Machine Learning and Medical Imaging

JOSH DUNBRACK, Bucknell University, Department of Computer Science

As seniors at Bucknell University, we were assigned a project that would take place over the course of our entire senior year. Our task was to select a problem in the field of medical imaging and develop a solution with modern machine learning techniques. We used the NIH Chest X-Ray Dataset [6] and worked to classify images based on what diseases they showed. For the diseases with physical indicators, we also created a system to identify the regions that these diseases were located. Our systems were built on top of existing, more general work in machine learning; our classifier used the features and architecture of ResNet [3], and our localization approaches were based on Mask R-CNN [2] and Grad-CAM [5]. Ultimately, Mask R-CNN provided better results than Grad-CAM since we used simplistic heuristics to create bounding boxes from the Grad-CAM heatmaps. Our classifier outperformed existing work on all diseases and our localization approach provided consistently superior results, although a direct comparison is difficult due to obfuscated results from the NIH paper. Both localization approaches offer the potential for successful applications applied to other data sets in future research.

Additional Key Words and Phrases: medical imaging, machine learning, transfer learning, classification, region detection

1 INTRODUCTION

For our senior design course, we aimed to solve a problem related to medical imaging using machine learning. More specifically, we were given the following prompt.

> For many medical problems there is an associated medical imaging problem. The explosive growth in healthcare imaging has led to a concomitant growth in the amount of time physicians must spend interpreting this data. Very generally in this problem space then, we want to train the computer to automatically analyze and quantify medical image data in a way that is consistent with that done by medical professionals. Recently released large-scale datasets in eye, lung, brain, and heart disease can allow for the development of deep learning based diagnostic systems.

This prompt has a significant amount of freedom built into its phrasing. We were able to pick an available dataset and a corresponding problem to solve. This freedom also came with risks, however; many of us had little experience with machine learning and even less with medical imaging in particular. Picking a problem with the appropriate scope was going to be a challenge. Our brief agile training highlighted the difficulties associated with estimating task length, even for those on the scale of hours, and yet we had to choose a medical imaging problem that would take us around six months of schoolwork as a team. This was required as a part of our project proposal, which would also come before our first actual work on the project, making these estimates even more challenging.

We were not working without guidance, however. Professor Joshua Stough took on the roles of our course professor, our technical advisor, and our client, as his prior research into the subject matter made him the clear choice. The lack of a clear distinction between these roles came with its own set of procedural challenges, which would manifest themselves in our fall development process. His perspective acted as a litmus test of some of our more creative ideas, providing an idea of some of the existing approaches that have been used to solve the same kinds of problems in other disciplines.

Machine learning techniques can be used to solve problems in many fields of study, with extremely similar techniques. There are some common elements that make certain topics more suitable for these types of techniques. Machine learning generally requires a large quantity of data in order to learn features effectively, building an understanding of the problem

Author's address: Josh Dunbrack, jtd028@bucknell.edu, Bucknell University, Department of Computer Science, , Lewisburg, PA, 17837.

space by looking at and practicing on numerous examples. This aligns particularly well with the problems faced in medical imaging. As referenced in the provided problem statement, technological improvements have generated more images and information over time, but doctors do not have the time to analyze this influx of data. A system that could automatically analyze this kind of data without requiring as much from trained doctors would be incredibly valuable. The key constraint of machine learning - having sufficient data - would be an important factor when choosing our data set.

Multiple data sets were considered before one was ultimately selected. One such example was the MRI data provided by OASIS, which seemed both interesting and promising [4]. However, most of the options led to a similar concern: insufficient data. OASIS offered only a few thousand images, which *might* be enough, but our lack of experience led us to be more conservative. This led to our interest in the NIH Chest X-Ray Dataset, which offered over 100,000 images of different diseased and healthy patients [6]. We would later find out that even this quantity of images did not provide as much data as it may first seem - the images were broken down into front-facing and back-facing x-rays, and the large variety of diseases meant fewer cases for each disease. Nonetheless, the large number of images was the primary motivation for selecting this dataset.

Disease	Our Method
	P / R / F
Atelectasis	0.99 / 0.85 / 0.91
Cardiomegaly	1.00 / 0.79 / 0.88
Effusion	0.93 / 0.82 / 0.87
Infiltration	0.74 / 0.87 / 0.80
Mass	0.75 / 0.40 / 0.52
Nodule	0.96 / 0.62 / 0.75
Normal	0.87 / 0.99 / 0.93
Pneumonia	0.66 / 0.93 / 0.77
Pneumothorax	0.90 / 0.82 / 0.86
Total	0.90 / 0.91 / 0.90

Fig. 1. A table showing the accuracy for the labels of each class. Not all images were labelled originally, and so most labels were generated by a different system [6]. This limits the practical use of our network since some of the data is incorrect. Columns represent precision, recall, and F1 score, respectively.

With our dataset chosen, we now had to pick the problem that we wanted to solve. The labels of the data set made this decision rather straightforward; each image had a series of one or more associated diseases, with the "No Finding" label if no diseases were present. This led to the natural task of classification. Our system would look at a chest x-ray, analyzing its physical features to determine which diseases were represented in the image. Machine learning is often used on these kinds of problems, so we felt confident that this would be achievable.

However, we were concerned about the scope of the project. An image classifier is such a common task in machine learning that it might not take particularly long to implement it in this domain. The prevalence of sample code and the similarity between the techniques across disciplines made us hesitant to set our sights so low. Fortunately, the data set was not only tagged with the disease labels. Some of the images had associated bounding boxes, labelled as the x- and y-coordinates of its four corners, which would outline the region that the disease was located. This only applied to some diseases and some images for each disease, but offered an interesting and unique problem to solve. We thus chose

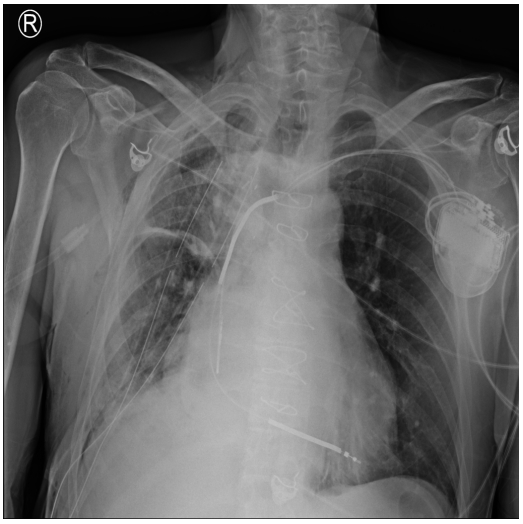

Fig. 2. A chest x-ray as provided in the NIH dataset [6], with the conditions of Emphysema and Pneumothorax.

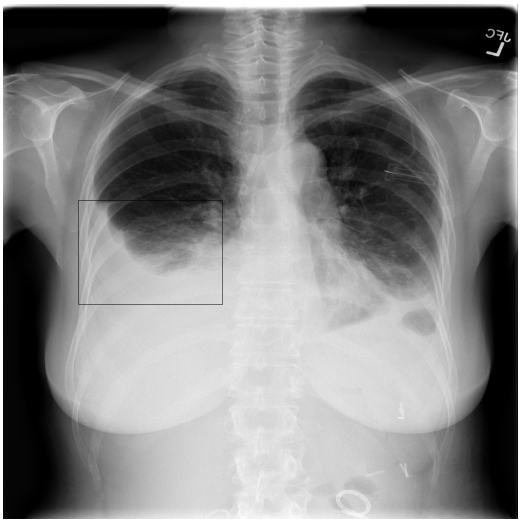

Fig. 3. An Atelectasis x-ray with the provided bounding box drawn.

to design a system that would determine which diseases were present in a chest x-ray, then construct a bounding box to locate the disease's physical symptoms.

2 BACKGROUND AND DESIGN

When developing a design for our system, we investigated a variety of machine learning approaches, both for classification and localization. We chose to use a convolutional neural network (CNN) as our classification technique at the start of our project [1]. It is the most common approach for image-analysis problems, offering an established means of observing image features and learning the patterns of a particular field of study. Moreover, multiple members of our team had experience working with CNNs before.

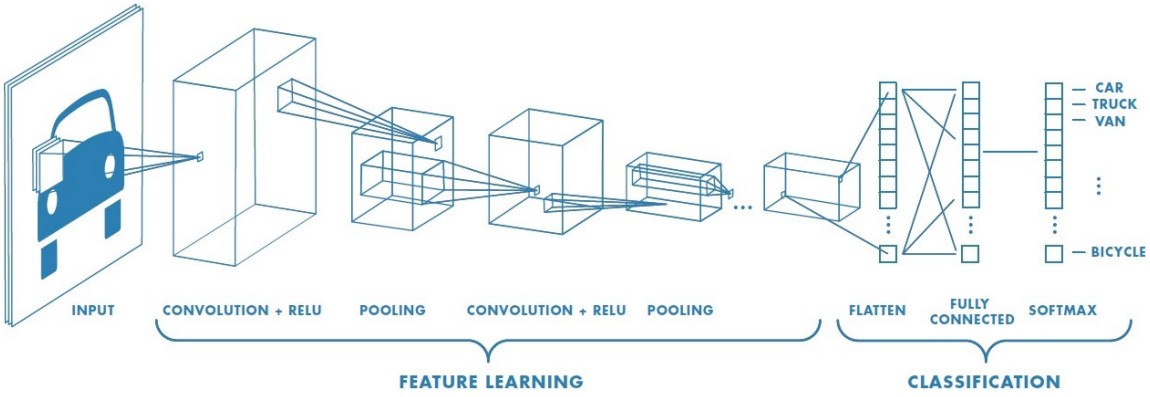

Fig. 4. A diagram showing the structure of a standard CNN. It first reduces an image to spatially coherent features, then weights those features to pick a classification.

A convolutional neural network has two main components. The network starts by breaking down an image into a series of features which are then used to classify the image. The first component, the feature learning, looks at each group of pixels in the image using a **kernel**. This allows it to learn the various structures of pixel groups that correspond with each class. The image size is reduced during this process to allow the learning of larger-scale features. After repeating this process a few times, the final set of features are flattened into a single, spatially-ignorant list of all of the identified features. These features are processed by fully connected network layers before being normalized to provide a particular classification.

While this approach is extremely powerful as-is, we would need to make some adjustments in order to apply it to our particular problem. Unlike traditional classification problems, images in our dataset are not limited to a single class due to patients having more than one disease. There are a few existing ways around this problem, including decoupling different classes by converting the final softmax normalization to a binary cross-entropy (BCE) loss or using a different binary classifier for each disease. These required adaptations did not deter us from proceeding with a CNN for our classifier.

Learning the features of our images from scratch has a few risks; it might take the network a very long time to understand what an appropriate feature might be, and the training could get caught in local minima during the optimization process. As such, it would potentially capture features that are good but could be better. Fortunately, we can take advantage of existing work in image-based classification. Networks such as ResNet offer a set of features that behaves well on the variety of image types presented in the ImageNet dataset [3]. By starting with these features, our network can focus its training time on understanding how these features correspond with each class rather than trying to learn the features from scratch at the same time. This procedure is an example of *transfer learning*, using a model for one problem to help solve a different problem.

Initially, the same technique was applied for bounding box generation, attempting to use our classifier to directly output a bounding box. Since the features would correspond with the various classes, we thought that the network might be able to use its internal knowledge of feature location to identify where the most important parts of the image were. If the network were truly learning how to recognize the disease, it would likely be focusing on the parts of the image where the disease is present. In our research, we found out that this concept of "network focus" had been previously explored using Gradient Activation Maps in Grad-CAM [5]. They were used to help avoid overfitting, ensuring via manual inspection that the network was basing its classification off of the appropriate component of the image rather than any confounding variables. As with the CNNs, this would require additional adaptation for our purposes, since our regions were only provided in a bounding-box format. Even if it ended up failing for localization, however, we could still use it for its intended purpose of understanding any potential overfitting.

Since there were no guarantees that the gradient activation map approach would work for localization, we simultaneously explored alternative options. We decided to move forward with Mask R-CNN in parallel with gradient activation maps, noting that it seemed the standard network for image segmentation problems [2]. Our problem was not exactly segmentation; there were no masks for what constituted each disease in our images, only bounding boxes. In contrast, segmentation relies on understanding each pixel as either inside or outside of a given object. However, existing localization approaches for the NIH dataset did not achieve particularly strong results. As such, constructing masks from the bounding boxes themselves may be able to perform sufficiently well.

Having investigated the existing work that we could use for our own purposes, we could now decide how best to implement our desired system. Our overall system structure was determined by the problems we had selected to solve.

Data preprocessing allows our system to read in the images from their provided format for use in the network. The images provided are 1024x1024, having already been normalized in size for their inclusion in the dataset. The original dataset has its training labels and bounding boxes stored in .csv files. We sorted our data on disk to make the training process easier. For a user image, however, preprocessing is quite simple, only requiring loading it in to our CNN.

We used custom CNNs with features learned from ResNet-18 to classify our images. ResNet-18 handles images of arbitrary size, including its own preprocessing to fit images into its 224x224 expectation. Each disease is evaluated with its own Binary CNN. Currently, these CNNs are trained on a complete lack of disease vs. an image with the given disease, meaning its behavior on other diseases is unspecified - this will (hopefully) be fixed by the final release.

Not all of the classified diseases have any bounding boxes at all; some of them are overall conditions as opposed to those with distinct physical lesions. For the localizable diseases, we use the two previously discussed approaches. We generate gradient activation maps from our classifier and create bounding boxes from the heatmaps. We also have a Mask R-CNN network trained on our bounding-box-masked images that generates its own bounding boxes. These bounding boxes are generated visually and saved to disk, while also providing the raw corner coordinates.

3 IMPLEMENTATION

As with many of our prior decisions, team and client experience motivated which tools we chose to use. In the machine learning space, most work is done in Python. There are a few primary reasons for this, including the availability of Python libraries and the ease of use. Also, the main downside of Python as a language, its slow computation, is offset by the optimized libraries and already high cost of neural network training. We chose to work with the PyTorch library over TensorFlow because despite the benefits of TensorFlow's zoomed-out perspective accessible through Keras, some team members had already done some image processing work with PyTorch prior. We used Jupyter notebooks to run our Python code due to its convenience in a remote world; Jupyter's web-based access allows team members to easily run code on Bucknell's computers, supported by powerful graphics cards, without requiring physical access to the referenced machines. It also supports combining Markdown-formatted text with the code segments, offering an easy way to provide useful, readable documentation of the code.

While our networks were trained on Bucknell's machines in the Academic East building, this was not our original plan. We had hoped to take advantage of BisonNet, a remote computing cluster with multiple GPUs accessible for a single task. This would have the theoretical advantage of offering more computing power to increase the training speed, with a few drawbacks. Submitting a request to BisonNet was not a trivial process, especially considering that it did

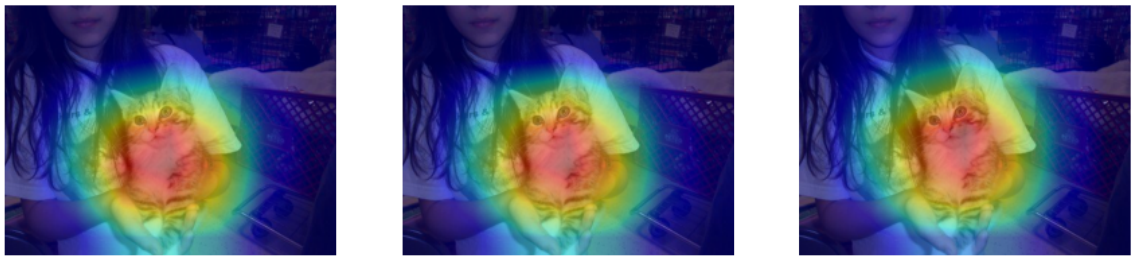

Fig. 5. Gradient activation maps applied to ResNet for the "cat" class, generated by our PyTorch implementation of Grad-CAM. The heatmap shows the parts of the image that most led the network to conclude that the image has a cat in each of the three color channels.

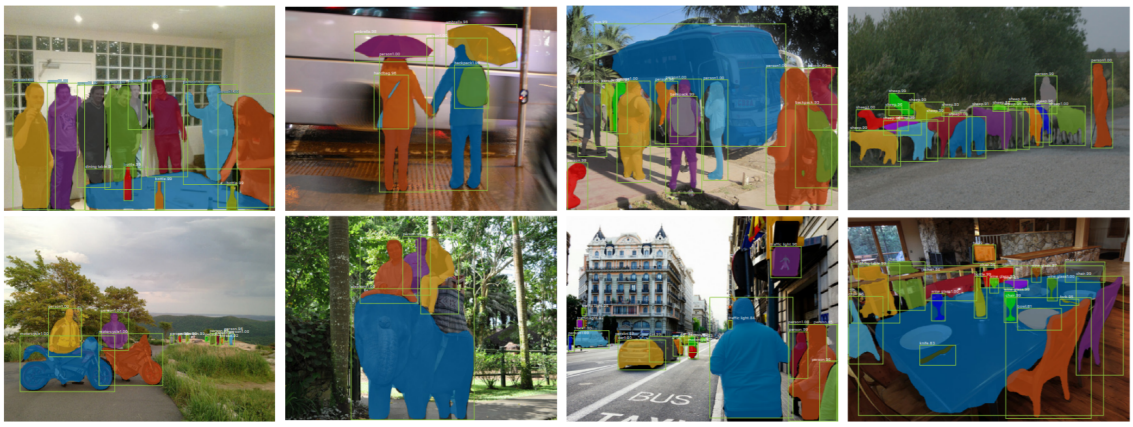

Fig. 6. A visual representation of Mask-RCNN's multiple outputs on sample images. We would only be using the bounding-box component of the output. [2]

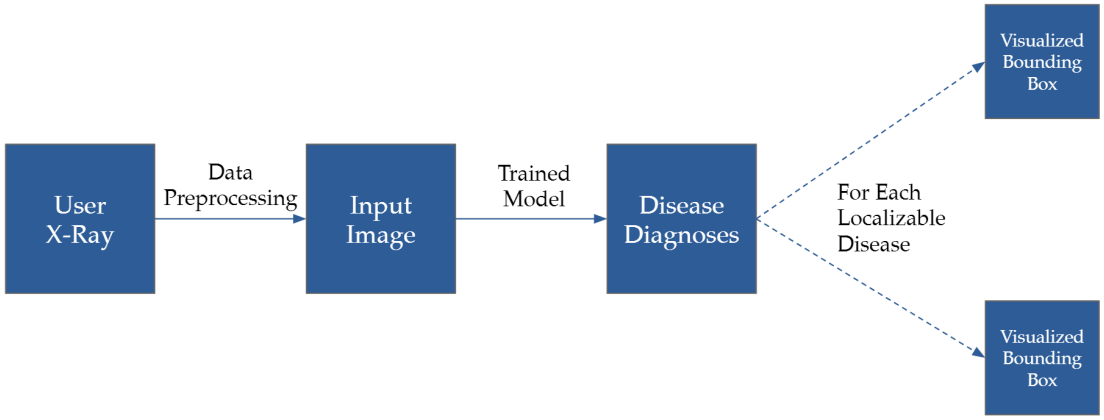

Fig. 7. A diagram representing the general structure of our solution. This is the natural data flow for attempting to classify and localize diseases in a given x-ray.

not work well with Jupyter notebooks in particular. Even when we surpassed that obstacle, however, the network still trained slower than on the original machines. There is an overhead cost to splitting training across multiple GPUs and handling that memory, but that did not explain the drastic difference in performance. Ultimately, we had to move on from the potential power of BisonNet in favor of a more practical and currently successful solution.

Most of our work is contained in an evolving set of Jupyter notebooks, some for temporary use and some needed for each network run. The dynamic, agile nature of our project combined with the need to explore many options that eventually fail means that many of our notebooks quickly lose their value. As such, most of our work is contained within a few key notebooks, each with a specific task. One notebook is used to generate the classification models, saving them to a folder. We then have two notebooks to use the classifier and localize found diseases, one for each of our localization approaches (gradient activation maps and Mask R-CNN).

Our data is stored in a custom-organized file structure to assist in the training process. The data directory has a subdirectory for each disease, each of which has two subdirectory corresponding to the two perspectives that the x-rays can be taken from. This allows us to load in a random assortment of images for each disease and perspective quite easily without referencing the original .csv file that contained such information.

Testing in the traditional software development sense was not a priority on this project as emphasized by our client. The technical pieces of our work, including continuing to optimize our final network's results, superseded the "nice-to-have" features that we could have worked on instead. That is, our client stressed that they cared most about the performance of our network, with nearly everything else optional. We did evaluate our network's performance with the discipline-standard performance metrics. For classification, the receiver operating characteristic curve (ROC curve) acts as a visual indicator of the network's performance, plotting the true positive and false negative rates for varying thresholds. The area under this curve (AUC, AUROC) quantifies how well separated the positive and negative classifications are for each class of disease. Localization used the standard metric of Intersection Over Union, comparing the overlap between the proposed region with ground truth to the total size of the two regions. To compare to the data from the NIH paper, we also evaluated localization by what proportion of the bounding boxes exceeded a particular overlap threshold for five distinct threshold values. This avoids the average being changed by outliers and provides insight into the network's consistency beyond a simple average.

4 RESULTS

Network	Atelectasis	Cardiomegaly	Effusion	Infiltration	Mass	Nodule	Pneumonia	Pneumothorax
Classification Score (AUROC)								
Deep MedIA	**0.74**	**0.87**	**0.81**	**0.68**	**0.78**	**0.74**	**0.68**	**0.83**
NIH	0.7069	0.8141	0.7362	0.6128	0.5644	0.7164	0.6333	0.7891
IoBB Average								
Grad Map	0.039	0.226	0.163	0.076	0.113	0.055	0.033	0.061
Mask R-CNN	**0.46**	**0.818**	**0.393**	**0.554**	**0.478**	**0.425**	**0.579**	**0.227**
T(IoBB) = 0.1								
Deep MedIA	**0.806**	**1.0**	**0.806**	**0.96**	**0.883**	**0.938**	**0.958**	**0.55**
NIH	0.7277	0.9931	0.7124	0.7886	0.4352	0.1645	0.7500	0.4591
T(IoBB) = 0.25								
Deep MedIA	**0.611**	0.967	**0.677**	**0.92**	**0.765**	**0.625**	**0.875**	**0.4**
NIH	0.5500	**0.9794**	0.5424	0.5772	0.2823	0.0506	0.5583	0.3469
T(IoBB) = 0.5								
Deep MedIA	**0.472**	**0.9**	**0.419**	**0.64**	**0.353**	**0.375**	**0.583**	**0.2**
NIH	0.2833	0.8767	0.3333	0.4227	0.1411	0.0126	0.3833	0.1836
T(IoBB) = 0.75								
Deep MedIA	**0.333**	**0.867**	0.129	0.28	**0.294**	**0.188**	**0.333**	0.0
NIH	0.1666	0.7260	**0.2418**	**0.3252**	0.1176	0.0126	0.2583	**0.1020**
T(IoBB) = 0.9								
Deep MedIA	0.083	0.3	0.032	0.12	**0.118**	**0.125**	0.0	0.0
NIH	**0.1333**	**0.6849**	**0.2091**	**0.2520**	0.0588	0.0126	**0.2416**	**0.0816**

Table 1. A collection of all of the numeric results for our network compared to the NIH paper [6]. Localization scores for Deep MedIA refer to the Mask R-CNN approach as it superseded the gradient activation map scores.

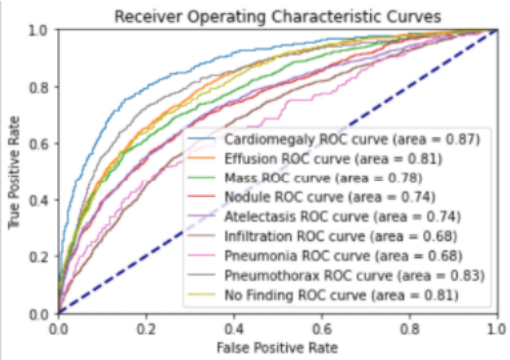

Fig. 8. The ROC curves for each disease's classifier along with the AUC for each curve. The curves can also be used to find an appropriate threshold to balance true positives and false negatives.

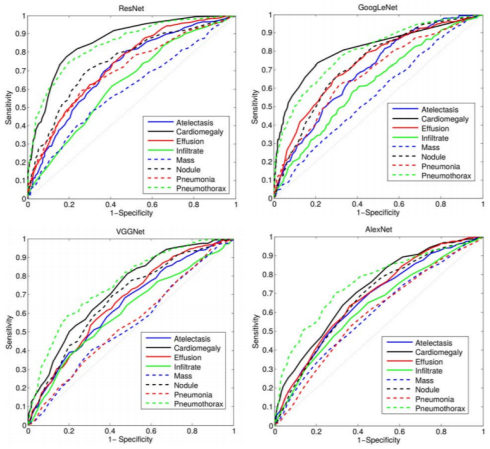

Fig. 9. The ROC curves of the multi-label classifier published with the NIH dataset [6]. The AUC values are reported in the table below.

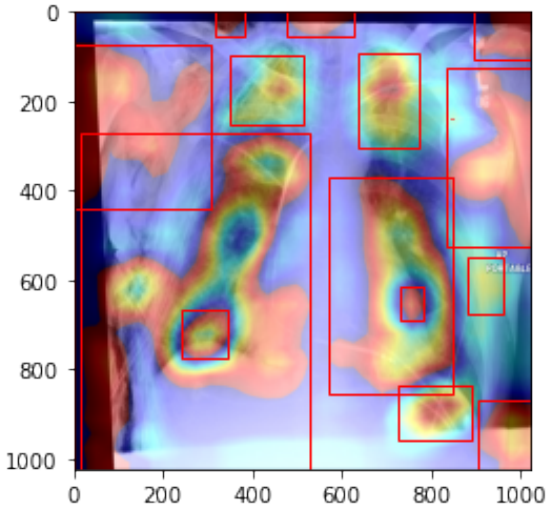

Fig. 10. An example gradient activation map of our network with bounding boxes. While there seems to be some error potentially involving a lack of normalization, it does seem as though the focus is in relatively the right spot. There do appear to be some segments outside of the body with high values, suggesting overfitting; data augmentation may be needed to remedy this.

For classification, our network outperformed all networks that were published with the NIH dataset, with higher AUROC values for all diseases. This is particularly important for any practical use of this system since localization of diseases only provides meaningful results if the disease is actually present. We tested two approaches for localization: gradient activation maps and Mask R-CNN. For this dataset, Mask R-CNN was much more successful. Gradient activation maps needed a series of heuristics and manual approaches to generate bounding boxes from the heatmaps, and so

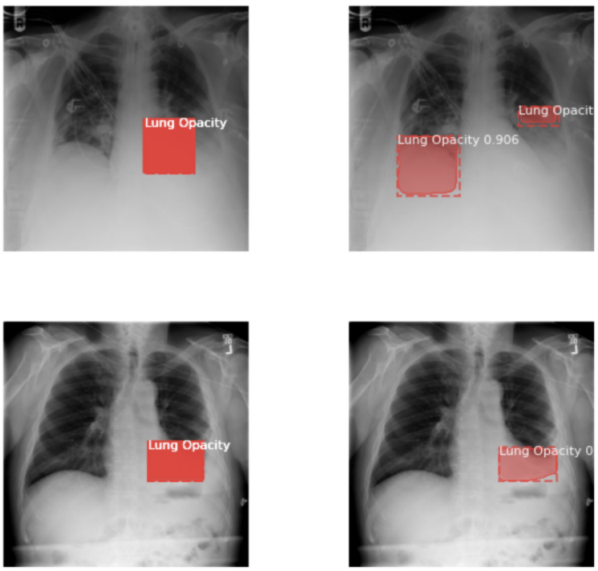

Fig. 11. Our more successful Mask R-CNN bounding box generation.
Left: the ground truth. Right: proposed regions.

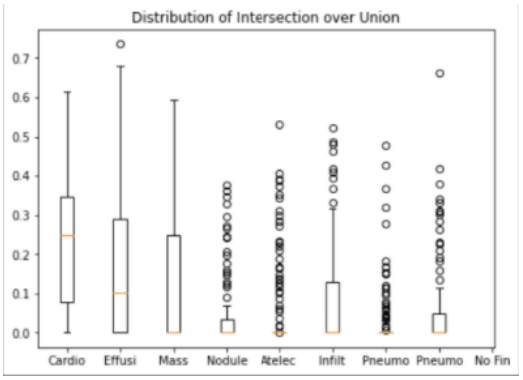

Fig. 12. Plots of IoU values for the gradient activation map approach

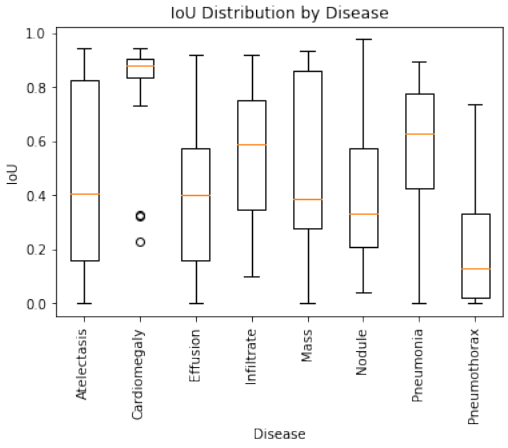

Fig. 13. Plots of IoU values for the Mask R-CNN approach

Fig. 14. Comparing the IoU graphs of the two approaches, Mask R-CNN outperforms the gradient activation maps significantly in all diseases. More work on constructing bounding boxes from a heatmap would be required to improve the performance of the gradient activation map system.

with the time we had Mask R-CNN was easier to use and performed better. As such, we compared the results from Mask R-CNN to the NIH paper with the metrics that the paper provided. Our network was much more consistent, outperforming the NIH system on maintaining low to moderate amounts of overlap. Our network fell short at providing

perfect bounding boxes, offering worse results at a 90% overlap threshold, but given the inconsistency inherent in the data set, consistent decent performance is more likely to be ultimately helpful.

It is difficult to evaluate the practical usefulness of our network. Without data on the classification rate offered by actual doctors, we cannot make a fair comparison to understand how well our network is performing. With that said, we can speculate a bit. I think the product as we created it will not be sufficiently accurate to support practical use in its current state. The limitations of the dataset combined with the compounding inaccuracies provide built-in flaws; any dataset issues lead to classifier mistakes, which in turn damage the region-detection process. With that said, I think the work is still valuable. The network can be easily adapted if a new, more thorough dataset is released, especially one with better or more bounding box data. It also offers a base for future research to build on, just as this network was built on existing work. As the paper that released with this data set wrote in its abstract: "[D]eep convolutional neural network based reading chest X-rays... remains a strenuous task for fully-automated high precision CAD systems." [6]

5 DEVELOPMENT PROCESS

Throughout this project and course, our work was structured by an agile development process, at least in theory. Our project's actual journey was not quite so simple. Our commitment to the agile process was intermittent at best, with most group communication occurring during our biweekly meetings rather than asynchronously through various chat mechanisms. Many of our agile artefacts were also created retroactively, especially for time-tracking. These issues were exacerbated by a general confusion about our project that permeated our starting weeks, trying to understand how to distinguish the advice from Professor Stough between his roles of client, professor, and technical advisor

We had mixed levels of success in our agile process. The frequent group meetings, especially those involving Professor Stough, gave us consistent technical guidance while also ensuring we were working on the pieces that provided the most value. We were able to investigate, start, and abandon many possible solutions throughout our project, ensuring that we were keeping our options open and proceeding with the best possible plan. When we stuck to the concept of agile and kept our processes flexible, we were quite successful. In contrast, however, the pieces of the agile process that were required as course artifacts often slowed us down and caused problems. Some of this is to be expected, since such work can take away from development time. Even if individual pieces were not particularly disruptive, there were some common themes that permeated our group's experience.

One such theme was being required to make commitments before we were at a place of understanding to do so. With the project proposal, for example, we had to determine our dataset and ultimate project goal before being able to learn the technology and properly evaluate task difficulty. This is a reasonable order of operations for most groups, since there are at least parts of the project that most groups know will be necessary, but our self-defined project combined with a lack of client interaction in the early weeks led to a significant amount of uncertainty. This continued into our initial attempt to write user stories for the entire project, at a time that we had barely determined what it was we were doing.

Similarly, the concept of creating a semester-long burndown chart, especially one with implied associated accountability, led to a significant struggle to stay agile. As our understanding of our project grew, so did our understanding of our future tasks, leading us to add, remove, and re-estimate tasks quite frequently. Modifying our burndown chart, however, was not such a simple process, especially since our "expected pace" trendline was based on our starting hour estimate. We chose to ignore GitLab's time tracking for a similar reason; if a two-hour task took six hours with no end in sight, there was not an obvious and clean way to handle it. It was also unclear how or if time-draining components like group meetings would be tracked, via burndown chart or otherwise. This also caused issues with sprint loading

in the first few sprints. We would lay out our tasks for the three-week sprint, and by the end of the first week we would realize that what actually needed to be done was very different from what we had thought. Adhering to the SCRUM-based agile process would require us to not change our course until the end of the sprint, however, which was not reasonable for us to meet our proof-of-concept deadline. These challenges led us to work around the agile process rather than working with it with the goal of hoping to stay truly flexible and agile throughout the semester.

Overall, our agile experience was hampered by a lack of experience and unclear expectations. Since the agile process could only be so monitored via the classroom aspect, we ultimately followed the pieces of the agile process for those components while often losing the spirit of agile development. Fixing this would be rather difficult, however, due to the time limitations of a Bucknell course and the experience limitations of students. I think an emphasis on agile principles and concepts rather than specific implementations could be a suitable replacement, both for the current course and for the prior education on SCRUM, since those concepts are what provide actual value. It would make the evaluation side more difficult, but I am uncertain as to the ease of evaluation in the current system as well. Despite the many specific problems that plagued our group, our commitment to staying agile was ultimately quite valuable to our project's success.

6 CONCLUSION

Creating this system was an effort of many parts. We researched a variety of datasets, problems, solutions, and adaptations that ultimately led us to our classification and localization system. Our network is built on previous work, including ResNet-18, Grad-CAM, and Mask R-CNN. We adapted these tools for our own purposes, creating a system that is more than the sum of its parts. Our classifier outperformed existing methods across the board. Grad-CAM provided some promising initial results, but required significant heuristic work to create meaningful bounding boxes from the provided maps. Mask R-CNN worked succeeded in its localization task, surpassing the existing results for providing consistently meaningful bounding boxes, despite the fact that bounding boxes were used as masks for the neural network. It also performed well across all diseases where the original paper was inconsistent on Mass and Nodule. Ultimately, our system outperformed existing designs, and Grad-CAM showed the potential for further improvement and use in similar applications.

REFERENCES

[1] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. 2012. Multi-column Deep Neural Networks for Image Classification. *CoRR* abs/1202.2745 (2012). arXiv:1202.2745 http://arxiv.org/abs/1202.2745

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2018. Mask R-CNN. arXiv:cs.CV/1703.06870

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:cs.CV/1512.03385

[4] Daniel S. Marcus, Tracy H. Wang, Jamie Parker, John G. Csernansky, John C. Morris, and Randy L. Buckner. 2007. Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults. *Journal of Cognitive Neuroscience* 19, 9 (09 2007), 1498–1507. https://doi.org/10.1162/jocn.2007.19.9.1498 arXiv:https://direct.mit.edu/jocn/article-pdf/19/9/1498/1756878/jocn.2007.19.9.1498.pdf

[5] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2019. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision* 128, 2 (Oct 2019), 336–359. https://doi.org/10.1007/s11263-019-01228-7

[6] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. 2017. ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3462–3471. https://doi.org/10.1109/CVPR.2017.369