Deep Learning approach for subtype classification and localization from non small cell lung cancer histopathology images

24th August, 2020

Team B19

Project Guide

Dr. M P Pushpalatha
Prof. & HOD of Dept.CS&E, JSSSTU

Students:

1. Hymavathi B U - 01JST16CS035
2. Santosh Umesh Shet - 01JST16CS132
3. Tejas H R - 01JST17CS421

Contents

Introduction

- Non-small cell lung carcinoma is a widely prevalent disease in which malignant cells form in the tissues of the lung. Adenocarcinoma (LUAD) and squamous cell carcinoma (LUSC) are two most predominant subtypes of non-small cell lung carcinoma.

- According to the National Cancer Institute , the US federal government's principal agency for cancer research and training, Lung cancer accounts for more deaths than any other cancer in both men and women which amounts to about 28% of all cancer deaths.

- In the US alone, approximately 228,820 new cases of lung cancer and approximately 135,720 deaths from lung cancer are estimated for the year 2020

- Non small cell lung carcinoma(NSCLC) amounts to 84% of all lung cancer cases.

- The analysis of the histopathology images of lung tissue is done to identify the type, subtype and the stage of cancer.

- But, this poses a challenge, as assessment of the histopathological slide even by a trained pathologist is time intensive and sometimes prone to error.

Aim / Statement of the Problem

- To develop a deep learning Convolutional Neural Network (CNN) model trained on histopathology images to accurately classify whole-slide pathology images into adenocarcinoma, squamous cell carcinoma or normal lung tissue to assist pathologists by providing a fast, accurate and inexpensive detection.

- Further, implementing an explainable model which provides localization and visualizes the model's results.

Objectives of the Project Completed

1. Data collection using GDC Tool - LUAD, LUSC and Normal Cells.

2. Classifying whole-slide histopathology images into normal tissue slide or Non Small Cell Lung Cancer(NSCLC) subtypes viz adenocarcinoma, squamous cell carcinoma.

3. Compare the results and performance of Inception v3 and our custom designed CNN model with relevant graphs.

4. Implementing Explainable model to highlight key features, that provides both specialists and patients a fast, accurate, easily understandable and inexpensive analysis of the whole-slide histopathology image.

5. Develop website and deploy our model as a service

6. Patient Report Generation

Existing Approach

- Inception v3 has been used to solve significant number of problems in the medical domain
- Inception v3 has been used in many automated medical image analysis tasks due to its robust architecture
- It has been used in the cancer detection, Alzheimer prediction, diabetic retinopathy identification and many more.
- The NSCLC problem domain is one of the major research areas due to the large scale prevalence of the disease
- Yu, Kun-Hsing et al.[1] in their study have extracted objective morphological information from thousands of whole slide non-small cell carcinoma images and then built a fully automated image-segmentation pipeline to identify the tumour nuclei and tumour cytoplasm from the histopathology images using the Otsu thresholding method. They extracted 9,879 quantitative image features and used regularized machine-learning methods to select the top features and to distinguish shorter-term survivors from longer-term survivors with an AUC of ~0.85.
- Nicolas Coudray et al.[5] have used end-to-end training and transfer learning on inception v3 for the subtype classification

Existing Approach

- Studies have been performed using conventional digital image processing techniques along with machine learning methods such as Support Vector Machines, Naive Bayes Classifier.

- These methods have accuracy at 75% levels in classifying LUAD and LUSC which leaves a lot to chance and makes the situation precarious.

- Deep learning methods have an edge over the conventional solutions. Convolutional Neural Networks based approach will outperform these methodologies.

- In a study by A.Esteva et al.[2], inception v3 was trained end-to-end with 129,450 clinical images to achieve two critical binary classification use cases: keratinocyte carcinomas versus benign seborrheic keratoses; and malignant melanomas versus benign nevi with AUC for each case over 91%.

System Requirements

Param System (Actual)

Description	Value
2x Nvidia Quadro P5000:	32GB
1x Nvidia Quadro P400:	2GB
Architecture:	x86_64
CPU op-mode(s):	32-bit, 64-bit
CPU(s):	72
On-line CPU(s) list:	0-71
Thread(s) per core:	2
Core(s) per socket:	18
Vendor ID:	GenuineIntel
CPU family:	6
Model name:	Intel(R) Xeon(R) Gold 6139 CPU @ 2.30GHz
CPU MHz:	2259.722
CPU max MHz:	2301.0000
BogoMIPS:	4601.79
NUMA node0 CPU(s):	0-17,36-53
NUMA node1 CPU(s):	18-35,54-71

Laptop (Used)

- Processor – Intel Core i7-9750H processor, turbo up to 3.10 GHz
- Memory - 12GB DDR4 Ram
- Storage - 2TB HDD + 256GB SSD
- Graphics - NVIDIA GeForce GTX 1660ti with 6 GB of Dedicated GDDR5 VRAM Graphics

Tools Used

- python 3.6.5
- tensorflow-gpu 1.9.0
- numpy 1.14.3
- matplotlib 2.1.2
- sklearn
- scipy 1.1.0
- openslide-python 1.1.1
- Pillow 5.1.0
- Pytorch 1.4.0
- Flask
- Network Drawing - NN-SVG

Dataset Collection

- Dataset of LUAD, LUSC and Normal Tissue slides were downloaded from GDC portal.
- The **GDC Data Transfer Tool** Client provides a command-line interface supporting both GDC data downloads and submissions.

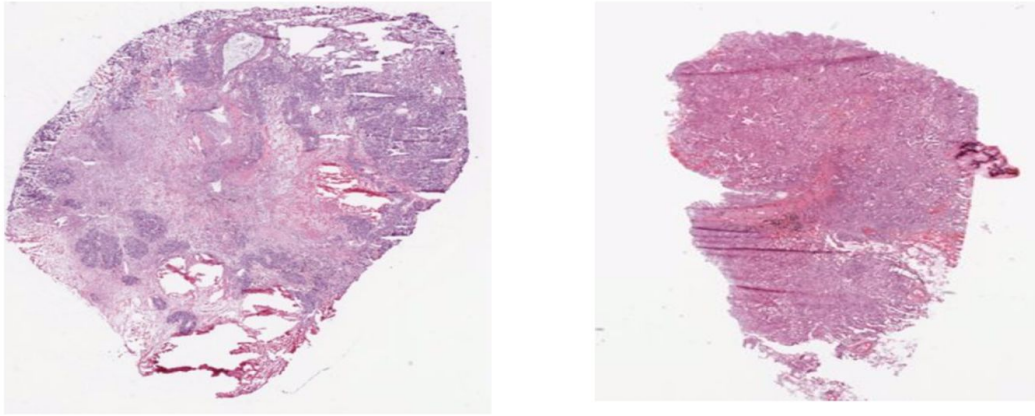
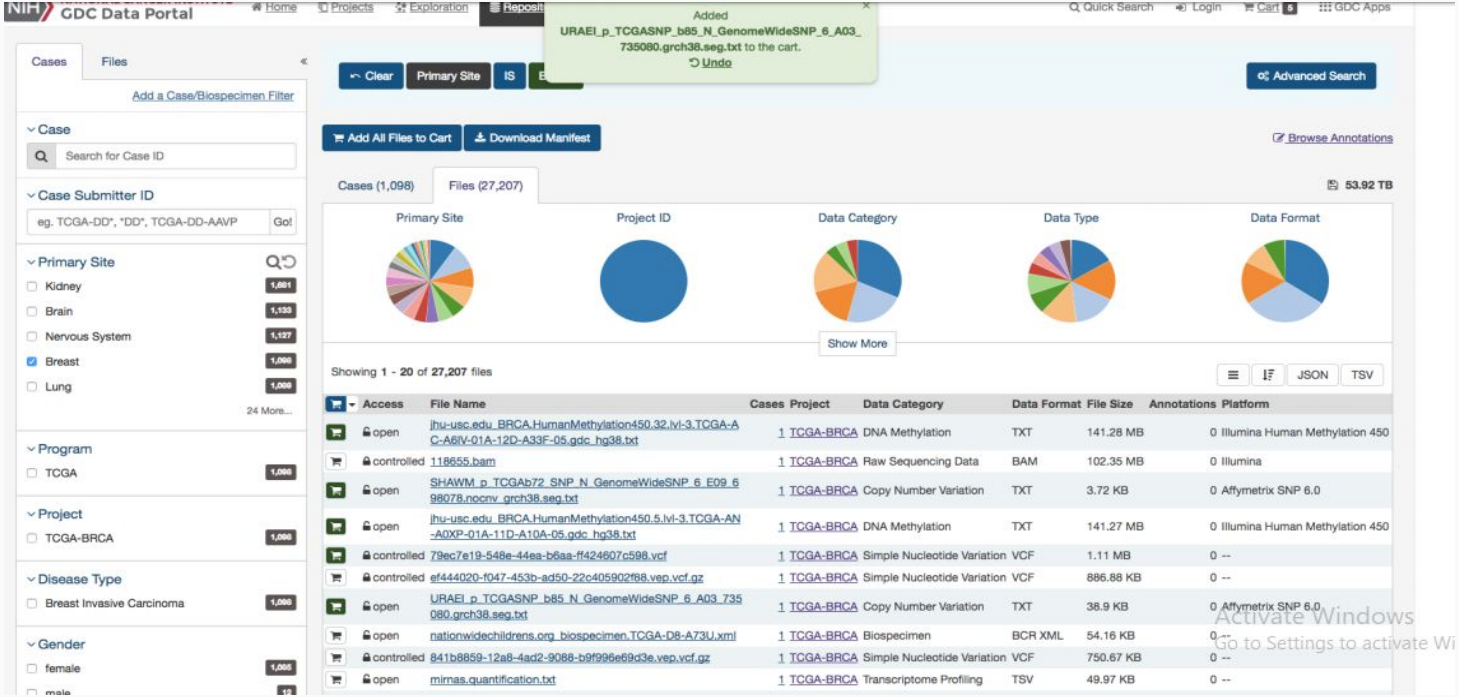


Figure 1. LUAD (left) , LUSC (right)

- **Downloading Data Using a Manifest File**
  A convenient way to download multiple files from the GDC is to use a manifest file generated by the GDC Data Portal.

GDC Portal

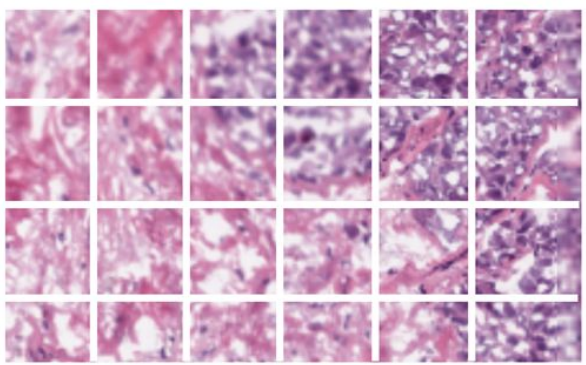


- **Downloading Data Using GDC File UUIDs**
  The GDC Data Transfer Tool also supports downloading of one or more individual files using UUID(s) instead of a manifest file.

Data Preprocessing

- Tiling of SVS files
- Conversion from SVS to JPG
- Sorting the tiles into folders
- Input Normalization
- Data Augmentation

Tiling



- Whole slide images are very high resolution images ranging from few MBs to GBs.
- Tile-based processing consists of dividing an image into small, rectangular pieces called "tiles," processing each tile, and reassembling the image. The principle of locality applies, so an image-processing operation usually requires input tiles of approximately the same area as the output tile.

- The tiles are non-overlapping, with background (white) less than 20%

- So, we use OpenSlide library to divide the .svs image to multiple jpeg tiles of shape (256x256).

Tiling

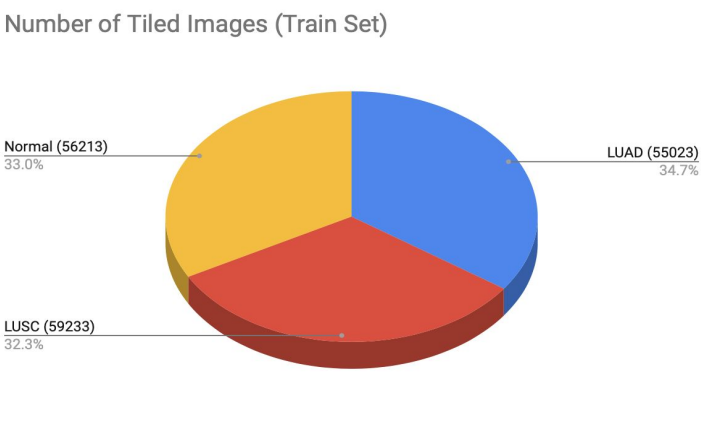
Dataset used with current approach:

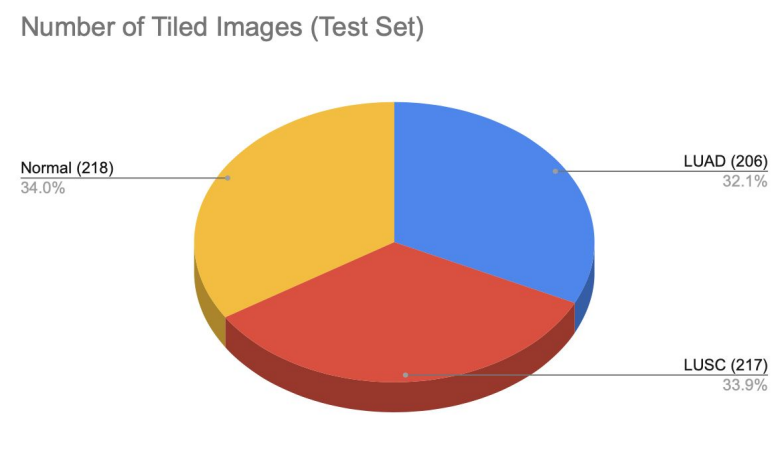
LUAD - 55,529 images

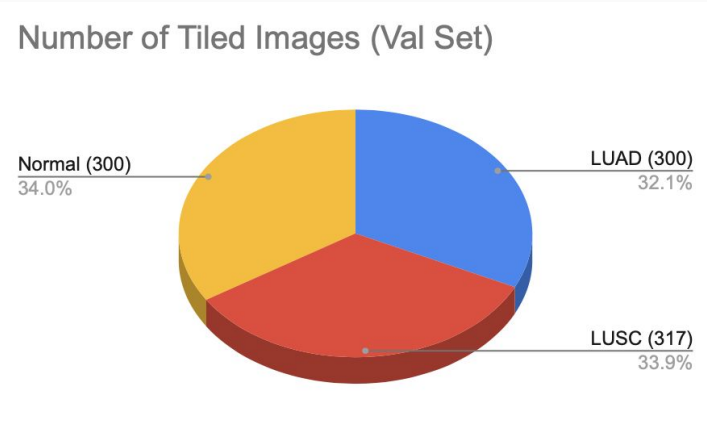
LUSC - 59,767 images

Normal - 56,731 images

Total tiled images are ~ **1,69,000**







Conversion from SVS to JPG format

- SVS is an image format associated with Aperio medical equipment, such as microscopes or scanscopes
- SVS files contain multiple images in a predefined order, including a baseline tiled image, a thumbnail, intermediate "pyramid" images (made up of tiles), an optional slide label image etc.
- JPG uses a type of compression that prioritizes the quality of some image sections over others, thus assuring that most favorable quality/size ratio.
- JPG image format is easy to compress just before feeding the tiles to the network. This format although is a lossy compression technique, the compressed images are sharper as only resolution is changed.

Sorting the tiles into folders

- The dataset downloaded from the TCGA are labeled and consists of metadata. In order to train the network each type of data manual sorting of images into respective class labeled folders using terminal command are done.

Folder Structure								
Train			Val			Test		
LUAD	LUSC	NORMAL	LUAD	LUSC	NORMAL	LUAD	LUSC	NORMAL

Normalizing the JPG images

- Normalization makes the convergence faster while training the network.
- The goal of normalization is to change the numeric values in the dataset to a common scale, without distorting differences in the ranges of values

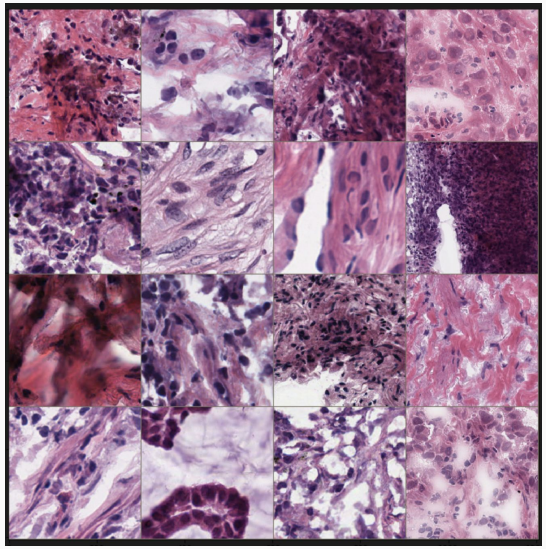
- Data normalization is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution.

- Data normalization is done by subtracting the mean from each channel and then dividing the result by the standard deviation.
  input[channel] = (input[channel] - mean[channel]) / std[channel]

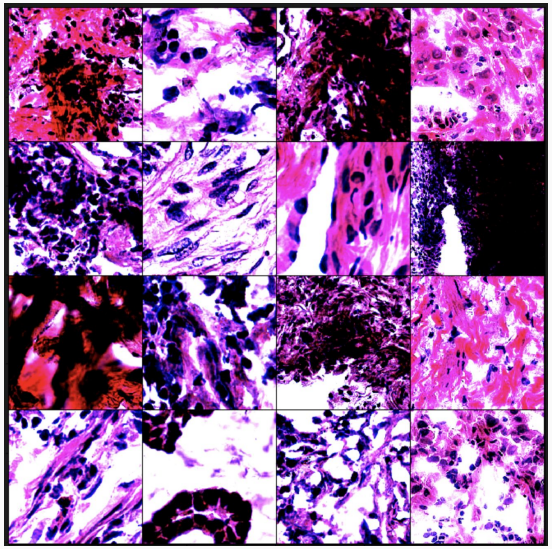
- The distribution of such data would resemble a Gaussian curve centered at zero

Normalizing the JPG images

- Normalization of input is done as it helps the model to converge quickly and speeds up the learning process. It removes large variations in the pixel values



Original Images



Normalized images

Data Augmentation

Another common pre-processing technique involves augmenting the existing data-set with perturbed versions of the existing images. This is done to expose the neural network to a wide variety of variations, so it generalizes well to the dataset and avoids high variance .

Transforms on input Image :

1. Resize - Resize the input PIL Image to the given size

2. RandomHorizontalFlip(p=0.5)
   Horizontally flip the given image randomly with a given probability. The image can be a PIL Image or a torch Tensor, in which case it is expected to have [..., H, W] shape, where .. means an arbitrary number of leading dimensions

Data Augmentation

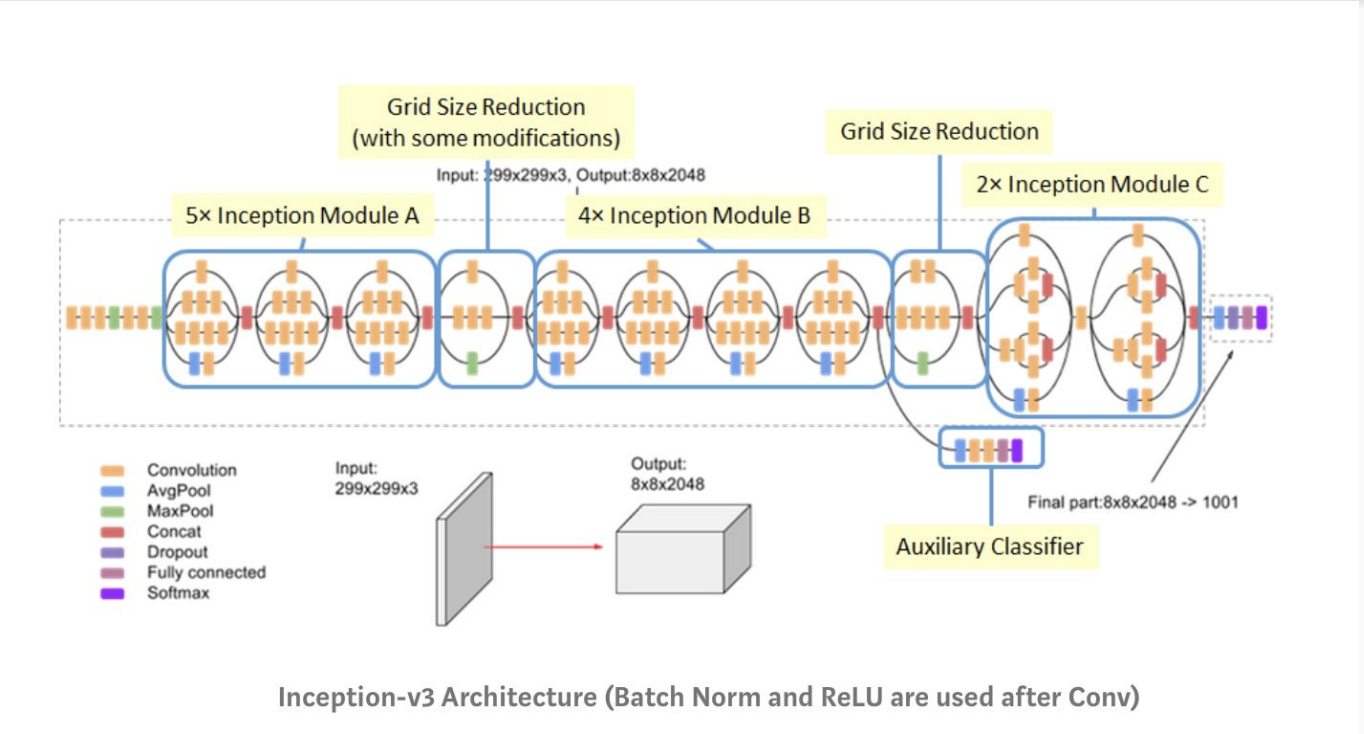
3. RandomResizeCrop - To account to various magnifications

Crop the given PIL Image to random size and aspect ratio.A crop of random size (default: of 0.08 to 1.0) of the original size and a random aspect ratio (default: of 3/4 to 4/3) of the original aspect ratio is made. This crop is finally resized to given size.

Parameters
size – expected output size of each edge
scale – range of size of the origin size cropped
ratio – range of aspect ratio of the origin aspect ratio cropped

Inception V3 Model

Architecture



Inception-v3 Architecture (Batch Norm and ReLU are used after Conv)

Network taken from [paper]

Architecture

- Inception-v3 consists of two parts:
  - Feature extraction part with a convolutional neural network.
  - Classification part with fully-connected and softmax layers.

- The Inception-v3 model achieves state-of-the-art accuracy for recognizing general objects in ImageNet dataset which comprises of ~1.48 million images belonging to 1000 classes

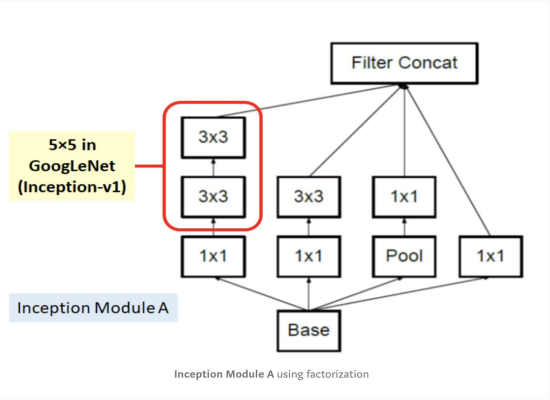
- The model extracts general features from input images in the first part and classifies them based on those features in the second part.

- With 42 layers deep, the computation cost is only about 2.5 higher than that of GoogLeNet, and much more efficient than that of VGGNet .

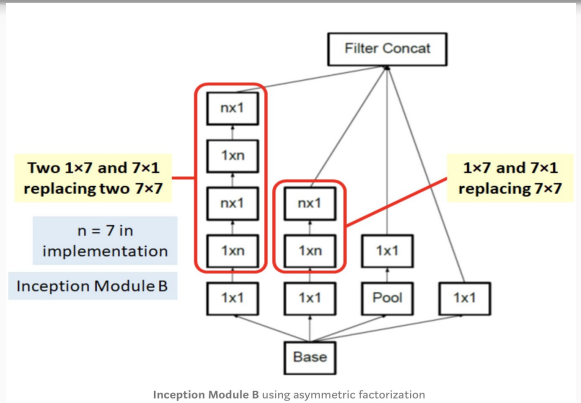
Architecture

- Very deep networks are prone to overfitting. It also hard to pass gradient updates through the entire network.
- Naively stacking large convolution operations is computationally expensive.
- To solve this, inception models have filters with multiple sizes operate on the same level. The network essentially will get a bit “wider” rather than “deeper”.
- Inception v3 exploits factorizing convolutions
- The aim of factorizing Convolutions is to reduce the number of connections/parameters without decreasing the network efficiency.
- The network architecture is composition of the modules - Inception module A, Inception module B, Inception module C, Grid size reduction and auxiliary classifier

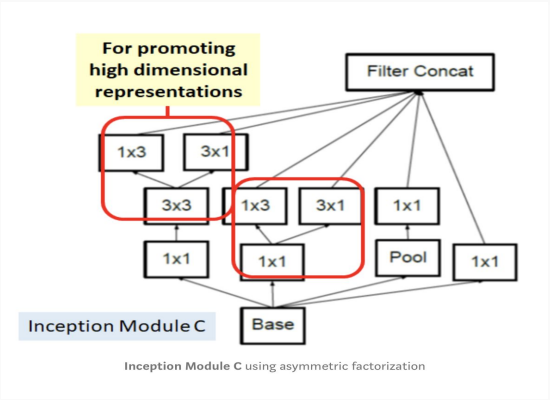
Architecture - Modules



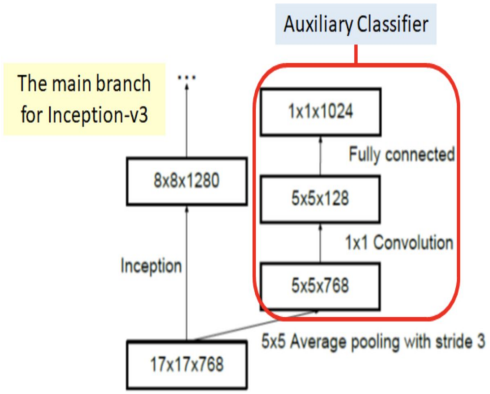
Inception Module A using factorization



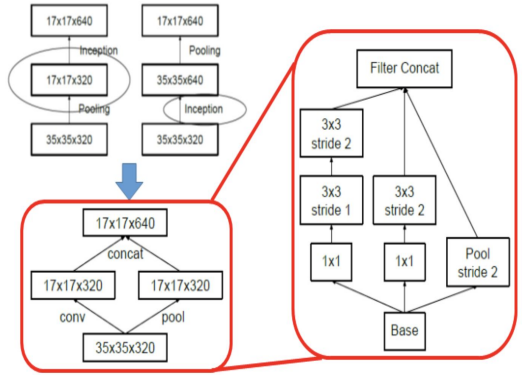
Inception Module B using asymmetric factorization



Inception Module C using asymmetric factorization



Auxiliary Classifier act as a regularization



Conventional downsizing (Top Left), Efficient Grid Size Reduction (Bottom Left), Detailed Architecture of Efficient Grid Size Reduction (Right)

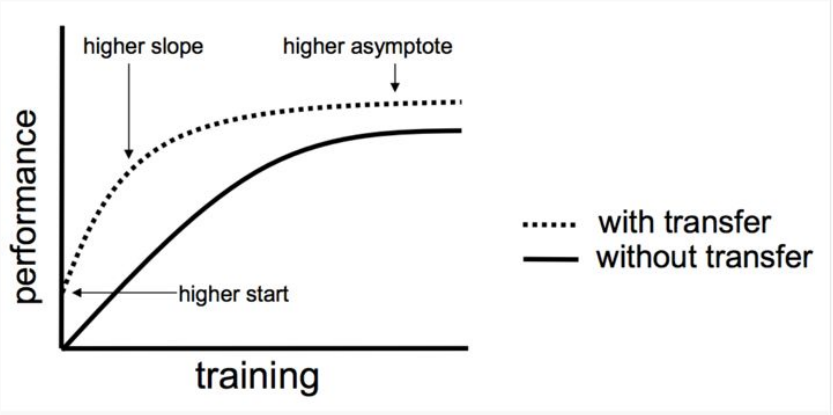
Performance of Inception v3

Network	**Top-1** Error	**Top-5** Error	**Cost** Bn Ops
GoogLeNet [20]	29%	9.2%	1.5
BN-GoogLeNet	26.8%	-	**1.5**
BN-Inception [7]	25.2%	7.8	2.0
Inception-v3-basic	23.4%	-	3.8
Inception-v3-rmsprop RMSProp	23.1%	6.3	3.8
Inception-v3-smooth Label Smoothing	22.8%	6.1	3.8
Inception-v3-fact Factorized 7×7	21.6%	5.8	4.8
Inception-v3 BN-auxiliary	**21.2**%	**5.6**%	4.8

Transfer Learning

- Transfer learning refers to the situation where information that has been learned in one setting is exploited to improve generalization in another setting
- Transfer learning allows model creation with significantly reduced training data and time by modifying existing rich deep learning models.
- In transfer learning, when a new model is built to classify an original dataset, the feature extraction part is reused and the classification part is re-trained with the dataset.
- Since the feature extraction part (which is the most complex part of the model) is not trained, the training process is accomplished with less computational resources and training time.
- The old top layer is removed and new top layer(s) are trained on the dataset of the task at hand.
- The reason final layer retraining can work on new classes is that it turns out the kind of information needed to distinguish between all the 1000 classes in ImageNet is often also useful to distinguish between new kinds of objects.

Transfer Learning



Benefits

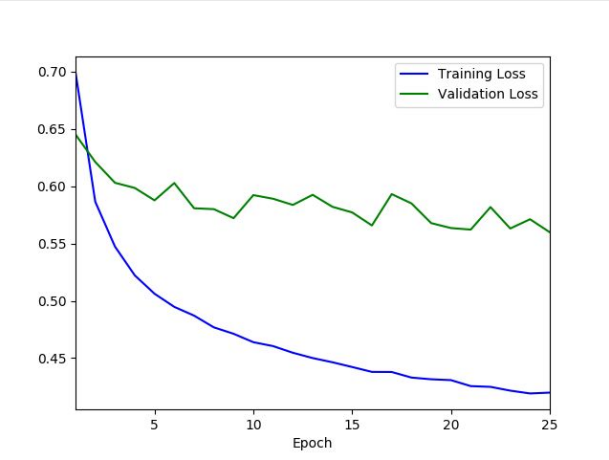
- **Higher start**. The initial skill (before refining the model) on the source model is higher than it otherwise would be.
- **Higher slope**. The rate of improvement of skill during training of the source model is steeper than it otherwise would be.
- **Higher asymptote**. The converged skill of the trained model is better than it otherwise would be.

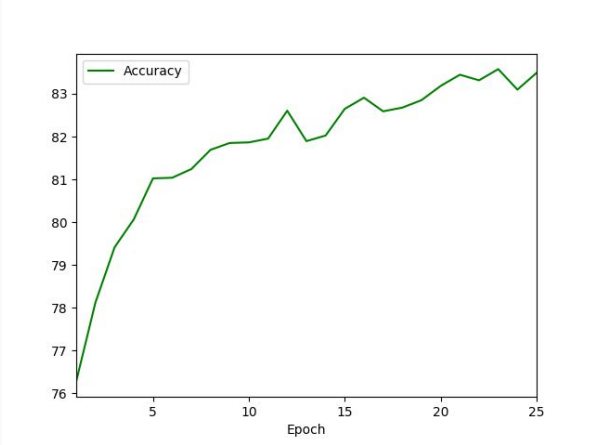
Training Process

The inception v3 architecture was trained by transfer learning using our training datasets and following the procedure described:

- We removed the final layers of inception v3 model, added 3 Fully Connected layers and an output layer
- We used cross entropy loss, and Adam optimization method along with the hyperparameters of the transfer learning case for training.
- In this approach we optimized the weights of the fully connected layers.
- This strategy was tested on the classification task of Normal vs LUAD vs LUSC.
- The training jobs were run for 25 epochs with batch size of 32.
- We computed the cross-entropy loss function on the train and validation dataset, and used the model with the best validation score as our final model.
- We did not tune the number of layers or hyper-parameters of the inception network such as size of filters.

Results for Inception V3





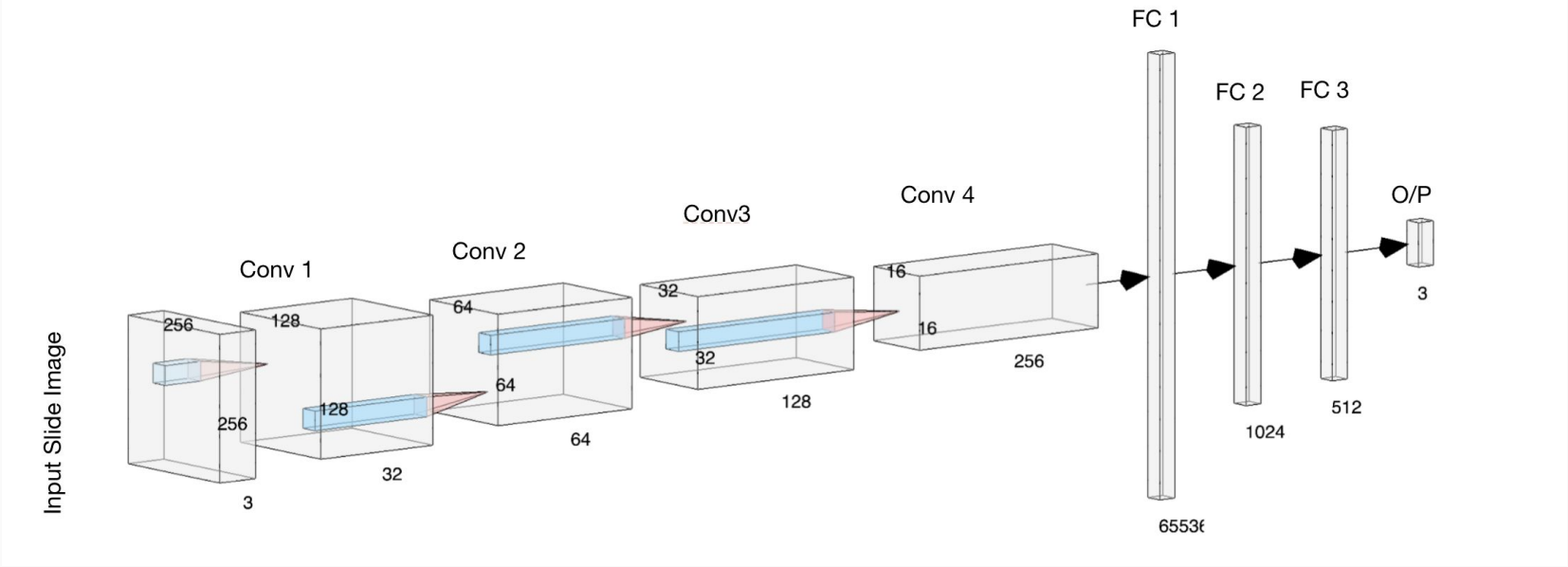
Techniques used to optimize the network:

- A scheduler to dynamically change learning rate based on validation loss.
- Weight decay method to overcome very early overfitting.
- A dropout node (probability ~ 0.8) was used to avoid overfitting.

Obtained accuracy ~84%

Custom Designed Neural Network

Architecture



- All layers activations ReLU
- MaxPooling after each Conv Layer
- DropOut in all the FC layers with 0.8 probability
- Output Layer activation Softmax

Network Figure Drawn using TOOL

Architecture

- The network takes in an already tiled whole slide image of shape (256,256,3) as input.
- The first convolution layer extracts multiple features of shape (128,128,32) with kernel shape (3,3) with stride and padding (1).
- The Conv2 layer takes input of shape (128,128,32) and outputs (64,64,64) where number of features extracted is 64 with kernel shape (3,3) with stride and padding (1).
- The Conv3 layer takes input of shape (64,64,64) and outputs (32,32,128) where number of features extracted is 128 with kernel shape (3,3) with stride and padding (1).
- The Conv4 layer takes input of shape (32,32,128) and outputs (16,16,256) where number of features extracted is 256 with kernel shape (3,3) with stride and padding (1).
- After each convolution layer we normalize the image pixels using 2D Batch Norm followed by Maxpool (2,2), which normalizes the pixels and reduces the features set size by a factor of 2.

Architecture

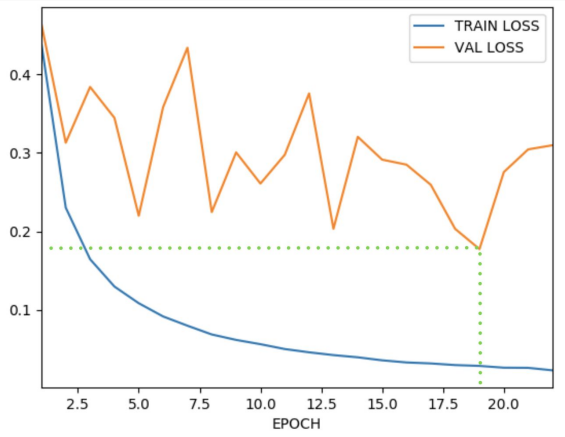
- Then the tensor is passed through fully connected linear layers of shape FC1 (65536), FC2 (1024), FC3 (512), and outputs tensor (3).
- Each of the linear layers are followed by 1D Batch Normalization and dropout with probability of 0.7. The activation function used is Relu for the entire network to enable faster training.
- The output layer has softmax activation, which gives us the probability for each type of class predicted by the model.

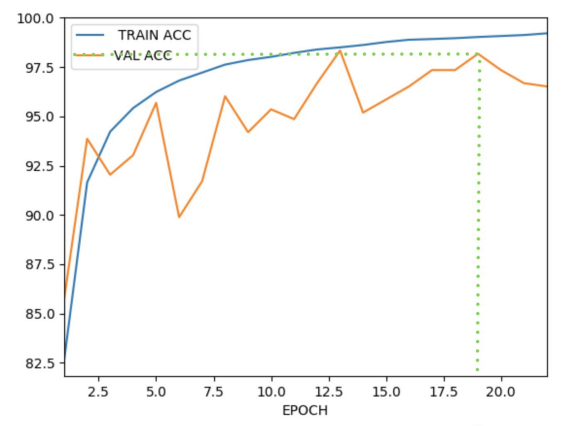
Training Process

The Custom Net was trained using our training set:

- The network was trained for 25 epochs using cross-entropy loss function with learning rate of 0.001
- Adam Optimizer was used to converge the model quickly
- BatchNormalization and Dropout techniques were used to prevent overfitting
- The validation set was used to measure the performance, and the model with the best validation score as our final model.

Results of Custom Network



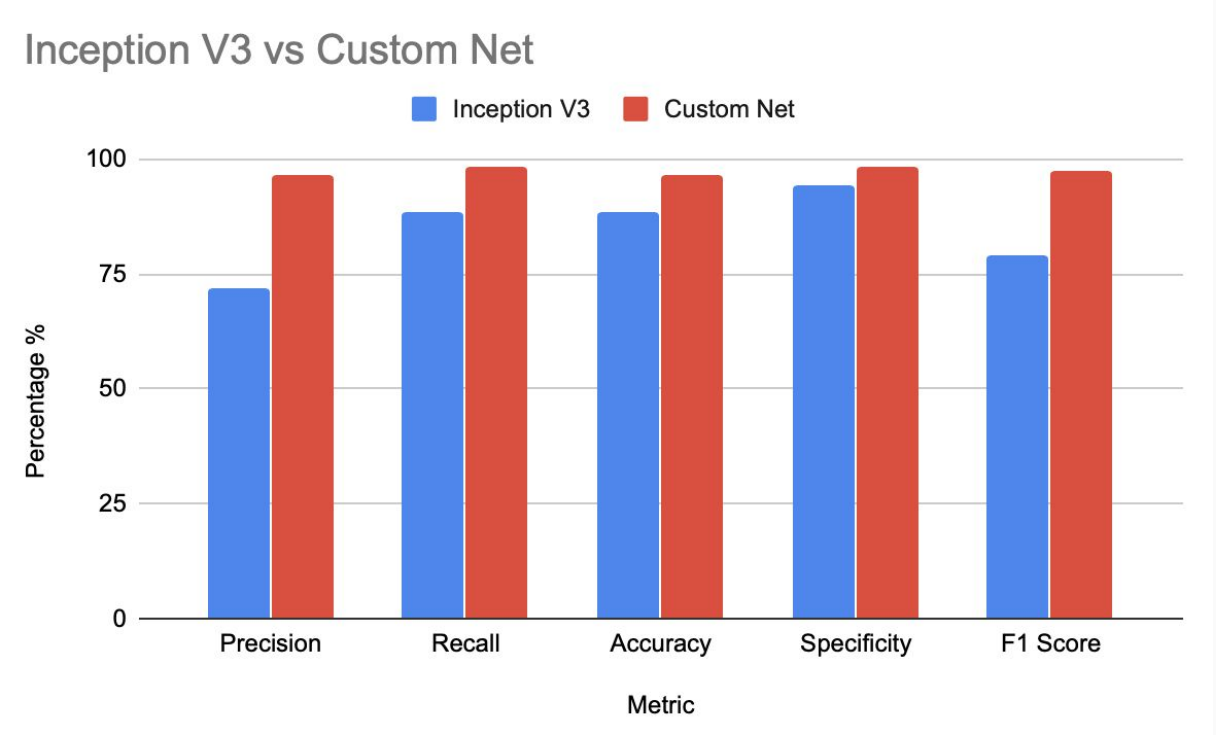


Key findings:

- The model learns well on train set and performs well on validation set.
- Even though the model fluctuates on validation set, at epoch 19 we get the highest accuracy and lowest loss.
- Only normalization and dropout techniques were used to prevent overfitting.
- The model performs well on our small dataset.

Obtained accuracy ~98%

Model Comparison



Explainable Results Using Heatmap For Patient Report Generation

Localisation and Explainability

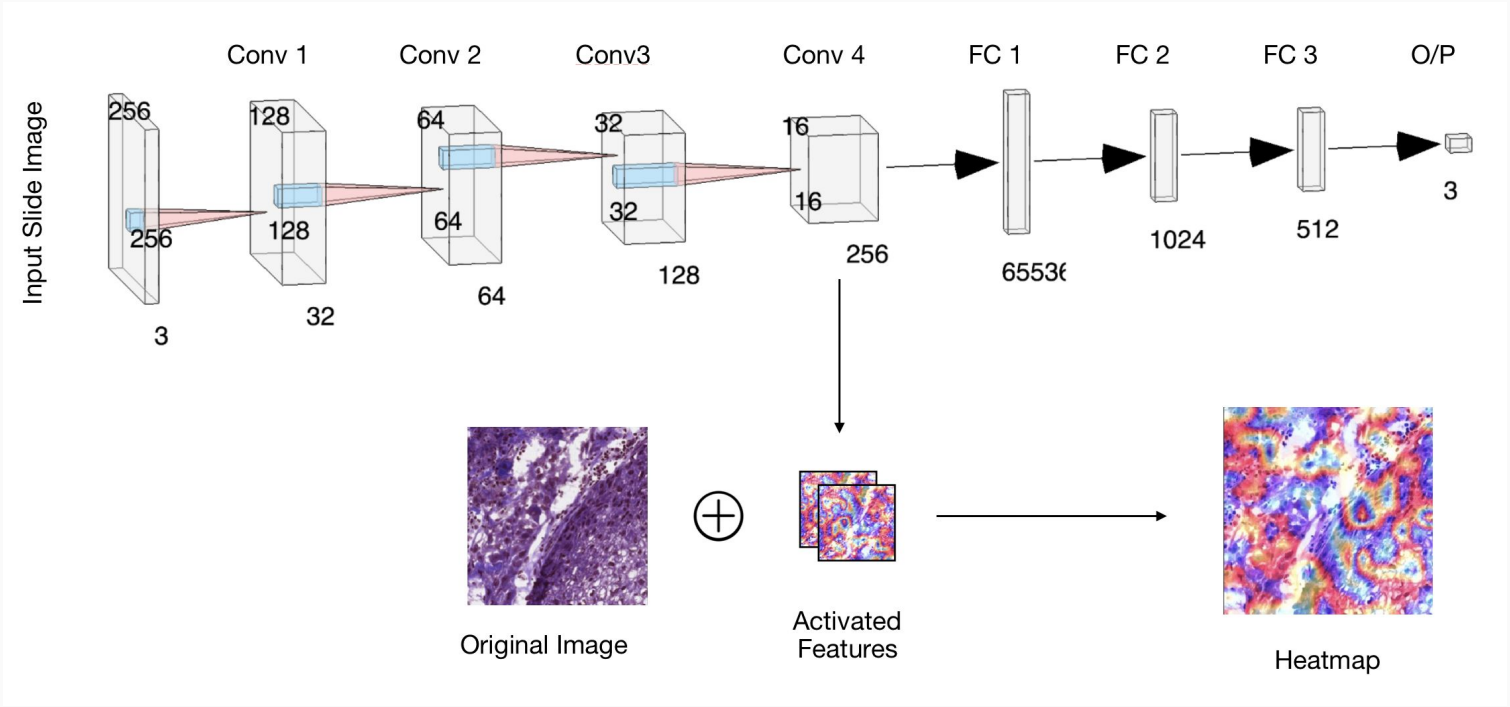
Explainability highlights activated features of a input object in a neural network during a forward pass.

These explainability can be used to understand which regions of the input caused the model to predict a particular output.

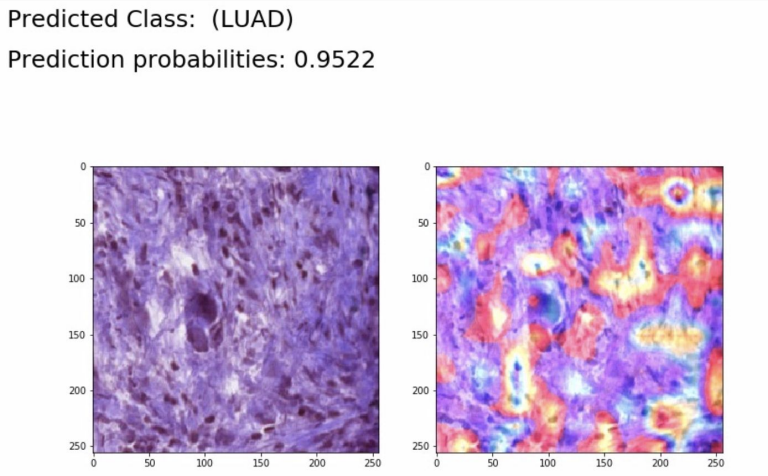
Example:

Image taken from paper LINK

Explainable Results Using Heatmap



Sample Tissue Slide Images With Explainability



Correct prediction of LUAD class by the model with 95% accuracy.

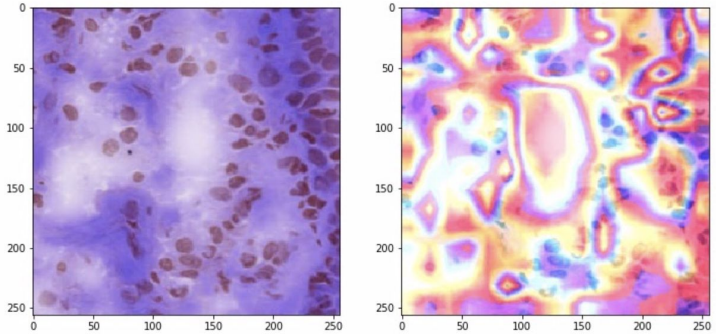
The yellow and red regions are regions where our model identifies region of high malignancy.

These explanations by our model helps pathologists pinpoint the area of malignancy.

Continued

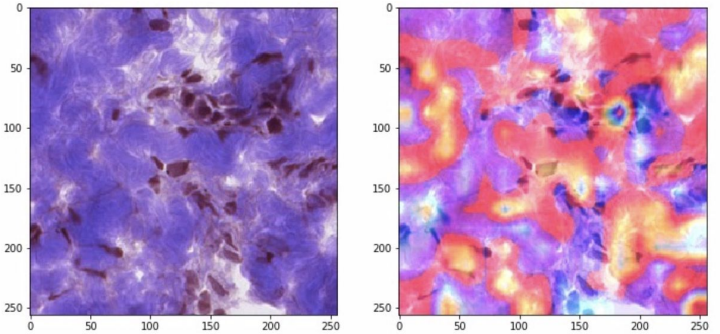
Predicted Class: (Normal)

Prediction probabilities: 1.0000



Predicted Class: (LUAD)

Prediction probabilities: 0.9621



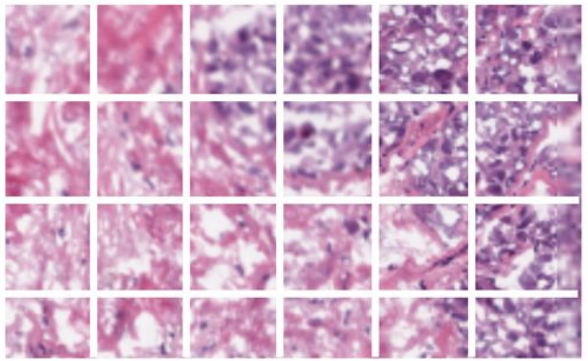
Correct predictions of respective classes with explanations by our custom model

Project Demo

Patient Report Generation

The process of report generation follows the following steps:

1. The uploaded svs file is tiled using the same process as above.
2. Each tile is given as input to the model to obtain the type of cancer and the corresponding prediction score
3. The count for each class and the corresponding sum of probabilities is stored as a vector.
4. Misclassification for a given tile is taken care by taking the class that has the maximum count
5. The corresponding average probability is taken as the prediction probability of the model.
6. Top three tiles of the resultant class with the highest probabilities are added to the report. This localizes the malignancies to a particular part of the whole slide.
7. If the user uploads only a single tile, then it is processed as such and the output is presented.
8. The report is available as a pdf to download.

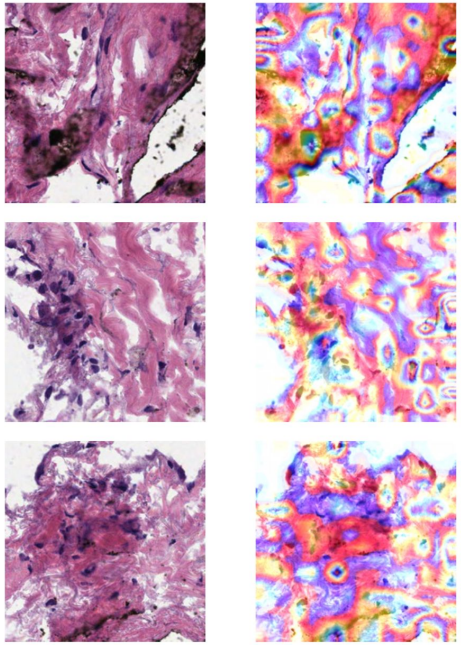


Patient Report

Report#: TCGA-55-7994-11A-01-TS1

Diagnosis: Positive for Adenocarcinoma(LUAD)
Model Confidence: 0.9764

Tile | Heatmap



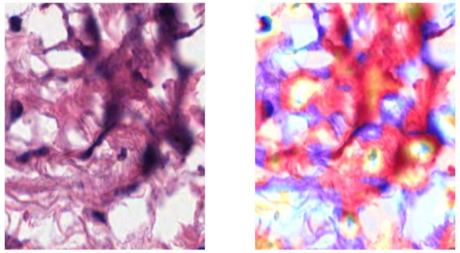
The diagnosis and the report is machine generated to aid interpretation by doctors and pathologists.Not to be used by patients for self-diagnosis.

Report generated on: 2020-08-23 14:53:35

Report#: 12_52

Diagnosis: Normal
Model Confidence: 0.5153

Tile | Heatmap



The diagnosis and the report is machine generated to aid interpretation by doctors and pathologists.Not to be used by patients for self-diagnosis.

Report generated on: 2020-08-23 14:54:03

Future Improvements

- Training the model with more data using Param System
- Train on DenseNet 121 architecture and check for performance
- Further improve patient report interpretation for doctors in collaboration with pathologists at JSS Hospitals
    - Show heatmap based on prediction scores or localization
    - Improve heatmap through segmentation networks

Publication

Publication details:

“Tracking and Preventing Diseases with Artificial Intelligence ” to be published by Springer Book Chapters - Link

Main Highlights
-Book series: Intelligent Systems Reference Library
-The books of this series are submitted to **ISI Web of Science, SCOPUS, DBLP and Springerlink**
-No publication fees

Important dates
-Chapter Submission Deadline: **15th October 2020**
-Acceptance Notification: 15th December 2020
-Camera Ready Submission: 31st December 2020

Achievements

- Classification of LUAD, LUSC, Normal with high accuracy and recall
- Explainability using CAM technique
- Localization at two levels
    - Show specific tile from whole slide image
    - Show specific regions within a tile the regions of malignancy
- Model deployed as service with report generation
- Patient Report generated in 1-2minutes (depends on size of whole slide) significant improvement as a pathologists takes upto a week to confirm

Q&A

Thank You

Appendix

References

1. YurRef Yu, Kun-Hsing et al. “Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features.” Nature communications vol. 7 12474. 16 Aug. 2016, doi:10.1038/ncomms12474 .
2. EstevaRef Esteva, A., Kuprel, B., Novoa, R. et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature 542, 115–118 (2017). https://doi.org/10.1038/nature21056
3. https://www.cancer.org/cancer/lung-cancer/about/ key-statisti cs.html
4. Classification and Mutation Prediction from Non-Small Cell Lung Cancer Histopathology Images using Deep Learning
5. Nicolas Coudray, Andre L. Moreira, Theodore Sakellaropoulos, David Fenyö, Narges Razavian, Aristotelis TsirigosbioRxiv 197574; doi: https://doi.org/10.1101/197574
6. Yu et al Yu, K., Zhang, C., Berry, G. et al. Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. Nat Commun 7, 12474 (2016). https://doi.org/10.1038/ncomms12474
7. AIHealthCare Esteva, A., Robicquet, A., Ramsundar, B. et al. A guide to deep learning in healthcare. Nat Med 25, 24–29 (2019). https://doi.org/10.1038/s41591- 018-0316-z
8. EstevaRef Esteva, A., Kuprel, B., Novoa, R. et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature 542, 115–118 (2017). https://doi.org/10.1038/nature21056
9. MednikovRef Y. Mednikov, S. Nehemia, B. Zheng, O. Benzaquen and D. Lederman, ”Transfer Representation Learning using Inception-V3 for the Detection of Masses in Mammography,” 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, 2018, pp. 2587-2590, doi: 10.1109/EMBC.2018.8512750.

References

10. WangRef C. Wang et al., ”Pulmonary Image Classification Based on Inceptionv3 Transfer Learning Model,” in IEEE Access, vol. 7, pp. 146533-146541, 2019, doi: 10.1109/ACCESS.2019.2946000.
11. GuanRef Guan, Qing et al. “Deep convolutional neural network Inception-v3 model for differential diagnosing of lymph node in cytological images: a pilot study.” Annals of translational medicine vol. 7,14 (2019): 307. doi:10.21037/atm.2019. 06.29
12. JinRef J. Li, P. Wang, Y. Li, Y. Zhou, X. Liu and K. Luan, ”Transfer Learning of Pre- Trained Inception-V3 Model for Colorectal Cancer Lymph Node Metastasis Classification,” 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, 2018, pp. 1650-1654, doi: 10.1109/ICMA.2018.84 84405.
13. YimRef Ding Y, Sohn JH, Kawczynski MG et al. A Deep Learning Model to Predict a Diagnosis of Alzheimer Disease by Using 18F-FDG PET of the Brain. Radiology 2019;290(2):456–464, doi: 10.1148/radiol.2020192224 12. GulshanRef Gulshan V, Peng L, Coram M, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA. 2016;316(22):2402–2410. doi:10.1001/jama.2016.1 7216