Deep Learning based Lung Cancer Detection

Aman Rana[1], Carlos W. Morato[2]

Abstract— **Early detection is the key to treating Lung Cancer, which accounts for 17% of the deaths caused due to cancer. Early detection can increase the survival rate drastically. Computer Aided Diagnostics (CAD), primarily based on image processing, help the doctors in providing a pre-screening and a secondary opinion resulting in expedited diagnosis. In the recent years, Deep Learning based solutions have gained popularity for several tasks. In this paper, an attempt has been made to develop an automated Deep Learning based solution that pre-processes the data, trains two DNN models and predicts the probability of lung cancer.**

I. INTRODUCTION

In a healthy body, normal cells are replaced by other healthy cells. Occasionally, abnormal cells in the body begin to develop and grow. If your body recognizes these cells as "abnormal," the body's defense mechanisms may kick into action to destroy the abnormal cells. In the case of cancer, your body sees these abnormal cells as part of your body, so it does not attack them and as a result, the cells begin to grow out of control. [1]

In a cancer cell, the DNA is damaged and is reproduced in other abnormal cells. In most types of cancer, these abnormal cells begin to stick together and form tumors, which in turn can be classified as Benign (non-cancerous) or Malignant (cancerous). [1]

There are many causes of lung cancer like carcinogens, environmental factors like smoker pollutants, radon gas, heavy metals, genetic mutation etc. Any one of them can secretly be the reason for lung cancer. Lung cancer is the leading cause of cancer deaths in the world. [1]

Lung cancer is hard to detect in early stages, but early discovery can lead to a jump in survival rate. Lung cancer occurs in the form of a tiny lesion (nodule) and is the size of a dime. By the time is is detected, it is already very late for the patient. A lot of experience and time has to be spent detecting, labeling and categorizing the potential lesions, which can be very inefficient if the number of diagnosis to perform is very large.

Computer Aided Diagnosis (CAD) solutions have come up in the recent past to aid the doctors in the form of a pre-screening step. The CAD system is able to detect possible lesions (nodules) and classify between benign and malignant based on the features captured for each lesion. This has helped to improve the success rate of early lung cancer detection. [2]

In the recent years, the availability of cheap hardware and the advancement of GPUs has ushered an era of Deep Learning based solutions that were previously impossible. Supervised Deep Learning solutions involve training a deep learning network via a huge training dataset. The learning occurs via change in internal weight values. After the network has been trained, it can be used to predict the outcome for inputs it has not seen before. Deep Learning networks can be used for classification, localization and segmentation. [3] [4]

This paper used supervised Deep Learning architectures to predict whether a patient has cancer or not by training a deep neural network.

II. RELATED WORK

Traditionally, the only available method to detect lung cancer was for trained doctors to manually find cancer nodules in lung scans. This process was slow and time consuming.

In recent years, Computer Aided Detection (CAD) based systems have been developed for lung segmentation and classification of nodule as benign and malignant. The goal of CAD systems is to improve the accuracy of radiologists with a reduction of time in the interpretation of images. CAD systems are classified into two groups: Computer-Aided Detection (CADe) systems and Computer-Aided Diagnosis (CADx) systems. CADe are systems geared for the location of lesions in medical images. Moreover, CADx systems perform the characterization of the lesions, for example, the distinction between benign and malignant tumors [5]. The CAD system includes segmenting the area of interest which is the lung, followed by analysis of area obtained for nodule detection. The last step is elimination of false positives. The benign tumor's area and perimeter value are large as compared to malignant tumor. But the irregularity index for benign is more important. Irregularity index gives an idea about the irregularity of the nodule boundary. Malignant tumor is more irregular than a benign tumor. A malignant tumor nodule has greater diameter than a benign one. Other factors like solidity value, entropy value, mean, variance, standard deviation, equivalent diameter are also taken into consideration in the CAD based detection system [6].

[7] proposes a method of creating a feature vector for every detected nodule and feeding that vector into an ANN or other classification technique for classification.

[8] gives a simplified description of the lung cancer detection system. The first stage is image enhancement to get better clarity from image (using Gabor kernel or FFT), image segmentation to segment lung from non-lung tissue (using

[1] Aman Rana is with Robotics Engineering Program, Worcester Polytechnic Institute arana@wpi.edu

[2] Carlos W. Morato is a Assistant professor with the Robotics Engineering Program at WPI cwmorto@wpi.edu

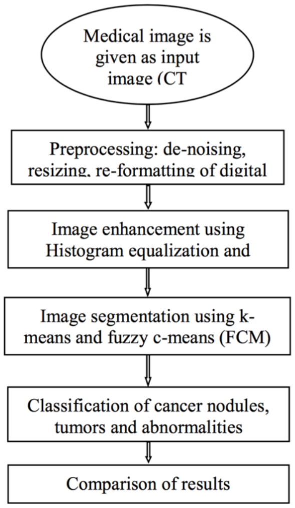

Medical image is given as input image (CT
Preprocessing: de-noising, resizing, re-formatting of digital
Image enhancement using Histogram equalization and
Image segmentation using k-means and fuzzy c-means (FCM)
Classification of cancer nodules, tumors and abnormalities
Comparison of results

Fig. 1. Typical flow diagram of a CAD based detection system

some type of thresholding technique) and finally feature extraction to extract useful information about the lung. This extracted feature set is then used to detect cancer or any abnormality. Finally, the false positives are eliminated.

Figure 1 shows the general work flow using the CAD based solutions.

III. DATASET

There are two datasets used for this research project - *Kaggle Data Science Bowl 2017 dataset* and *LUNA 2016 Challenge Dataset*. Both datasets consist of low dose (chest) CT scan images from high-risk patients. Each scan can be considered a stack of 2D images (*slices*), with the number of images in the stack varying between 90 to 430. Each slice has a cross sectional area of 512 x 512 pixels and the value of each pixel varies between -2000 to 1300 according to the standard Hounsfield Scale (HU). [9]

A. LUNA 2016 Challenge Dataset

The **LU**ng **N**odule **A**nalysis (LUNA) challenge [10] focuses on large-scale evaluation of automatic nodule detection algorithms performed on the publicly available LIDC/IDRI dataset. The dataset consists of 888 low dose CT lung scans., which were collected during a two year annotation phase using 4 experienced radiologists. Each radiologist marked the nodules (>=*3mm*) and the non-nodules (< *3mm*). All nodules marked by at-least 3 radiologists was considered a cancerous nodule (*malign*) and other were marked as irrelevant findings. Along with the scans, two csv files have been provided - one which contains the nodule candidates (both the malign and benign candidates) and an annotation.csv file which contains only the malign nodules. The objective of the LUNA 2016 competition is to find the location of cancerous nodules and the probability that it is cancerous. The *candidates.csv* contains nodule candidate per line. Each line holds the scan name, X, Y, and Z position of each candidate in world coordinates, and the corresponding class (cancer/non-cancer) value. There are over 50,000 candidates in the csv file.

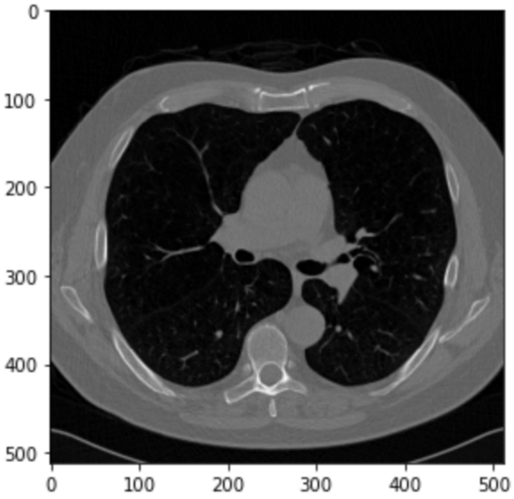


Fig. 2. Cross-section of a lung CT scan

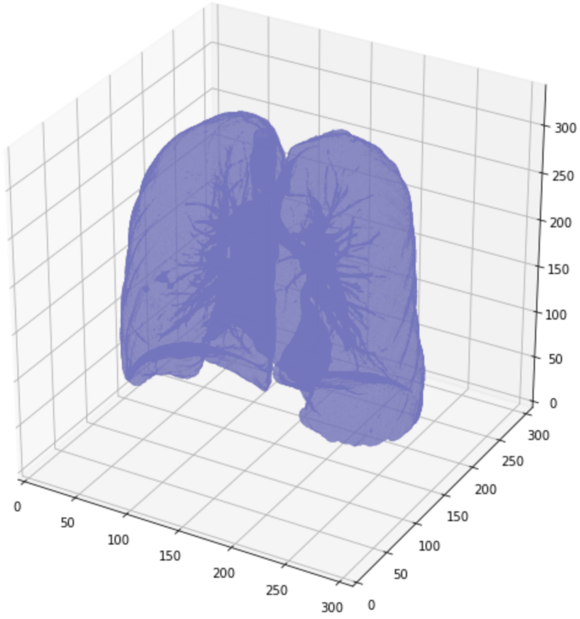


Fig. 3. 3D Scan of a patient

B. Kaggle Data Science Bowl 2017 Dataset

The Kaggle Data Science Bowl [11] dataset consists of more than a thousand scans from various high-risk patients along with their labels - cancer or non-cancer. The dataset was released in two stages. Stage 1 dataset consisted of **ABC** scans while the stage 2 dataset consisted of **ABC** scans.

A cross-sectional images from the scans can be seen in figure 2. A 3D scan of a patient can be seen in figure 3

IV. METHODOLOGY

The small size of the kaggle dataset (less than 2000 samples) proved to be a major hurdle in training any suitable

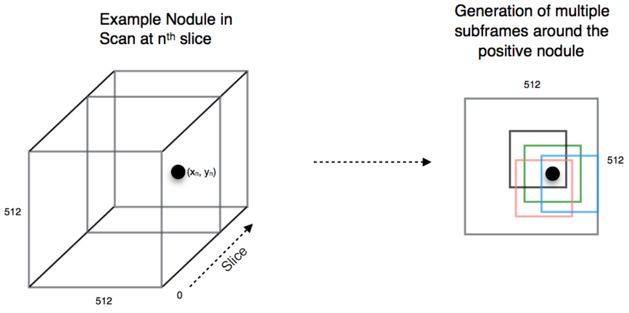

Example Nodule in Scan at n^th slice
512
512
0
Slice
(x_n, y_n)
Generation of multiple subframes around the positive nodule
512
512

Fig. 4. Creation of multiple positive sample from one cancer nodule using random 3D volumes around the nodule (*with the nodule always inside the 3D volume*). Shown in 2D for simplicity. The actual box is a 3D volume space.

Deep model. For this reason, the LUNA dataset [10], along with the included *candidates.csv* file were used to create a processed dataset derived from the LUNA dataset.

The initial step was to segment the lungs from non-lung tissue in the scan. V. This was followed by processing the LUNA dataset by cropping small 3D volumes from the scans around the candidates; as explained below.

The candidates.csv file contained valuable information about possible candidates locations (candidates) in the scans including x, y, and z positions along with the label for the candidate in the LUNA dataset. This spacial position was used to create small 3-dimensional volumes around the candidates and store them on disk, including an additional processing step. The number of cancer samples were far more than the non-cancer samples, leading to a huge class imbalance in the dataset. To overcome this, a simple technique was adopted. If the candidate was labeled as cancerous, multiple small volumes (*3D crops*) were extracted containing the nodule, at random distances from the centers of the nodule and the 3D volume, in the xyz direction. This enabled the creation of multiple positive samples using just a single cancer nodule. This is explained graphically in 4.

After the generation of the processed LUNA dataset, a 3D ConvNet model was trained on the generated dataset. The architecture of the model is described in VI-B. The output of the model learns to predict the probability of a 3D volume space of size [10x64x64] containing cancer nodule. The trained model was stored on disk along with the trained weights.

The kaggle dataset was preprocessed as describes in V, followed by cropping 3D volumes of size [10x64x64] sequentially with a stride of [5, 32, 32] in the [zyx] directions respectively. This lead to the creation of multiple 3D volumes from a single scan. Each 3D volume was sent to the trained model (explained above) for prediction of cancer and this value is stored in an empty 3D array for every scan, placed according to the cropped position. These 3D arrays were stored on disk to train a second 3D ConvNet model. The architecture of the second 3D ConvNet is described in VI-C. The output of the second model is the desired cancer prediction.

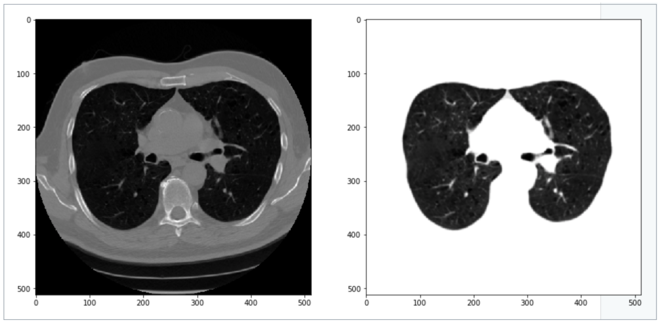


Fig. 5. Before and after lung segmentation step

The inference of test scans include the following steps:

1) Segmenting the lung tissue as described in V
2) Cropping 3D volumes of size [10x64x64] from scan sequentially with a stride of [5, 32, 32] in the [zyx] direction.
3) Sending a 3D volumes into the first 3D ConvNet for predicting the presence of cancer nodule.
4) Storing the results of all predictions in an empty 3D array
5) Predicting the cancer probability using the second 3D ConvNet model by feeding the 3D array.

V. LUNG SEGMENTATION

[12] describes general methodology and tips to identity Lung Cancer nodule. The cancer nodule is only found in the lung tissue and not in the surrounding tissues. We can thus treat the surrounding tissue as background and segment out the lung tissue. Lung segmentation involved the following steps:

- Applying Gaussian blur using a kernel of size [5x5].
- Applying a threshold on the scan [-300] and converting to uint8.
- Find contours in every slice using OpenCV.
- Filling the contours in an empty array (*mask*) using cv2.FillPoly() function
- Set the pixel values in the binary image to zero wherever the mask values are zero.
- Applying morphological operation like opening, closing, erosion and dilation on the resultant binary image.
- Set the pixel value in the original image to zero wherever the mask values are zero.

The above method successfully segments the lung tissue from the surrounding tissue. The results of lung segmentation can be seen in figure 5.

VI. DNN MODELS

A. *AutoEncoder + LSTM*

An autoencoder was used as one of the first models to predict cancer in the scans. The idea was to train an autoencoder to encode each slice of the scan in a 1D array, giving a 2D representation of the whole scan. Then, a LSTM would be trained to classify cancer in the scans. The lack of huge quantity needed to train an autoencoder was a factor in

	3D ConvNet Architecture for LUNA dataset
	Input [?x10x512x512x1]
Layer 1	Conv. Layer [kernel size=(3x3x3), stride=(1x1x1)]
	Conv. Layer [kernel size=(3x3x3), stride=(1x1x1)]
	ReLU
	Max Pooling [kernel size=(1x2x2), stride=(1x2x2)]
	Batch Normalization
Layer 2	Conv. Layer [kernel size=(3x3x3), stride=(1x1x1)]
	Conv. Layer [kernel size=(3x3x3), stride=(1x1x1)]
	ReLU
	Max Pooling [kernel size=(1x2x2), stride=(1x2x2)]
	Batch Normalization
Layer 3	Conv. Layer [kernel size=(3x3x3), stride=(1x1x1)]
	Conv. Layer [kernel size=(3x3x3), stride=(1x1x1)]
	ReLU
	Max Pooling [kernel size=(1x2x2), stride=(1x2x2)]
	Batch Normalization
	Fatten
	Fully Connected [n=1024]
	Fully Connected [n=256]
	Fully Connected [n=2]

TABLE I

3D CONVNET ARCHITECTURE TRAINED ON THE PROCESSED LUNA DATASET

	3D ConvNet Architecture for LUNA dataset
	Input [?x10x512x512x1]
Layer 1	Conv. Layer [kernel size=(3x3x3), stride=(1x1x1)]
	ReLU
	Max Pooling [kernel size=(1x2x2), stride=(1x2x2)]
	Batch Normalization
Layer 2	Conv. Layer [kernel size=(3x3x3), stride=(1x1x1)]
	ReLU
	Max Pooling [kernel size=(1x2x2), stride=(1x2x2)]
	Batch Normalization
	Fatten
	Fully Connected [n=1024]
	Fully Connected [n=64]
	Fully Connected [n=1]

TABLE II

3D CONVNET ARCHITECTURE TRAINED ON THE PROCESSED LUNA DATASET

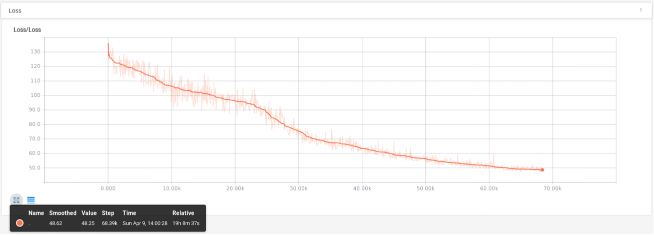
the failure in this approach. The architecture can be seen in figure 14

B. Model 1

The first DNN models is a 3D ConvNet with the architecture described in table I. The model tries to classify the input 3D volume between two classes - Cancer/Non-cancer. The input size is [Nx10x64x64x1] and the output size is [Nx2]. Adam Optimizer along with softmax_cross_entropy loss function has been used in the model. A custom generator has been written which makes a batch of N=128 samples at a time, with equal number of positive and negative samples.

C. Model 2

The second DNN models is also a 3D ConvNet. The architecture can be seen in table II. The input size is [Nx125x15x15x1] and the output is the probability of cancer, of size [Nx1] for a batch size of N=128. Adam Optimizer along with softmax_cross_entropy was used in the model.



	Name	Smoothed	Value	Step	Time	Relative
●	.	48.62	48.25	68.39k	Sun Apr 9, 14:00:28	19h 8m 37s

Fig. 6. Loss function plot for model 1

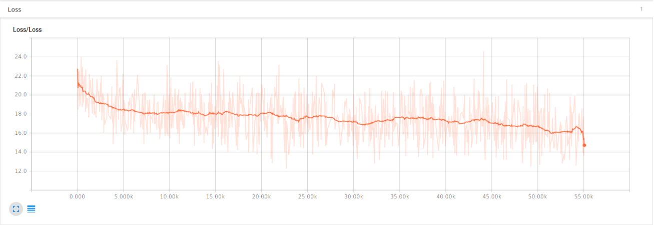


Fig. 7. Loss function plot for model 2

VII. EXPERIMENTS

The code was written in python with Tensorflow as the deep learning library with GPU support. The training was done in 20 epochs and each epoch had around 8000 iterations. Each epoch took a little over one hour.

One of the first models used was the modified U-Net model 16. The U-Net model was intended to perform segmentation of the nodules and locate them for further classification.

An AutoEncoder-LSTM model was developed as described in VI-A. But due to lack of dataset, the autoencoder model was not sufficiently trained.

The method described in IV was used next. The loss function plots of the first and the second 3D ConvNet respectively, can be seen in figure 6, figure 7. This architecture proved to be the most successful.

VIII. RESULTS

Model 1 was able to achieve an accuracy of 95.25% while Model 2 achieved an accuracy of 64%. After extensive research about the low accuracy level of the second model, it was found that the pixel values in LUNA dataset and Kaggle dataset were a little bit off. The first model was trained on the LUNA dataset but it predicted values for the Kaggle dataset which resulted in predictions that were not all correct. This led to bad training dataset being created for the second model.

The accuracy can be improved by finding a way to scale the scans in the two datasets so that the prediction of the first model are more accurate and thus improving the accuracy of the whole process. Another method to improve the accuracy would be to improve the preprocessing steps, to segment the possible candidates in the lung which will lead to much better accuracy.

IX. CONCLUSION

An architecture to detect Lung Cancer was developed, involving two 3D CNN models. Other DNN models like AutoEncoders and LSTM were also tested. Efforts were made to automate the process and to make the code so that it would become easy for the user to change the hyper-parameters and other variables. A shell file was created to run the whole pipeline sequentially. Despite the low accuracy of the second model, the endeavor can be treated as a good learning ground. There was a lot of experimentation about various DNN architectures (3D CNN, AutoEncoders, LSTM, Fully Convolutional Networks), learn Tensorflow and experiment with medical data. Overall the project can be treated as successful.

REFERENCES

[1] *ALCF Handbook 2nd Edition*, http://www.lungcancerfoundation.org/downloads/ALCF_Handbook_2nd_Edition.pdf.

[2] F. Li, H. Arimura, K. Suzuki, J. Shiraishi, Q. Li, H. Abe, R. Engelmann, S. Sone, H. MacMahon, and K. Doi, “Computer-aided detection of peripheral lung cancers missed at ct: Roc analyses without and with localization 1,” *Radiology*, vol. 237, no. 2, pp. 684–690, 2005.

[3] R. Gruetzemacher and A. Gupta, “Using deep learning for pulmonary nodule detection & diagnosis,” 2016.

[4] R. Anirudh, J. J. Thiagarajan, T. Bremer, and H. Kim, “Lung nodule detection using 3d convolutional neural networks trained on weakly labeled data,” in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2016, pp. 978 532–978 532.

[5] *Biomedical Engineering Online*, https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-015-0120-7.

[6] X.-Y. Jin, Y.-C. Zhang, and Q.-L. Jin, “Pulmonary nodule detection based on ct images using convolution neural network,” in *Computational Intelligence and Design (ISCID), 2016 9th International Symposium on*, vol. 1. IEEE, 2016, pp. 202–204.

[7] *CAD for Lung Cancer Detection: A Review*, http://www.ijmter.com/papers/volume-2/issue-7/cad-for-lung-cancer-detection-a-review.pdf.

[8] M. S. AL-TARAWNEH, “Lung cancer detection using image processing techniques,” *Leonardo Electronic Journal of Practices and Technologies*, vol. 20, pp. 147–58, 2012.

[9] *Hounsfield scale*, https://en.wikipedia.org/wiki/Hounsfield_scale.

[10] *LIDC-IDRI Dataset*, https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI.

[11] *Kaggle Data Science Bowl 2017*, https://www.kaggle.com/c/data-science-bowl-2017.

[12] *Lung Cancer Details*, https://www.kaggle.com/c/data-science-bowl-2017/discussion/27922.

APPENDIX

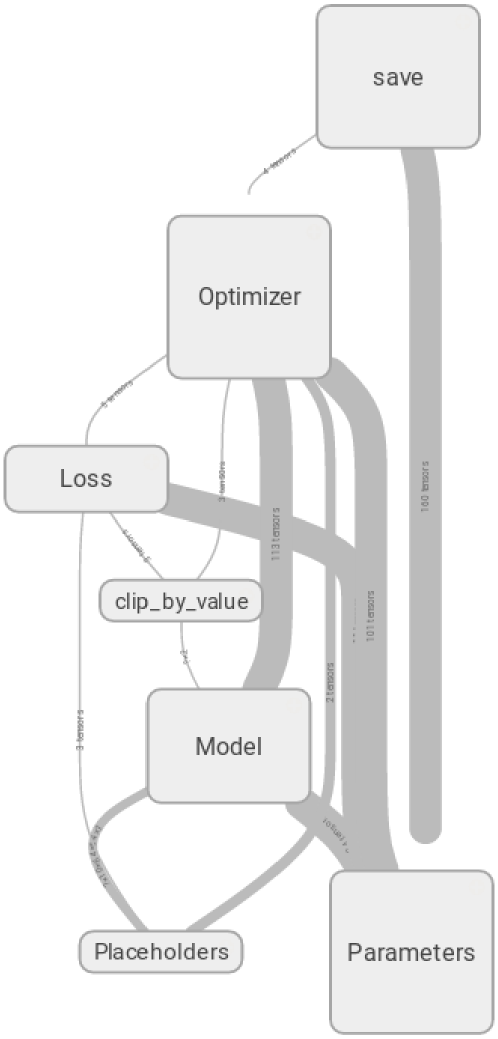


Fig. 8. Model 1 layout in Tensorflow

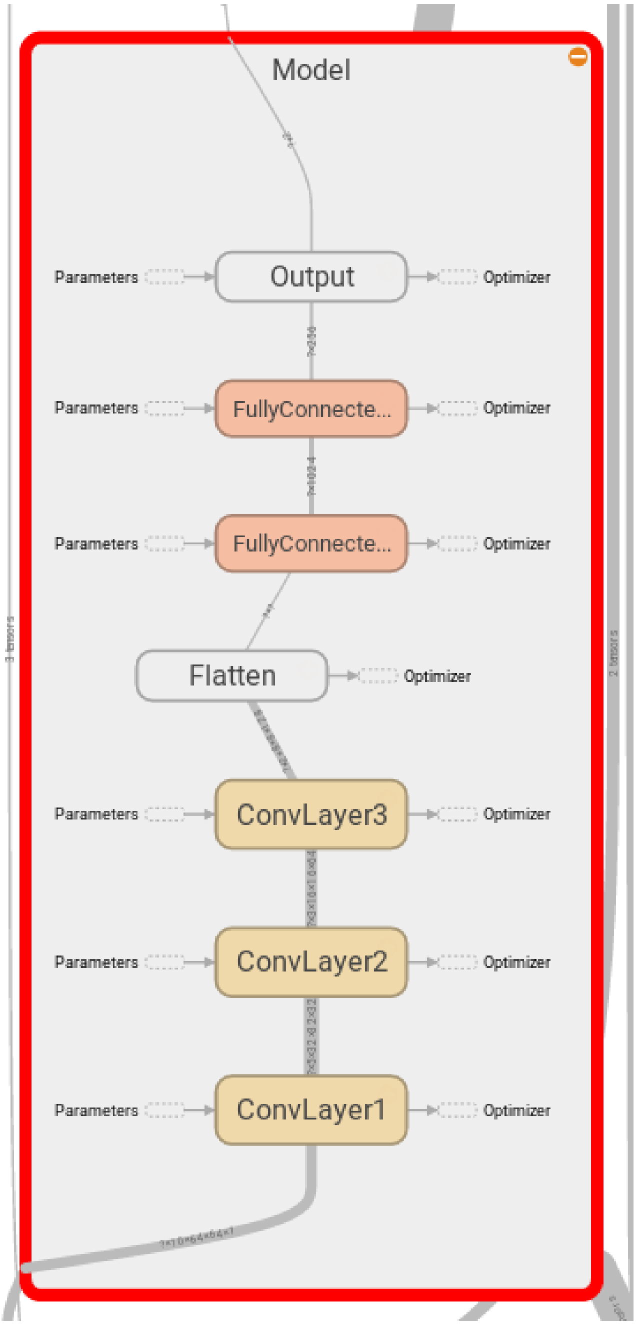


Fig. 9. Model 1 architecture (Tensorflow)

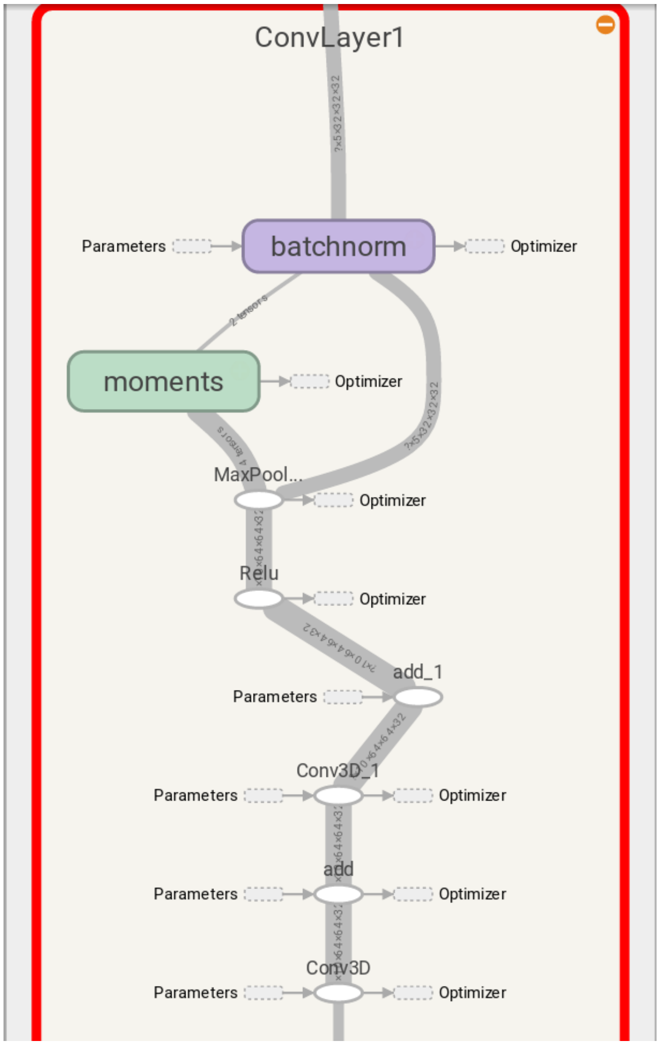


Fig. 10. Conv. Layer 1 architecture (Tensorflow)

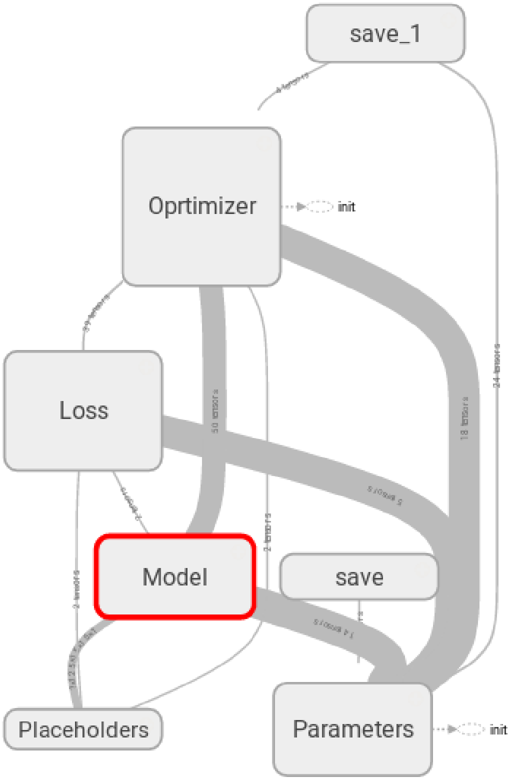


Fig. 11. Model 2 layout (Tensorflow)

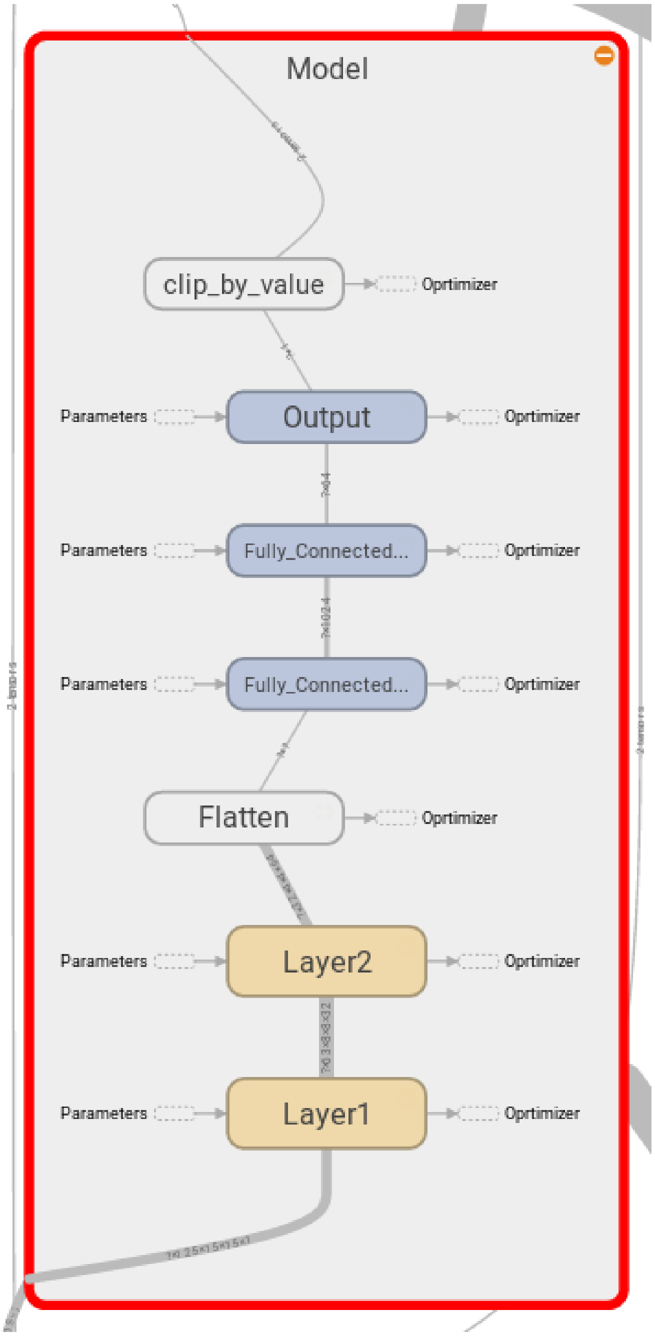


Fig. 12. Model 2 architecture (Tensorflow)

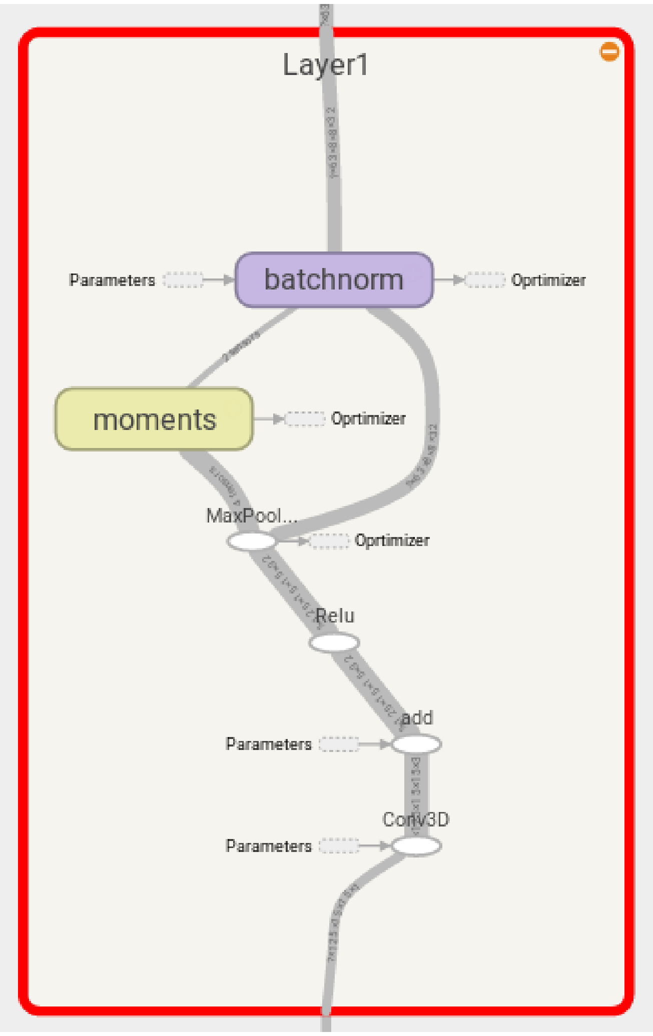


Fig. 13. Conv. Layer 1 architecture (Tensorflow)

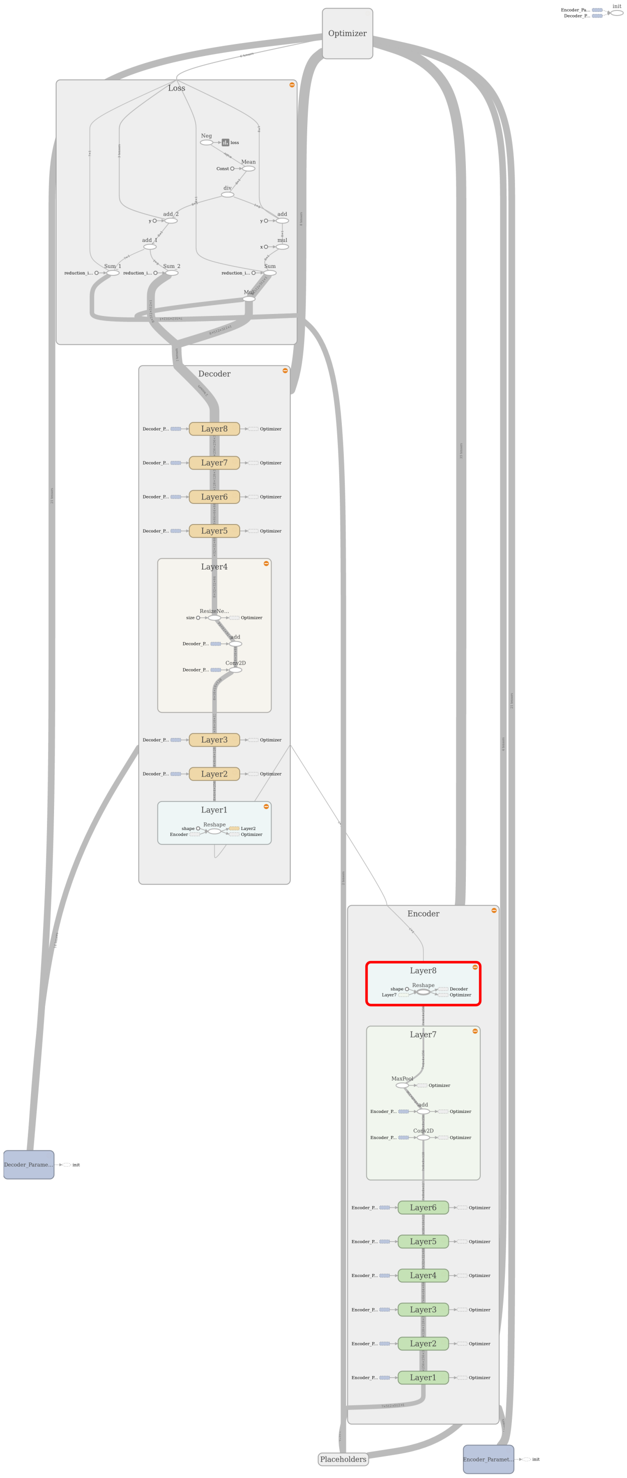


Fig. 14. AutoEncoder Architecture (Tensorflow)

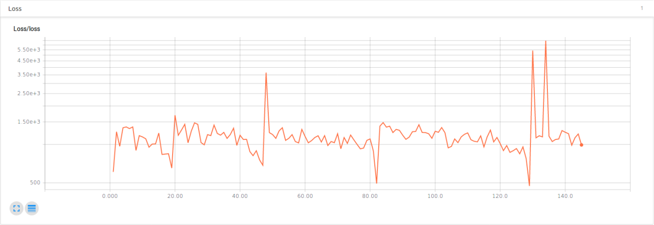


Fig. 15. AutoEncoder loss function plot (Tensorflow)

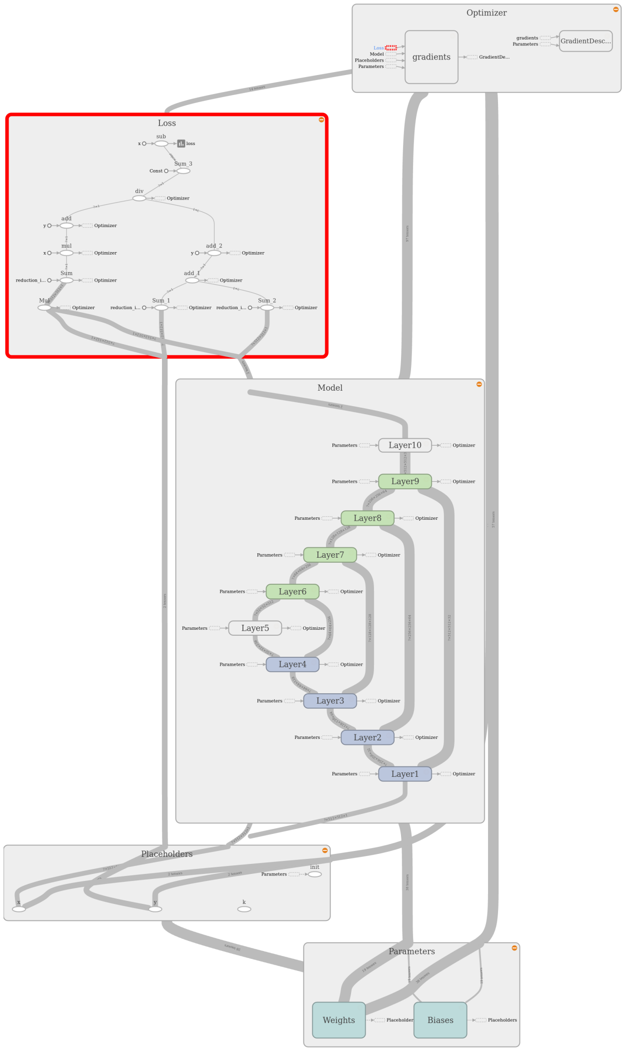


Fig. 16. Modified U-Net layout (Tensorflow)

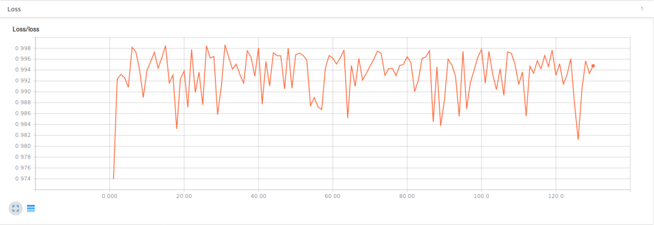


Fig. 17. Modified U-Net loss function graph (Tensorflow)