1. Cellular Response to Perturbations with Batch Effects

In the study of biological systems, microscopy images are important to identify and understand intracellular organelle functionality, differentiate cell types, and identify effects of assays on cells. Here, cells are subjected to chemical and genetic perturbations, painted with 6 stains and imaged in 5 channels (highlighting 8 cellular departments). The types of chemical perturbations (drug discovery) included using new products, drugs and uncharacterized compound libraries. The genetic perturbations (functional genomics) took advantage of the techniques to produce overexpression, CRISPR, RNAi and deletion of strains. Thus, the first challenge of this project is to characterize images into these different assay classes. Along with image analysis, there is a methodology called profiling that is used to classify these assay classes. In biological experiments there are common issues with imaging data such as batch effects and undesired artifacts. When given two batches of microscopy images subject to the same assay but subject to different technical conditions then there is likely a quantitative difference observed. Such differences are not due to meaningful biological variations and should be removed using computational methods. The ultimate challenge is to align the information content of two batches by correcting for batch effects and then making profiles of the same assay such that it does not distort the relationships among other assays. Success will ultimately be measured based on the correct association of assay type. A subset of assays will be chosen from the larger dataset available in the cytodata dataset. The general thought process behind how to approach this project is as follows:

- Build deep convolutional neural network to analyze cellular images.
- Train neural network to classify images based on assay type.
- Observe how batch effect alter the prediction of assay classes. Tune neural network parameters and perform any preprocessing on images depending on what is learned about batch effects.
- Train model with new preprocessed images and/or tuned parameters and assess final approach accuracy.

There are a large number of laboratories and industries that would benefit from the results of this project. Batch effects, in particular, have challenged machine learning biologists for a long time and therefore any progress in this field would be useful. The dataset for this project is called the Cell Painting Image Collection and is part of the research conducted at the Broad Institute of MIT and Harvard. The data is hosted on Amazon Web Services. The project deliverables will be a functioning python code (with data wraggling, exploratory data analysis and a machine learning model), a project report document and a power point presentation report.

2. **Data wrangling/ image preprocessing**

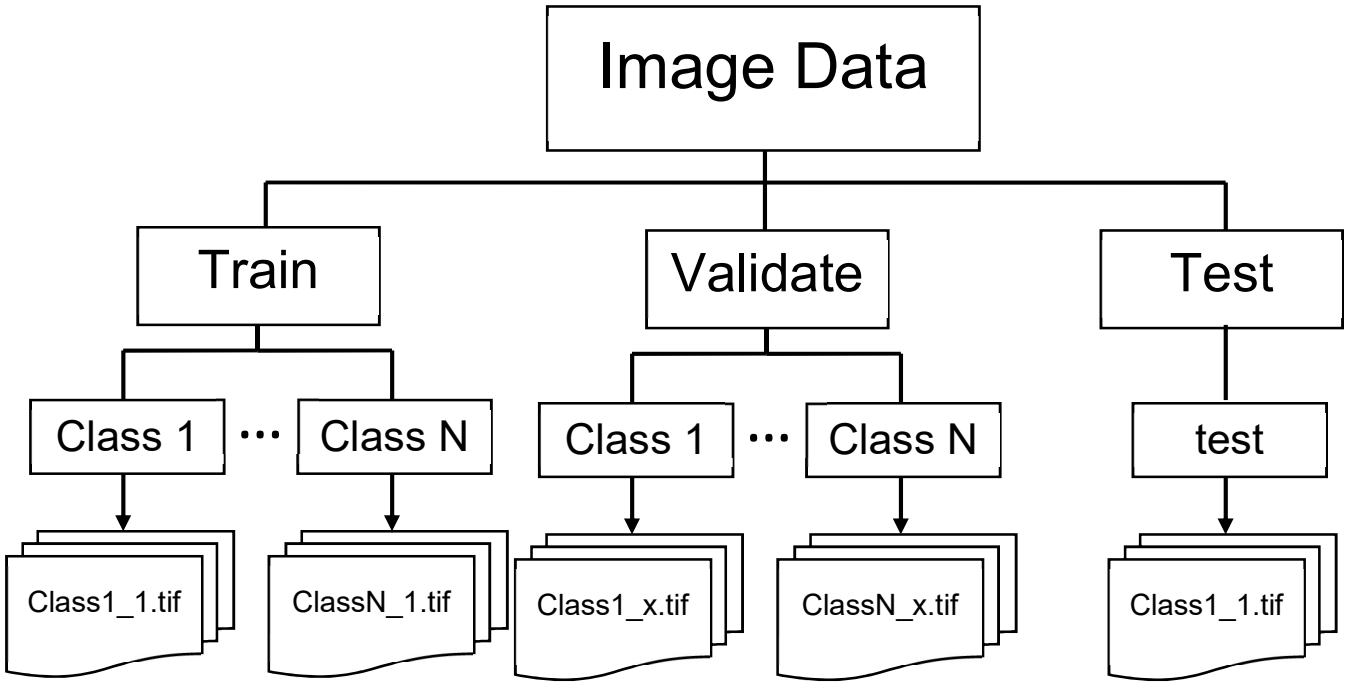
Data used in this project are image file available from research conducted at the Broad institute of MIT and Harvard. The data is freely available on Amazon Web Services (AWS) and in the Cell Painting Image Collection and was accessed using AWS command line interface. An example of downloading a specific set of images using this interface is:

```
aws s3 cp s3://cytodata/datasets/Bioactives-BBBC022-Gustafsdottir/images/Bioactives-BBBC022-Gustafsdottir/20585/ local --recursive --exclude "*" --include "IXMtest_A02*"
```

Here a series of images with a specific starting file name are downloaded to computer local drive. Notice, that s3://cytodata/datasets/ is the bucket on AWS that contains the Cell Painting Images. None of the data was corrupted and therefore cleaning was not necessary. For this project, the starting number of assays to classify was 10 (5 chemical and 5 genetic). Each chemical assay contains 9 samples per batch, 4 batches and 5 channels (each an individual image) per sample. That means there are 36 total samples per assay and 180 images per assay. Each genetic assay contains 9 samples per batch, 5 batches and 5 channels per sample. Therefore, there are 45 samples per assay and 220 images per assay.

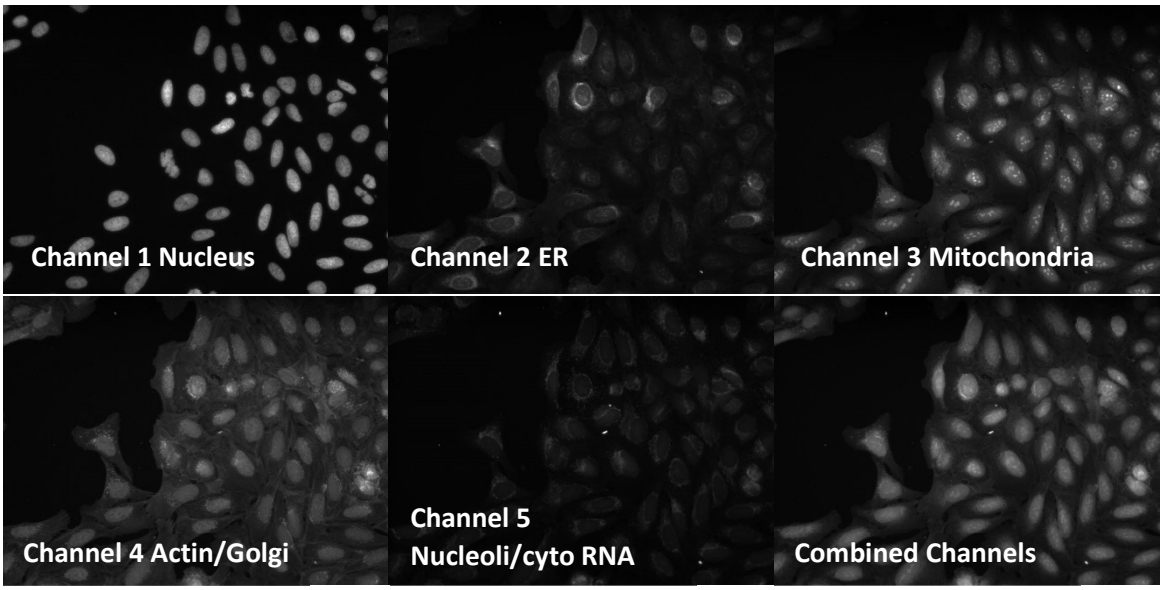
There were several image preparation steps that needed to be done before trying work with these images further. Each image contains one channel (5 channels makes all cellular organelle) of one sample of a particular assay within a batch. First, the images needed to be renamed to make it simpler to keep track of image classification (assay), channel number and batch number. A sample image filename before renaming is taoe005-u2os-72h-cp-a-au00044859_a04_s1_w1_thumbed683e83-abe8-403c-a032-e886d0a92dd8.tif and after converting to new filename it is classa04_s1_w1_b1.tif. An algorithm was written to automatically do this for all files.

To make it easier to flow image data from the directories during modeling there images have to be collected into a specific directory location.



The images are arranged into folders that are arrange into a training folder, a validation folder, and a test folder. Within the training and validation folder are subfolders that are named for different classes (assays). Within the class folders are the images for those particular classes. The directory hierarchy makes it easier to access the data when model.

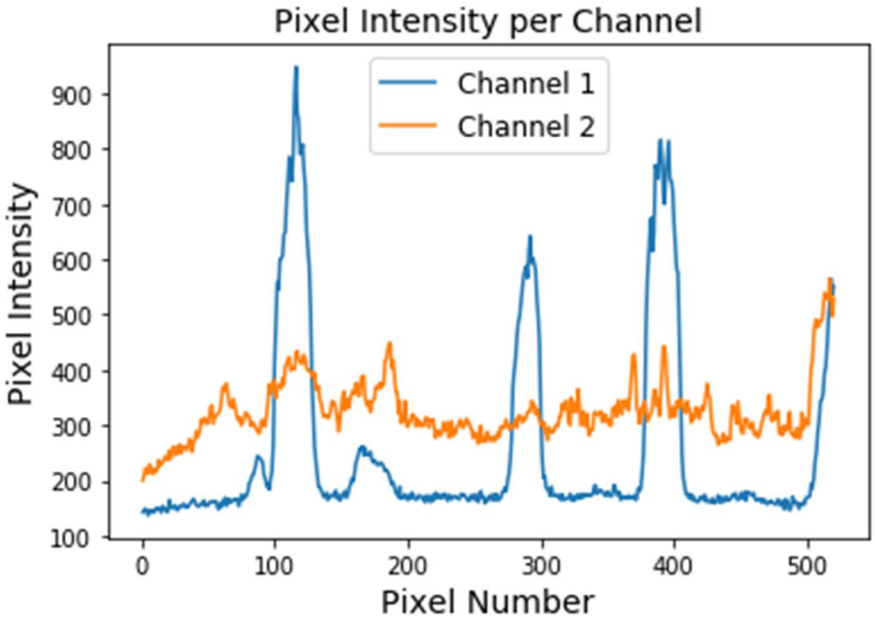
There are a number of ways that the different image channels can be handled during modeling. Several methods will be tested during modeling. Some ideas for channel handling are treating them the same way RGB information is treated during modeling, creating a model for each channel and creating an ensemble result from these models or combine the channels into one channel. Here, combine the channels into one channel is demonstrated and new images are created to use in modeling later. Each channel highlights a different portion of the cellular environment. To create one channel from all 5 channels the values in the image array for all 5 channels are summed together. Then, the resulting 2D image matrix is normalized by taking the maximum value of the array and multiplying by the largest image value of 255. This process is summarized as 255*image_matrix.sum(axis=2)/image_matrix.sum(axis=2).max().max(). Below is an example of the 5 channels separated and then the resulting combining of all of the channels.



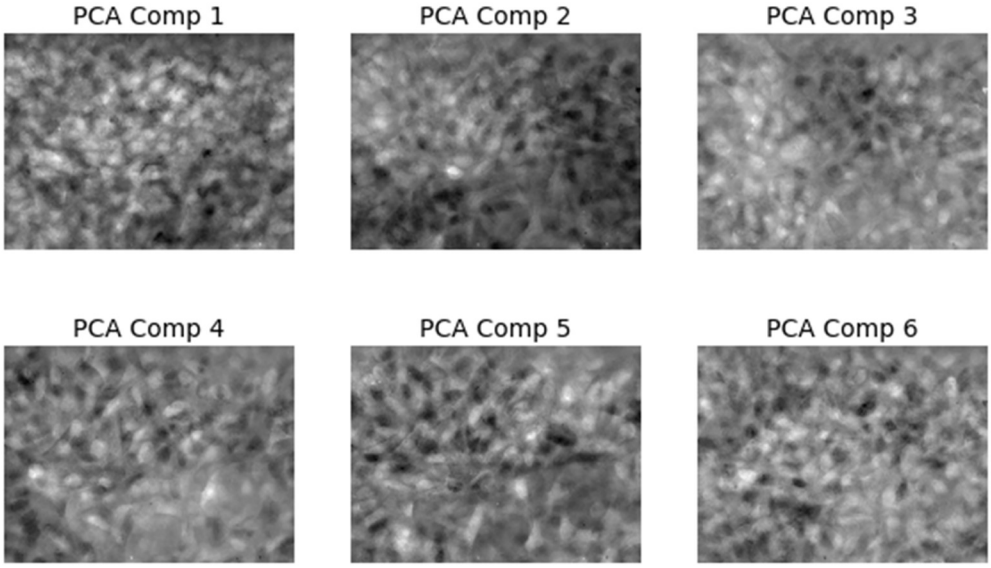
3. Image pixel intensity and image structure

It is apparent from the images that each channel highlights different aspects of the cellular environment. The first and second channel are plotted over the length of an image at a particular y-location to demonstrate the pixel intensity variations due to different types of structures. Notice, in channel 1 the pixel intensity is low and then peaks a specific points. This is indicative of a bright object and, in this case, a nucleus.

While channel 2 which does not highlight the nucleus but shows other organelle more evenly disbursed throughout the image.

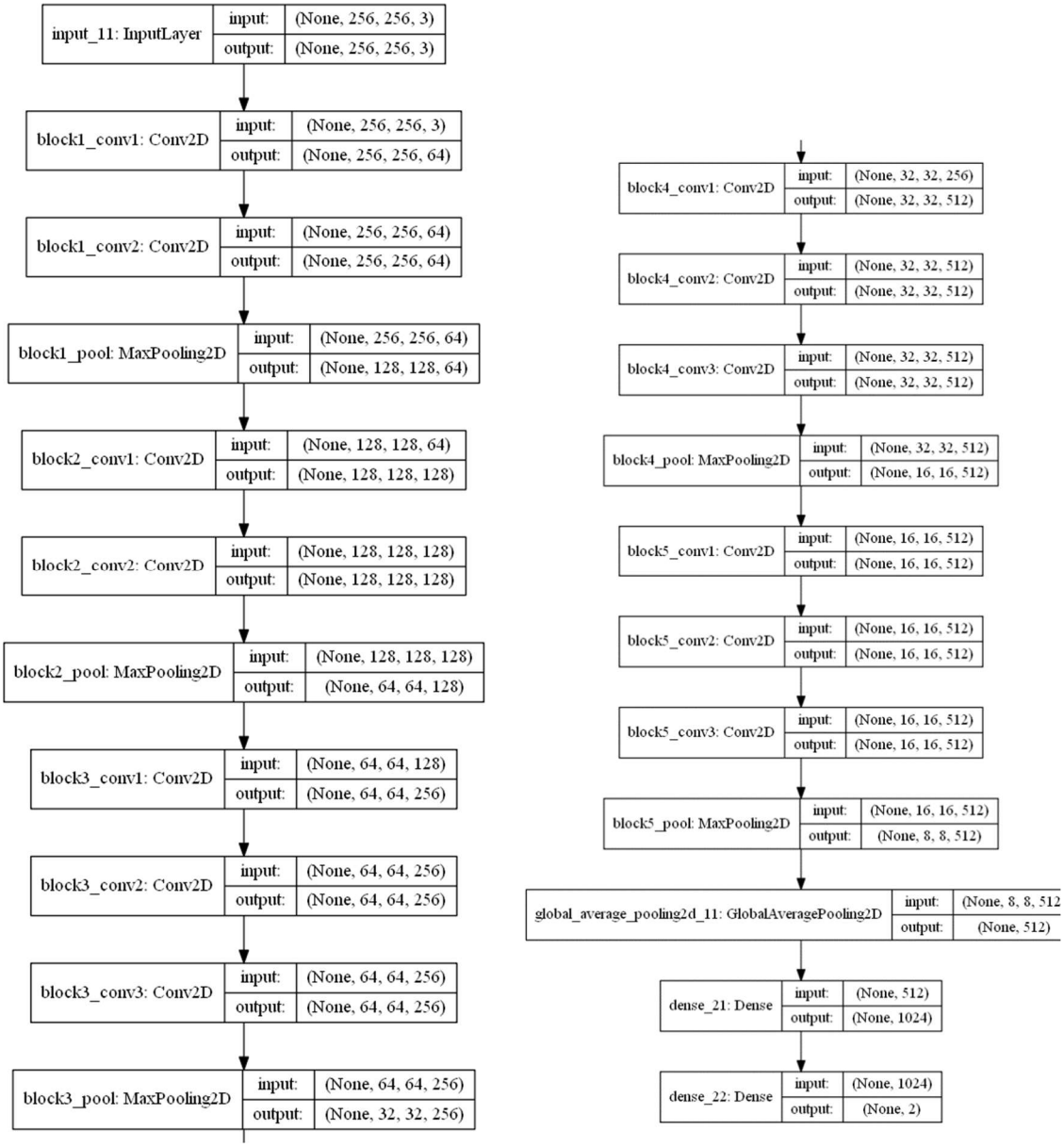


Another thing that can be looked at is to see if there is a particular reoccurring structure in the cells. To do this one assay's images underwent principle component analysis (PCA). Though it is an intriguing exercise the results were unspectacular. The images of the principle components show that there is a repeating granular structure of the size of the cells.



4. Machine learning to distinguish cellular assays

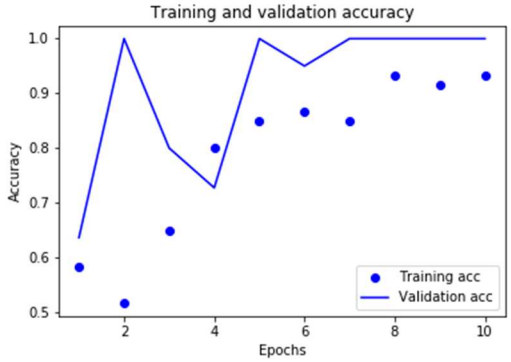
The data being worked with for this project are all microscopy images of cells. The natural model to use for image recognition/classification are typically convolutional neural networks. The challenge is to determine the deep network structure and the best way to determine the network weights given the sample size being used. The number of microscopy images per assay is very small (around 36 per assay) therefore there is a challenge associated with training a full deep network using so few images. A way to solve the issue with a small sample size is to take advantage of transfer learning. The basic principle is to use a base model that has been trained using a different set of data. There are many models available and models that were explored for this project but the final model chosen was a base model input directly from the Keras library. The model used is the VGG16 model initialized with the ImageNet weights. See model layout in the diagram below.

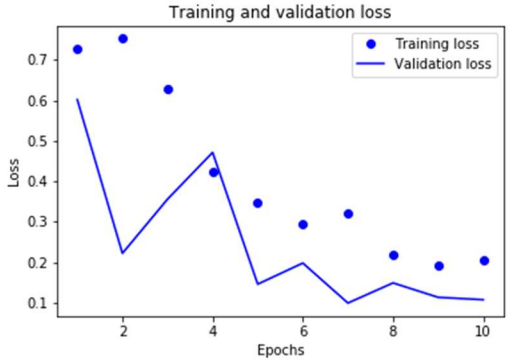


Notice, that an additional dense layer with 1024 nodes an additional dense layer with 'softmax' were added to account for the number of output classifications. Depending on the experiment performed there were 2, 5 or 10 output classes. Also, several iterations were performed to optimize the layers that needed to be trained in the network. In addition to the last two layers that were added, the last 3 layers of the VGG16 network were set such that they were trained while the remaining layers were frozen such that they maintained the ImageNet weights.

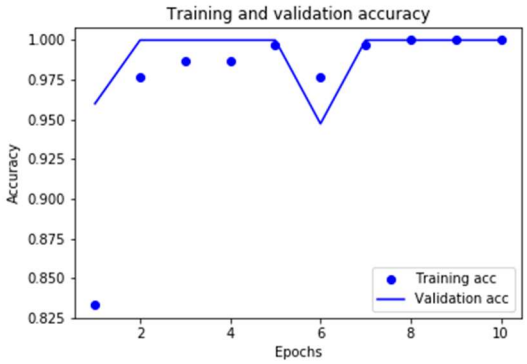
Several experiments were done to see how effective the neural network would be at differentiating different cellular assays. There are two different types of assays; one type is a chemical assay and the type is a genetic assay. To test the effectiveness of predicting between the different types of assays two experiments were run; one experiment tested differentiating one chemical assay from one genetic assay and the other experiment looked at differentiating 5 chemical assays from 5 genetic assays. These results are shown below. Notice, the validation accuracy eventually reaches 100% for both experiments but the image set with more assays converged to the result quickly. This is because for each epoch the model was trained using more images and therefore the weights of the model adjusted closer to the ideal value over each epoch relative to the 2 image set experiment. The results indicate that the model very effectively differentiates between chemical and genetic assays.

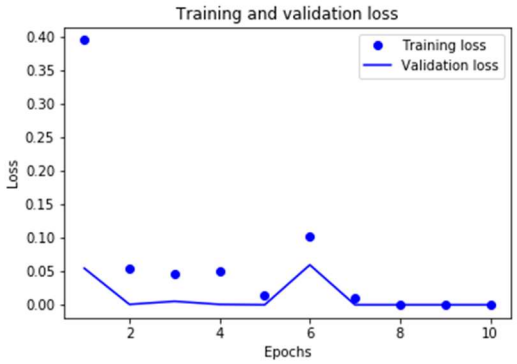
1 chemical assay and 1 genetic assay (2 classes)





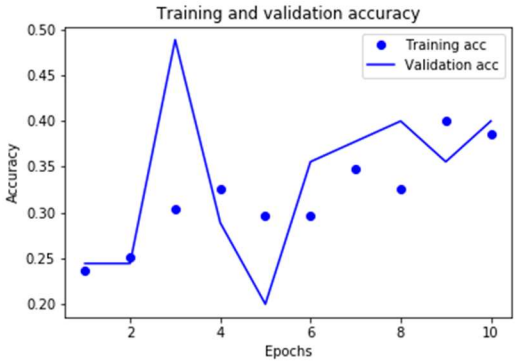
5 chemical assays and 5 genetic assays (2 classes)

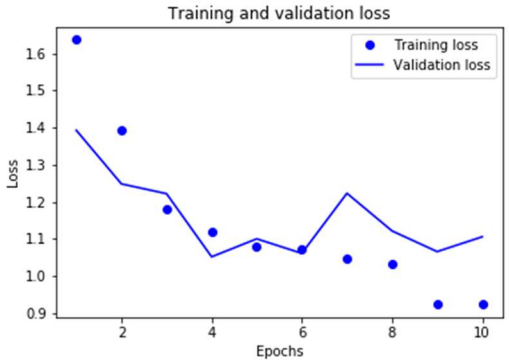




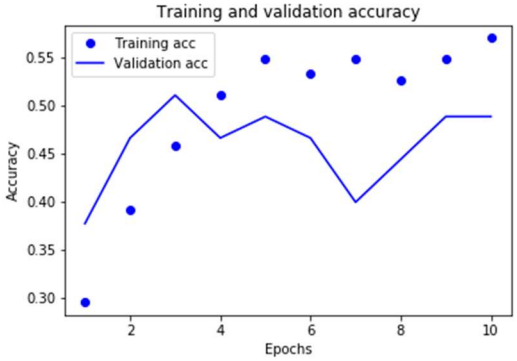
The model performs well in differentiating chemical and genetic assays. The next question is, can the model effectively differentiate different chemical assays and different genetic assays from one another. Taking 5 chemical assays and 5 genetic assays the validation accuracy only reaches slightly above 40% before overfitting begins to be observed. To better understand if the model can predict the a particular set of assays better than another the 5 chemical assays are modeled separate from the 5 genetic assays. It is observed that the chemical assays can be classified properly with about 45-50% accuracy before overfitting is observed while the genetic assays are classified with accuracy of about 40% before overfitting.

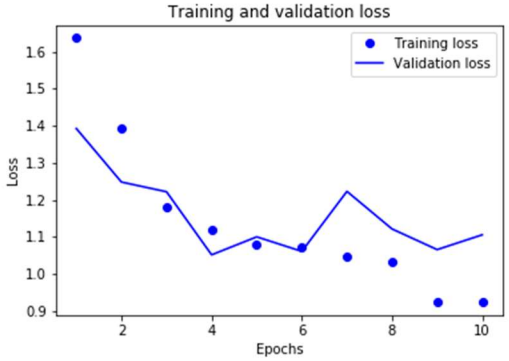
5 chemical assays and 5 genetic assays (10 classes)



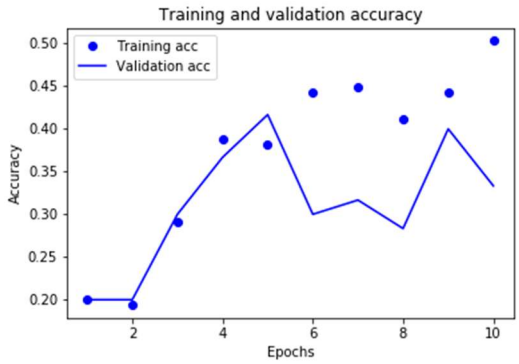


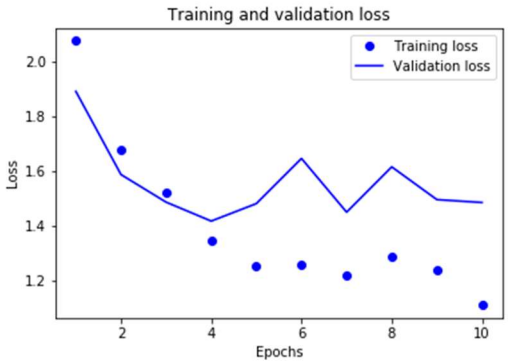
5 chemical assays (5 classes)





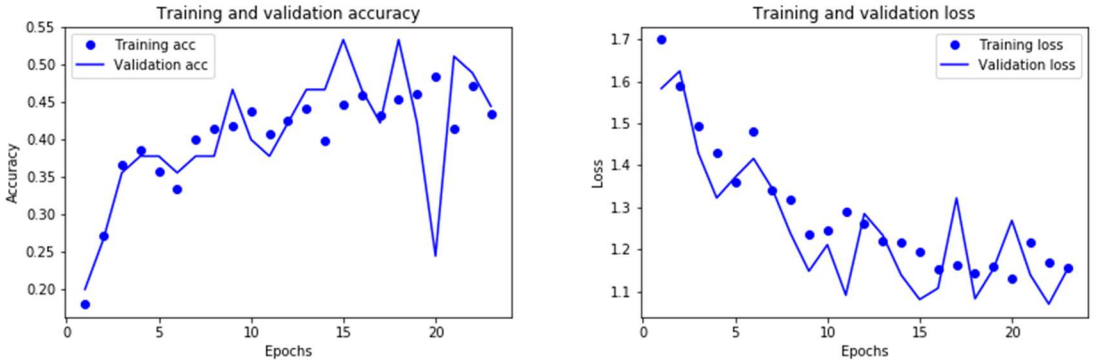
5 genetic assays (5 classes)





Overfitting and the low number of images in the data set means that other methods could be to improve model accuracy. Here the set of 5 chemical assays is modeled using image augmentation techniques to expand the image set. The images were augmented by using variable zooming (scaling), rotating the images, flipping the image horizontally and vertically, and changing brightness. Apply all of these changes in different permutations resulted in about 2 orders of magnitude increase in the number of images to use in training. The image augmentation was not applied to the validation data set. Notice, the validation accuracy is around 50% but doubling the number of epochs does not result in overfitting. For this set of images data augmentation improves model accuracy and reduces the chance of overfitting.

5 chemical assays (5 classes)



5. Visualizing intermediate activation

The activation of each convolutional layer can be viewed to see how the trained neural network successively breaks down an image to interpret what is being seen. To visualize the individual layer activation an example image is processed through the trained neural network and then the activation at different layers is observed. The

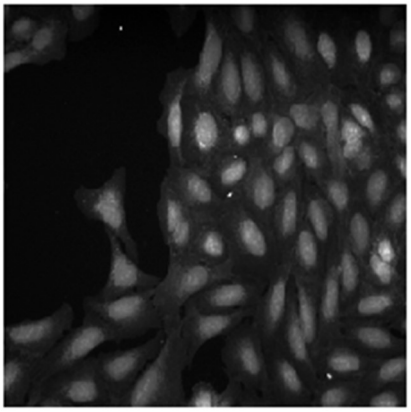
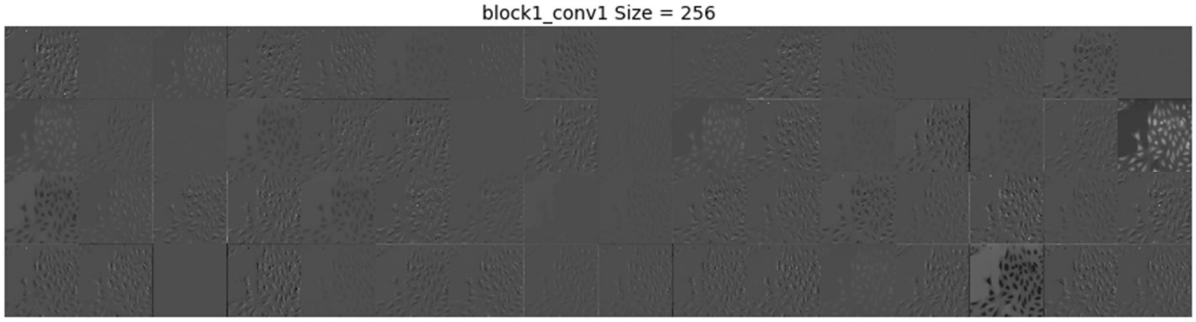
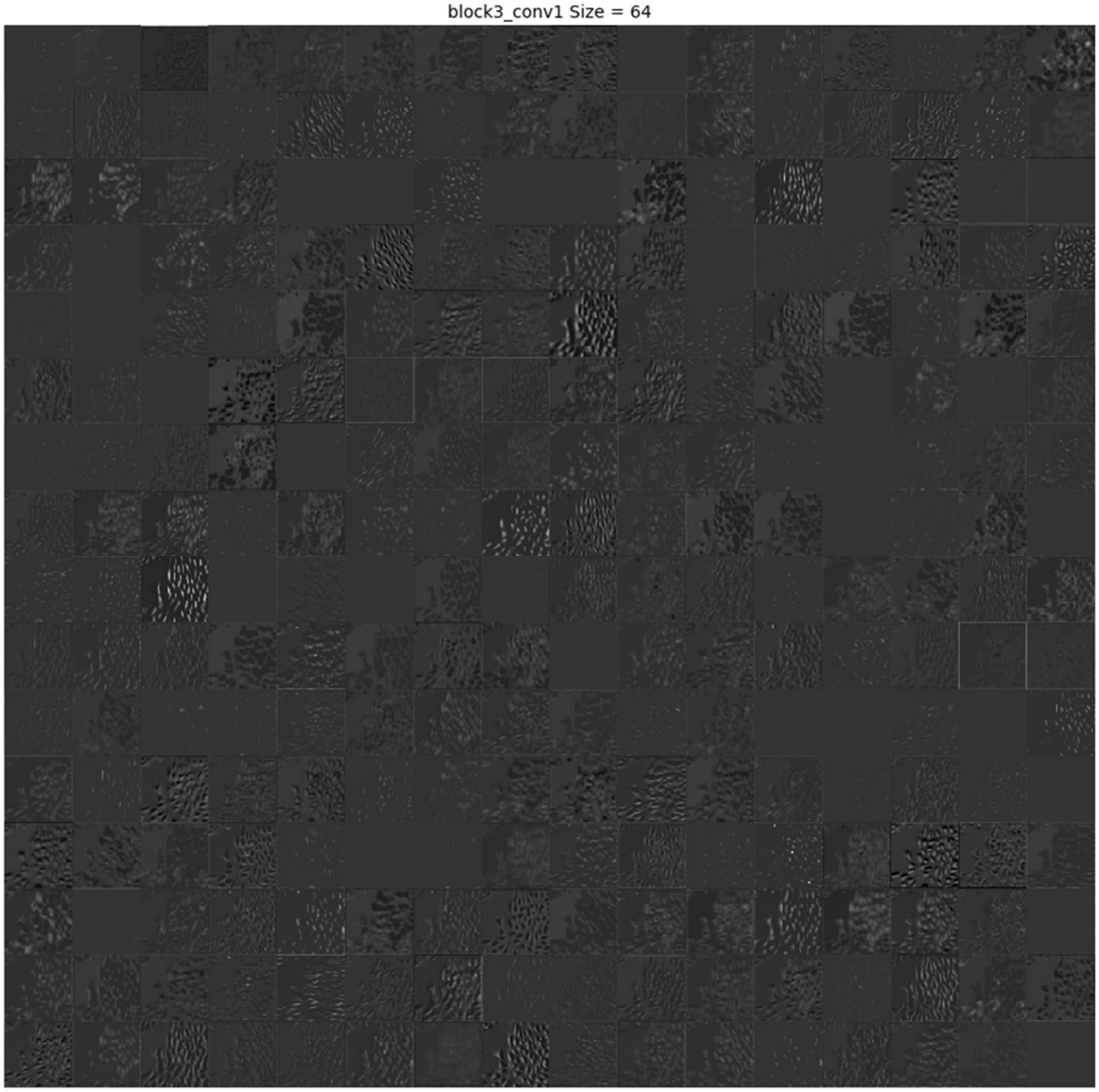


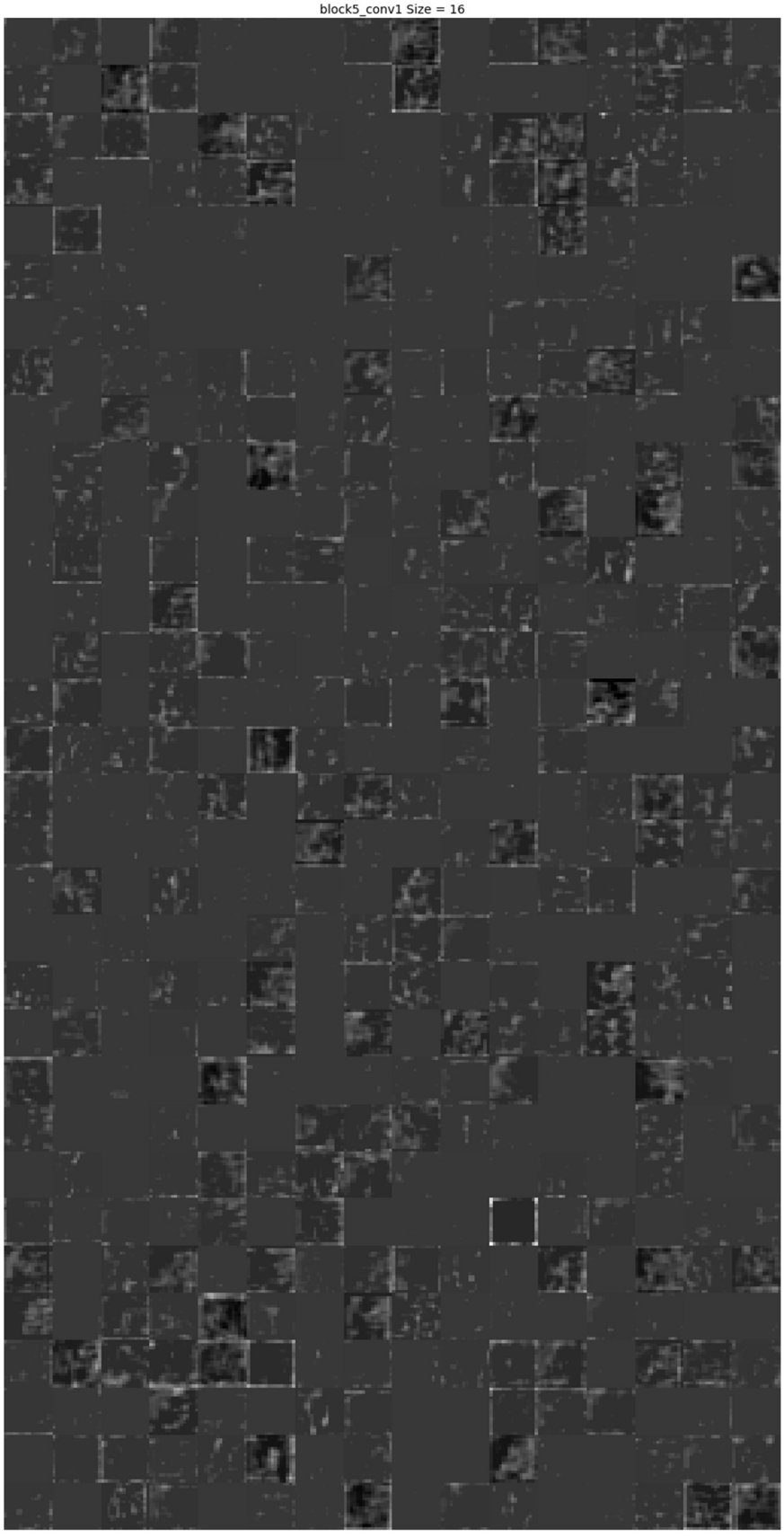
Figure 1 Example image used in activation layer samples.

example image used for observing individual layer activation is shown below. Several sets of convolutional layer activations are shown. The first shows a convolutional layer with a filter size of 256 pixels which is the same size of the resized input images. The



next activation layer example has a filter size of 64 pixels while the third image shows the activations for filter of size 16 pixels. Notice, the layers that have larger pixel filters (and are earlier layers in the neural network) tend to capture different levels of features





that are the same size of the image. Hence, the activation examples just look like

different shades of the original image. However, the convolutional layer with pixel sizes of 16 hone in on shapes that are indicative of the individual cells. The smaller pixel convolutional layers (which come in the later layers in the neural network) are also the layers that were trained while the earlier layers were frozen. This makes sense when looking at the convolutional activated layers. The finer details of the images need to be trained for the particular case while the earlier neural layers can be more general.

6. Conclusions and Recommendations

Images from the cellular painting project were collected and using a convolutional neural network with transfer learning different cellular assays were distinguished. Categorizing assays into chemical or genetic assays was very efficient with a validation and testing accuracy or 100% after training. The challenge was distinguishing assays within the same category (i.e. assays within the chemical category or assays with genetic category). The categorizing 5 chemical assays and 5 genetic assays into their specific assay resulted in a 41% accuracy. While categorizing just 5 chemical assays resulted in a 40% accuracy. When applying data augmentation techniques to the images (i.e. rotating images, zooming in on images, image brightness, etc.) the accuracy of distinguishing the 5 chemical assays improved to 52%. This ultimately can mean several things. First, distinguishing between chemical and genetic assays is very easy and this may be because of the dramatic difference in the images. Distinguishing between different assays within the chemical or genetic category are more difficult. Most likely this is because the difference between the different assays are too subtle. While image augmentation improved the accuracy, it saturated at around 50% accuracy which is better than guessing but still not good. There are several things that could be done to improve this project. First, the number of images used for individual assays was a small sample. It would be better to have a larger number of images per individual assay. The assays for this particular data set do not cause significant physical shifts in the cells and therefore are very difficult to identify using images. The set of assays used in the project may not have been ideal to be used in trying to distinguish assays using images. Further versions of this project would include more targeted assays to make sure that image recognition techniques could distinguish assays better. By doing this, one could then look deeper into distinguishing between different batches of assays and therefore develop techniques to remove batch effects from the data.