Histopathologic Cancer Detection

Team Name: C.O.D.E
URL: https://www.kaggle.com/c/histopathologic-cancer-detection/team

Rucha Waghulde
Artificial Intelligence for ECE
Stevens Institute of Technology
NJ, USA
rwaghuld@stevens.edu

Siddharth Mandgi
Computer Engineering
Stevens Institute of Technology
NJ, USA
smandgi@stevens.edu

Pranati Kaza
Electrical Engineering
Stevens Institute of Technology
NJ, USA
pkaza@stevens.edu

***Abstract*—Abundant accumulation of digital histopathological images has led to the increased demand for their analysis, such as computer-aided diagnosis using machine learning techniques. Histopathological grading of cancer not only offers an insight to the patients prognosis but also helps in making individual treatment plans. Pathologists perform this grading by manual examinations of a few thousand images for each patient. Hence, finding the tumor figures from these images is a tedious job and prone to observer variability due to variations in the appearances of the metastatic tumor tissues. In this project, we developed an algorithm to classify metastatic cancer in small image patches taken from larger digital pathology scans.**

***Index Terms*—Histopathology, cancer, metastatic tissues, convolutional neural networks**

I. Introduction

A. Motivation

Cancer is often thought of as an untreatable, unbearably painful disease with no cure. However popular this view of cancer may be, it is exaggerated and over-generalized. Cancer is undoubtedly a serious and potentially life-threatening illness. However, it is a misconception to think that all forms of cancer are untreatable and deadly. The truth of the matter is that there are multiple types of cancer, many of which can today be effectively treated to eliminate, reduce or slow the impact of the disease on patients' lives. While a diagnosis of cancer may still leave patients feeling helpless and out of control, in many cases today there is cause for hope rather than hopelessness.

The cure for cancer has always been a major concern in the field of medicine. Cancer that is diagnosed at an early stage while it has not spread, is more likely to be treated successfully. If cancer spreads, effective treatment becomes more difficult, and generally a persons chances of surviving are much lower. A clinical diagnosis alone is most often made in the context of advanced malignancy where anti-cancer therapy would neither improve quality of life nor survival. Thus, most of the patients have the diagnosis of cancer confirmed on tissue pathology along with a clinical treatment.

Tumors come in two forms; benign and malignant. Benign tumors are not cancerous, thus they do not grow and spread to the extent of cancerous tumors. Benign tumors are usually not life threatening. Malignant tumors, on the other hand, grow and spread to other areas of the body.

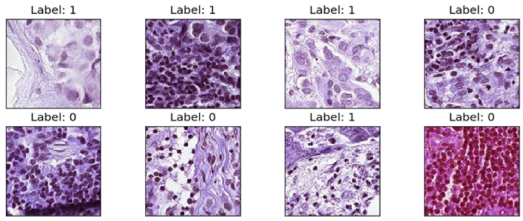

Fig. 1: Classification based on labels

Machine Learning methods help us classify cancer tissues based on these labels; Malignant or Benign which is important in the detection process (as shown in Fig. 1:). Therefore, these techniques have been utilized as an aim to model the progression and treatment of cancerous conditions. In addition, the ability of ML tools to detect key features from complex datasets reveals their importance. A variety of these techniques, including Convolutional Neural Networks (CNNs), Transfer Learning, Support Vector Machines (SVMs), Decision Trees (DTs), etc have been widely applied in cancer research for the development of predictive models, resulting in effective and accurate decision making.

B. Project Definition

The goal of this project is to create an algorithm that will identify the metastatic tissues in histopathologic scans of the lymph node sections using one of the deep learning techniques Convolutional Neural Network. We aim to classify cancer tissues based on the labels - Malignant or Benign. For simplicity, we have given Malignant as 1 and Benign as 0. In this project, we tried to understand the cancer detection process based on the given dataset. Our dataset includes many small pathology images to classify as labels - 1 or 0. The project is implemented using Python.

C. Project Flow

Understanding the dataset: In this dataset, we are provided with training and testing folders that contain cell images

named with their image id. The trainslabels.csv file provides the ground truth for the images in the train folder. Labels are in a binary format of 0 and 1.

Data pre-processing and Data Augmentation: This step involved importing the images and using their paths to access these images and apply augmentation techniques.

Training: Here, we train our model using Convolutional Neural Networks.

Testing: In this step, we validate and test our model.

II. Related Work

This section consists of research work from several authors who have worked on classifying and detecting cancer using different machine learning algorithms. Many of these researches from different authors have aided us in developing our project so far.

One of these publications is on Lung Cancer Detection using Deep Convolutional Networks. This is an insightful abstract published by Jelo Salomon and Bianca Schoen Phelan. In this paper, the author proposes a method of detecting lung cancer using a 2DUNet model on a web application. The author cropped 2D cancer masks on its reference image using the centre of the lung cancer given in the dataset and trained a model with different techniques and hyperparameters. Finally, the result is evaluated using a dice coefficient and confusion matrix metrics. The author reaches a 65.7 percent accuracy on the dice coefficient and an average 0.88 percent true positive rate and 0.71 percent false positive rate on a test set of positive and negative samples.

Another cancer related research was carried out by authors - Alexander Rakhlin, Alexey Shvets, Vladimir Iglovikov, Alexander A. Kalinin on Convolutional Neural Networks for Breast Cancer Histology Image Analysis. In this work, the author has developed a computational approach based on deep convolution neural networks for breast cancer histology image classification. Haematoxylin and eosin stained breast histology microscopy image dataset have been provided as a part of the ICIAR 2018 Grand Challenge on Breast Cancer Histology Images. Their approach utilizes several deep neural network architectures and gradient boosted trees classifier.

In recent years, deep learning has attracted great attention in artificial intelligence (AI) due to its successes in various research fields, such as pattern recognition, computer vision and big-data analysis. As one of the most successful techniques in deep learning, DCNN achieved outstanding performance in recognition of natural images. Several researchers studied and proposed methods for breast mass classification in mammography images. One of the recent one being by Suzuki in 2016 used the deep convolutional neural network (DCNN) for mass detection. This study introduced the transfer learning in the DCNN. The sensitivity achieved when differentiating between mass and normal lesions was 89.9 percent using the digital database for screening mammography. Their study was the first demonstration for the DCNN mammographic CAD applications.

Some other publications on cancer detection which made use of different ML algorithms are shown in the below table:

Publication	Method	Cancer type	Accuracy (AUC)	Validation method
Ayer T.	ANN	Breast cancer	0.89	10-fold cross validation
Waddell M.	SVM	Multiple myeloma	0.85	Leave-one-out cross validation
Listgarten J.	SVM	Breast cancer	0.85	20-fold cross validation
Stajadinovic.	BN	Colon carcinomatosis	0.79	Cross-validation

These researches and publications have helped us understand the importance of detection of cancer at an early stage. The success of a disease prognosis is undoubtedly dependent on the quality of a medical diagnosis; however, a prognostic prediction should consider more than a simple diagnostic decision. Upon careful analysis of the performance metrics of different machine learning algorithms, we have realized the need to create an efficient model which would reduce misclassifications and ensure a highly accurate performance. Furthermore, there are nowadays separate subgroups among the same type of cancer based on specific genetic defects that have different treatment approaches and options as well as different clinical outcomes. This is the foundation of the individualized treatment approach, in which computational techniques could help by identifying less costly and effectively such small groups of patients.

III. Solution

A. Dataset Description

In this dataset, we are provided with many small pathology images to classify. Files are named with an image id. The trainslabels.csv file provides the ground truth for the images in the train folder. We are predicting the labels for the images in the test folder. A positive label indicates that the centre 32x32px region of a patch contains at least one pixel of tumour tissue. Tumour tissue in the outer region of the patch does not influence the label. This outer region is provided to enable fully convolutional models that do not use zero padding, to ensure consistent behaviour when applied to a whole-slide image. The original PCam dataset contains duplicate images due to its probabilistic sampling, however, the version presented on Kaggle does not contain duplicates. However, we have been provided with the same data and splits as the PCam benchmark.

B. Machine Learning Algorithm Used

Our project is based on Convolutional neural networks which is a very efficient model in deep learning used to classify images. Artificial Intelligence has been witnessing a monumental growth in bridging the gap between the capabilities of humans and machines. Researchers and enthusiasts alike, work on numerous aspects of the field to make amazing things happen. One of many such areas is the domain of Computer Vision.

The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image and Video recognition, Image Analysis and Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one algorithm- a Convolutional Neural Network (as shown in Fig. 2:).

Convolutional Neural Networks:

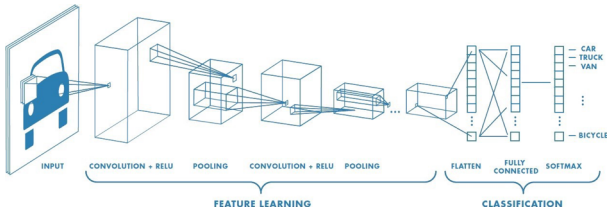

Fig. 2: Convolutional Neural Network

Definition: A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets could learn these filters/characteristics. Therefore, we are using convolutional neural networks to classify metastatic tissues as malignant or benign (1 or 0).

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area.

Working of CNN: There are four main operations in the CNN (as shown in Fig. 3:)

- Convolution
- Batch Normalization
- Pooling
- Dropout

These operations are the basic building blocks of every Convolutional Neural Network.

The primary purpose of Convolution is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. We still need to specify parameters such as number of filters, filter size, architecture of the network, etc. before the training process). The greater number of filters we have, the more image features get extracted and the better our network becomes at recognizing patterns in unseen images.

Batch Normalization is a method to reduce internal covariate shift in neural networks. In principle, the method adds an additional step between the layers, in which the output of the

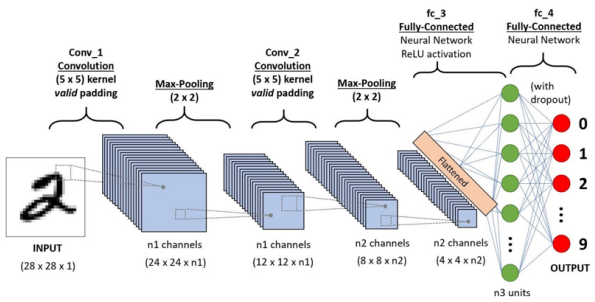

Fig. 3: Working of CNN

layer before is normalized. Batch normalization (BN) consists of two algorithms. Algorithm 1 is the transformation of the original input of a layer to the shifted and normalized value. Algorithm 2 is the overall training of a batch-normalized network. Due to batch normalization, networks train faster converge much more quickly.

A pooling layer is another building block of a CNN. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network. Pooling layer operates on each feature map independently. The most common approach used in pooling is max pooling.

Dropout is a regularization technique for neural network models. It is a technique where randomly selected neurons are ignored during training. They are dropped-out randomly. The effect is that the network becomes less sensitive to the specific weights of neurons. This in turn results in a network that is capable of better generalization and is less likely to overfit the training data. Dropout is implemented in Keras. It is only used during the training of a model and is not used when evaluating the skill of the model.

C. Preliminary Implementation

Implementation: This project was done in four stages illustrated in the project flow. This section will elucidate all these stages.

Understanding the Dataset:

As stated before we have used images from training dataset and testing dataset. Along with a trainlabels.csv which provides a ground truth for all the image ids in the training dataset.

We have 220k training images and 57k evaluation images. This dataset is a subset of the PCam dataset and the only difference between these two is that all duplicate images have been removed.

According to the data description, there is a 50/50 balance between positive and negative examples in the training and test splits. However, the training distribution seems to be 60/40 (negatives/positives). A positive label means that there is at least one pixel of tumor tissue in the center region (32 x 32px) of the image. Tumor tissue in the outer region of the patch does not influence the label. This means that a negatively labeled image could contain metastases in the outer region. Thus, it would be a good idea to crop the images to the center region.

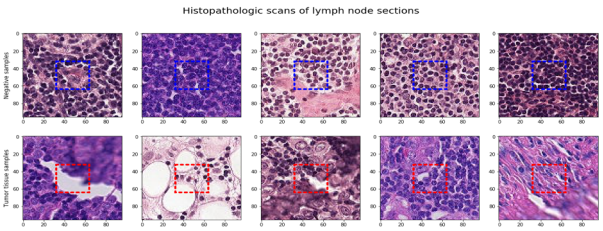

Fig. 4: Creating a 32x32 patch

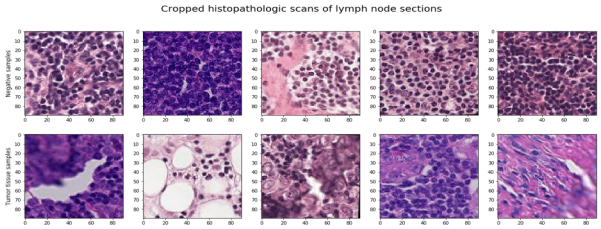

Fig. 5: Cropped Images

Importing the data: The libraries required for this project are as follows: numpy, pandas, matplotlib, sklearn, cv2, random and keras. We began our project by creating a pandas data frame containing the path of all the files in the trainpath folder and then read the matching labels from the provided csv file.

Data Visualization and Data Augmentation: We then proceeded to Exploratory Data Analysis (EDA). This was carried out in three steps-

- Loading the images for visualization
- Understand the distribution of the two classes (malignant / benign)
- Data Augmentation

With the help of OpenCV, we read and separated the images into positively and negatively classified samples from the training path. We then created a rectangular patch of size 32x32px (as shown in Fig 4). This implies that tumor tissue outside the region of the patch does not influence the label which in turn means that a negatively labelled image could contain metastases in the outer region.Thus, cropping the images would reduce any misclassifications (as shown in Fig 5).

Moving onto data visualization, we designed plots of some randomly generated images with and without cancer tissue for comparison.

We then proceed towards data augmentation. Here we will define what image augmentations to use and add them directly to our image loader function. Note that if we apply augmentation here, augmentations will also be applied when we are predicting (inference). This is called test time augmentation (TTA) and it can improve our results if we run inference multiple times for each image and average out the predictions.

Data Augmentation consists of the following processes - (as shown in Fig 6)

- random rotation

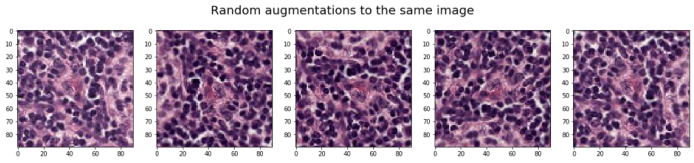

Fig. 6: Random Augmentations

- random crop
- random flip
- random contrast
- random brightness

After data augmentation, the focus of our project shifted to creating and setting up the model. This was the most important step in our project.

We then split the data into a training and validation dataset. We used 80 percent of the data for training and 20 percent to validate that our model can generalize to new data. After that, to avoid any influence of a possible previous sorting of data we shuffled the data (in-place).

We then setup a neural network architecture. This used keras, which made it very easy to setup a neural network and start training it. The convolutional neural network consists of three blocks of convolutional layers, batch normalization, pooling and dropout.

Once our model was created by using these CNN blocks, we then proceeded to training our model with keras. We used a batch size of 50, i.e., fed the network 50 images at once. We set the learning rate of 0.001 for now. As output, we got the classification accuracy of the model.

After training our model, we have at present created a submission by predicting the labels of the test data.

D. Ensemble Learning

We have tried CNN algorithm with different approaches. So, we have total three submission files. In order to improve our accuracies, we combined all these submission files with ensemble learning.

- In ensemble algorithm, we can ensemble the results of various algorithms and generate the new prediction.
- Various algorithms have their own approaches with some advantages and disadvantages. Therefore, sometimes a method can have corner cases in which it cannot predict properly.
- In this scenario, if we ensemble different outputs in one method then it can eliminate the corner cases of other methods and can help each other to improve the results.
- In this project, we took the prediction results from the various previous results and tried to combine up to three prediction files results using Ensemble Algorithm.
- There are different methods in ensemble learning. In this project, we have used averaging and weighted averaging methods of ensemble learning.

Sr No.	Methods	Accuracy
1.	Approach 1	88%
2.	Approach 2	90%
3.	Approach 3	88%
4.	Ensembling (Averaging)	91%
5.	Ensembling (Weighted Average)	93%

Fig. 7: Results

IV. Results

These results imply that our model will be able to predict if the level of tumor and hence cancer in a metastatic tissue is malignant or benign with an accuracy of 0.93. (as shown in Fig. 7)

We have also plotted a ROC curve with its AUC value = 0.958. (As shown in Fig. 8)

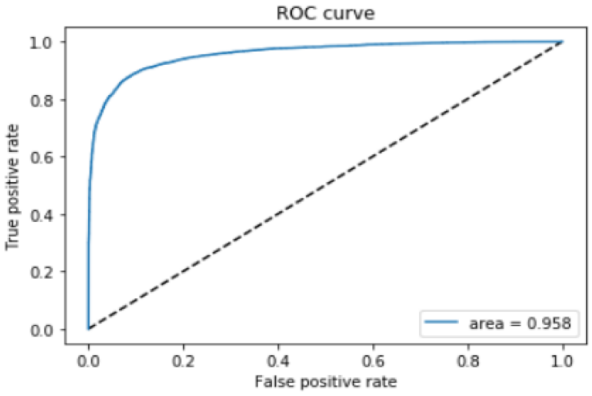

Fig. 8: ROC

V. Comparision with top ranking kaggle teams

In our project, the framework which we have used is tensorflow and keras. We obtained an accuracy of 0.93. Some of the top-ranking kernels from Kaggle worth appreciation are as follows:

A. Joni Juvonen

This kernel obtained an accuracy of 0.9622. Framework used in this kernel includes scikit learn and fastai.

Following are the advantages and disadvantages of the frameworks used:

Advantages of using tensorflow:

I think the biggest advantage of using Tensorflow over Scikit Learn is the ability to do automatic differentiation. Tensorflow works on a neat idea that you build a computation graph for doing any computation and you always end up working on that graph. The nodes on the graph are the different operations and the edges are the tensors. This structure of visualizing a problem enables Tensorflow to provide us with automatic differentiation to perform backpropagation easily. Tensorflow provides other low-level operations. Thus, you can literally build any machine learning model.

Disadvantages of using tensorflow:

TensorFlow is a library for array data calculations and computations that can be used to conduct neural network and deep learning. It doesn't provide other machine learning methods, like decision tree, logistic regression, k-means or pca.

Advantages of using scikit learn:

We can use scikit learn to provide the above machine learning methods that tensorflow doesnt provide. It is robust, fast, easy to use, comprehensive, and well documented.

Disadvantages of using scikit learn:

On the contrary to TensorFlow, it doesn't have deep learning framework. Although scikit learn is a great ML library, its deep learning functionality is quite limited. They introduced shallow networks quite recently, and to my knowledge do not have convolutional or recurrent networks yet. If we want a simpler solution it would be better to use keras. If you need more flexibility for designing the architecture, we can use TensorFlow. They have used Fast.ai V1 software library that is built on PyTorch.

Advantages of fastai include:

- Much less code for us to write for most common tasks
- More best practices baked in, so normally faster to train and higher accuracy
- Easier to understand
- Handles tabular data much better
- Fits in with wider python ecosystem better (e.g pandas)
- Fastai library has implemented a training function for one cycle policy that we can use with only a few lines of code. The policy brings more disciplined approach for selecting hyperparameters such as learning rate and weight decay. This can potentially save us a lot of time from training with suboptimal hyperparameters.

Disadvantages of fastai:

- Not much documentation.
- Relies on pytorch, which doesnt have such mature production (mobile or high scalability server) capabilities compared to tensorflow.
- Pytorch doesnt run on as many devices yet (e.g Googles TPU)
- Some parts still missing or incomplete (e.g object localization APIs)
- I think the biggest issue with fastai, or even standalone pytorch, would be production deployability.
- Also skills in Keras/TF could be much more easily marketable.

Run-time for their code: 22575.3 seconds
Run-time for our code: 2156.3 seconds

B. Ben Fung

This kernel obtained an accuracy of 0.9704. Framework used in this kernel includes scikit learn and fastai. All the advantages and disadvantages of using scikit learn and fastai

are covered in the previous section. In this kernel, different activation function is used. They used softmax function whereas we have used relu and sigmoid. Softmax function is used for multiclass classficiation unlike sigmoid function which is used for binary classification.

Run-time for their code: 91.9 seconds

Run-time for our code: 2156.3 seconds

C. SEED=323

Framework used in this kernel is Pytorch and scikit learn

Accuracy: 0.9744

Runtime: 29223.3 seconds

VI. CONCLUSION

- We have implemented several approaches in our project.
- Each approach has yielded a classification model with a different accuracy.
- Although every approach did make use of CNN, there were different augmentation and processing techniques along with different attributes which contributed to the betterment of our results.
- Finally, we have implemented the ensemble algorithm in which we have combined all our approaches to create a model which has given us the maximum accuracy.
- We applied the ensemble algorithm in two methods: Averaging and weighted average.
- Using all the approaches mentioned above, we were able to successfully classify the metastatic tissues in histopathologic scans of the lymph node sections into the labels of malignant (1) or benign (0).

ACKNOWLEDGMENT

It is with great satisfaction and achievement that we have completed this project and we would like to take this opportunity to acknowledge everyone who contributed towards our work. We express our sincere gratitude towards Prof. Shucheng Yu, for his guidance and support. We are grateful for his cooperation and his valuable suggestions.

REFERENCES

[1] Mentalhelp.net. (2019).Overview: Introduction to Cancer. [online] Available at: https://www.mentalhelp.net/articles/overview-introduction-to-cancer/ [Accessed 10 May 2019].

[2] Prevention,C. (2019). Cancer Detection and Prevention. [online] Journals.elsevier.com.Available at: https://www.journals.elsevier.com/cancer-detection-and-prevention [Accessed 12 May 2019].

[3] Ieeexplore.ieee.org. (2019). Mitosis Detection for Invasive Breast Cancer Grading in Histopathological Images- IEEE Journals and Magazine. [online]Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=7165640 [Accessed 17 May 2019].

[4] Kashyap, A., Gunjan, V., Kumar, A., Shaik, F. and Rao, A. (2019). Computational and Clinical Approach in Lung Cancer Detection and Analysis.

[5] Kourou, K., Exarchos, T., Exarchos, K., Karamouzis, M. and Fotiadis, D. (2019). Machine learning applications in cancer prognosis and prediction.

[6] H. Irshad, Automated mitosis detection in histopathology using morphological and multi-channel statistics features, J. Pathol. Informat., vol. 4, no. 1, p. 10, 2013.

[7] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, Mitosis detection in breast cancer histology images with deep neural networks, in Proc. 16th Int. Conf. MICCAI, 2013, pp. 411418.

[8] Verrill, C. (2019). Histopathological assessment of lymph nodes in colorectal carcinoma: does triple levelling detect significantly more metastases?.

[9] Komura, D. and Ishikawa, S. (2019). Machine Learning Methods for Histopathological Image Analysis.

[10] Veeling, B. and Amsterdam, t. (2019). PCam: histopathology dataset for fundamental machine learning.. [online] Bas's Blog. Available at: http://basveeling.nl/posts/pcam/ [Accessed 17 May 2019].