Comparing Sparse and Dense Neural Networks: Using AI to Detect Cancer.

New Mexico Supercomputing Challenge

Final Report

April 8, 2020

Team 1005

Los Alamos High School

Team Members:

- Charles Strauss

Mentor:

- Dr. Garrett T. Kenyon

Comparing Sparse and Dense Neural Networks: Using AI to Detect Cancer.

Charles M. S. Strauss, Austin M. Thresher, Garrett T. Kenyon

Executive Summary

Pathologists view slides containing millions of cells, and gigabytes of data, to discover tumors [3]. Even highly trained pathologists, however, may still come to different diagnoses [3]. To assist pathologists, deep learning models have been developed to provide an "extra set of eyes" [5]. However, deep learning is a black box algorithm, in which the reasons for a given decision are often unclear. Moreover, adversarial examples demonstrate that deep learning often relies on meaningless or non-semantic features [2]. Here, we apply a type of transfer learning based on autoencoders for annotating pathology slides. Such an approach indicates which pixels the models uses as evidence for tumors, making their decisions more explainable. Two types of autoencoder are examined: a deep denoising bottleneck autoencoder, and a sparse autoencoder. We find that both autoencoders did well at tumor discovery at the single pixel level, supporting AUC ROC scores of approximately 0.85 on holdout slides. To better visualize what features the models were using for classification, we preformed standard adversarial attacks against the deep denoising bottle neck autoencoder. We found that while some attacks appeared semantically reasonable, others did not. Small amplitude attacks against the deep denoising bottle neck autoencoder transferred poorly to the sparse autoencoder, suggesting that the two classifiers may use different criteria for classification. We conclude that autoencoders represent an approach to developing tools for assisting pathologists.

1 Introduction

Tumor discovery in biopsies is currently a task done by human pathologists, requiring an enormous amount of training and manual viewing of millions of cells. Even highly trained pathologists can miss small tumors, or disagree on the interpretation of an ambiguous region [3]. To help pathologists be more accurate by providing "an extra pair of eyes", deep learning models have been trained on datasets of annotated pathology slides [3]. On whole-slide-classification, some deep learning models have outperformed human pathologist.

However, trained pathologists can not only spot a tumor in tissue slices (pathology slides), but they can also explain how they classified the tissue. In particular, pathologists can annotate which regions show evidence of tumor and which regions do not, and do so at the cellular level. Although these complex deep learning models appear to have human-pathologist-level performance, nobody can currently tell exactly how these complex deep neural networks find cancer, and the FDA does not allow black box (inexplicable) models for use in medicine. It frequently occurs that minute pixel-level changes to an image can "trick" the classifier into emitting the wrong classification. These alterations can be so slight that humans can't even perceive them. Thus the combination of not knowing what a deep neural net uses to find tumor and unexpected extreme sensitivity to small artifacts in images (such as a lens glint or unevenness in staining the tissue) is a grave concern for AI medical diagnosis.

Our work addresses both of these issues. First, we compare the ability of a deep neural network to extract information from the slides to a different type of neural network, based on sparse coding, (We choose this approach because sparse coding concentrates the decision features into a smaller set than a deep network, and thus simplifies interpretation.) we show that this (simpler) sparse coded approach performs equally well to a (complex) deep neural net for identifying cancer regions in pathology slides. Second, we will compare the two networks on their robustness to small image artifacts. Not every random perturbation will cause miss-classification. Instead, we discover a small perturbation that causes miss-classification (known as an Adversarial Example). These allow quantitative measures of robustness, and insight into what features the models uses to make decisions.

Our approach is not intended to replace the Doctor but instead guide and assure their own assessment. Notably, both the deep and sparse models we create identify the pixels that are tumorous. Pixel-level classification is critical because it directly shows the Doctor what in the slide is causing the cancer diagnosis, and thus aids the Doctor's judgment of the diagnosis. In contrast, most other approaches to pathology slide analysis do not resolve cancer at the pixel level, but instead predict whether a tumor is likely in the slide somewhere, without identifying the source of the evidence for the Doctor to assess.

We demonstrate two models capable of making lesion-level annotations: a Deep Convolutional Neural Network (DCN) model, and a Sparse Coding (SC) model. Adversarial examples are generated on the DCN model, and transferred to the SC model with varying amounts of the perturbation. Next, AUC ROC and AUC PR were used to determine how well each model could classify on the lesion-level.

2 Methodology

2.1 Dataset

We use the Camelyon 2016 dataset. We segmented each pathology slide into tiles of size 512x512. The slides also consist of large areas of white space, so areas that were found to be mostly white were left out entirely.

Labels were rasterized from the XML annotations provided by the Camelyon 2016 dataset. Each slide that was not meant for testing has an XML file associated with it, containing the actual pathologist annotations. A program that checked if a point was inside the tumor annotation area was written, and used to create black and white tiles of the annotations. These labels are of the shape 128x128x1.

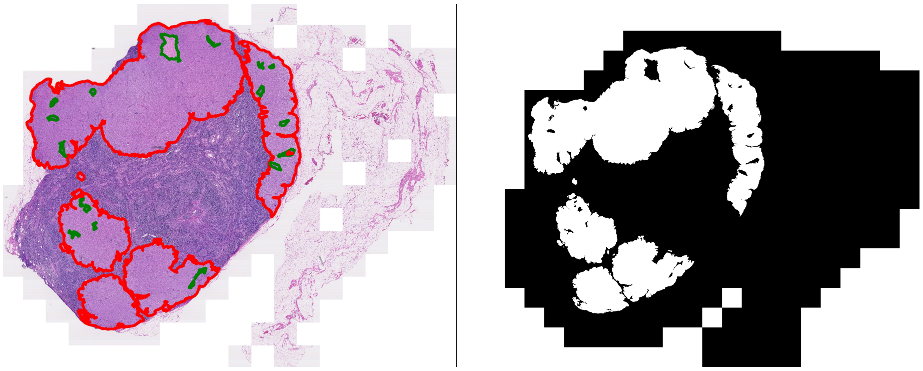

Figure 1: Left: Pathologist annotations. Red lines encircle tumorous areas. Green lines encircle non-tumorous islands. Right: Labels created for training. White is tumor, black is non-tumor.

2.2 The Basic Model

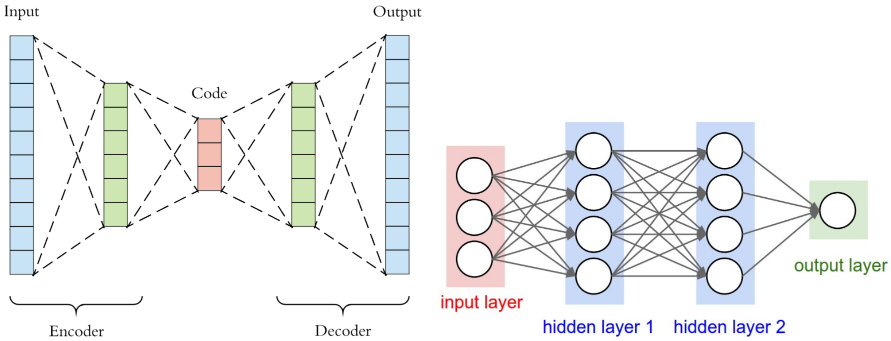

Figure 2: Autoencoder neural network and classifier are trained separately. Autoencoder compresses the tile into the red latent representation. The latent representation is then used to predict a tumor heat map in the classifier.

The basic model is an autoencoder, which feeds a classifier. The autoencoder is trained to reconstruct pathological tiles.

The classifier then takes the latent representation from the autoencoder, and returns confidence levels in a 128x128x1 map of the slide. The classifiers were trained using Binary Cross Entropy loss. Finally, their final outputs were compared to the actual annotations through ROC and PR curves. ROC curves were used by others who did the same task with deep learning, thus allowing us to compare our results to theirs. PR is used for its capacity to better represent highly unbalanced datasets, and to further compare the performance of the DCN and SC models.

2.3 The DCN model

The DCN model consists of a deep denoising autoencoder, and deep classifier. Because this model will be attacked with adversarial noise, it is trained as a denoiser, meaning that it learns to remove noise from tiles. See Figure 3 for performance on reconstruction.

2.4 The SC model

The SC model uses a sparse autoencoder to compress input tiles, and a deep classifier to generate tumor heat maps. The sparse encoder was originally built by [1], and adapted for use with our deep classifier. Sparsity enforces a constraint where the minimum number of neurons which best represent the given information are used, hopefully making a sparse representation more difficult to alter. See Figure 4 for performance on reconstruction.

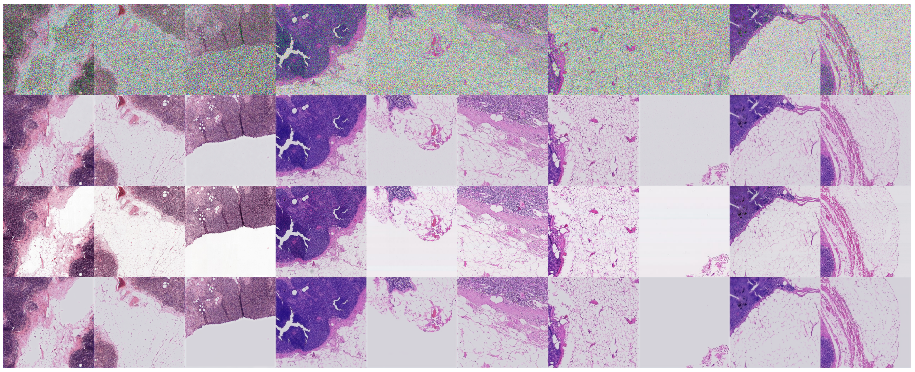

Figure 3: Top Row: noisy inputs to deep denoising autoencoder (DCN). Row 2: Autoencoder reconstructions from noisy data. Row 3: noiseless inputs to deep denoising autoencoder. Bottom Row: Autoencoder reconstruction from noiseless inputs. Note: Details are preserved across the bottleneck, validating the autoencoders latent representation.

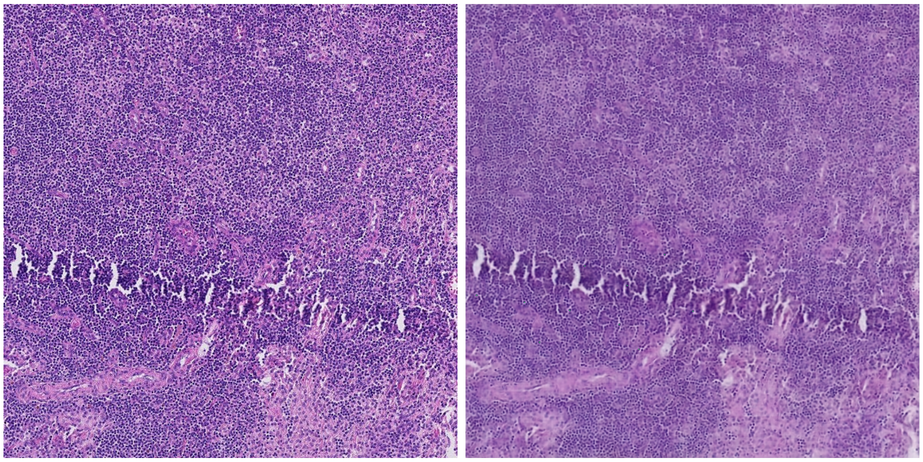

Figure 4: Left: original tile. Right: reconstructed tile. Although the reconstruction may look more blurry than the original, detail on the cellular level and higher is clearly preserved.

2.5 Adversarial Examples

Adversarial Examples were generated using the Fast Gradient Sign Method (FSGM). After the gradient was taken, only a fraction was added to the input tile, at 1%, 10%, and 15%. Both models were subjected to the adversarial examples generated through the deep model. The 1% perturbation that was added represents a change so small that it should be meaningless. No adversarial examples were found for the sparse model. Outputs for original tiles, and tiles with 1% and 15% of the perturbation are shown.

3 Results

We found that the sparse model was more robust to adversarial examples than the deep model. Although, further research is warranted to determine if adversarial examples generated for the sparse model can fool the deep model, and if any adversarial examples generated could fool a human pathologist.

3.1 The DCN model

Figures 5 & 6 show how adding the 1% perturbation to the input of the DCN model caused a 5% change in AUC ROC, and 14% change in AUC PR. The perturbations added make no sense to a human (the added noise does not reflect any features seen in pathological slides), so the DCN model can not be verified as explainable.

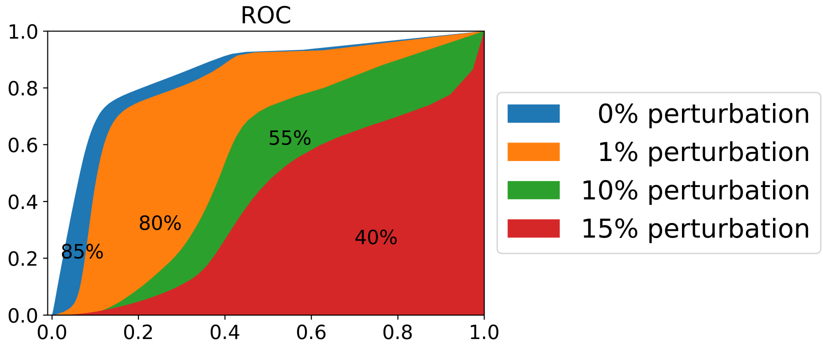

Figure 5: Blue: AUC ROC score of 0.85 for tiles that had no adversarial perturbation (straight from the dataset). Red: AUC ROC score of 0.40 on tiles with the 15% perturbation. A true random guesser would get an AUC ROC of 0.50 here.

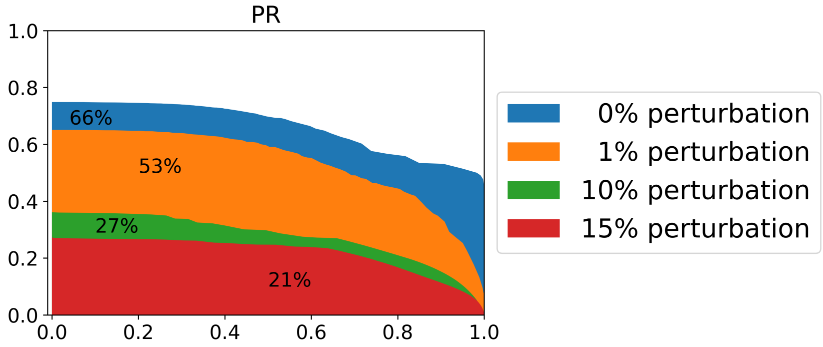

Figure 6: AUC PR is reduced to a third of its original AUC by the 15% perturbation.

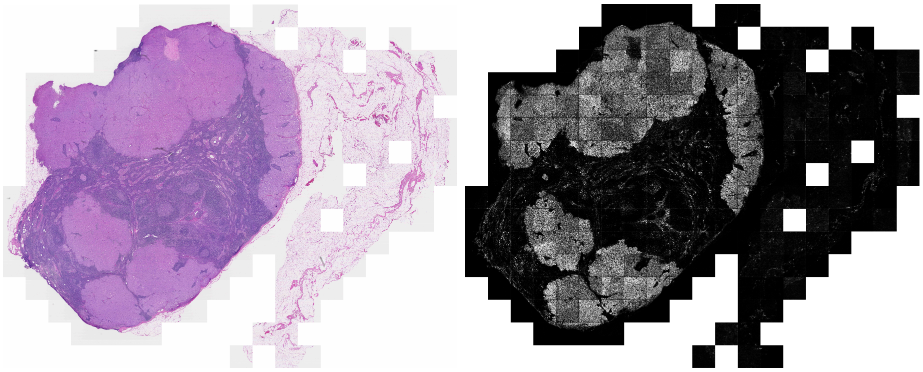

Figure 7: Left: Original tile with no perturbation. Right: Predictions on tiles with no perturbation. This slide has an AUC ROC of 0.85, and an AUC PR of 0.67.

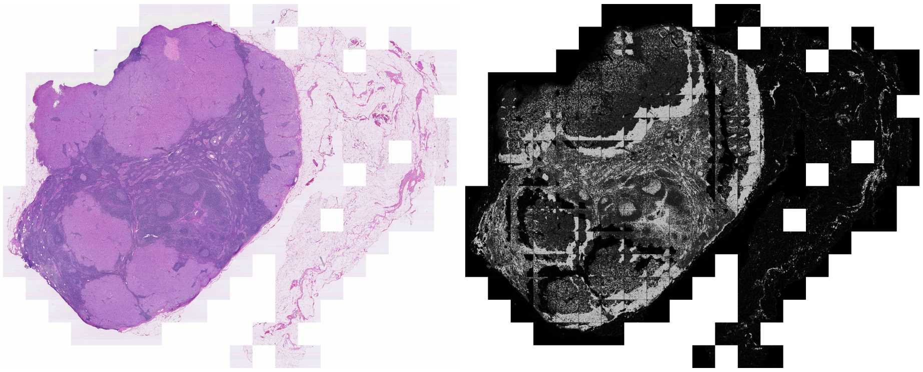

Figure 8: Left: Slide with 1% perturbation. Right: Predictions on slide to the left. This slide has an AUC ROC of 0.80, and AUC PR of 0.53.

Figure 8 is nearly identical to Figure 7 in input slides, however the predictions are nearly inverted (tumor is where non-tumorous annotation should be and visa versa). This demonstrates how little of a change is required to fool the DCN.

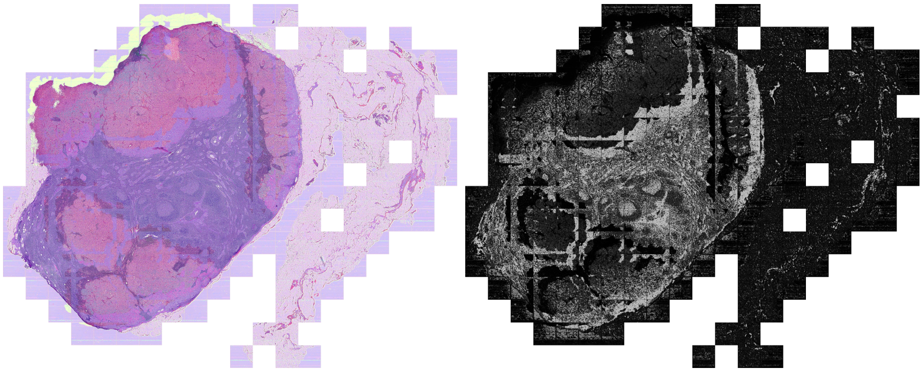

Figure 9: Left: Slide with 15% perturbation. Right: Predictions on slide with 15% perturbation. This slide has an AUC ROC of 0.40, and AUC PR of 0.21. The predictions are completely inverted.

3.2 The SC model

The same three adversarial examples shown earlier were tested on the SC model. Below are predictions for the 15% and 1% added perturbation slide, and original slide.

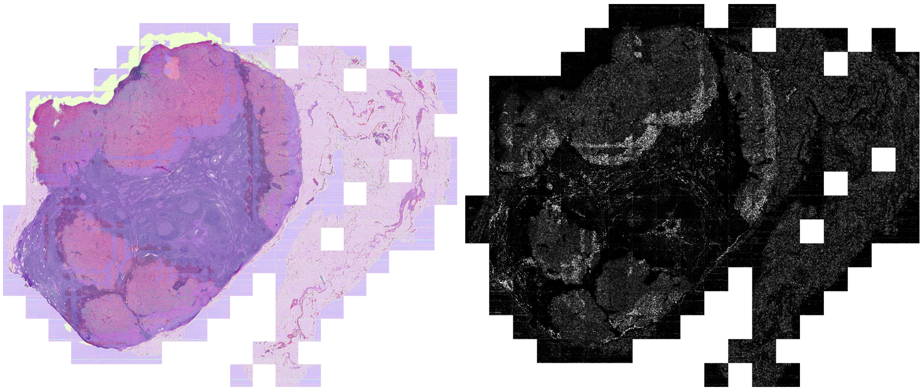

Figure 10: Left: Slide with 15% perturbation. Right: SC model predictions on slide with 15% perturbation. Tumor is no longer recognised with much confidence. However slide is not inverted, so is more robust than DCN model.

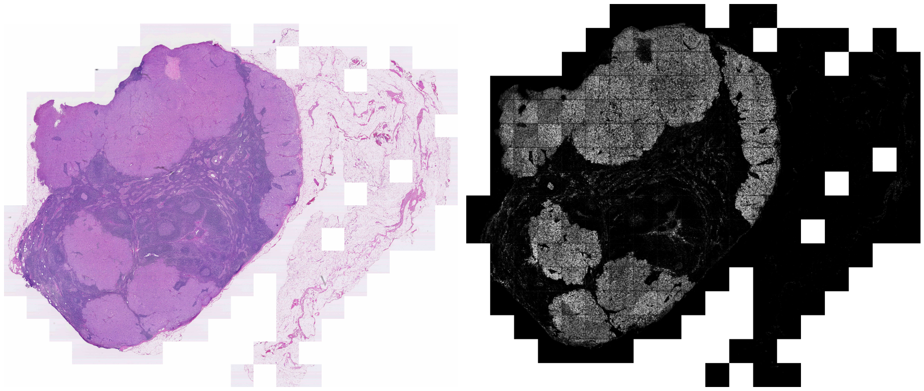

Figure 11: Left: Slide with 1% perturbation. Right: SC model predictions on slide with 1% perturbation. The DCN model was nearly inverted by this adversarial examples, yet the SC model is unaltered.

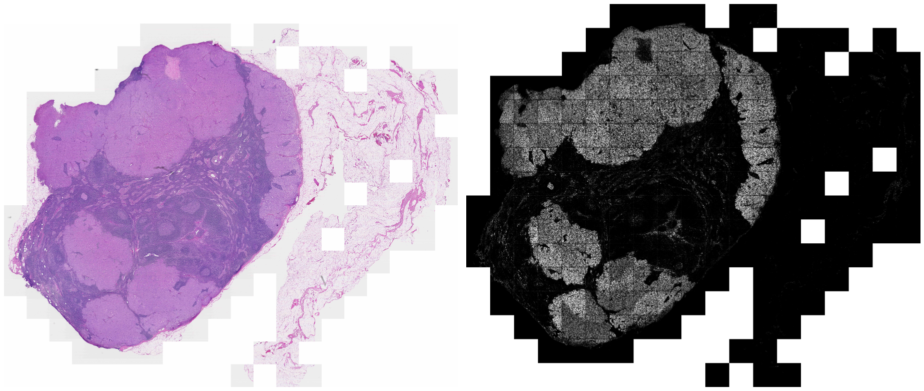

Figure 12: Left: Original Slide with no perturbation. Right: SC model predictions on original slide.

Figures 11 & 12 appear to match, which makes sense, considering that the tiles used to create them appear to match.

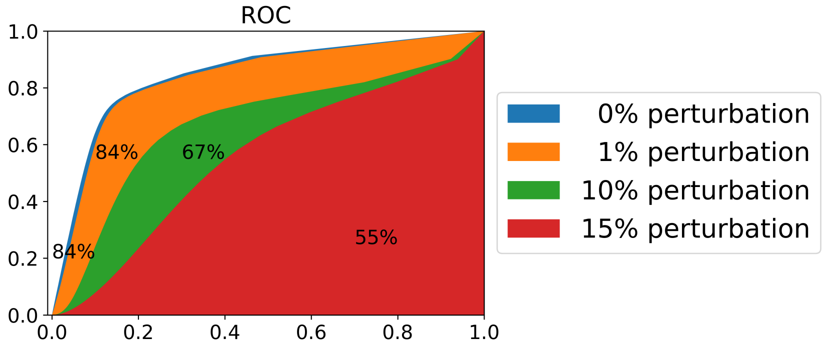

Figure 13: Blue: AUC ROC score of 0.84. Red: AUC ROC score of 0.55.

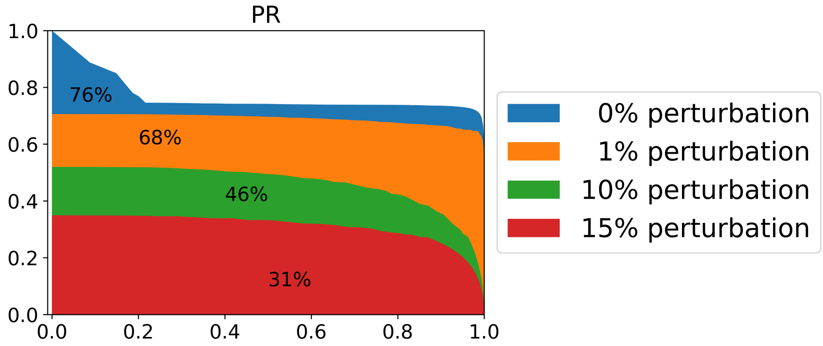

Figure 14: Blue: AUC PR score of 0.76. Red: AUC ROC score of 0.31.

The SC model was effected by the adversarial examples generated for the DCN model. It was not effected quite as adversely, though was effected a noticeable amount by the 15% perturbation in Figure 10, where its confidence levels were severely weakened.

4 Discussion

In a previous study, we found that small transferable adversarial perturbations do not transfer well to sparse coding [4]. This conclusion was validated by the fact that the SC model was unaltered by the 1% perturbation, while the DCN model was nearly inverted. Here, we also see that amplifying the adversarial perturbation wound up decreasing metrics for the sparse model as well. One reason for this may be that the adversarial example also contained meaningful features which, when amplified, actually turn images of tumors into non-tumors.

The adversarial examples we viewed did add meaningful changes to the tiles, such as changing the color from pink to purple, or even adding cell-like shapes. Below is an example of meaningful and meaningless adversarial perturbations.

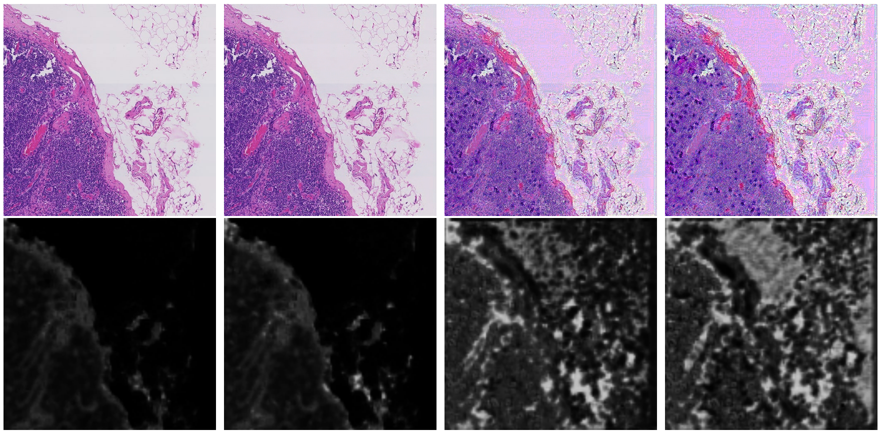

Figure 15: First Column: original tile. Second Column: 1% perturbation. Third Column: 10% perturbation. Fourth Column: 15% perturbation. Each corresponding prediction tile was inferred using the deep model. Notice the appearance of what could be described as nuclei in the third and fourth columns. This could be an explainable feature to add, which may increase the confidence in a cell being tumorous. At the same time, the white space in the upper center area of the tile gains a bar-code like pattern, and the resulting predictions indicate tumor. This is an example of a non-meaningful perturbation.

Compared to 1% perturbation, 15% perturbation caused the SC model to lose confidence. We believe that the adversarial examples may have used meaningful features, which tricked the SC model at this percentage of added change. Even so, the SC model proved to be more robust than the DCN model on meaningless changes. A next step would be to generate adversarial examples to the SC model, by teaching a Generative Adversarial Neural Network to construct fake sparse codes, and use the sparse decoder to turn these into tiles.

5 Conclusion

We created a DNN that achieves state of the art accuracy, and a sparse coding neural network with almost equal performance. By attacking the DNN classifier we showed that it is, like almost all DNN classifiers, susceptible to adversarial examples. We discovered that the very same meaningless adversarial examples that disrupt the DCN model do not alter the SC model. However, with higher perturbations, meaningful features arose in the tiles, which altered predictions in both models.

6 Personal Statement

I, Charles Strauss have written over 4,000 lines of python code for this project, which I started in June of 2019. Dr. Garrett T. Kenyon, my mentor, brought me onto this project after having already built the Sparse Coding Autoencoder with another team. I wrote all of the deep autoencoders, and all classifiers used in this project myself. I also wrote the code that did the analysis, with guidance on what algorithms to implement and how they work. Recently, I have been given even bigger supercomputers to run my code on. I am planning on scaling up my problem by using larger images. Previously, I used tiles, however now, I will input entire slides.

References

[1] Will Fischer, Sanketh S. Moudgalya, Judith D. Cohn, Nga T. T. Nguyen, and Garrett T. Kenyon. Sparse coding of pathology slides compared to transfer learning with deep neural networks. *BMC Bioinformatics*, 19(18):489, 2018.

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2014.

[3] Geert Litjens, Peter Bandi, Babak Ehteshami Bejnordi, Oscar Geessink, Maschenka Balkenhol, Peter Bult, Altuna Halilovic, Meyke Hermsen, Rob van de Loo, Rob Vogels, Quirine F Manson, Nikolas Stathonikos, Alexi Baidoshvili, Paul van Diest, Carla Wauters, Marcory van Dijk, and Jeroen van der Laak. 1399 H
amp;E-stained sentinel lymph node sections of breast cancer patients: the CAMELYON dataset. *GigaScience*, 7(6), 05 2018. giy065.

[4] Jacob M. Springer, Charles S. Strauss, Austin M. Thresher, Edward Kim, and Garrett T. Kenyon. Classifiers based on deep sparse coding architectures are robust to deep learning transferable examples. *CoRR*, abs/1811.07211, 2018.

[5] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H. Beck. Deep learning for identifying metastatic breast cancer. 2016.