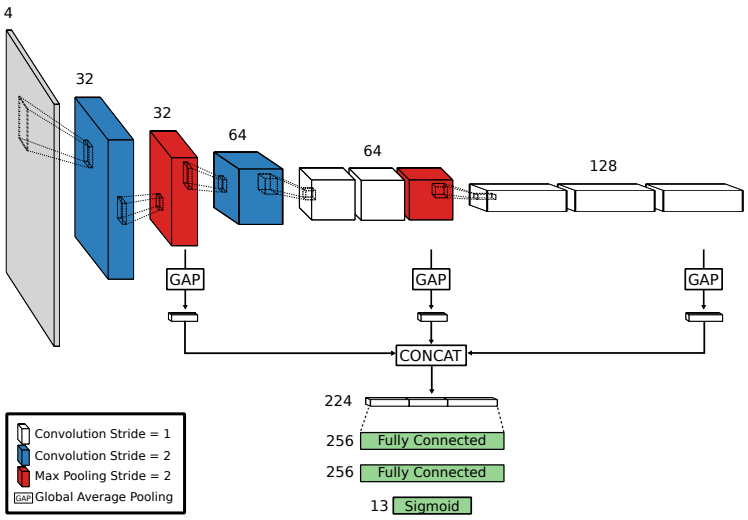

Figure 4: GapNet-PL architecture

and therefore has the possibility to learn from fine structures within images as they do not have to be downscaled. We achieve this via a two-step approach. In a first step we use an encoder consisting of several convolutional layers, some with a stride of two, interspersed with max-pooling layers to learn abstract features on different spatial resolutions.

In the second step, we reduce the feature maps from three different layers via *global average pooling* to a size of one pixel and concatenate the resulting feature vectors. This pooling operation can also mitigate the influence of noisy labels. The resulting features, representing different spatial resolutions, are then passed on to a fully connected network with two hidden layers for the final prediction. This architecture has a relatively low number of parameters (in our case around 600,000) and can be tailored to a specific task very easily, e.g., by adding more convolutional blocks in the encoder and passing more features to the second stage. As a side effect, this architecture can process inputs of arbitrary spatial resolution on inference.

For efficient training and especially to reduce memory consumption, which is a problem specific to high-resolution images, we use the SELU activation function (Klambauer et al., 2017) instead of Rectified Linear Unit (ReLU) with Batch normalization. As a result, our architecture can be trained on high-resolution images with very short training times when compared to other architectures while achieving new state-of-the-art performance on our task. For our experiments we use an initial learning rate of 0.01, dropout in the fully connected layers of 30% and a batch size of 40.

DenseNet Densely Connected Convolutional Networks (Huang et al., 2017) are currently among the best architectures for a variety of image processing tasks. The basic idea of DenseNet is to reuse features learned on early layers of a network, containing fine-grained localized information, on higher layers which have a more abstract representation of the input. This is achieved by passing feature maps of a layer to all consecutive layers (within certain boundaries). A stated benefit of this architecture is that it does not have to re-learn features several times throughout the network. Hence, the individual convolutional layers have a relatively small number of learned filters.

The authors present different variations of the architecture. Due to memory restrictions the smallest variant, DenseNet-121 (see Appendix Table 5), was chosen for the comparison in this work. The reduction rate of transition layers is set to 0.5 and the growth rate is $k = 32$. As proposed, Batch normalization and ReLU activation function are applied before every convolutional layer. For training a batch size of 3 and an initial learning rate of 0.1 was determined.

Convolutional Multiple Instance Learning (Convolutional MIL) In Kraus et al. (2016) the authors introduce a CNN designed specifically for HTI data with a focus on the problem of noisy labels, i.e. that microscopy images not only contain cells of the target or labeled class but also cells that do not comply with the label. The authors propose to tackle this problem with *multiple instance*

learning (MIL), where cells belonging to the class label of an image are identified automatically while the influence of other cells on the result of the model is down-weighted by using a special pooling function called *noisyAND*.

This function follows the assumption, that a sample belongs to a class if the number of cells of this class exceeds a threshold which is learned for each class individually. The authors implement their model using a fully convolutional approach (FCN) allowing them to train on full images with noisy labels and apply this model to images of single-cell crops.

Furthermore, the pooling function introduces an additional hyper-parameter a controlling the slope of the function. Suggested values for a in Kraus et al. (2016) are 5, 7.5 and 10. We use a = 5 for our final model after a comprehensive hyper-parameter search showed this to be the best value for our data. We train the model with an initial learning rate of 0.01 and a batch size of 32.

Multi-scale Convolutional Neural Network (M-CNN) The M-CNN model (Godinez et al., 2017) is designed for phenotype classification of human HTI data and was benchmarked on 8 separate datasets, half of them dealing with protein localization for different cell types (HeLa, CHO, A-531 and MCF-7). The main idea of the architecture is to combine features extracted from the input at several spatial resolutions. This is achieved by scaling the original image dimensions to $width/s$ and $height/s$, where $s \in \{1, 2, 4, 8, 16, 32, 64\}$. These scaled versions of the input are processed by three convolutional layers and the feature maps of the last layer, respectively, are downscaled via pooling to the smallest resolution. Then, the feature maps are concatenated and combined via 1x1 convolution and passed on to a fully connected layer and the output layer. For this model we determined a batch size of 22 and an initial learning rate of 0.01.

DeepLoc DeepLoc (Kraus et al., 2017) was designed for subcellular protein localization in yeast cells. The network consists of 8 convolutional layers in total, with max pooling layers after the second, fourth and last layer. The proposed model was trained on small crops of size 64x64 centered on x and y coordinates of cells extracted via CellProfiler (Carpenter et al., 2006) from larger images. Since the model was designed for considerably smaller inputs we had to adapt the architecture to make training possible (see Appendix Table 6 for detailed changes to the original architecture). Furthermore, we replaced ReLU with Batch normalization with SELU since it performed better in our experiments. The model is trained with batch size of 2 and an initial learning rate of 0.001.

3.1 Model Training and Evaluation

To conduct a fair comparison we re-implemented all methods and optimized the most relevant hyper-parameters for each method. Since not all methods are designed for large inputs, a comparison on full resolution images is not feasible due to memory restrictions. However, the nature of our data imposes limitations on small crops. First, very large or elongated cells are often not captured in full. Second, the problem of noisy labels gets amplified, as a label might not conform to the whole crop, whereas on high resolution the label at least corresponds to parts of the image. Experiments with different crop sizes corroborate this (see Appendix Figure 9). Furthermore, Liimatainen et al. (2018), the last years' challenge winner, also trained on crops containing multiple cells with relatively high resolution.

We train all models on 1024x1024 pixel crops, where one random crop per training sample is extracted in every epoch. With the exception of DeepLoc, every method in our comparison is designed for processing such large inputs. For DeepLoc, we also tested smaller input sizes of 72x72 and 224x224 pixels and subsequent averaging the predictions of those smaller crops, which performed worse (see Appendix Figure 9). For validation and testing the whole image is used by cropping several 1024x1024 pixel patches covering the whole image. A final prediction for a sample is derived by applying average pooling over the predictions of all patches. All methods consider this problem as a multi-task setting of 13 binary classification tasks. Therefore, all networks contain an output layer with 13 units, sigmoid activations and cross entropy loss.

The batch size for the models depends on their memory consumption and is chosen as large as possible to fit on a typical GPU with 11GB memory (NVIDIA GTX 1080 Ti). As an optimizer we use Stochastic Gradient Descent (SGD) with momentum of 0.9. To stabilize training we use gradient clipping where the gradients are normalized if their global L2 norm exceeds a certain threshold

(which is set to 5). For optimal results we use a learning rate schedule, specifically the learning rate is halved when model performance plateaus. The initial learning rate is determined via a hyper-parameter search on the validation set. To avoid overfitting the following regularization techniques are applied: L1 norm of $1e^{-7}$, L2 norm of $1e^{-5}$. Dropout is used as proposed in the respective publications. All models are trained until convergence and the checkpoint with best performance on the validation set is used for inference on the test set.

3.2 Comparison with Human Expert

To determine the practical applicability of our model, we conducted a comparison of the GapNet-PL performance with a human expert in pathobiochemistry who frequently works with flourescence microscopy images. For this comparison, we consider only samples where the protein binds to a single location within cells in order to simplify the task and decrease workload for the human expert. We fine-tuned the last layer of our model, which was trained in a multi-task setting, on a multi-class subset (11,848 samples) of the training set.

In a first interactive session, the expert was given the opportunity to investigate the training set *ad libitum*. More precisely, we provided the labels and all channels separately, and the possibility to create overlays with selected channels for the samples. The human expert stated that he would be capable to localize proteins with the provided data. He chose to obtain four images, each of the three channels with protein, and the protein channel alone, per sample in the test set (see Appendix Figure 7). The expert was then given 200 samples from our held-out test set for labeling in a second non-interactive session in his typical working environment. We are aware that the design of this study could suffer from potential biases (see discussion Section 5) similar to other comparisons of AIs against human experts.

4 Results

The results of our method comparison are shown in Table 1, which also includes the results of last years' winner of the "Cyto Challenge 2017". Liimatainen et al. (2018) propose a fully convolutional network (FCN) trained on single-cell crops of multi-class samples which produces confidence maps that are then individually thresholded per task to get final predictions.

We found that the average performance across the 13 tasks of the compared methods ranges from an area under ROC (AUC) of 91% to 98%, with a mean AUC of 95%. Since for imbalanced learning scenarios the AUC may provide a too optimistic view of the models we also report the average performance of F1 score, precision and recall. The average F1 score over all tasks comprises a larger range from 47% to 78% with a mean of 65%. The table shows that the predictive performance of GapNet-PL, M-CNN and Dense-Net-121 are closer together and the remaining methods are behind by a large margin. We used a one-sided paired Wilcoxon test to test if our results are statistically significant. Overall, GapNet-PL significantly outperforms all compared methods

Method	F1 Score	p-value	Precision	Recall	AUC
GapNet-PL	**0.78 ± 0.09**		0.84	**0.75**	**0.98 ± 0.02**
M-CNN	0.75 ± 0.12	0.0232	**0.90**	0.66	0.97 ± 0.02
DenseNet-121	0.73 ± 0.12	0.0434	0.75	0.74	0.97 ± 0.02
DeepLoc	0.52 ± 0.28	0.0009	0.79	0.45	0.91 ± 0.07
Convolutional MIL	0.47 ± 0.31	0.0007	0.71	0.40	0.94 ± 0.06
Liimatainen et al. (2018)[1]	0.51	-	0.45	0.68	-

[1] performance estimate calculated on a different test dataset

Table 1: Performance comparison of different CNN architectures for protein localization. For each method, the table reports the average F1 score and its standard deviation across 13 prediction tasks. The third column, p-value, reports the result of a one-sided paired Wilcoxon test with the null hypothesis that GapNet-PL and the respective method perform equally. The consecutive columns report average precision, recall and area under ROC (AUC). GapNet-PL has outperformed all competing methods.

Method	GapNet-PL	M-CNN	DenseNet-121	DeepLoc	Convolutional MIL
Actin filaments	0.682	0.651	**0.719**	0.125	0.000
Centrosome	**0.659**	0.522	0.598	0.039	0.000
Cytosol	0.776	**0.805**	0.789	0.784	0.735
Endoplasmic reticulum	**0.672**	0.629	0.645	0.538	0.338
Golgi apparatus	**0.754**	0.689	0.750	0.428	0.331
Intermediate filaments	**0.800**	0.625	0.490	0.000	0.000
Microtubules	0.767	**0.901**	0.853	0.696	0.712
Mitochondria	**0.866**	0.818	0.843	0.758	0.608
Nuclear membrane	**0.922**	0.907	0.756	0.675	0.692
Nucleoli	**0.873**	0.816	0.826	0.613	0.802
Nucleus	**0.949**	0.945	0.943	0.925	0.945
Plasma membrane	**0.766**	0.756	0.656	0.597	0.538
Vesicles	0.683	**0.685**	0.682	0.551	0.435
Mean	**0.782**	0.750	0.735	0.517	0.472

Table 2: F1 scores per task for each method. Overall, GapNet performs best for 9 tasks, M-CNN for 3 tasks and DenseNet wins one task.

Method	Accuracy	F1 Score	Precision	Recall
GapNet-PL	**0.91**	**0.82**	**0.75**	**0.95**
Human Expert	0.61	0.58	0.73	0.58

Table 3: Predictive performance of GapNet-PL and the human expert. The table shows average performance metrics over 13 tasks.

The performance of the compared methods also varies across the 13 localization tasks corresponding to cell organelles. Table 2 shows that GapNet-PL has the best performance on 9 tasks, M-CNN on 3 tasks and DenseNet-121 on 1 task. DeepLoc and Convolutional MIL are competitive for some tasks but show an inconsistent performance over all 13 tasks. Considering only the top 3 methods, on average the most challenging tasks are Centrosome, Intermediate filaments and Endoplasmic reticulum, while the easiest tasks for the 3 methods are Nucleus, Nuclear membrane and Mitochondria. We hypothesize that both the number of positive examples for each task, as well as, the complexity of the task itself influences the predictive performance (see Discussion Section 5).

In our comparison of the best performing CNN against the human expert, we found that the computational method could reach an accuracy of 91% and the human expert reached an accuracy of 61% (see Table 3). To test if there is a significant difference between the performance of GapNet-PL and the human expert we applied a McNemar test with a resulting p-value of $2e^{-10}$. We discuss potential biases this comparison could suffer from in Section 5. Table 4 shows the confusion matrix for the comparison of GapNet-PL with the human expert.

In Figure 5 we show a more detailed break down on a per-task level. The performance of the human expert has a large variability across tasks with very good results for, e.g., Actin filaments, Nuclear membrane and Nucleus and moderate results for, e.g., Centrosome, Cyotsol and Endoplasmic reticulum. Especially, the class Vesicles exhibits a high number of false positives. In comparison, GapNet-PL shows a consistent performance and can predict most of the classes perfectly.

Both Correct	GapNet-PL Correct	Human Expert Correct	Both Incorrect
121	60	7	12

Table 4: Confusion matrix for comparison of GapNet-PL with human expert. McNemar's test for equality of row and column marginal frequencies gives a p-value of $2e^{-10}$.

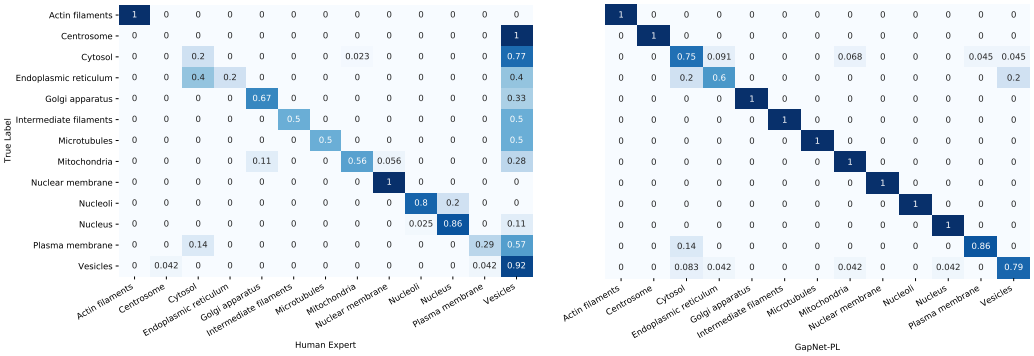

Figure 5: Confusion matrices for multi-class estimates of human expert (left) and GapNet-PL (right). While the performance of the human expert is commendable for most of the classes there are a lot of false positives for the class Vesicles. The CNN exhibits good performance across all classes, predicting 9 classes perfectly.

5 Conclusion and Discussion

In this work, we introduced a new neural network architecture *GapNet-PL* that was designed to overcome the challenges introduced by the nature of HTI data such as high resolution and noisy labels. Furthermore, we conducted a large study comparing convolutional neural networks on a public dataset for protein localization in human cells. Our method performs nearly perfect, achieving an average AUC of 98% and an F1 score of 78%, outperforming all compared methods, as well as, a human expert while requiring less computational resources (see Appendix Table 7). Considering the individual tasks separately a similar picture emerges. Our method is superior to the other methods in most of the tasks. M-CNN and DenseNet-121 show a competitive performance and win the remaining tasks. The overall performance of the top 3 methods is close to perfect in terms of AUC. However, the performance across the tasks varies, some of the classes are harder to learn (Centrosome, Intermediate filaments and Endoplasmic reticulum), while Nucleus, Nuclear membrane and Mitochondria are quite easy tasks for the best performing CNNs. The relative frequency seems not to be the only driver for this variability since Nuclear membrane is a strongly underrepresented class which can be predicted well with our tested methods.

With respect to our comparison with the human expert (HE), we are aware that our estimate of human performance at this task could suffer from potential biases such as (and not limited to) the following: a) Anchoring effect: The HE investigated only a subset of the training data points and could be biased towards those; b) Confirmation bias: The HE knew he would compete against an AI and probably thought the AI would be better, which could have decreased his motivation and thereby his performance; c) Response bias: There was interaction between HE and the experimenter in the first interactive session, such that the HE knew that the experimenter's goal was to show that the AI can outperform humans and might have subconsciously supported this effort or – vice versa – may have motivated the HE to perform even better than routinely; d) The HE could also be performing better than usual since he knew he was in a testing situation; e) Limited sample size and selection bias: we only tested the performance of a single HE and the HE was selected to be someone who is familiar with protein localization in microscopic images, but it is unclear how representative his performance is for possible human performance. It would be a tremendous effort to find a completely fair experimental setting, if this is possible at all, which is beyond the scope of this manuscript.

We have shown that GapNet-PL could be used to automatically annotate protein locations in immunostained HTI data and envision that it could be developed into a routine clinical application. It is a generic and robust method which can process input images of arbitrary size capable of learning images from various heterogeneous cell lines. It already exhibits a high predictive performance on a large and diverse dataset and could get even better with more data from different stainings and additional cell lines produced by other biotechnological devices from various labs.

REFERENCES

Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I., Friman, O., Guertin, D. A., Chang, J., Lindquist, R. A., Moffat, J., Golland, P., and Sabatini, D. M. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10):R100.

Dürr, O. and Sick, B. (2016). Single-Cell Phenotype Classification Using Deep Convolutional Neural Networks. *Journal of Biomolecular Screening*, 21(9):998–1003.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118.

Godinez, W. J., Hossain, I., Lazic, S. E., Davies, J. W., and Zhang, X. (2017). A multi-scale convolutional neural network for phenotyping high-content cellular images. *Bioinformatics*, 33(13):2010–2019.

Haenssle, H. A., Fink, C., Schneiderbauer, R., Toberer, F., Buhl, T., Blum, A., Kalloo, A., Hassen, A. B. H., Thomas, L., Enk, A., Uhlmann, L., Alt, C., Arenbergerova, M., Bakos, R., Baltzer, A., Bertlich, I., Blum, A., Bokor-Billmann, T., Bowling, J., Braghiroli, N., Braun, R., Buder-Bakhaya, K., Buhl, T., Cabo, H., Cabrijan, L., Cevic, N., Classen, A., Deltgen, D., Fink, C., Georgieva, I., Hakim-Meibodi, L.-E., Hanner, S., Hartmann, F., Hartmann, J., Haus, G., Hoxha, E., Karls, R., Koga, H., Kreusch, J., Lallas, A., Majenka, P., Marghoob, A., Massone, C., Mekokishvili, L., Mestel, D., Meyer, V., Neuberger, A., Nielsen, K., Oliviero, M., Pampena, R., Paoli, J., Pawlik, E., Rao, B., Rendon, A., Russo, T., Sadek, A., Samhaber, K., Schneiderbauer, R., Schweizer, A., Toberer, F., Trennheuser, L., Vlahova, L., Wald, A., Winkler, J., Wölbing, P., and Zalaudek, I. (2018). Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Annals of Oncology*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 971–980. Curran Associates, Inc.

Kraus, O. Z., Ba, J. L., and Frey, B. J. (2016). Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52–i59.

Kraus, O. Z., Grys, B. T., Ba, J., Chong, Y., Frey, B. J., Boone, C., and Andrews, B. J. (2017). Automated analysis of high-content microscopy data with deep learning. *Molecular Systems Biology*, 13(4):924.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc.

Liimatainen, K., Valkonen, M., Latonen, L., and Ruusuvuori, P. (2018). Cell organelle classification with fully convolutional neural networks. Technical report.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440.

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

Pepperkok, R. and Ellenberg, J. (2006). High-throughput fluorescence microscopy for systems biology. *Nature Reviews Molecular Cell Biology*, 7(9):690.

Poostchi, M., Silamut, K., Maude, R. J., Jaeger, S., and Thoma, G. (2018). Image analysis and machine learning for detecting malaria. *Translational Research*, 194:36–55.

Stadler, C., Rexhepaj, E., Singan, V. R., Murphy, R. F., Pepperkok, R., Uhlén, M., Simpson, J. C., and Lundberg, E. (2013). Immunofluorescence and fluorescent-protein tagging show high correlation for protein localization in mammalian cells. *Nature Methods*, 10(4):315.

Swamidoss, I. N., Kårsnäs, A., Uhlmann, V., Ponnusamy, P., Kampf, C., Simonsson, M., Wählby, C., and Strand, R. (2013). Automated classification of immunostaining patterns in breast tissue from the human protein atlas. *Journal of Pathology Informatics*, 4(Suppl).

Thul, P. J., Åkesson, L., Wiking, M., Mahdessian, D., Geladaki, A., Ait Blal, H., Alm, T., Asplund, A., Björk, L., Breckels, L. M., Bäckström, A., Danielsson, F., Fagerberg, L., Fall, J., Gatto, L., Gnann, C., Hober, S., Hjelmare, M., Johansson, F., Lee, S., Lindskog, C., Mulder, J., Mulvey, C. M., Nilsson, P., Oksvold, P., Rockberg, J., Schutten, R., Schwenk, J. M., Sivertsson, Å., Sjöstedt, E., Skogs, M., Stadler, C., Sullivan, D. P., Tegel, H., Winsnes, C., Zhang, C., Zwahlen, M., Mardinoglu, A., Pontén, F., von Feilitzen, K., Lilley, K. S., Uhlén, M., and Lundberg, E. (2017). A subcellular map of the human proteome. *Science*, 356(6340).

Uhlen, M., Oksvold, P., Fagerberg, L., Lundberg, E., Jonasson, K., Forsberg, M., Zwahlen, M., Kampf, C., Wester, K., Hober, S., et al. (2010). Towards a knowledge-based human protein atlas. *Nature Biotechnology*, 28(12):1248.

6 APPENDIX

6.1 DATASET

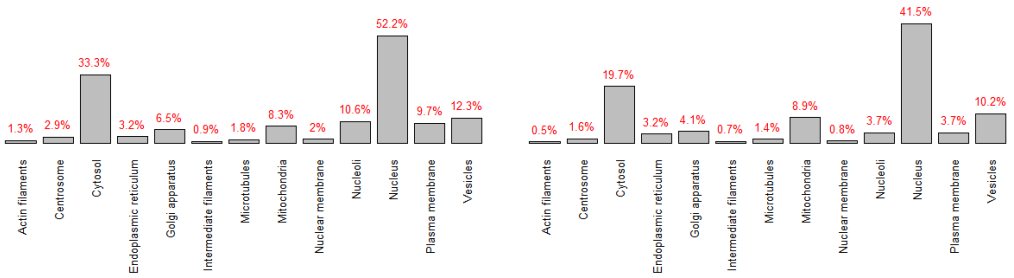

Figure 6: Class distributions for the whole dataset with 18,756 samples (left) and for all 11,484 multi-class samples (right).

6.2 HUMAN EXPERT COMPARISON

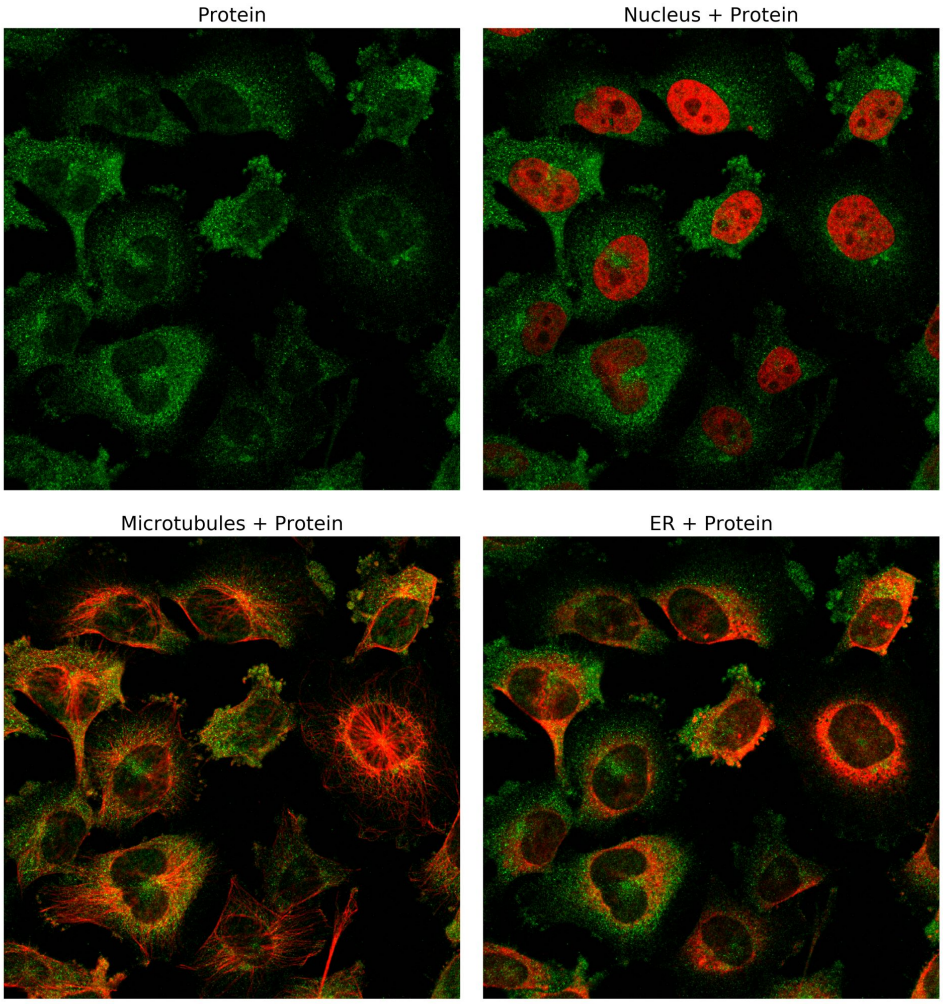

Figure 7: Visualization of samples provided to the human expert. Every sample was displayed in 4 separate images with the protein in green and the reference channels in red.

6.3 ARCHITECTURES

6.3.1 DENSENET

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Table 5: DenseNet architecture variants ((Huang et al., 2017)). The authors present different variations of the architecture. Due to memory restrictions the smallest variant, DenseNet-121, was chosen for the comparison in this work. The reduction rate for the 1x1 convolution transition layer is set to 0.5 and the growth rate is k = 32.

6.3.2 DEEPLOC

Layers	Convolution	Convolution	Max Pooling	Convolution	Convolution	Max Pooling	Convolution	Convolution	Convolution	Convolution	Max Pooling	Fully Connected	Fully Connected	Output
Units (original)	64	64	64	128	128	128	256	256	256	256	256	512	512	19
Units (adapted)	32	32	32	64	64	64	96	96	96	96	96	128	128	13

Table 6: The architecture configuration taken from Kraus et al. (2017) and the adaptations for our comparison.

6.3.3 MULTI-SCALE CONVOLUTIONAL NEURAL NETWORK (M-CNN)

	Scale factor s	*Convolutional layer 1*	*Convolutional layer 2*	*Convolutional layer 3*	*Pooling factor*	*Convolutional layer 4*	*Pooling factor*	*Fully-connected layer*	*Output layer*
Input image	1	16 kernels 5×5 pixels	16 kernels 5×5 pixels	16 kernels 5×5 pixels	64	1024 kernels 1×1 pixel	2	512 units	N_p units
	2	16 kernels 5×5 pixels	16 kernels 5×5 pixels	16 kernels 5×5 pixels	32				
	4	16 kernels 5×5 pixels	16 kernels 5×5 pixels	16 kernels 5×5 pixels	16				
	8	32 kernels 5×5 pixels	32 kernels 5×5 pixels	32 kernels 5×5 pixels	8				
	16	32 kernels 5×5 pixels	32 kernels 5×5 pixels	32 kernels 5×5 pixels	4				
	32	32 kernels 5×5 pixels	32 kernels 5×5 pixels	32 kernels 5×5 pixels	2				
	64	64 kernels 5×5 pixels	64 kernels 5×5 pixels	64 kernels 5×5 pixels	1				

Figure 8: The architecture configuration taken from the appendix of Godinez et al. (2017).

6.4 Model Training and Evaluation

Method	Training Time/Epoch (sec)	Parameters
Convolutional MIL	413	593,143
GapNet-PL	605	626,221
M-CNN	895	34,193,853
DenseNet-121	2,176	6,970,317
DeepLoc	3,497	201,716,967

Table 7: Training times per epoch in seconds and number of parameters for all compared methods.

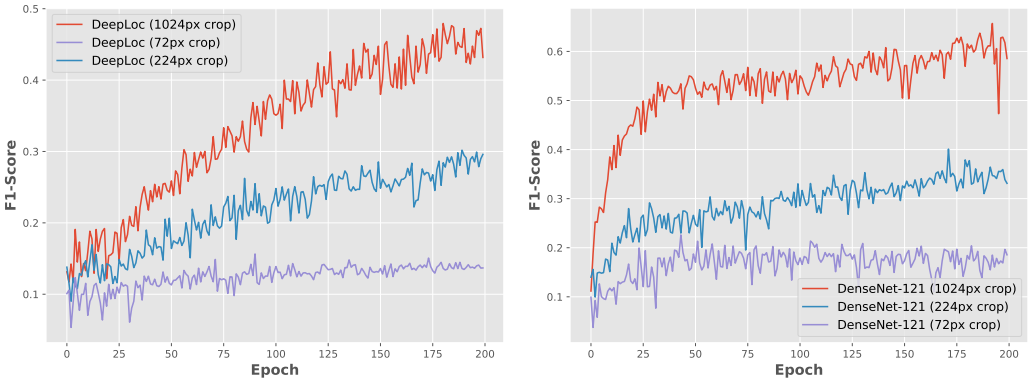

Figure 9: Comparison of different input sizes (72x72, 224x224 and 1024x1024 pixel) for DeepLoc and DenseNet on the validation set.