LA-UR-21-21957
Approved for public release; distribution is unlimited.

Title: Transfer Learning using Denoising Auto-Encoders for Cellular-Level Annotation of Tumor in Pathology Slides

Author(s): Strauss, Charles Shelby Murton
Kenyon, Garrett

Intended for: Web

Issued: 2021-02-26

Transfer Learning using Denoising Auto-Encoders for Cellular-Level Annotation of Tumor in Pathology Slides

Charles M. S. Strauss[1,2] and Garrett T. Kenyon[1,2]

[1] *Los Alamos National Laboratory, Los Alamos, NM*
[2] *New Mexico Consortium, Los Alamos, NM*

Abstract

Adversarial examples can produce altered classifications using only seemingly innocuous, imperceptible perturbations to the original image. The imperceptibility of adversarial perturbations suggests that the corresponding classifiers use decision criteria different than those of a human. In a medical setting, inexplicable decision criteria confound a pathologist's willingness to trust machine-generated annotations. Here, we analyze denoising tumor detection models to see if they are robust to imperceptible adversarial perturbations. Moreover, to be more fully trusted by pathologists, we require tumor detectors that generate interpretable annotations which segment pathology slides into tumorous and normal regions at the cellular level. We therefore compare transfer learning based on two different autoencoder architectures, one derived from a deep denoising bottleneck autoencoder and one from an over-complete sparse autoencoder. Both autoencoders were first trained in an unsupervised manner on a set of pathology slides drawn from the Camelyon16 dataset. The latent representations produced by each autoencoder were then passed to separate neural networks that were trained in a supervised manner on binary tumor-normal masks generated by pathologists at cellular resolution. Both tumor detectors supported better than 90% AUC PR as measured by the area under the precision/recall curve on a held-out pathology slide. To assess the underlying decision criteria used by both tumor detectors, we constructed imperceptible adversarial examples which reduced the AUC PR of both models to less than 70%. Random noise of the same amplitude had almost no effect on the AUC PR of either model. Additionally, each tumor detector was resistant to adversarial "transfer" attacks targeting the other. The adversarial perturbations showed strong characteristic differences: the deep denoising models perturbations were a very diffuse, seemingly unrecognizable pattern while the sparse coding models perturbations showed traces of tissue cells.

1 Introduction

Tumor discovery is difficult, requiring highly trained pathologists to devote hours of time for a single pathology slide. Moreover, human pathologists can, and do, make mistakes(Litjens et al., 2018; Liu et al., 2017; CAM). Deep learning algorithms can be trained to detect the presence of tumors with near-human accuracy and could potentially be used to reduce time and mistakes(Liu et al., 2017; Wang et al., 2016; Khosravi et al., 2018; Xu et al., 2017). Unfortunately, deep learning algorithms in general are often susceptible to imperceptible perturbations added to an input image which alters the resulting classification, known as adversarial examples(Szegedy et al., 2014; Goodfellow et al., 2014). The effectiveness of imperceptible perturbations implies that deep learning algorithms trained to detect the presence of tumors are using unphysiological features in their decision criteria, and thus would be hard for a pathologist to trust. In addition, deep learning algorithms are typically optimized for labeling either whole slides, or large regions thereof, with a binary label, either tumor or no-tumor, whereas pathologists inspect tissue at the level of single cells. In an effort to construct classifiers that are robust against adversarial perturbations and which yield local detections at the cellular level, we tested transfer learning protocols employing two autoencoders known to remove random noise perturbations (denoising): a deep denoising (deep learning based) bottleneck autoencoder and an over-complete sparse autoencoder.

Classifiers based on latent representations inferred for optimal sparse coding have been previously shown to be robust to transferable adversarial examples targeting deep learning models designed for whole image classification(Springer et al., 2018). Here, we seek to extend these results to classifiers optimized for local detections and to compare alternative transfer learning approaches based on distinct autoencoders. Using the latent representations learned in an unsupervised manner by the two autoencoders, we used pathologist-provided annotations to train two fully-supervised deep neural networks to generate cellular-level heatmaps indicating tumor probability. Cell-level predictions are important to the pathologist for human interpretability, providing an understanding of where tumors are predicted down to the level of single cells, as opposed to classifiers that only detect the presence of tumors somewhere in the slide. We crafted adversarial examples to both tumor detection neural networks, using a novel method for adversarially attacking the sparse latent representation directly. Classification results are quantified using the area under the precision-recall curve (AUC PR).

Related Work Deep learning has been previously applied to pathology slides and tumor detection. Khosravi et al. (2018) explored many deep learning architectures(Google's Inceptions, ResNet, and two Inceptions), along with a breadth of training strategies (fine-tuning, transfer learning, training from scratch) to the detection of cancer, cancer subtypes, and biomarkers in lung cancer, bladder cancer, and breast cancer. Furthermore, the Camelyon16 dataset (CAM), which we use here, has been routinely used for training successful tumor detector neural networks on various architectures(Wang et al., 2016; Liu

et al., 2017; Litjens et al., 2018). One of these many successful approaches, by Liu et al. (2017), conducted transfer learning by pretraining an Inception V3 architecture on ImageNet before fine-tuning on pathology slides. They concluded that pretraining with ImageNet did not improve performance, likely because of the substantial differences between pathology slides and natural images. In agreement, Fischer et al. (2018) found that transfer learning from ImageNet was inferior to directly optimizing a set of kernels for convolutional sparse coding of pathology slides. Xu et al. (2017) applied transfer learning from ImageNet CNNs to brain and colon histopathology images. Visualization of the features in their last hidden layer exhibited pathologist-verified insights but no attempt was made to construct adversarial examples.

Here, we extend applications of deep learning in tumor detection by constructing adversarial examples to the fully-trained classifiers built upon latent representations inferred by two distinct autoencoders, both trained in a *de novo*, unsupervised manner on pathology slides, as opposed to using transfer learning from ImageNet.

This Paper We present the architecture of our tumor detectors in Sections 2.3, 2.4, and 2.5, introducing our tumor detector neural networks and their key differences in Section 2.2, and diagramming the neural network layers in Figure 1. Next, in Section 2.6, we explain how our dataset was processed from the Camelyon16 dataset, display a piece of a pathology slide in Figure 3, and describe the partitioning of data and how metrics displayed here were made in Section 2.7. In Section 2.8, we introduce our own adversarial attack against the sparse coding based tumor detector, and explain how attacks were carried out against the deep learning based tumor detector. Displaying our results in Section 3, we go over each autoencoder's features and how this may have impacted robustness to adversarial examples in Section 3.1. We then visualize the adversarial attacks to each model in Sections 3.3 and 3.4, and compare the attacks in Section 3.5. In Figure 9, we display adversarial examples next to the original pathology tile, and in Figure 10, we quantify the effects of the same adversarial attacks on a holdout slide. We discuss possible problems and differences in the tumor detectors in Section 4, and conclude our work in Section 5.

2 Methods

2.1 Hypothesis Regarding Robustness to Adversarial Examples

The novel questions we address here are: first, can we adversarially attack tumor detection models based on denoising autoencoders, and second, if so, are the resulting adversarial perturbations interpretable by the pathologist, and does the character of the perturbations depend on the type of autoencoder employed. We hypothesize that imperceptible adversarial attacks will reveal non-semantic decision criteria that do not align with those used by human pathologists. In

order to test this hypothesis, we first construct a deep denoising autoencoder and a sparse autoencoder, both trained in an unsupervised manner. Then, we use each autoencoder's latent representation as the input to a deep classifier neural network trained in a fully supervised manner, which returns cell-level predictions for an entire pathology tile. We obtain pathology tiles using the Camelyon16 dataset and evaluate both resulting tumor detector models at the cellular level. If our hypothesis is true, we will have demonstrated that both autoencoders use non-semantic decision criteria. Furthermore, the differences in adversarial attacks and their cross-model transferability will display how much their decision criteria overlap.

2.2 Model Architectures

Sparse coding has been shown to be more robust to transferable adversarial examples targeting deep learning based classifiers(Springer et al., 2018). Therefore, we tested both a deep learning based tumor detector and sparse coding based tumor detector. A key difference between these two models is in how their latent representations hold information. A sparse code is comprised of an over-complete latent space, which, because it's sparse, causes an information bottleneck. In contrast, the deep denoising based model attains an information bottleneck by being dimensionally under-complete in its latent layer. The latent representations are passed to corresponding deep neural networks that return tumor-prediction heatmaps. Sparse coding was done in Petavision(Lundquist et al., 2017), and all deep neural networks were built in Tensorflow(Abadi et al., 2015). A diagram depicting the architectures of the sparse coding based, and deep denoising based, tumor detector neural networks can be seen in Figure 1. Reconstructions of clean and noisy images from both autoencoders can be seen in Figure 2.

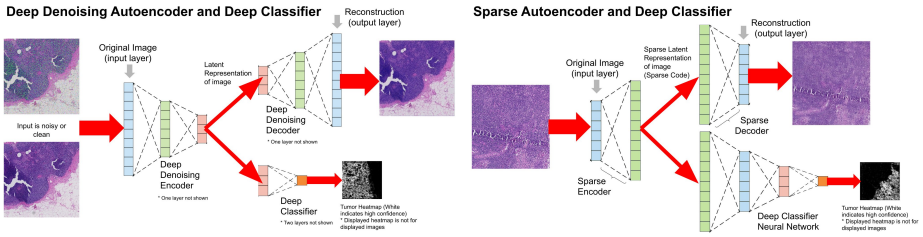

Figure 1: Neural network architectures. **Left:** deep denoising based tumor detector. **Right:** sparse coding based tumor detector. Each diagram shows the autoencoder, its latent space, and the deep neural network developed on the corresponding latent space.

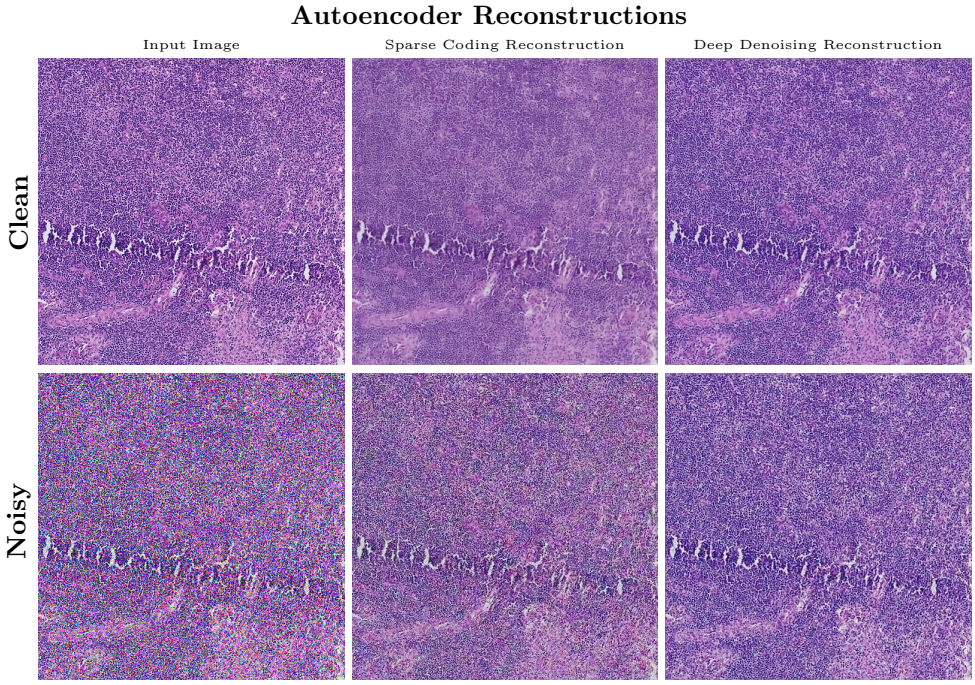

Figure 2: Example reconstructions. **Top (left to right):** clean image, sparse reconstruction, deep denoising reconstruction. **Bottom (left to right):** image with added noise, sparse reconstruction on noisy image, deep denoising reconstruction on noisy image. Details are well-preserved from image to reconstruction in both autoencoders.

2.3 The Deep Denoising Autoencoder

We trained our deep autoencoder to denoise images. The denoising task is expected to be a first line of defense to small perturbations. Additionally, random noise has been shown to regularize the features within a deep neural network (An, 1996; Matsuoka, 1992; Bishop). We accomplished this by explicitly adding noise from a normal distribution with a standard deviation of 0.1 to the input image, and by using the original "clean" image as the target for the reconstruction.

Previous approaches to tumor detection rescaled pathology tiles to the range [0, 1]. Here, we do the same, rescaling our images from the range [0, 255]. All $K \times K$ kernels mentioned here are in pixels. We settled on an autoencoder encoder portion consisting of a convolutional layer with 36 8×8 kernels followed by a max pooling layer with a 2×2 window and then a 10% dropout layer. Next, we applied a second convolutional layer with 18 4×4 kernels, another 10% dropout layer, and a final convolutional layer with 9 3×3 kernels. Each convolutional layer used a stride of 1, biases, a ReLu activation function, "same" padding, and had its weights initialized using a Glorot uniform. For the decoder

portion, we mirrored the structure but did not share weights or biases. We trained this autoencoder for 4 epochs on our pathology tile training dataset using Mean Squared Error loss; this produced the reconstructions seen in Figure 2, column 3. Our autoencoder latent representation was 25% under-complete. An input image of 512×512 three channel color-pixels was represented in our autoencoders latent representation by an array of the shape $256 \times 256 \times 9$ (9 latent channels).

2.4 The Sparse Coding Autoencoder

Sparse coding has been shown to be highly robust to not just random noise perturbations, but also to adversarial perturbations targeting deep learning based classifiers(Springer et al., 2018), thus why we employ it here. This robustness to specific adversarial perturbations is due to the process of sparse coding which drives the inputs towards an attractor basin. The same process causes sparse coding to be naturally denoising—denoising is only a byproduct of our sparse autoencoder, as it was never explicitly trained to denoise images by adding noise to the inputs. As a result of the sparsity constraint, the latent representation is 99% zeros, making a sparse code 99% under-complete (informationally, not dimensionally). Thus, it is no surprise that the resulting reconstructions are slightly blurrier than the deep denoising autoencoders reconstructions.

An in-depth explanation of sparse coding via a convolutional locally competitive algorithm (LCA) was done by Kim et al. (2017). However, the basic approach used here for using convolutional LCA to sparse code pathology slides follows Fischer et al. (2018). More specifically, the sparse autoencoder encoder portion we settled on used 512 24×24 kernels in its only convolutional layer, with a fixed bias (threshold) followed by a ReLu activation function. The same set of kernels were shared by the decoder portion.

We trained the sparse autoencoder for 1.4 epochs over tiles without tumorous tissue in the training set. This produced the reconstructions seen in Figure 2, column 2.

2.5 The Cell-level Classifier Neural Networks

Both autoencoders produced compressed versions of images in their latent representations. To take advantage of these compressed latent representations, and any features learned by the autoencoders, we applied transfer learning. We did this by training a supervised deep neural network on the autoencoder latent representation to produce tumor probability heatmaps for the given image. The deep classifier neural networks consisted of three convolutional layers with "same" padding, initialized with the Glorot uniform, built using Tensorflow Keras like the deep denoising autoencoder. The classifier consisted of 36 4×4 kernels in its first convolutional layer, which fed a 50% dropout layer, then 18 3×3 kernels in its second convolutional layer. Next, we applied an average pooling layer with a 2×2 window, and finally a singular 2×2 kernel in its last convolutional layer. Heatmaps are binary, so a sigmoid activation function was

chosen for the last layer. All other convolutional layers used a ReLu activation function and a stride of 1.

We trained each deep classifier neural network on the binary tumor-normal tissue masks provided by the Camelyon16 dataset for 10 epochs. Because the human-interpretability of machine-predictions increases by specifying the pixels which cause the classification, our tumor detectors produce probability heatmaps, not just binary yes/no tumor classifications. The heatmaps, being 16 times smaller than the original image in surface area, contain predictions for each 4×4 region (an area smaller than a human cell in a standard pathology slide). We call these predictions cell-level heatmaps.

2.6 Dataset

Pathology slides are gigapixel images consisting of tissue and empty background slide(Litjens et al., 2018). In order to train on images containing tissue and avoid training on empty regions, we preprocessed each slide into image-tiles containing tissue and excluded tiles that were mostly empty. We used pathologist-annotated, H&E stained pathology slides from the Camelyon16 challenge(CAM). This slide set consists of 399 slides, with 240 containing no tumorous cells and 159 containing at least some tumorous cells(Litjens et al., 2018). The slides were annotated by human pathologists though label noise existed within the dataset, such that entire slides were incorrectly labeled as being free of tumors(Liu et al., 2017), duplicate tissue slices were labeled once not twice, and micrometastases were left out of the labels altogether(Litjens et al., 2018). We gridded the whole slide images into tiles at the 3rd highest resolution level available in the dataset and saved out only tiles that contained enough purple-looking tissue (average hue matching $R - B < 32, R - G > 16, B - G > 16$, where R, G, and B are red, green and blue values for a pixel). By visually checking the resulting tiles, we confirmed that this sorted out most empty white tiles. Tiles were stored as PNG images with three 8-bit color channels. Corresponding cell-level binary tumor-normal tissue masks were made for each tile, by converting the pathologist vector tumor annotations to masks.

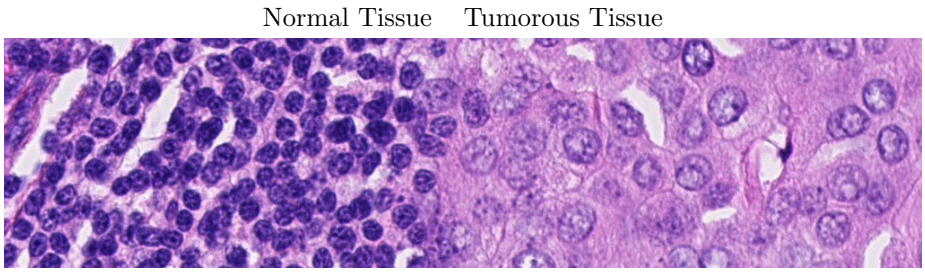

Figure 3: Zoomed in patch of a whole slide image from the Camelyon16 dataset(CAM). Dark purple (left) cells are normal tissue. Pinker (right) cells are tumorous.

2.7 Evaluation

The Camelyon16 dataset includes human pathologist-drawn cell-level annotations for all training data. To produce a holdout set, we partitioned slide tumor_110 (a slide without any major errors known to the authors) away from the main training set and used only this holdout data for results shown here. The task of tumor detection requires high sensitivity and specificity, and deals with a rare positive class—tumor cells. Others have used FROC scores to evaluate their models, though we believe that highlighting every tumorous cell, and not just pointing to a pixel in a tumorous lesion, is more useful and interpretable to the pathologist. We calculated precision-recall(PR) curves on the cell-level masks and heatmaps, and report this metric after different perturbation using the area under the curve (AUC scores) in Figure 10. For comparison, a null model achieved a cell-level AUC PR of 27% on the holdout set, while our models each attained a cell-level AUC PR score above 90% on original images on the holdout set. Thus both models had similar performance.

2.8 Generating Adversarial Examples

Adversarial examples are images with tiny, imperceptible changes that cause a classifier to miss-classify(Szegedy et al., 2014; Goodfellow et al., 2014). We crafted adversarial examples to target tumor detectors in two ways: by attacking the latent representation of the autoencoder, and by attacking the image layer directly.

2.8.1 Attacking the Sparse Latent Representation

Because decomposing an image into a sparse code requires an iterative solution to a nonlinear objective function, direct gradient descent on the image is not a viable strategy for producing adversarial examples. We therefore devised a novel method for adversarially attacking the sparse coding based tumor detector by attacking not the image itself but rather attacking the sparse latent representation of the image. Following the standard method for generating an adversarial example for a given input image, we used backprop to compute the gradient of the local detections generated by the deep neural network classifier with respect to the latent representation at each hidden layer, yielding a vector of adversarial perturbations associated with the sparse latent representation of that image. However, simply adding this perturbation to the sparse latent representation would have resulted in a latent representation that was no longer sparse. Therefore, we restricted perturbations to active, non-zero coefficients, ensuring that the resulting adversarial example remained closer to the natural image manifold. We repeated this process of modifying the non-zero support of the original sparse latent representation for 100 iterations and then used the attacked sparse latent representation to construct an adversarial example image. Using the original sparse reconstruction derived from the original image, we subtracted the reconstructed adversarial image to determine the adversarial

perturbation in image space. Although there is no guarantee that the same adversarial sparse latent representation results from sparse coding the adversarial image, our procedure nonetheless produced effective adversarial examples.

2.8.2 Attacking the Deep Denoising based Tumor Detector Image Layer

Unlike the sparse coding based tumor detector, the deep denoising based tumor detector is simple to backpropagate through and attack on its image layer. Therefore, we used the Fast Gradient Sign Method(FGSM) with a step size of 1/255 for 100 iterations.

2.8.3 Selecting a Perturbation Size

We measured adversarial perturbations as the L2 norm, which gives a measure of the distance an image must be moved away from the original image for misclassification to occur. Adversarially attacked images were all of the size $512 \times 512 \times 3$ and we set all perturbations shown here (unless otherwise stated) to have an L2 norm of 25, which generally appeared visually imperceptible although produced greatly altered heatmaps and AUC PR scores. For comparison, original images had an L2 norm of around 500.

3 Results

3.1 Autoencoder Feature Comparison

To better understand the decision criteria used by both models, we looked at their first-layer convolutional features. We visualize all the features in the first layer of both autoencoders in Figures 4 and 5. A cursory look over these Figures reveals many insights also visible in the images, and some which, we believe, do not fit in as well. A more in-depth analysis of the deep denoising autoencoders features (Figure 4) reveals many features which appear random, with not much noticeable structure (top two rows), and also some features which contain cell-like features, and gaussians (bottom two rows). Why the top two rows appear to be representing noise is unknown, however, the resolution of the kernels we display likely played a large role in the features used by this autoencoder. In comparison to the deep denoising autoencoders features, the sparse dictionary has higher resolution kernels and many more of them. The top two rows in the sparse dictionary visualization (Figure 5) correctly represent artifacts visible in the data, while the remaining rows contain gaussians and cell-like features. A problem with comparing these two feature sets lies within the resolution of the kernels. We believe the deep denoising autoencoder may have been more susceptible to non-semantic adversarial perturbations because of its smaller kernel size and the mentioned random-looking features.

Deep Denoising Autoencoder First Layer Features

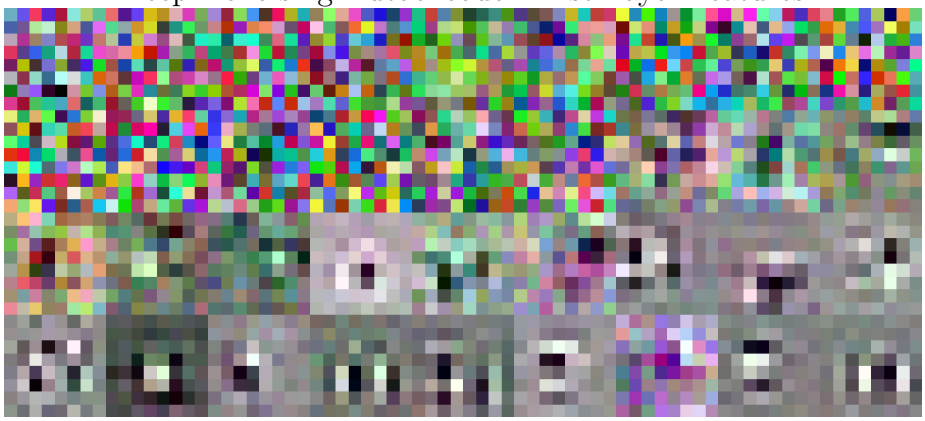

Figure 4: The features used by the first layer of the deep denoising autoencoder. This layer used 36 8×8 color kernels.

Sparse Dictionary

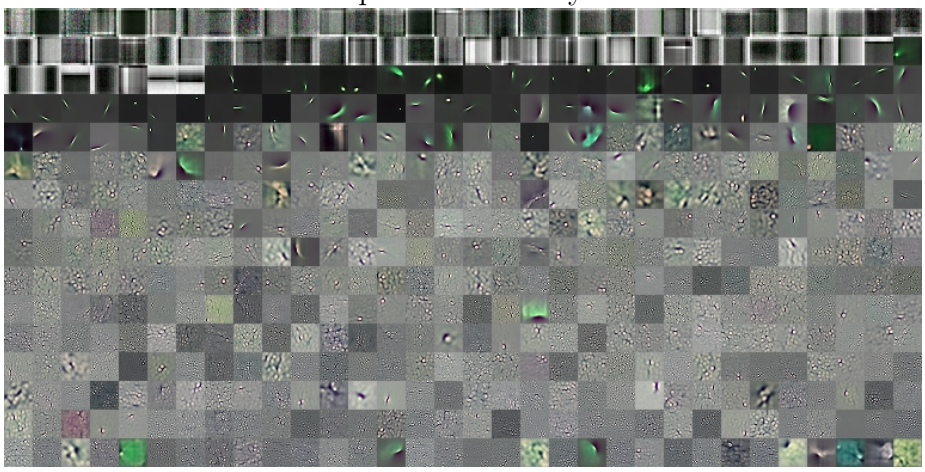

Figure 5: The the sparse dictionary. The single-layer sparse coding autoencoder used 512 24×24 color kernels.

3.2 Cell-level Heatmaps

Knowing which cells cause a tumorous classification for a slide is important to a pathologist, as it provides a more interpretable explanation for the diagnosis, and would allow them to check by reviewing the identified regions of interest. We displayed cell-level tumor predictions and the binary ground truth in Figure 6. These heatmaps are on original images, which we did not intentionally adversarially attack. Interestingly, in the example depicted by Figure 6, a small set of cells are identified as having a high probability of being tumorous by both

models in the upper left quadrant of the heatmaps. These same cells were not identified by the pathologist-drawn mask shown by the same figure. However, it is known that some tumorous regions were intentionally not annotated by the pathologists because they were too small (micrometastasis).

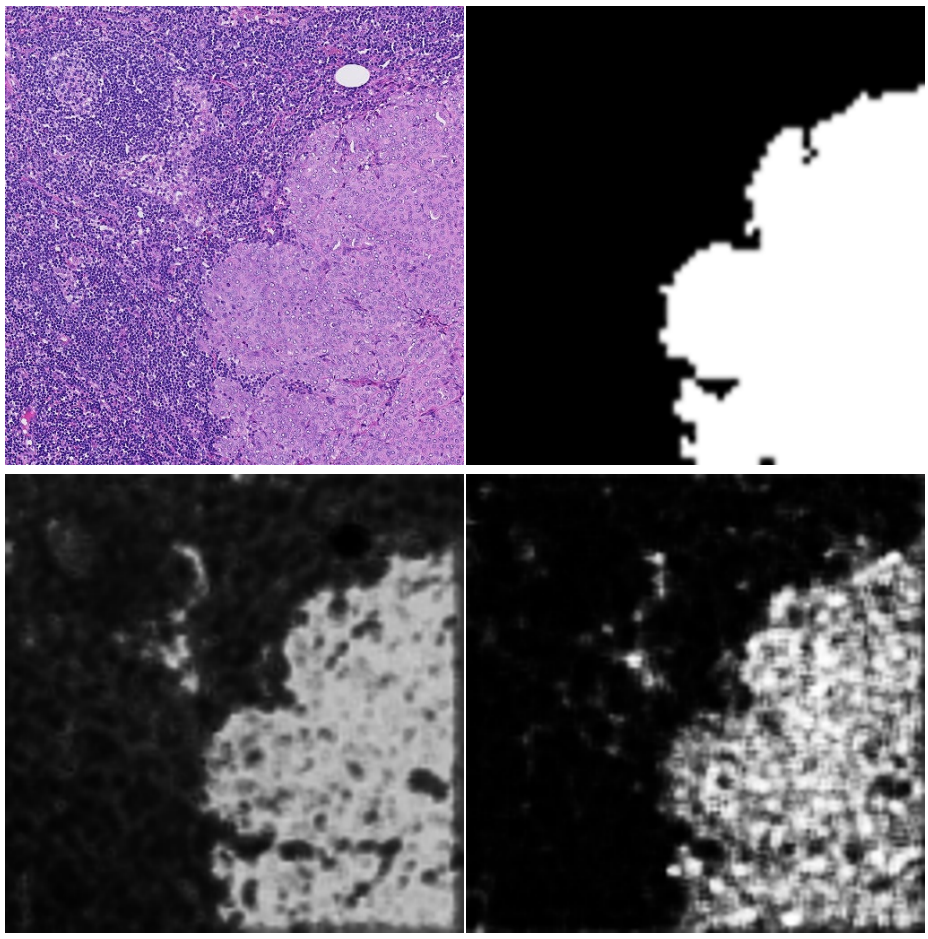

Figure 6: Cellular-level classification. **Row 1 (left to right):** original image and the binary pathologist annotations. **Row 2 (left to right):** the deep denoising based tumor detector's heatmap and the sparse coding based tumor detector's heatmap. Whiter areas indicate higher predicted probability of tumor. Note that not all tumorous cells were annotated by pathologists, so the white area in the upper left quadrant of the heatmaps may not be a false positive.

3.3 Applying FGSM to the Deep Denoising Based Tumor Detector

The adversarial perturbations targeting the deep denoising based tumor detector did not contain recognizable features, as visualized by Figure 7. These perturbations caused the AUC PR of the deep denoising tumor detector to drop to 69%, from 97%, although caused a 4% drop in the AUC PR of the sparse coding based tumor detector, as shown by Figure 10.

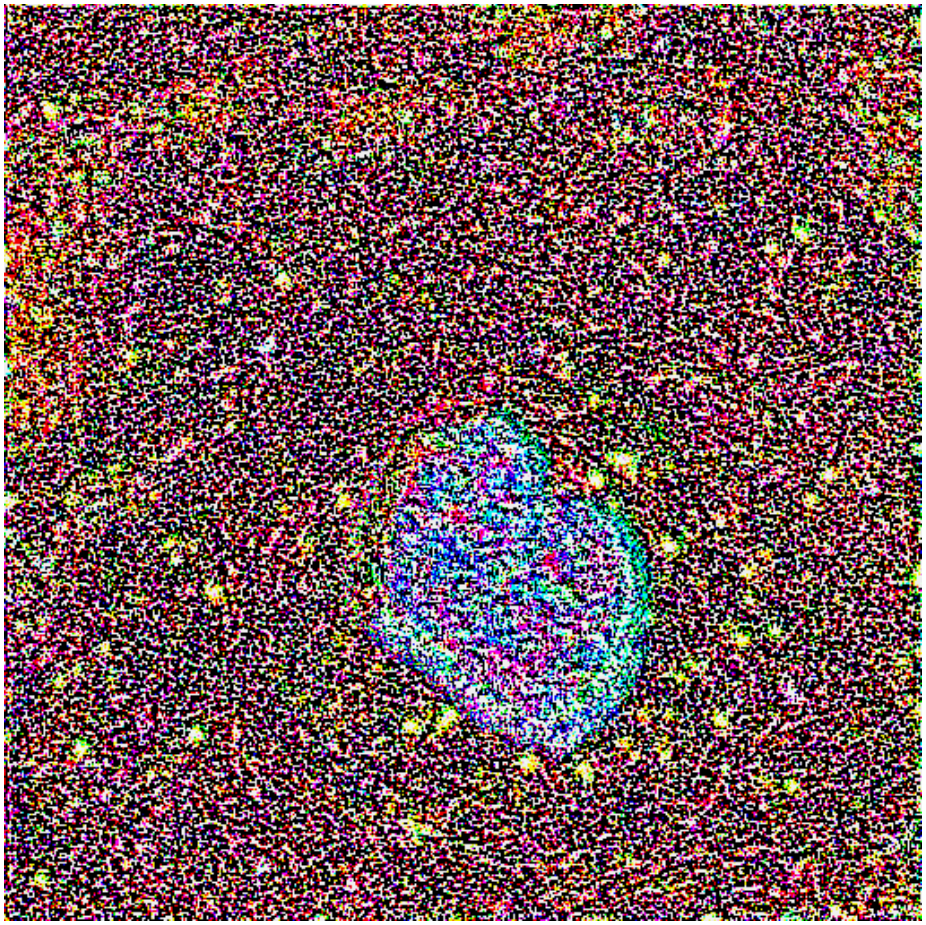

Figure 7: Amplified adversarial perturbation targeting the deep denoising based tumor detector. Generated via FGSM on the input image layer.

3.4 Adversarially Attacking the Sparse Coding Based Tumor Detector

The adversarial perturbations targeting the sparse coding based tumor detector did contain recognizable features, as visualized in Figure 8. These more meaningful perturbations caused the AUC PR of the sparse coding based model to drop to 65% from 94%, although caused a 6% drop in the AUC PR of the deep denoising based tumor detector, as shown by Figure 10.

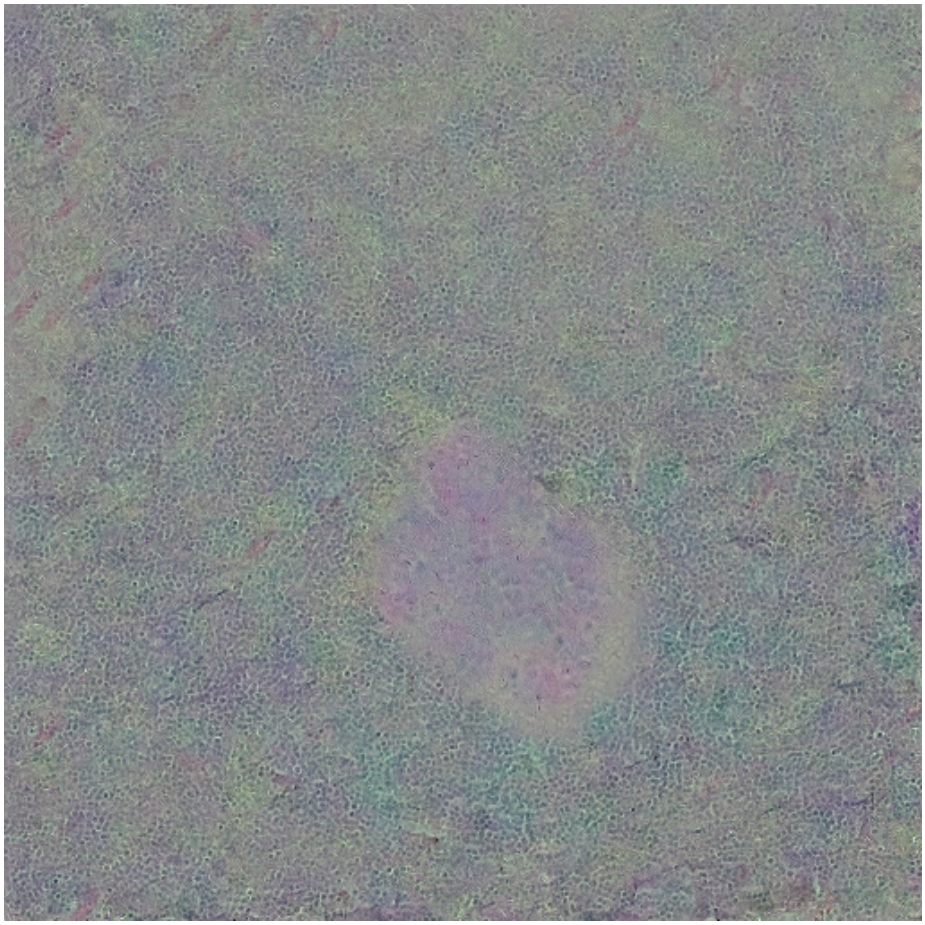

Figure 8: Amplified adversarial perturbation targeting the sparse coding based tumor detector. Generated via attacking sparse latent representation, like explained in 2.8.

3.5 Attack Comparison

The adversarial perturbations we tested were nearly invisible to the human eye. We argue that a human pathologist would not have altered their predictions, given the almost imperceptible nature of the perturbations. Figures 7 and 8 display amplified versions of these perturbations, while Figure 9 shows the original images next to the adversarial examples. Even though no noticeable change is visible in the right two images of Figure 9, the AUC PR scores on the holdout slide are negatively effected by the adversarial attacks targeting each model. The AUC PR scores go from greater than 90% to less than 70% when a targeted attack is applied to a tumor detector as shown by Figure 10. Interestingly, each model is robust to the opposite model's adversarial perturbations, and to random noise of the same amplitude as the adversarial perturbations. Robustness to random noise perturbations but not targeted adversarial perturbations shows that our perturbations were significant.

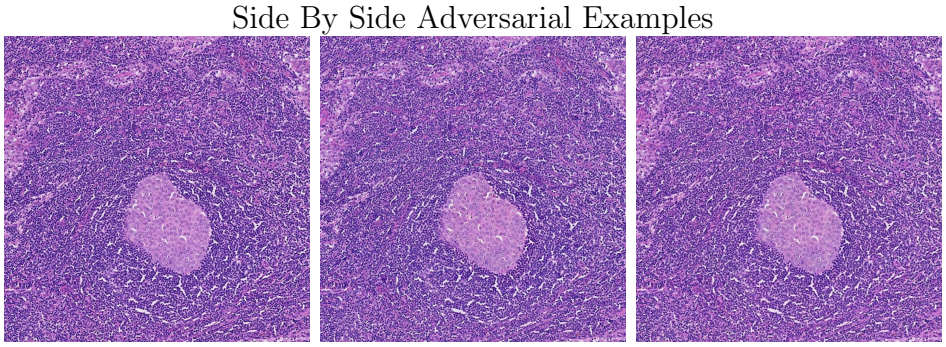

Figure 9: Side by side adversarial examples. **Left:** original image. **Center:** adversarial example targeting the deep denoising based tumor detector generated using FGSM (uses perturbation visualized by Figure 7). **Right:** adversarial example targeting the sparse coding based tumor detector generated by attacking the sparse latent representation (uses perturbation visualized by Figure 8).

4 Discussion

4.1 Attacking a Sparse Code

A standard adversarial attack, such as FGSM, is difficult to apply to a sparse autoencoder. A simpler approach, which we demonstrated above, is to train a traditional neural network on the sparse latent representations, attack these sparse latent representations through the traditional neural network, and reconstruct the attack from the perturbed sparse latent representation. Applying adversarial attacks to hidden layers within a deep neural network is not a novel

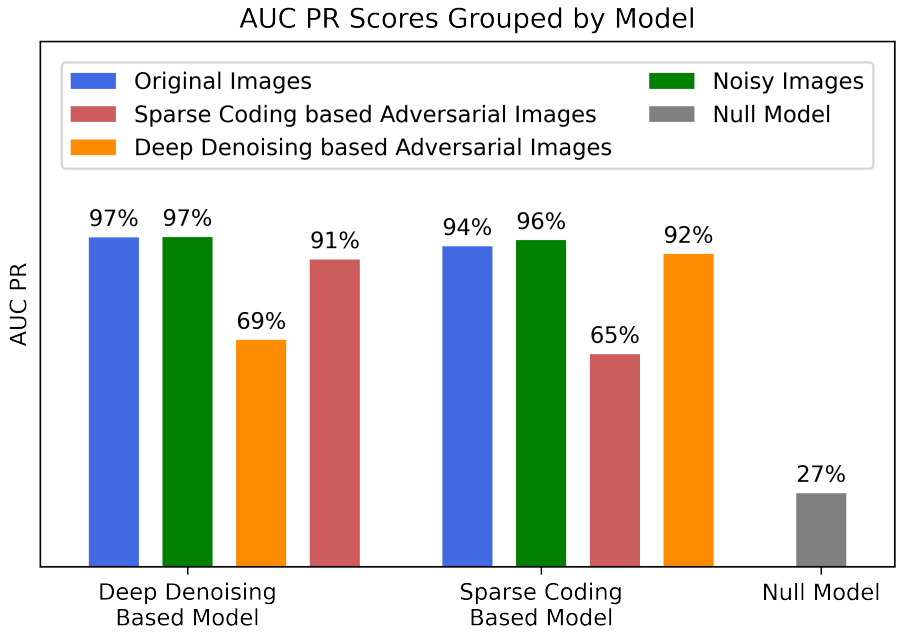

Figure 10: AUC PR scores (area under the precision-recall curve) for both tumor detectors, and a null model for comparison, labelled along the x-axis. Blue bars represent each models AUC PR score before perturbations are added. Green bars represent each models AUC PR after noise perturbations were added. Orange bars represent AUC PR after the adversarial perturbations targeting the deep denoising based tumor detector were added. Red bars represent the AUC PR after the adversarial perturbations targeting the sparse coding based tumor detector were added. The grey bar is the AUC PR of a null model tested on the same holdout set.

idea. However, attacking a sparse code in this manner may be. Furthermore, because of the sparse code separation from the input layer, there is no guarantee that the optimal, adversarial sparse latent representation will be reached again from the reconstructed image. The attacked reconstruction is the optimally bad change for that reconstruction, not necessarily for the original image. Using the mentioned method, we showed that this is a viable method of attack, even though it is indirect.

4.2 Comparing Adversarial Perturbations

The two adversarial attacks tested here create perturbations that effectively exist in two separate domains. This makes them difficult to compare beyond their effects on AUC PR. Because the FGSM—Fast Gradient *Sign* Method—

takes the *sign* of some change, it is expected that the resulting perturbation would only have three values (-1, 0, 1). This resulted in the strange, featureless changes seen in Figure 7. Whereas, our method for attacking a sparse coding layer returns a more image-like perturbation because this perturbation is the residual between an adversarial reconstruction and an original reconstruction. Our method resulted in the imperceptible, yet meaningful feature changes seen in Figures 8 and 9. Furthermore, because the adversarial sparse latent representation is kept sparse, the resulting perturbation is likely forced to stay closer to the natural image manifold than those produced through the FGSM. Even so, they still appear imperceptible in Figure 9, and do not successfully transfer between tumor detectors in Figure 10. This robustness to the opposite tumor detector neural networks adversarial examples demonstrates that each uses a differing set of decision criteria, that the perturbation we tested did not transfer between models well. Because features that fooled the sparse coding based tumor detector appeared more semantic in human terms, we say that this model is more robust. We did not show imperceptible non-semantic adversarial perturbations to the sparse coding based tumor detector.

5 Conclusion

We hypothesized that tumor detection neural networks would use decision criteria that do not align with human pathologists. To test this hypothesis, we created two tumor detectors capable of detection on the cell-level and found that they were susceptible to imperceptible adversarial attacks. Through a novel approach, we found and verified adversarial examples to our sparse coding based tumor detector. We quantified how the adversarial attacks appeared in human terms and saw how they affected the AUC PR of each tumor detector. Furthermore, we saw that each tumor detector was less susceptible to adversarial attacks targeting the opposite model. We conclude that imperceptible adversarial examples exist to both tumor detectors studied, and that sparse coding is less susceptible to inexplicable changes than traditional deep learning, though imperceptible adversarial perturbations exist to both models studied.

References

Camelyon 2016. https://camelyon6.grand-challenge.org/. Accessed: 2019.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan

Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

G. An. The effects of adding noise during backpropagation training on a generalization performance. *Neural Computation*, 8(3):643–674, 1996. doi: 10.1162/neco.1996.8.3.643.

Christopher M. Bishop. Training with noise is equivalent to tikhonov regularization. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/bishop-tikhonov-nc-95.pdf.

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. ISSN 1471-2105. doi: 10.1186/s12859-018-2504-8. URL https://doi.org/10.1186/s12859-018-2504-8.

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

Pegah Khosravi, Ehsan Kazemi, Marcin Imielinski, Olivier Elemento, and Iman Hajirasouliha. Deep convolutional neural networks enable discrimination of heterogeneous digital pathology images. *EBioMedicine*, 27:317–328, January 2018. doi: 10.1016/j.ebiom.2017.12.026. URL https://doi.org/10.1016/j.ebiom.2017.12.026.

Edward Kim, Darryl Hannan, and Garrett T. Kenyon. Deep sparse coding for invariant multimodal halle berry neurons. *CoRR*, abs/1711.07998, 2017. URL http://arxiv.org/abs/1711.07998.

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&E-stained sentinel lymph node sections of breast cancer patients: the CAMELYON dataset. *GigaScience*, 7(6), 05 2018. ISSN 2047-217X. doi: 10.1093/gigascience/giy065. URL https://doi.org/10.1093/gigascience/giy065. giy065.

Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E. Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q. Nelson, Greg S. Corrado, Jason D. Hipp, Lily Peng, and Martin C. Stumpe. Detecting cancer metastases on gigapixel pathology images. 2017.

Sheng Lundquist, Garrett Kenyon, Boram Yoon, and Scot Halverson. Petavision version 2.0. [Computer Software] https://doi.org/10.11578/dc.20180315.3, mar 2017. URL https://doi.org/10.11578/dc.20180315.3.

K. Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):436–440, 1992. doi: 10.1109/21.155944.

Jacob M. Springer, Charles M. 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. URL `http://arxiv.org/abs/1811.07211`.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

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

Yan Xu, Zhipeng Jia, Liang-Bo Wang, Yuqing Ai, Fang Zhang, Maode Lai, and Eric I-Chao Chang. Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. *BMC Bioinformatics*, 18(1), May 2017. doi: 10.1186/s12859-017-1685-x. URL `https://doi.org/10.1186/s12859-017-1685-x`.

Acknowledgements

Research presented in this paper was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20210043DR, by NIH project number 5R01EB025703-02, and by the generous support of the New Mexico Consortium.

We thank Jacob M. Springer, for critical reading and suggestions, and Austin M. Thresher for dataset preparation.