Color-Optimized One-Pixel Attack Against Digital Pathology Images

Joni Korpihalkola, Tuomo Sipola, Tero Kokkonen
JAMK University of Applied Sciences
Jyväskylä, Finland
{joni.korpihalkola, tuomo.sipola, tero.kokkonen}@jamk.fi

***Abstract*—Modern artificial intelligence based medical imaging tools are vulnerable to model fooling attacks. Automated medical imaging methods are used for supporting the decision making by classifying samples as regular or as having characters of abnormality. One use of such technology is the analysis of whole-slide image tissue samples. Consequently, attacks against artificial intelligence based medical imaging methods may diminish the credibility of modern diagnosis methods and, at worst, may lead to misdiagnosis with improper treatment. This study demonstrates an advanced color-optimized one-pixel attack against medical imaging. A state-of-the-art one-pixel modification is constructed with minimal effect on the pixel's color value. This multi-objective approach mitigates the unnatural coloring of raw none-pixel attacks. Accordingly, it is infeasible or at least cumbersome for a human to see the modification in the image under analysis. This color-optimized one-pixel attack poses an advanced cyber threat against modern medical imaging and shows the importance of data integrity with image analysis.**

I. Introduction

Correct functioning of health care technology is essential for modern societies. As stated in the EU's Cybersecurity Strategy for the Digital Decade [1], cross-sector interdependencies are very strong in the health care domain, and the health care domain is heavily reliant on interconnected networks and information systems. In that sense, cybersecurity has an important role in the digitalized health care domain. Our dependence on it has made it a lucrative target for malicious actors. The International Criminal Police Organization (INTERPOL) has made the remarkable observation that during the ongoing COVID-19 pandemic, cyberattacks are re-targeted against the critical health infrastructure [2].

The global digital transformation with the development of Machine Learning (ML) and Deep Learning (DL) based solutions has provided a possibility to automatically process pathological samples. Nam et al. define Computer-Aided Pathology (CAP) as *"computational diagnosis system or a set of methodologies that utilizes computers or software to interpret pathologic images"* [3]. Digital pathology can be defined as analyzing digitalized whole-slide images of tissue samples using Artificial Intelligence (AI) [4]. Solutions based on ML and DL (subsets of AI) are progressively utilized for decision making and prediction in medical imaging [5]. Among other things, enablers for such utilization are: (i) there exists an abundant amount of medical data available globally for research and development activities [6], and (ii) medical data is reasonable formed and labeled [7].

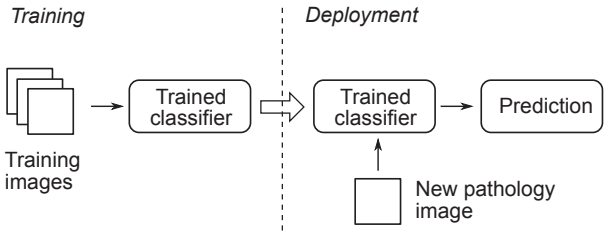

Fig. 1. Basic idea of normal use of AI-based digital pathology

A typical digital pathology task includes classification of whole-slide images as being either healthy or containing signs of cancer [4]. This fast pre-screening of the images increases efficiency and saves the doctor's and patient's time. The classifiers ordinarily use an ML solution to automatically learn and make predictions about the images. Typically, the input consists of digitized images, and the ML solution outputs a prediction in the form of confidence score. This score is sometimes equivalent to a probability of the image containing abnormalities. As with all machine learning, training a ML model is never perfect. Getting false positives is a real problem, and because of it, the results should not be trusted without expert interpretation. Fig. 1 illustrates the basic idea of an AI-based digital pathology solution. A classifier is trained using existing images, and then the classifier is deployed to make predictions about new images.

As can be seen, ML and DL solutions are widely used for diagnosis using digital imaging. This increases the threat of using digital imaging as a target surface for cyberattacks. If an attacker reaches access to the image data, the automated diagnosis can be fooled. Adversarial examples are images that cause unwanted behavior in the image classifier. Furthermore, these threats are real in medical imaging [8], [9]. Following categories of threats have been identified for model fooling against medical imaging: (i) adversarial images, (ii) adversarial patches, (iii) one-pixel attacks and (iv) training process tampering [10]. The first three are in the category of adversarial examples. They are specifically crafted images that deceive a classifier to make false predictions about input images. If such an attack does not need knowledge of the inside workings of the classifier, it is known as a black-box attack, because the only output needed is the prediction confidence score of the classifier [10], [11]. When an adversarial example is given

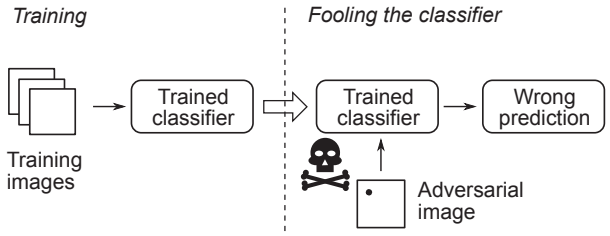

Fig. 2. Adversarial attack against a classifier

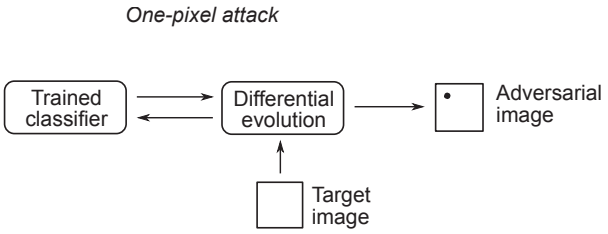

Fig. 3. Finding an adversarial image where only one pixel is changed

to a classifier analyzing digital pathology images, the subtle changes cause an image of healthy tissue to be classified as having signs of cancer. There have been successful attempts at fooling image classifiers with imperceptible adversarial examples. For example, Deng et al. used multi-objective optimization to attack against black-box classifiers [12]. Fig. 2 illustrates the idea of using malicious adversarial examples to fool the pre-trained classifier. The training set can be the same as in the ordinary scenario. The deployed classifier has not been changed in any way; it is simply receiving an image that has been altered somehow to produce wrong predictions. In practical terms, if an attacker can modify an image before it is input to an in-production classifier, the medical system would give an incorrect answer, thus endangering the health of the patient. Such a system could be related to whole slide image analysis, X-ray image analysis, or any other imaging modality where an automated classifier is used, especially in the case of a neural network.

One-pixel attack based on evolutionary optimization was introduced by Su et al. [13]. The basic idea is to find the coordinates and color values for a pixel that is placed on an image so that the classification result of the image is flipped. Evolutionary optimization is one way to find the best possible pixel that causes this effect. This heuristic black-box attack has been successful in deceiving image classifying neural networks [14]. One-pixel attacks can be studied using adversarial maps which record pixels vulnerable to this attack. Recent research has proposed that these areas correspond closely to the neural network's saliency maps [15]. Although there is an emerging understanding on how a single pixel modification influences the neural network [16], this lack of robustness leaves room for imaginative ways of exploiting them while doing image classification [17]. A related method to the single pixel modification is the use of sparse perturbations [18]. It should noted that pixel attacks are effective against well-known standard datasets and neural network architectures [19].

This research paper aims to present a way of creating more imperceptible one-pixel attacks. In our earlier study [20], we showed that the TUPAC16 dataset [21], [22] containing breast cancer tissue samples could be modified so that a single pixel could deceive a breast cancer classifier. However, in the original attack the color scheme of the attack pixel was modified so dramatically that the result was easily perceptible by a human observer. The bright yellow pixels were easy to spot; although, when looked at a distance, the pixel became less prominent. In this current study, the advanced color-optimized one-pixel attack against medical imaging is demonstrated. Within this state-of-the-art attack the color of a particular pixel is subtly changed in order to fool the automatic classifier, while still concealing the modification from human eyes.

II. Methods

A. Finding adversarial examples

We use evolutionary optimization to find adversarial examples where only one pixel is changed, following the example of Su et al. [13]. However, our optimization objective also contains a component that skews the results towards imperceptible pixel colors. The chosen optimization method is effective enough for the purpose of finding attack pixels, as the main bottleneck is the performance of the classifier. In other words, the main approach of this research is to find one-pixel perturbations $e(\mathbf{x})$ that cause an image $\mathbf{x}$ to be misclassified by classifier g while still being imperceptible to the human eye. Fig. 3 illustrates the idea behind the one-pixel attack against a trained classifier. The target image is used as a starting point on which a randomized set of one-pixel attacks is performed. Then this population of attacks is evaluated using the trained classifier. Based on the results, the differential evolution will iterate and find the most suitable one-pixel attack so that the confidence score produced by the classifier differs maximally from the confidence score of the plain target image. As a result, an adversarial image is created, which can be used for one-pixel attack to fool the classifier.

Differential evolution was chosen as the multi-objective optimization method to find attack vectors that best lower the confidence score of the neural network model while also optimizing the color values of the attack vector. The goal of the color value optimization is to blend the attacked pixel into the image as unnoticeably as possible. It is assumed that the pixel is more unnoticeable when its color is close to the mean value of the neighboring eight pixels. Below, we follow the definitions of Su et al. [13] and introduce a novel objective function resembling the one used by Su et al. [19].

B. Cost function

First, a basic mathematical framework for image classification and additive attacks is presented so that the objective function can be crafted. Input vectors $\mathbf{x} = \{x_1, \ldots, x_n\}$ describe the raw n pixels of a target image belonging to the class k. Let g be the classifier that discriminates between

classes. The output of $g_k(\mathbf{x})$ is the probability of input $\mathbf{x}$ belonging to the class k. A perturbation attack against $\mathbf{x}$ is represented by the attack vector $e(\mathbf{x}) = \{e_1, \ldots, e_n\}$, which contains the n pixels corresponding to the target image. This presentation would allow for a more general perturbation attack. However, in this research the one-pixel attack is in focus, so only one of the pixels in the vector differs from zero.

To create a more imperceptible alteration, we assume that the closer the color of the attack pixel is to the surrounding eight pixels, the more imperceptible it is. A perfect attack would have a color as close to their average as possible. We propose the following color scoring function. Here $h(\mathbf{x})$ is the color score, and it is calculated as the root-mean-square error (RMSE):

$$h(\mathbf{x}) = \sqrt{\frac{(c_r - c_{r\mu})^2 + (c_g - c_{g\mu})^2 + (c_b - c_{b\mu})^2}{3}},$$

where c_r, c_b, c_g are the color values of the attack vector scaled within the range $[0, 1]$ and $c_{r\mu}$, $c_{g\mu}$, $c_{b\mu}$ are the means of the attack vector's surrounding pixels' color values. The used root-mean-square error color scoring function was adequate to penalize large color differences. The L_2 norm and mean squared error (MSE) were also considered but the results did not change considerably.

Consequently, the multi-objective cost function f for the wanted class k can be defined as a combination of the neural network's confidence score and the color score:

$$f_k(\mathbf{x}) = w_{nn} g_k(\mathbf{x}) + w_c h(\mathbf{x}),$$

where the neural network's confidence score $g(\mathbf{x})$ is multiplied by the weight w_{nn} associated to the score, given the input $\mathbf{x}$, and $h(\mathbf{x})$ the corresponding color score, which is multiplied by the weight w_c. These weights can be used to make one objective more desirable than the other. Finally, the optimization problem can be defined as follows. The term $||e(\mathbf{x})||_0$ expresses the number of non-zero elements in the vector. Here the constraining number $d = 1$, so that we want to find only one pixel.

$$\begin{aligned} \underset{e(\mathbf{x})^*}{\text{minimize}} \quad & f_k(\mathbf{x} + e(\mathbf{x})) \\ \text{subject to} \quad & ||e(\mathbf{x})||_0 \leq d \end{aligned}$$

In practical terms, the one-pixel attack vectors $e(\mathbf{x})$ are presented with 5-dimensional image modification vectors $\hat{e}$ that contain the x and y coordinates and the three RGB values c_r, c_g, c_b. Conversion from $\hat{e}$ to e can be thought as a one-pixel additive operator E, which creates a one-pixel attack mask image at the specified coordinates and with the corresponding color. Thus, the practical optimization problem is to find the optimal $\hat{e}(\mathbf{x})^*$:

$$\begin{aligned} \underset{\hat{e}(\mathbf{x})^*}{\text{minimize}} \quad & f_k(\mathbf{x} + E(\hat{e}(\mathbf{x}))) \\ \text{subject to} \quad & ||E(\hat{e}(\mathbf{x}))||_0 \leq d. \end{aligned}$$

C. Differential evolution

The differential evolution is initialized by creating the population $\mathbf{X}_0 = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \in \mathbb{Z}^{N \times D}$. Here N denotes the size of the population and D the dimension of a population vector. In our case, $D = 5$. The coordinate vectors (x, y) are randomly sampled from a discrete uniform distribution between the interval $[1, 62]$ due to image width and length belonging to interval $[0, 63]$. The surrounding pixels around these coordinates are extracted from the original image, and the mean c_μ of each color value is calculated based on the eight surrounding pixels. The corresponding RGB values for each coordinate vector are sampled from a discrete uniform distribution inside the interval $[c_\mu - 50, c_\mu + 50]$. This is carried out to help the color optimization by initializing the attack vector's color values to match the adjacent color values in the target image.

After the initialization, the differential evolution process evolves the vector population towards the best attack vector through two processes: (i) mutation and (ii) crossover. We use the SciPy implementation of the method [23] with some modifications to obtain information about evolution progress and to create a new population initialization method. During the differential evolution, new trial vectors $\mathbf{a}$ are created by mutating the currently best attack vector $\mathbf{a}_{\text{best}}$ (meaning the one that has achieved the highest cost function score) by the difference of two random vector components $\mathbf{a}_1, \mathbf{a}_2$ from the population:

$$\mathbf{a} = \mathbf{a}_{\text{best}} + m * (\mathbf{a}_1 - \mathbf{a}_2),$$

where m is the so-called mutation value that controls the number of mutations happening during the evolution. Random indices r_i (where $i \in \{1, \ldots, D\}$) are drawn from a continuous uniform distribution over the interval $[0, 1)$. The random numbers r_i are then compared to the crossover factor C, which determines if the new mutant component continues to the trial vector or whether the currently best component carries over. The fitness of the population and the new vectors is evaluated using the cost function f, and the best N candidates are accepted to the next population [24], [25].

Maximum iterations for the evolution are set to 20, and if the standard deviation of the cost function f values across the population is smaller than the mean of the cost function values

$$\mu = \frac{1}{N} \sum_{i=1}^{N} f_i$$

multiplied by the tolerance factor t, the evolution convergence check is carried out so that progress is stopped before reaching the iteration limit:

$$\sqrt{\frac{1}{N} \sum_{i=1}^{n} (f_i - \mu)^2} \leq t * |\mu|,$$

where f_i is the cost function value for a population member and t is the tolerance factor [25].

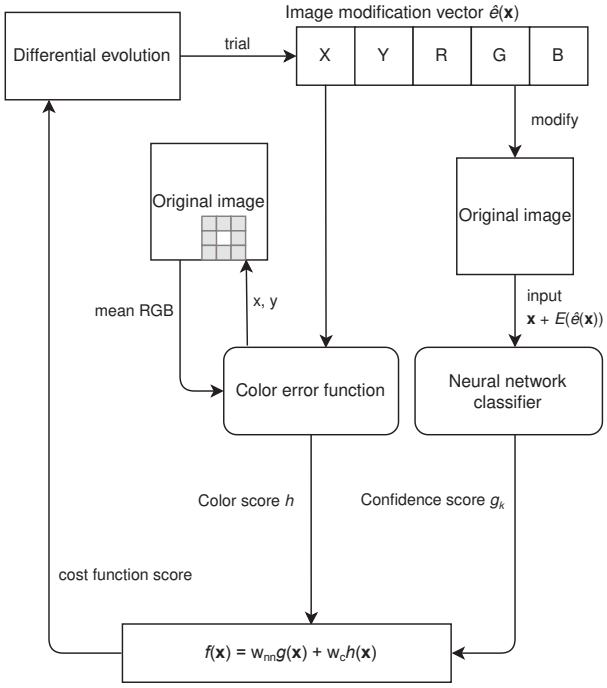

Fig. 4. Optimization process

Fig. 4 shows a schematic presentation of the optimization process presented above. The trial image modification vectors have five components as seen in the upper part of the figure. The two optimization objectives are measured, and finally the scores are combined as the result of the multi-objective cost function. This way the differential evolution process gets an evaluation of how successful the trial vectors are in the task.

III. Experiment setup

The goal of the experiment was to find one-pixel attacks against several pathology images. The attack was targeted against a classifier trying to identify images containing mitosis activity. An attack means that one target image is taken from the pool of preprocessed images, and the optimization procedure is used against that image. We conducted thousands of attacks in the experiment.

A. Dataset

The dataset used in the experiment was the TUPAC16, which contains pathology images of breast tissue [21], [22]. As the image format is very versatile with multiple zoom levels, preprocessing was performed to extract correctly sized images. The whole-slide images of the dataset were preprocessed into 64 by 64-pixel images in PNG format. Due to the number of images and the time-consuming evolution, subsets of the dataset are randomly chosen during each experiment. The dataset contains labeling for the images, either being normal, or containing signs of mitoses, which could indicate the presence of cancerous tissue.

B. Environments and tools

We have used the IBM MAX Breast Cancer Mitosis Detector and the deep-histopath framework it is based on as our attack target [26], [27]. Its main purpose is to classify pathology images, predicting the possibility of cancerous growth. The Breast Cancer Mitosis Detector takes images of 64×64 pixels as input, and returns a confidence score as an output via a REST API. The output represents the probability of the image containing mitosis. It was specifically built for the TUPAC16 dataset utilizing a modified ResNet-50 neural network model as the classifier. The network was trained using preprocessed images that were centered at labeled mitotic activity. The deep-histopath framework provides the preprocessing code and the code needed to train the classifier [27]. However, we did not train our own classifier but used the pre-trained model.

The Docker packaged classifier was wrapped so that it acted as a black-box classifier via Python code. We have chosen this classifier because it has been made publicly available, and it uses the same TUPAC16 dataset as its training material. This should mean that it will classify these images with high precision in normal circumstances. Our purpose is not to criticize this classifier in particular but to use one that has been developed independently from us.

The experiment was run on a High Performance Computing (HPC) server, which utilizes four Tesla V100 32GB GPUs, with the official driver version 450.80.02, and four 64-core Xeon Gold 6130 CPUs and 768 GBs of RAM for computing. The model was loaded using deep-histopath framework and MAX Breast Cancer Mitosis Detector model assets were loaded to the framework. The experiment script was run using Python version 3.7.2 programming language framework. The same Python framework was used to save the resulting attack vectors and adversarial images, and to produce the figures in this paper.

C. Two attacks

The goal of the research is to flip the classification of one class to the other by the one-pixel attack. The experiments are divided into two attack categories: (i) mitosis-to-normal and (ii) normal-to-mitosis. Firstly, the goal of the optimization during mitosis-to-normal experiment is to find an attack vector that reduces neural network's confidence the most while also keeping the color error low. Secondly, in the normal-to-mitosis experiment, the confidence score needs to be maximized, while the color error is again minimized. The confidence score is subtracted from 1, since it is the maximum value for the confidence score, and it allows to the optimization function to reduce the total score of the functions.

Multiple experiments were run, where combinations of different weights w_{nn} and w_c and color score functions were tried, along with other parameters such as mutation factor, crossover factor and population size in the differential evolution algorithm. The success rate of the attacks were statistically analyzed by measuring the confidence scores of the target images and the adversarial images.

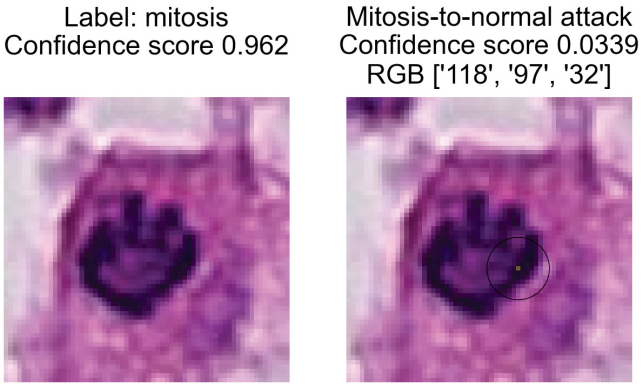

Fig. 5. Example of mitosis-to-normal attack. The original image on the left and the modified image on the right. Notice the circled brown pixel on the center-right.

IV. Results

A. Mitosis-to-normal attack

It was found during mitosis-to-normal attack experiments that the neural network confidence score was minimized the most and the modified pixel was less distinguishable when w_{nn} was set to 0.6 and w_c to 0.4. These weights were empirically selected as they produced the most visually concealed attack pixels. The more important objective of fooling the neural network is given a bigger weight, while still giving a quite large weight to the objective of finding the most unnoticeable color. The differential evolution hyperparameters' mutation value m was set to 0.5, crossover value C to 0.7 and the number of members in the population to 1,500. At the beginning, 1,771 images were randomly selected from the dataset, and the experiment ran for 36 hours.

Fig. 5 shows an example of a successful attack, where the confidence score was significantly lowered to the point that instead of labeling the image with mitosis activity, the neural network was manipulated to label the image as normal cell activity. In this example, the brown pixel is much closer to the characteristic red of the raw image.

The second component of the objective function aims to alter the color of the pixels. The colors of the pixels that are inserted into the original image during the attack vary widely, but the most successful attacks are brown colored pixels with red and green values around 100 and blue values near zero.

Fig. 6 shows the confidence scores before the attack and after the attack in boxplot visualization. The scores before the attack are close to the classification results of 1.0. It can be seen that the attack has a significant effect to some of the target images. The original scores of the target images are near the 1.0 level, which signifies that the image contains mitoses. The one-pixel attacks are able to change the label of the best 20% down below at least 0.67, some reaching almost 0.

After conducting the attack, the minimum confidence score achieved in the experiment was 0.02, and 10th percentile of

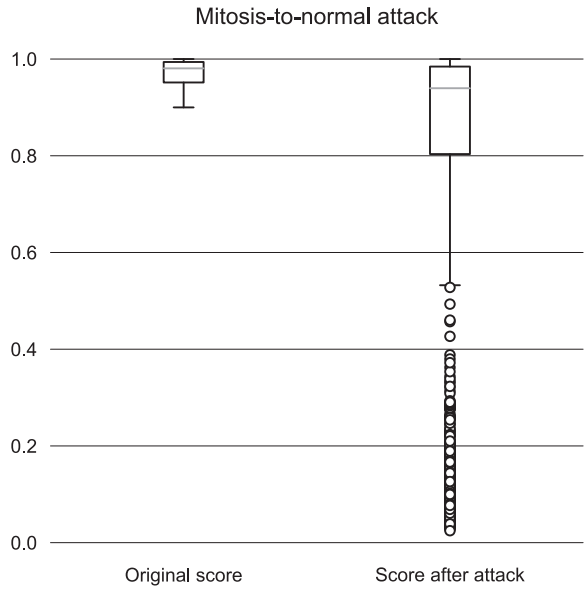

Fig. 6. Original confidence scores and confidence scores after the attack, with 25th percentile to 75th percentile inside the boxes, the whiskers indicating no more than the $1.5\times$ interquartile range and outliers drawn as circles.

TABLE I. Neural network confidence score statistics for mitosis-to-normal attack

	Before attack	After attack
Maximum	0.99	0.96
20th percentile	0.94	0.67
10th percentile	0.92	0.12
Minimum	0.90	0.02
Mean	0.97	0.78
Median	0.98	0.94
Standard deviation	0.03	0.32

the score is near 0.1. The majority of the attacks fail to achieve scores below 0.5, with 20th percentile of scores reaching 0.67. The exact statistical values of image classification before and after the attack are listed in the Table I.

B. Normal-to-mitosis attack

The normal-to-mitosis attack proved much more difficult; the impact to neural network confidence on target images was much lower than in the previous attack. The attack procedure is the same but here the target images have a confidence score near normal tissue and the goal is to make it higher with the one-pixel attack. The confidence score difference between attacked and original image is already low, so the weight w_{nn} associated with the neural network was set high at 0.7 and color weight set at 0.3. This means that the goal of optimizing the color might not be as successful. The differential evolution hyperparameters were set the same as in the previous attack. The attack experiment ran for 48 hours, and an attack was

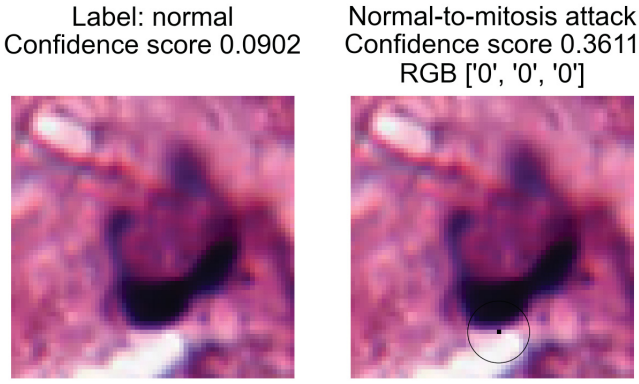

Fig. 7. Example of normal-to-mitosis attack. The original image on the left and the attacked image on the right. Notice the circled black pixel at the bottom-center.

TABLE II. NEURAL NETWORK CONFIDENCE SCORE STATISTICS FOR NORMAL-TO-MITOSIS ATTACK

	Before attack	After attack
Maximum	0.090	0.43
99th percentile	0.0074	0.038
95th percentile	0.00086	0.0063
Minimum	0.000001	0.000001
Mean	0.00048	0.0021
Median	0.00001	0.00004
Standard deviation	0.0042	0.016

performed on 2,849 randomly selected images from the pool of preprocessed whole-slide images.

Fig. 7 shows an attack where the confidence score was raised to the point that the neural network is no longer fully confident that the image could be labeled as normal, but the attack fails to twist the neural network's prediction to the opposite label. The color seems to be close to the environment, but the environment itself gives the pixel away. This, combined with the difficulty of flipping the classification, causes the attack to be less successful.

Again, the characteristics of the attack pixels proved to be interesting. The attacks mostly replaced a pixel in the image with a completely black pixel that is injected close to other dark colored pixels.

Fig. 8 shows the confidence scores before the attack and after the attack in boxplot visualization. The difference to the previous scenario is clearly visible. The initial scores are almost all near 0, so the boxes are barely visible. The attack scores do not achieve even the 0.5 level.

The maximum confidence score from an attack reaches 0.43, a vast majority of the attacks do not manage to impact the neural network's predictions; 99th percentile reaching 0.038. Other statistical values are listed in the Table II. Please note that here the attack direction is from low to high scores.

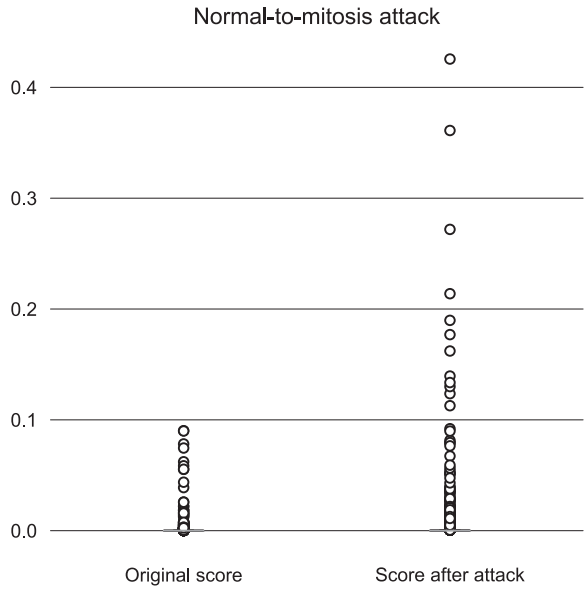

Fig. 8. Original confidence scores and confidence scores after the attack, with 25th percentile to 75th percentile inside the boxes, the whiskers indicating no more than the 1.5× interquartile range and outliers drawn as circles. Note that most of the data is close to 0, so only outliers are visible.

V. CONCLUSION

Medical imaging applications based on AI are extensively used in the medical imaging to recognize tumors. Although neural network based classification solutions are effective for detecting cancerous cell growth, there is an option to mislead the classification algorithms and cause false prediction results. The analysis methods related to medical imaging are not safe from model fooling attacks. Furthermore, different imaging modalities such as X-ray images are as feasible targets as the breast cancer tissue samples used in this research. One-pixel attack is a state-of-the-art example of the modern model fooling attacks. Automatic classification can be fooled by affecting nothing but one pixel of the image under analysis. This also raises concerns about the robustness of automatic analysis systems.

This study shows that a one-pixel attack against medical imaging can be modified to appear more imperceptible to a human observer, while the attack is still effective. This result can be achieved by changing the optimization cost function to take into account the pixel's color scheme in order to preserve the modification to the medical image unnoticeable to a human observer. As seen in the displayed examples and in the statistical analysis, the attacks can be successful in fooling the classifier.

The multi-objective cost function was useful because it turned the adversarial images and the attack pixels in them to resemble the natural coloring surrounding the pixel. It seems that it is possible to skew the one-pixel attack towards a state where the attack pixel appears more imperceptible. However, the attacks are not always as successful as when using only the

one-pixel objective. This is expected as the two objectives are conflicting, since the plain one-pixel attack seems to produce brighter pixels.

These adversarial images can appear quite natural to the eye, especially in the case of mitosis-to-normal attacks. Even in the normal-to-mitosis case the pixels near high-contrast edges can appear deceiving. If such a classifier is used for pre-screening, and a human tries to quickly assess the situation, the pixel might go unnoticed. At the least, the conflicting opinion of the system and the human could cause confusion and misuse of resources.

As the attack method relies on the non-robustness of the classifier, this methodology is limited by the vulnerability of the classifier. The methodology the classifier utilizes has an effect on how successful the one-pixel attack is. One could imagine that neural networks, as in this paper, are more vulnerable but, nevertheless, other types of classifiers could have robustness issues.

Our results show the variable nature of the classifier: not all attacks were successful. The mitosis-to-normal attacks produced some quite successful adversarial images. The normal-to-mitosis attack proved again to be more difficult. This could be caused by the generally lighter color scheme in those images. As the classifier is trained to identify areas containing dark mitoses, the normal images could have a more general and variable look, which is more difficult to make look like an actual mitosis situation.

The brown color of the attack pixels suggests that in order to fool the network, the pixel needs to be sufficiently bright. However, with the objective of fooling a human observer the pixel needs to blend into the red-brown surroundings. On the other hand, the black attack pixels might indicate that it is indeed the sharp color contrast impulse that fools the classifier. As for the human observer, dark pixels among other dark areas or edge areas could be difficult to recognize.

The recent real-life attacks have demonstrated the motive for attacking against the medical domain and the medical domain can be seen as valuable target. This study demonstrates a vulnerability of the artificial neural network technology. These results should not be seen as pessimistic against the usage of automated image analysis systems. Rather, these constraints should be understood more deeply when utilizing these systems in real-life scenarios. Moreover, an extreme concern should be focused on the requirement of data integrity because even small changes can produce severe changes in predictions.

As a next step, we propose further study of the objective function to find better ways to optimize the conflicting goals of accuracy and imperceptibility. Further developments in the optimization methods might reveal a faster way of finding the most unnoticeable attack. Another path forward is to study the mechanisms by which one-pixel attacks succeed, and if the form of the objective function affects this understanding. Different classifiers could be tested to evaluate whether the attack is successful in a wider context. Understanding both the attack and defense helps us to create more robust classifiers.

ACKNOWLEDGMENTS

This work was funded by the Regional Council of Central Finland/Council of Tampere Region and European Regional Development Fund as part of the Health Care Cyber Range (HCCR) project of JAMK University of Applied Sciences Institute of Information Technology.

The authors would like to thank Ms. Tuula Kotikoski for proofreading the manuscript.

REFERENCES

[1] European Commission, "The EU's Cybersecurity Strategy for the Digital Decade," https://ec.europa.eu/digital-single-market/en/news/eus-cybersecurity-strategy-digital-decade, Dec 2020.

[2] INTERPOL, The International Criminal Police Organization, "INTERPOL report shows alarming rate of cyberattacks during COVID-19," Aug 2020, accessed: 11 February 2021. [Online]. Available: https://www.interpol.int/en/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19

[3] S. Nam, Y. Chong, C. K. Jung, T.-Y. Kwak, J. Y. Lee, J. Park, M. J. Rho, and H. Go, "Introduction to digital pathology and computer-aided pathology," *The Korean Journal of Pathology*, 2020.

[4] F. Ghaznavi, A. Evans, A. Madabhushi, and M. Feldman, "Digital imaging in pathology: whole-slide imaging and beyond," *Annual Review of Pathology: Mechanisms of Disease*, vol. 8, pp. 331–359, 2013.

[5] J. Latif, C. Xiao, A. Imran, and S. Tu, "Medical imaging using machine learning and deep learning algorithms: A review," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1–5.

[6] S. M. Sasubilli, A. Kumar, and V. Dutt, "Machine learning implementation on medical domain to identify disease insights using tms," in *2020 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2020, pp. 1–4.

[7] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018.

[8] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019.

[9] S. G. Finlayson, H. W. Chung, I. S. Kohane, and A. L. Beam, "Adversarial attacks against medical deep learning systems," *arXiv preprint arXiv:1804.05296v3*, 2019.

[10] T. Sipola, S. Puuska, and T. Kokkonen, "Model Fooling Attacks Against Medical Imaging: A Short Survey," *Information & Security: An International Journal (ISIJ)*, vol. 46, pp. 215–224, 2020.

[11] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 151–178, 2020.

[12] Y. Deng, C. Zhang, and X. Wang, "A multi-objective examples generation approach to fool the deep neural networks in the black-box scenario," in *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2019, pp. 92–99.

[13] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[14] R. Paul, M. Schabath, R. Gillies, L. Hall, and D. Goldgof, "Mitigating adversarial attacks on medical image understanding systems," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 2020, pp. 1517–1521.

[15] W. Wang, J. Sun, and G. Wang, "Visualizing one pixel attack using adversarial maps," in *2020 Chinese Automation Congress (CAC)*. IEEE, 2020, pp. 924–929.

[16] D. V. Vargas and J. Su, "Understanding the one-pixel attack: Propagation maps and locality analysis," *arXiv preprint arXiv:1902.02947*, 2019.

[17] M. Afifi and M. S. Brown, "What else can fool deep learning? Addressing color constancy errors on deep neural network performance," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 243–252.

[18] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, "Sparsefool: a few pixels make a big difference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9087–9096.

[19] J. Su, D. V. Vargas, and K. Sakurai, "Attacking convolutional neural network using differential evolution," *IPSJ Transactions on Computer Vision and Applications*, vol. 11, no. 1, pp. 1–16, 2019.

[20] J. Korpihalkola, T. Sipola, S. Puuska, and T. Kokkonen, "One-pixel attack deceives automatic detection of breast cancer," *arXiv preprint arXiv:2012.00517*, 2020.

[21] Medical Image Analysis Group Eindhoven (IMAG/e), "Tumor proliferation assessment challenge 2016," http://tupac.tue-image.nl/node/3, 2016.

[22] M. Veta, Y. J. Heng, N. Stathonikos, B. E. Bejnordi, F. Beca, T. Wollmann, K. Rohr, M. A. Shah, D. Wang, M. Rousson, M. Hedlund, D. Tellez, F. Ciompi, E. Zerhouni, D. Lanyi, M. Viana, V. Kovalev, V. Liauchuk, H. A. Phoulady, T. Qaiser, S. Graham, N. Rajpoot, E. Sjöblom, J. Molin, K. Paeng, S. Hwang, S. Park, Z. Jia, E. I.-C. Chang, Y. Xu, A. H. Beck, P. J. van Diest, and J. P. Pluim, "Predicting breast tumor proliferation from whole-slide images: The TUPAC16 challenge," *Medical Image Analysis*, vol. 54, pp. 111–121, 2019.

[23] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[24] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2010.

[25] "SciPy v1.6.0 reference guide: Differential evolution documentation," https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html, 2020.

[26] "IBM code model asset exchange: Breast cancer mitosis detector," https://github.com/IBM/MAX-Breast-Cancer-Mitosis-Detector, 2019.

[27] M. Dusenberry and F. Hu, "Deep learning for breast cancer mitosis detection," 2018.