The answers for all the critiques are provided together in this document.

In the data analysis, you mentioned the problem of class imbalance in the labels of the data. However, I do not recognize a solution of this being presented. Is this problem addressed in some way?

The dataset that was prepared to train the classification CNN was prepared in a manner that it didn't have the class imbalance problem. Thus we didn't have to do special tricks to handle the class imbalance. In the traditional setting the class imbalance problem would be handled by a weighted focal loss function.

What is the purpose of the custom head in general? The goal of the batch norm layer and the dropout layer was discussed, but is there a specific purpose for the custom head as a whole?

Custom head is usually added to the backbone architecture. This is a place where various regularization methods can be employed, such as dropout, L1 regularization, etc. In addition to that for training the triplet metric loss, you need to extract embedding for anchor, positive and negative image, which is also done using a custom head. Pre trained resnet is used as a feature extractor because the lower and second order spatial features (corners, boundaries) are usually same across datasets.

The results show an average precision score of 0.35. Is this considered a good result for this topic, or do you think you can improve in some way?

The average precision on the top of the leaderboard was 0.49-0.55. In comparison to that we have a good average precision value. There are ways of improving the same. We could generate a far bigger classification dataset using the initial dataset and then use the weighted focal loss and a custom backbone architecture trained from scratch. This will ensure higher average precision value. We use this metric since this is the metric used in the competition.

How are you training a 4 channel dataset on a model that was originally trained on 3 channels? The small dataset might lead the ResNet to overfit, were simpler CNNs first trained from scratch to set a baseline?

The dataset isn't a small dataset. We consider the first three channels, since in the first competition the maximum discriminability comes from the RGB channels.

Since each image can have multiple labels, and the model is only being used to predict a single label per image, how was it determined for each image which label to be considered the correct one for consistency?

The final result will have multiple labels. As mentioned in the presentation, it is a two stage process, one is where the HPA cell extractor extracts the cells and another where the classifier

classifies the cells. Thus the final result will be the union of all the cell classifications from one single image.

GROUP 30
SEGMENTATION AND CLASSIFICATION OF INDIVIDUAL CELL-TYPE BASED ON THE DISTRIBUTION OF ORGANELLE PROTEIN PATTERNS

Mudit Jain, Shruti Joshi

University of California San Diego

ABSTRACT

Protein locations in genetically identical cells can be used to identify their functional subtypes.Our aim in this project was to build a classifier to predict cell subtypes based on the protein localization patterns found in the cells. We used a Resnet based architecture for classification. Using this model, we achieved a average precision score of 0.35, where the top performing models on the Kaggle competition received an average precision scores between 0.49 and 0.55.

1. INTRODUCTION

Proteins are used by the cells in our bodies to carry out a variety of tasks. Genetically identical cells can often perform different tasks, based on the localisation of proteins within different organelles in the cells. These genetically identical cells can be grouped according to their functionality. One way to do this is by imaging the cells through microscopy and classifying them into different groups based on the locations of different proteins within the cells. Identifying these functional subtypes of cells can provide great benefit in understanding the complex biology of human cells and diseases related to the dysfunction of these proteins. This, in turn, will lead to better treatment options for patients.

We aimed to automate this process of classifying cell subtypes based on protein localizations using images obtained from the Human Protein Atlas database. The input to our model were images of cells and the outputs were the labels for each cell in the image. We use Resnet as the backbone architecture to our model and add a custom head on top of it.

This is a multi-label classification problem. Since we are given only the image level labels and are tasked with predicting cell-level labels, this is a weakly supervised classification problem.

2. RELATED WORK

Previous work to identify cells based on protein localization was done manually. Although this was a particularly labor intensive task, the performance achieved by human experts has not been replicated by any algorithm so far. [1]

In order to scale up this manual method, researchers converted the annotation task into an online mini-game, named Project Discovery. This game was played by around 300,000 gamers over a span of 1 year and generated nearly 33 million patterns of subcellular localization patterns, some of which were not previously annotated by the Human Protein Atlas. The player performance was based on the F1 score, a suitable measure for multi-label classification problem. [2]

In [3], Support Vector Machine (SVM) classifiers radial basis function kernel was used to classify cells by their subcellular protein locations. Two levels of nested 5-fold validation was used to optimize the training parameters. They refined their classification performance by retraining the SVM classifiers for the classes that were far away from the decision boundary and reannotate them using human annotations. This achieved accuracies per label ranging from 30% to 95%

Random Forest classifiers were also used for this task. 500 trees with each tree consisting of 11 decision nodes were used. The accuracies per label for this approach ranged from 45% - 98%. [4]

A kaggle competition was conducted for image level classification of cells. Most teams used deep learning models, most common being variations of Resnet, Densenet or Inception models as backbone architectures. Even though these models performed better than previous methods, they has problems of generalisation, picking up unintentional variations. [5]

A limitation of automated approaches seems to be that the dataset is limited in terms of the variability actually observed in real biological systems. Also, human experts at HPA have consistently performed better than all the automated approaches so far.

3. DATASET

The Dataset was obtained from The Human Protein Atlas (HPA), an initiative based in Sweden aimed at mapping proteins in human cells, tissues and organs. The data was collected using the method of confoncal microscopy. This dataset comprises of cells from 17 different cell types, with highly different morphologies affecting the pattern of protein localization within organelles.

The data is separated into train and test sets. Each image sample in the dataset is represented by 4 channels: Red for Microtubule channels, blue for Nuclei channels, yellow for Endoplasmic Reticulum channels and green for the protein of interest. Each sample contains images of multiple cells. The labels for training samples is present in the train.csv file, and each sample in the dataset can have multiple labels. There are 19 labels corresponding to 19 different protein patterns observed in the samples. Some or all of these protein patterns may be present in individual cells in the sample. The labels are represented as numbers from 0 to 18, with 0 to 17 representing specific patterns and 18 representing a negative or nonspecific result.

Our task was to predict the protein pattern (label) for each cell in the sample, given the labels associated with the entire sample. This makes it a weakly supervised classification task. The green channel (protein of interest) is used to predict the label, while the other channels are used as references.

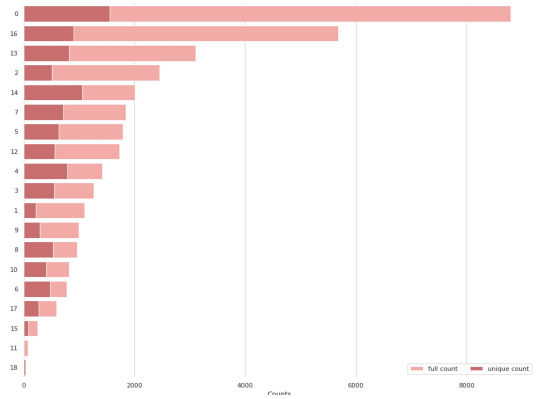

Fig. 1. Number of Images for each Label

We analysed the distribution of labels in the dataset. We found that labels 0 and 16 were the most common, and 11 and 18 were the least common, with barely any images for these, as seen in figure 1. This disparity points to class imbalance.

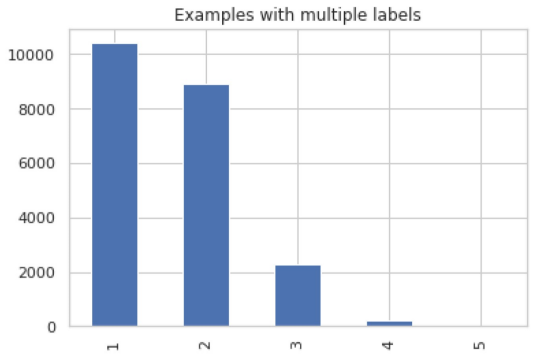

Fig. 2. Number of Labels per Image

Each image has multiple labels associated with it and the number of unique label combinations in the train set was 432. We also noticed that most images had either 1 or 2 labels and the maximum number of labels for any image was 4. This can be seen in figure 2

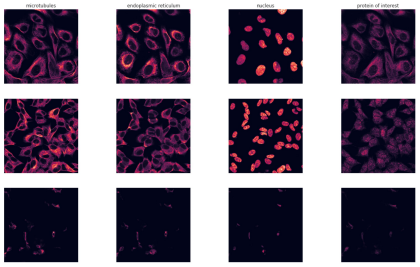

Fig. 3. Cell Images from Different Channels

In figure 3, we can see the image through the different filters, with some images showing very faint staining. We can combine these filters together to get figure 4. The combination is as follows: Red for Microtubules, Blue for Nucleus, Yellow for ER, and Green for the protein of interest.

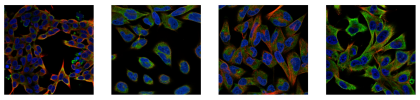

Fig. 4. Cell Images with all Channels Combined

3.1. Feature Extraction

We used the HPA Cell Segmentation module to create masks for individual cells. The images and their corresponding masks can be seen in figure 5 The cells extracted using this mask form the dataset for our multilabel classifier.

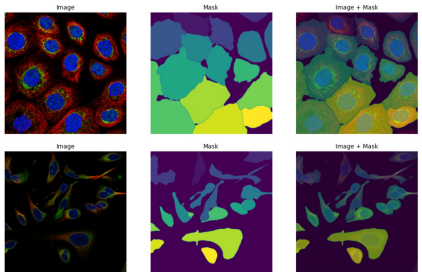

Fig. 5. Cell Images and Masks obtained by HPA Cell Segmentor

We can see the images of single cells extracted from the HPA Cell Segmentation Module in figure 6

Since it is a weakly supervised problem, we only consider images which have single labels for training our classifiers.

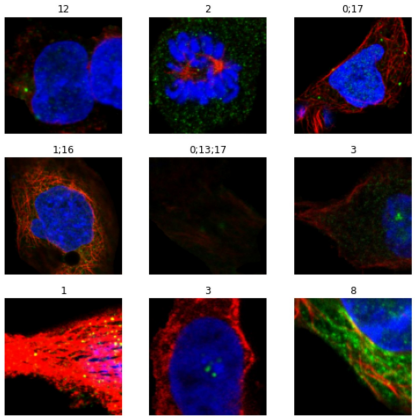

Fig. 6. Single Cell Images extracted using HPA Cell Segmentor and Labels associated with their Images

4. METHODS

The Feature Extraction was done using the previously developed/trained HPA Cell Segmentation Module. This module internally uses the DPN(Dual Path Network)-Unet based architecture to segment cells from images. DPN picks the advantages of ResNet and DenseNet ans is shown to provide state of the art results for object detection and semantic segmentation.

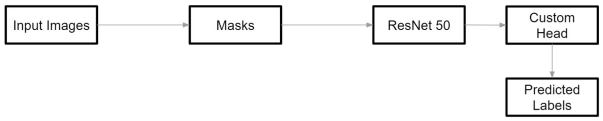

Fig. 7. The Model

For the multi-label classification problem, we used the Resnet-50 with a custom head. ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. This model overcomes the problem of vanishing gradient that is common in deep architectures by using residual layers/blocks. The flow chart description of the method is given in Figure7

The custom head converts the features extracted by Res Net to our desired classification output using full connected layers. We have also added Dropout to help decrease the variance of the model. Batch Norm is also added so as to restrict the co variance shift of the activation layers. In addition to the basic categorical cross entropy loss we also formulate another custom loss function. This is a combination of the triplet loss and the categorical cross entropy loss.

For computing the triplet loss, a custom data loader gives out an anchor image, a positive image and a negative image. The formulation of the triplet loss essentially brings the embedding of the anchor and positive image closer to each other and the anchor and the negative image farther from each other.

In order to get good performance in a reasonable amount of time, it is important to choose the correct learning rate. If the learning rate is too small, the model will learn too slowly and will take a long time to reach optimal performance. If the learning rate is too high, we could overshoot the optimal model performance, and settle on a sub-optimal model. For this reason, we used the learning rate finder from the fastai API to settle on an optimal learning rate. This learning rate finder works as follows:

1. To start, we make a guess on the learning rate based on the loss vs learning rate graph as shown in figure 8. Looking at the figure, the slope of the graph is steepest at 0.01 and hence we choose this value.
2. We then train the model using this learning rate.
3. We re-run the learning rate finder after training, to fine-tune the learning rate. Here, we see the loss vs learning rate graph in 9. We choose the value 0.01 again so that the loss doesn't increase with further training.

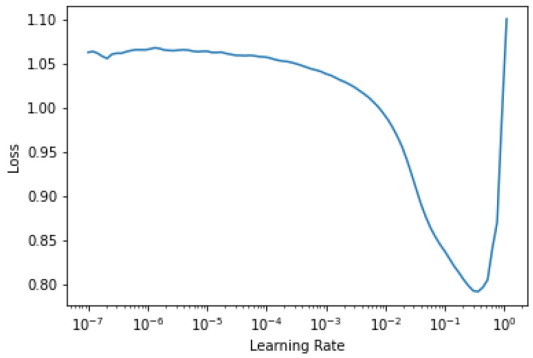

Fig. 8. Loss vs Learning Rate

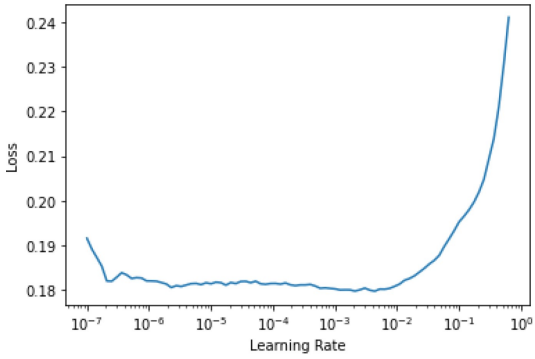

Fig. 9. Loss vs Learning Rate

5. RESULTS

Our model achieved an accuracy of 95.6% on the training data, with a precision of 0.81 after fine-tuning the learning

rate as mentioned in Methods.

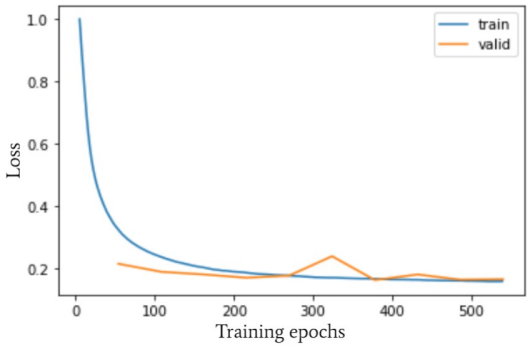

Fig. 10. Loss Plot

The training and validation loss plots are shown in figure 10. We can see the loss decreasing for the training as well as validation data with training epochs, showing that overfitting has not occurred.

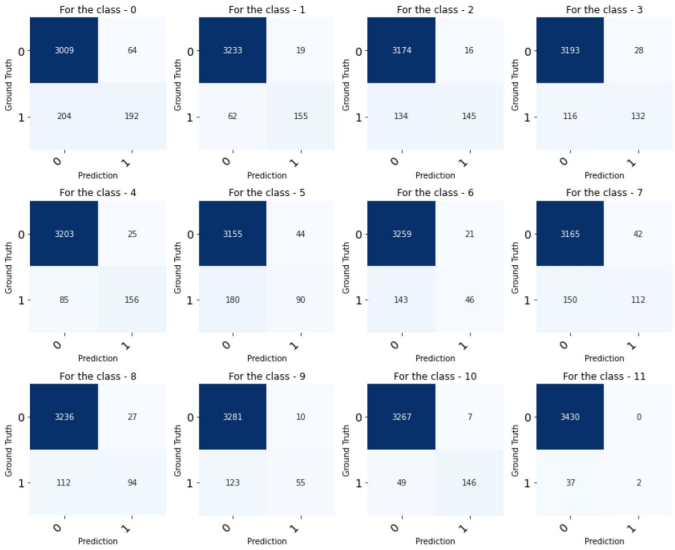

Fig. 11. Confusion Matrices for Each Label

The confusion matrices for each label is shown in figure 11 and 12. From the confusion matrices we can see that even though we have moderately good precision scores, the recall values can be worked on.

In figure 13, we can see the individual cells and their true and predicted labels. If you look at the image in row 1, column 2, we see that the image level labels are 10 and 16. However, the predicted label for the cell shows only 16. The case here may be that this particular cell does not contain the pattern with label 10 rather than being an incorrect prediction. This example shows the weakly supervised nature of the classification task.

The averaged precision rate, micro-averaged over all classes was found to be 0.35 for the categorical cross entropy loss and 0.28 for the custom loss function. This metric is used throughout since this is the metric used in the Kaggle competition.

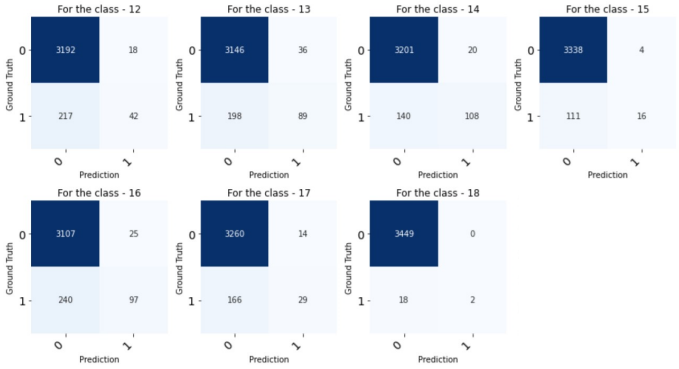

Fig. 12. Confusion Matrices for Each Label

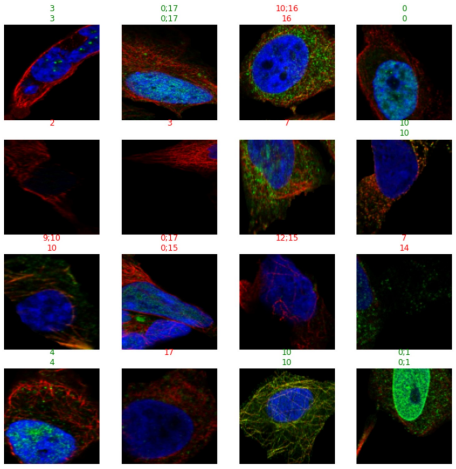

Fig. 13. Single Cell Images with True and Predicted Labels

6. CONCLUSION

We used a Resnet based classification model for classifying individual cells based on their protein localizations, since it has shown past success on image based classification tasks. This method achieved an average precision of 0.35 with the categorical cross entropy loss function and 0.28 with the custom loss function. We see that with the custom loss function even without a deep architecture we sort off come close to the resnet precision value. The top performing models on the kaggle competition achieved average precision of 0.55.

The next steps would be to prepare a bigger dataset for classification and handle the problem of class imbalance using weighted focal loss function in conjuction with the triplet loss. We could also look into end to end training/fine tuning where the entire volume of mask outputs from HPA Cell segmentor can be processed at once.

7. INDIVIDUAL CONTRIBUTIONS

Shruti worked on understanding the dataset that was given and extraction of the masks using the HPA Cell segmentor. A custom dataset for training the classifier was also generated. Finding the correct learning rate for the training process and testing the inference of the classifier was done by Shruti.

Mudit worked on developing and training both the baseline model with cross entropy loss function and worked on the code for the custom loss function as well. Visualization of the training/testing was also done by Mudit. Presentation Report were worked on together.

8. REFERENCES

[1] Peter J. Thul, Lovisa Åkesson, 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, Åsa 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):eaal3321, May 2017.

[2] Devin P Sullivan, Casper F Winsnes, Lovisa Åkesson, Martin Hjelmare, Mikaela Wiking, Rutger Schutten, Linzi Campbell, Hjalti Leifsson, Scott Rhodes, Andie Nordgren, Kevin Smith, Bernard Revaz, Bergur Finnbogason, Attila Szantner, and Emma Lundberg. Deep learning is combined with massive-scale citizen science to improve large-scale image classification. *Nature Biotechnology*, 36(9):820–828, October 2018.

[3] Jieyue Li, Justin Y. Newberg, Mathias Uhlén, Emma Lundberg, and Robert F. Murphy. Automated Analysis and Reannotation of Subcellular Locations in Confocal Images from the Human Protein Atlas. *PLoS ONE*, 7(11):e50514, November 2012.

[4] Justin Newberg, Robert Murphy, and Fredrik Ponten. Automated Analysis of Human Protein Atlas Immunofluorescence Images. *Proc IEEE Int Symp Biomed Imaging.*, 2009(5193229):1023–1026.

[5] Wei Ouyang, Casper F. Winsnes, Martin Hjelmare, Anthony J. Cesnik, Lovisa Åkesson, Hao Xu, Devin P. Sullivan, Shubin Dai, Jun Lan, Park Jinmo, Shaikat M. Galib, Christof Henkel, Kevin Hwang, Dmytro Poplavskiy, Bojan Tunguz, Russel D. Wolfinger, Yinzheng Gu, Chuanpeng Li, Jinbin Xie, Dmitry Buslov, Sergei Fironov, Alexander Kiselev, Dmytro Panchenko, Xuan Cao, Runmin Wei, Yuanhao Wu, Xun Zhu, Kuan-Lun Tseng, Zhifeng Gao, Cheng Ju, Xiaohan Yi, Hongdong Zheng, Constantin Kappel, and Emma Lundberg. Analysis of the Human Protein Atlas Image Classification competition. *Nature Methods*, 16(12):1254–1261, December 2019.