Comparative Analysis of various Deep Learning models applied on CTPA images to detect Pulmonary Embolism

Aditya Varshney[1]
B.Tech, CSE
Bennett University
Greater Noida, India
av1965@bennett.edu.in

Anshuman Agarwal[2]
B.Tech, CSE
Bennett University
Greater Noida, India
aa6493@bennett.edu.in

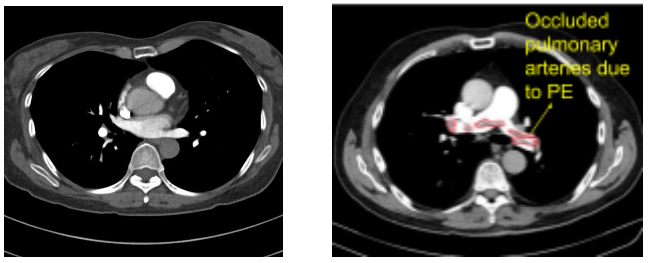
Arnav Bansal[3]
B.Tech, CSE
Bennett University
Greater Noida, India
ab6573@bennett.edu.in

***Abstract* — Pulmonary Embolism (PE) is an extremely life-threatening disease caused by a clot of blood in the lung artery or one or more branches of them. While it is a common disease, it is still hard to diagnose and providing an early diagnosis can improve the odds of survival of the patient drastically. This study aims to provide a comparative analysis of the various state of the art Deep Learning models currently used. We have tested the models on CTPA images in order to detect the presence of Pulmonary Embolism across various patients. This study encouraged the use of CAD systems for the detection of PE in CTA images. Out of the six different models deeply evaluated for this study, we have found the best performing model - XCeption achieved a sensitivity of 0.95 while maintaining an ROC-AUC of 0.819**

Keywords* — *Computer Aided Diagnosis (CAD), Computed Tomography (CT), Pulmonary Embolism (PE), Deep Learning

I. Introduction

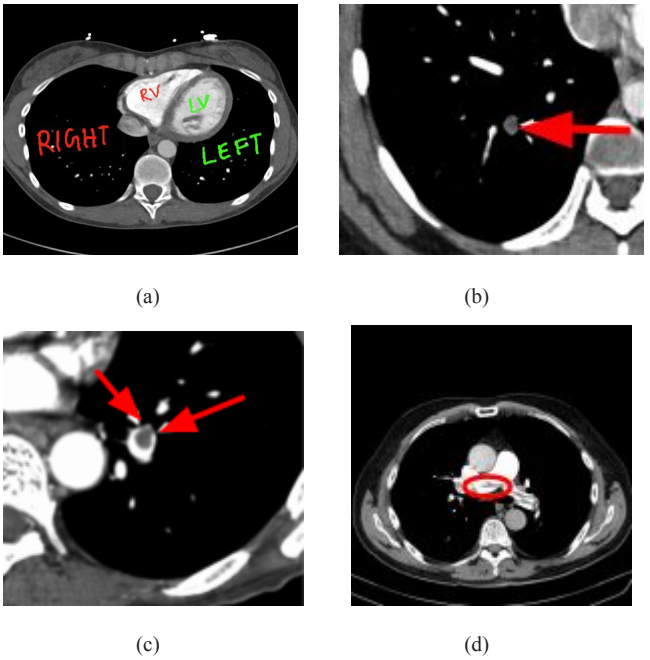
Pulmonary Embolism(PE) is a virulent and a life threatening cardiovascular disease caused by formation of blood clots in the pulmonary arteries present in the heart, inhibiting sufficient blood flow to the lungs [1]. When the right ventricle supplies oxygenated blood to the lungs, the blood clots present in pulmonary arteries hinders its flow and can even travel into the lungs. This leads to restricted blood flow, and hence damage to the lungs. This disease, if left untreated, could become fatal in an acute period of time. Prompt diagnosis and immediate medical attention is required to fight this life threatening situation. Pulmonary Embolisms can occur to people who have been inactive or bed-ridden for a long period of time due to lack of inactivity. Common symptoms of PE include problems breathing, chest pain, persistent cough, dizziness and frequent sweat across the body [2].



(a) (b)

Fig. 1. CT scan of lungs (a) with no sign of PE, (b) with PE annotated using red markers

Current practice mainly relies on a methodology called Computer Tomography Pulmonary Angiography (CTPA), where intravenous contrast is administered into the patient's body and a CT scan of the pulmonary blood vessels are captured by the radiologists [3-5]. Pulmonary embolism are classified either according to their location (Right PE ie. PE present in the right side of the heart, Left PE ie. PE present in the Left side of the heart and Central PE ie. PE present around the center of the heart) or based on their time of occurrence (Acute ie. PE that has occured in a short period of time or Chronic ie. PE that has formed over a longer duration of time) [6].



(a) (b)

(c) (d)

Fig. 2. CT scan of lungs (a) with all major parts of lungs labelled including right and left ventricle, (b) with Right PE, (c) with Left PE, (d) with Central PE

Not All PE's show symptoms, and Acute PE are said to be more prominent and fatal due to the fact that the body hasn't had sufficient time to adapt to the changes taking place in the body. An estimated 630,000 cases [7] and 100,000 - 200,000 deaths every year are said to be caused by this life threatening situation [8]. This year the Radiological Society of North America hosted a challenge on Kaggle which required the competitors to apply novel Computer Vision Methods to accurately classify whether a patient was infected with Pulmonary Embolism, and to also

detect the location along with the type of the disease. (The dataset consisted of 9 different classes that helped describe the exact type and location of the disease in the given images) [9].

To solve the issue of accurately detecting PE and in order to build computer models that can assist doctors, we aim to apply Deep Learning Methods and Techniques to automate the process of detecting a PE from a CTPA scan to help reduce the morbidity rate of this fatal disease. In this study, we have outlined our findings after performing a detailed comparison between some of the various commonly known Deep Learning models after training and inference on the provided dataset.

TABLE I

OUTLINE OF THE DATASET FIELDS: THE DATASET IS COMPOSED INTO 17 COLUMNS, WITH EACH ROW DESCRIBING THE FEATURES OF AN IMAGE. NO NULL VALUES WERE PRESENT.

S.No	Column	Data Type
0	StudyInstanceUID	object
1	SeriesInstanceUID	object
2	SOPInstanceUID	object
3	pe_present_on_image	int64
4	negative_exam_for_pe	int64
5	qa_motion	int64
6	qa_contrast	int64
7	flow_artifact	int64
8	rv_lv_ratio_gte_1	int64
9	rv_lv_ratio_lt_1	int64
10	leftsided_pe	int64
11	chronic_pe	int64
12	true_filling_defect_not_pe	int64
13	rightsided_pe	int64
14	acute_and_chronic_pe	int64
15	central_pe	int64
16	indeterminate	int64

II. MATERIALS

In this study, the data has been taken from above mentioned Kaggle competition [9]. The images are grouped by the study and further by the series. Each series then has a collection of images, uniquely identifiable by their unique identifier known as the "SOPInstanceUID". This competition requires to predict a number of labels at both image and study levels. The label hierarchy of the dataset is defined in fig 3.

The images provided in this competition are DICOM (Digital Imaging and Communications in Medicine) images, which is an internationally standardized format related to the exchange, storage and communication of digital medical images. This format eases the transfer of medical scans significantly.

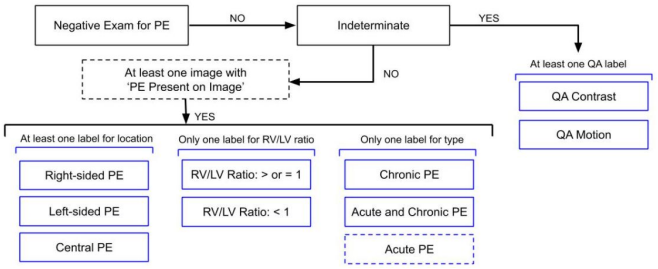


Fig. 3. Flowchart outlining the relationships between labels

CT-scan captures information about the radiodensity of an object or tissue exposed to x-rays. More the density of tissue, the more x-rays are absorbed. Current CT machines use 'Spiral CT', the information obtained from that is reconstructed to form a 3D volume which can be digitally sliced to obtain thinner slices as well as slices in different planes [10]. These 3D volumes are mapped to grey-level by manipulating the greyscale component of the CT image via the CT numbers, which specifically highlights particular structures hence changing the appearance of the overall image. This procedure is also known as Windowing. Windows are crucial in a radiologist's workflow. Just looking at the bare images provided in the raw CT-scans provides a good looking image for non-med people however, from a radiologist's point of view, that provides little information about what he is actually looking for [11].

While exploring the dataset, we experienced major class imbalance. As the data were essentially slices of a 3D scan, where it was scanned at the middle portion of the body (ie. the frames where the lungs were correctly aligned to the CTPA sensor) while taking the scan that had detected the PE. This consisted of a very small percentage of scans (5%) actually consisting of PE.

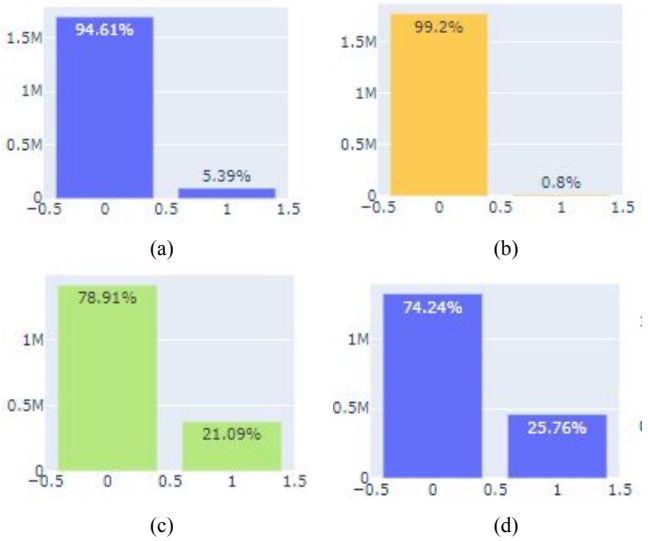


Fig. 4. Bar plots showing the class imbalance in (a) 'pe_present_on_image', (b) 'true_filling_defect_not_pe', (c) left_sided_pe, (d) 'right_sided_pe'

III. Methodology

This study aims to compare the performance of various state of the art deep learning models. As stated in the previous section the provided dataset was of DICOM images of the CTPA scans of various patients. The CTPA scans are 3D reconstructions of the lungs of each patient. We feed these scans to our Neural Network module by converting the DICOM images into RGB images and then slicing them into a stack of 2D images. This significantly reduces computation costs due to the effective reduction of a dimension.

A. Preprocessing

In order to preprocess the images and make them favourable for our Neural Network module, we started by removing all metadata that was tied up to the DICOM image files as they were irrelevant for the task at hand. Additionally, we used the VTK library [12], to convert the 2D DICOM slices into Numpy arrays which are very versatile and are much easier to manipulate while being more memory efficient. Furthermore, we resized all the RGB images to a uniform 512 x 512 dimension so as to provide a uniform input to all our models. We then applied sampling on our train set in order to induce randomness in the dataset and eliminate any sort of memorization by the model. We additionally separated 20% of our train data to be used as the validation set after the completion of the model training to generate our metrics, as Kaggle did not provide us with ground truth values of the test data.

B. Neural Network Module

This module contained our complete neural network designs. As the aim of this study is to establish a comparative analysis of the performance of various deep learning models, therefore we selected well known classification models such as MobileNet, XceptionNet, InceptionV3, ResNet-50, VGG-19 to extract features from the images. The Input Layer had the same dimensions as of the image i.e ((512,512,3)). Further, we experimented by freezing certain layers, adding batch normalization and dropout layers in order to boost performance. The default top layer was removed and replaced with the following set of layers: 1 Global Averaging Pooling, 1 Dropout layer with 0.25 dropout rate, 3 Dense Layers activated with the 'ReLu' activation function and then finally closed with the 9 nodes activated using the 'sigmoid' activation function, each representing the 9 output class values which are required to be predicted per image.

C. Training parameters

The major problem we faced while taking part in the kaggle challenge was the hardware limitation. Medical data, by nature is an extremely detailed and heavy set of data, making it extremely hard to process with generic computational resources. Each kaggle kernel assigned to us could run upto a maximum of 9 hours at a stretch providing each user with 12 GB RAM and only 42 hours of GPU training time per week on the Nvidia K80. For a dataset like this, which consisted of approximately 1 TB of data, these specification seemed to be rather incompetent.

In order to overcome this major constraint, the Learning rate was set to 10^{-3}. This was chosen due to the large dataset, as we didn't want our model to get stuck at a local minima for a long time. We then divided the entire dataset (~4,00,000 images) into batches of 1000 and ran 3 epochs each batch. Each time the Validation Loss decreased, we updated our model weights and saved them as a checkpoint. This way we were able to restrict our kernels from crashing when they exceeded the kernel limits and managed to save the best possible weights before it happened.

D. Models used for Comparison

To create a comparative study we needed to decide which models should be chosen for the same. We chose to apply a wide range of Deep Learning Neural Networks which have historically given very promising results on image classification tasks. We started out the study from light-weight models such as the MobileNet and slowly increased our scope to one of the very computationally demanding models - VGG19, along with other approaches such as XceptionNet, InceptionNet and ResNet50.

The sole reason for which MobileNetV2 was chosen was due to its extremely small and compact size which is obtained by using depthwise separable convolution as efficient building blocks. It has given exceptional results with the limited number of parameters that it trains.

Consequently we used the VGG-19 because it was developed with the sole purpose of classifying images across multiple classes and hence trains a high number of parameters. It is one of the most expensive models in our study in terms of computational costs.

Furthermore we used the ResNet-50 because it uses 'skip connections' which tend to carry forward information learnt by a previous layer to the next layer. This helps the model to learn complex patterns and allows it to create deeper neural networks. Because aur dataset consists of lung images which are similar to each other in terms of looks, using residual blocks seemed like an apt approach to tackle this problem.

The Inception Network consists of inception modules which tend to go 'wider' instead of 'deeper. Inception modules are basically a set of various sized filters through which our data is passed, a technique that helps in solving the problem of image localization. Using these inception modules, we don't need to worry about the size of the object we are trying to find in the input image, which was extremely helpful as the blood clots are actually small anomalies, which is what makes them hard to detect with a naked eye.

Finally, the XceptionNet was chosen because it has a modified depthwise separable convolution, which has historically performed even better than the Inception-v3 in classification tasks.

E. Feeding data to the Neural Network Module

Once our individual modules were complete, We divided our entire train dataset for validation and training purposes. 20% of the data was given to validation and the rest was kept for training purposes.The next task in hand was to assemble all the modules together. A simple for loop was used to call the image generator module which picked up the train set with the ground truth values.After this, the

train set was ingested into the neural network module in batches of 1000. Each batch of these images were run for 3 epochs each and each of these batches were further divided into 8 batches in order to avoid any sort of bottleneck with the memory.

F. Post Processing

While training the model, we saved the history of each training of the batch, this information was used to generate loss and accuracy curves for each of the labels. Once the training was complete, the generated outputs were to be post processed, this was achieved by creating a threshold barrier of 0.5. Values which were predicted to be greater than this threshold were rounded off to 1 and those with values less than 0.5 were rounded off to 0.

The Evaluation Metric for the RSNA-STRA Pulmonary Embolism Detection Challenge was the weighted log loss, which is formulated as follows:

$$L_{ij} = -w_j * [y_{ij} * \log(p_{ij}) + (1 - y_{ij}) * \log(1 - p_{ij})]$$

Once these metrics were obtained, the weighted average of each of the feature labels were taken in order to create the submission file for the competition. The weights for calculating the weighted average are as follows:

TABLE II

WEIGHTS FOR CALCULATING WEIGHTED AVERAGE

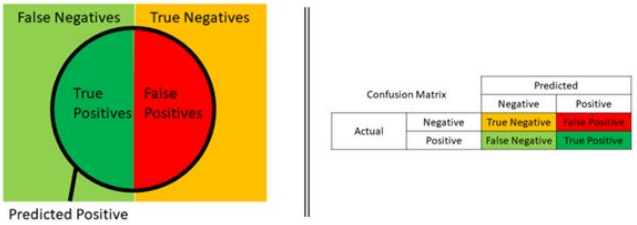
Label	Weight
Negative for PE	0.0736196319
Indeterminate	0.09202453988
Chronic	0.1042944785
Acute & Chronic	0.1042944785
Central PE	0.1877300613
Left PE	0.06257668712
Right PE	0.06257668712
RV/LV Ratio >= 1	0.2346625767
RV/LV Ratio < 1	0.0782208589

IV. EVALUATION METRICS

A. Confusion Matrix,

A confusion matrix is an N x N matrix that is used to understand the performance of the classification model on hand. This can only be performed on the set of test data for which the true values are known or given. Largely speaking, in binary classification, the predicted samples can be divided into 4 different categories-

→ True positive (TP) — actual = 1; predicted = 1
→ False positive (FP) — actual = 0; predicted = 1
→ False negative (FN) — actual = 1; predicted = 0
→ True negative (TN) — actual = 0; predicted = 0



B. Accuracy,

Accuracy, on the most common metric used for classification, is the fraction of samples predicted correctly.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

C. Precision Score,

Precision can be defined as the fraction of predicted positive events that are actually positive in nature.

$$\text{Precision} = \frac{TP}{TP+FP}$$

D. Recall,

Recall (also known as sensitivity or True Positive Rate) is the fraction of positive events that are predicted correctly.

$$\text{Recall} = \frac{TP}{TP+FN}$$

Precision and Recall when used individually are not as trustworthy, and hence our model can fool us easily by following the metrics mentioned above. We require a metric that can combine both 'precision' and 'recall' together and that metric is known as the "F1 Score".

E. F1 Score

F1 Score is the harmonic mean of recall and precision, and a higher score generally represents a better model.

$$\text{F1} = \frac{2}{\left(\frac{1}{precision}\right)+\left(\frac{1}{recall}\right)} = 2 * \frac{precision*recall}{precision+recall}$$

If either of the precision or recall values are low, the F1 score is immediately affected. Only when both of them are high, the F1 score achieved will be high.

F. ROC Score

- ROC curves are extremely helpful in understanding the balance between true-positive rate and false positive rates of a model.
- It is calculated using 3 different modules - thresholds, FPR and TPR.
- thresholds are all unique prediction probabilities generated by the model in descending order.
- FPR, or the false positive rate (FP / (FP + TN)) are the False Positives for each threshold
- TPR, similar to FPR are the true positive rate (TP / (TP + FN)) values for each threshold
- It tells how much a model is capable of distinguishing between classes.
- Higher the Area Under the Curve of ROC, better the model is at predicting 0s as 0s and 1s as 1s, ie. predicting True as True values and False as False values.

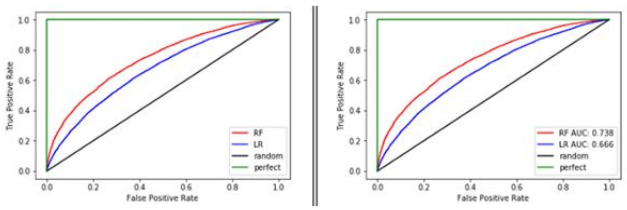


Fig5: The change in 'tpr' and 'fpr' can be plotted through the ROC curve and can be measured through ROC AUC Score.

Here, the green line represents the perfect prediction or the ground truth itself, red and blue are two possible threshold values for prediction, and black is just some random prediction. The quality of the curve is quantified by AUC Score, if the area under the curve is more, it is considered a better algorithm. So, here, we will choose the red predictor threshold as it is nearest to the ground truth and has maximum AUC Score

TABLE III

CALCULATED METRICS FOR PE_PRESENT_ON_IMAGE

Model Name	Accuracy	Precision	Recall (Sensitivity)	AUC Score	F-1 Score
MobileNetV2	0.91	0.95	0.96	0.612	0.95
VGG-19	0.92	0.96	0.95	0.749	0.96
InceptionNet	0.89	0.94	0.95	0.783	0.94
ResNet	0.92	0.96	0.95	0.710	0.95
Xception	0.91	0.95	0.95	0.819	0.95

TABLE IV

CALCULATED METRICS FOR RV_LV_RATIO_GTE_1

Model Name	Accuracy	Precision	Recall (Sensitivity)	AUC Score	F-1 Score
MobileNetV2	0.80	0.88	0.89	0.563	0.88
VGG-19	0.77	0.86	0.88	0.682	0.87
InceptionNet	0.79	0.89	0.87	0.727	0.88
ResNet	0.78	0.88	0.88	0.68075	0.88
Xception	0.77	0.87	0.87	0.751	0.87

TABLE V

CALCULATED METRICS FOR LEFTSIDED_PE

Model Name	Accuracy	Precision	Recall (Sensitivity)	AUC Score	F-1 Score
MobileNetV2	0.71	0.82	0.81	0.517	0.82
VGG-19	0.71	0.84	0.80	0.659	0.82
InceptionNet	0.67	0.81	0.79	0.672	0.80
ResNet	0.69	0.81	0.80	0.689	0.81
Xception	0.67	0.78	0.79	0.738	0.78

TABLE VI

CALCULATED METRICS FOR CHRONIC_PE

Model Name	Accuracy	Precision	Recall (Sensitivity)	AUC Score	F-1 Score
MobileNetV2	0.92	0.95	0.96	0.421	0.95
VGG-19	0.92	0.96	0.96	0.549	0.96
InceptionNet	0.92	0.95	0.96	0.596	0.96

ResNet	0.93	0.96	0.96	0.53	0.96
Xception	0.93	0.96	0.96	0.583	0.96

TABLE VII

CALCULATED METRICS FOR RIGHTSIDED_PE

Model Name	Accuracy	Precision	Recall (Sensitivity)	AUC Score	F-1 Score
MobileNetV2	0.69	0.82	0.78	0.57	0.8
VGG-19	0.68	0.82	0.77	0.693	0.79
InceptionNet	0.61	0.72	0.75	0.712	0.73
ResNet	0.66	0.79	0.76	0.684	0.77
Xception	0.65	0.79	0.75	0.763	0.77

TABLE VIII

CALCULATED METRICS FOR ACUTE_AND_CHRONIC_PE

Model Name	Accuracy	Precision	Recall (Sensitivity)	AUC Score	F-1 Score
MobileNetV2	0.96	0.98	0.98	0.521	0.98
VGG-19	0.96	0.98	0.98	0.749	0.98
InceptionNet	0.96	0.98	0.98	0.775	0.98
ResNet	0.96	0.98	0.98	0.709	0.98
Xception	0.96	0.98	0.98	0.791	0.98

TABLE IX

CALCULATED METRICS FOR CENTRAL_PE

Model Name	Accuracy	Precision	Recall (Sensitivity)	AUC Score	F-1 Score
MobileNetV2	0.91	0.95	0.95	0.517	0.95
VGG-19	0.90	0.94	0.95	0.623	0.95
InceptionNet	0.90	0.94	0.94	0.632	0.94
ResNet	0.90	0.94	0.95	0.616	0.95
Xception	0.90	0.94	0.94	0.691	0.95

TABLE X

CALCULATED METRICS FOR INDETERMINATE

Model Name	Accuracy	Precision	Recall (Sensitivity)	AUC Score	F-1 Score
MobileNetV2	0.96	0.98	0.98	0.511	0.98
VGG-19	0.96	0.98	0.98	0.673	0.98
InceptionNet	0.96	0.98	0.98	0.691	0.98
ResNet	0.96	0.98	0.98	0.652	0.98
Xception	0.96	0.98	0.98	0.715	0.98

V. RESULTS AND DISCUSSION

As this study is aimed to be a comparative analysis, We now try to determine the results obtained by our models and generate an intuition as to why a certain model is preferred in comparison to each other. To achieve this task, we compared each of these models by a number of metrics, including their Accuracy, Precision, Recall, AUC and F1 Score for each of the output labels as a metric for comparing the performance of all our Neural Networks.

Alongside, we also noted the loss obtained by each of the models during training in order to create a high level intuition of which model generated the best results and get a pre-sense of how they may perform with our test data.

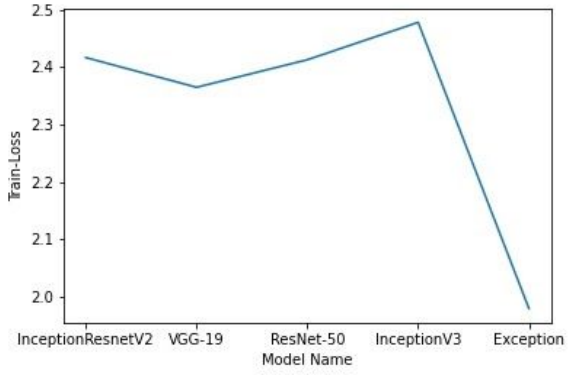


Fig 6 (a)

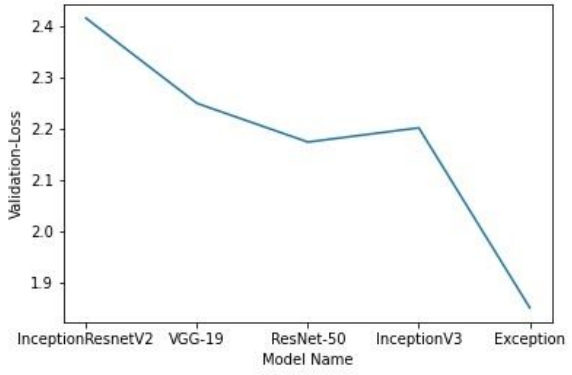


Fig 6 (b)

Fig. 6. Log Loss of each of the model during (a) Train time (b) Validation

The results procured in Fig 6 tell us that all the models have performed very similar to each other except for the InceptionResNet and the Xception. The InceptionResnet has obtained the highest loss during both train time as well as validation time. This might have happened due to the large amount of layers present in this model. Xception shows an immediate drop in the train_loss and val_loss, and this sharp decline definitely suggests that this model would perform much better than the rest.

While submitting these models for evaluation purposes in the competition, the Xception Network also provided us with the lowest Public score (0.503) amongst all the other models. The same has been evident through tables III to X. Smaller models such as the MobileNet give high Accuracy and Precision Values but fail to give robust AUC and F1 scores. This might be due to the class imbalance or a case of overfitting in these small models. Image level predictions had the best results in terms of the AUC score, which might be due to the reason that models such as the InceptionNet and the XceptionNet are used as feature extractors. The depthwise separable convolutions present in the Xception Network tend to work better in feature extraction tasks as compared to the regular Inception modules, which has been made apparent by their performance among the different classes as well.

VI. ACKNOWLEDGMENT

We would like to extend our thanks to Dr. Vipul Kumar Mishra for his constant guidance during this behemoth task, along with the Department of Computer Science of Bennett University for giving us the right motivation and resources to accomplish writing this study. Without their constant support, it would have been impossible to finish this.

VII. REFERENCES

[1] S. Z. Goldhaber and R. B. Morrison, “Pulmonary embolism and deep vein thrombosis,” Circulation, vol. 106, pp. 1436–1438, 2002.

[2] D. Manganelli, A. Palla, V. Donnamaria, C. Giuntini, “Clinical features of pulmonary embolism: doubts and certainties” Chest; 107 (1 suppl): pp. 22-32, 1995.

[3] M. Domingo, L. Martí-Bonmatí, R. Dosdá, and Y. Pallardó, “Interobserver agreement in the diagnosis of pulmonary embolism with helical CT,” Eur. J. Radiol., vol. 34, pp. 136–140, 2000.

[4] Perrier A, Bounameaux H, Morabia A, et al. Diagnosis of pulmonary embolism by a decision analysis-based strategy including clinical probability, D-dimer levels, and ultrasonography: a management study. Arch Intern Med 1996; 156:531-536.

[5] Heijboer H, B#{226}uller HR, Lensing AW, Turpie AG, Colly LP, ten CateJW. A comparison of real-time compression ultrasonography with impedance plethysmography for the diagnosis of deep-vein thrombosis in symptomatic outpatients. N EnglJ Med 1993; 329:1365-1369.

[6] Jan Bělohlávek, A., 2020. *Pulmonary Embolism, Part I: Epidemiology, Risk Factors And Risk Stratification, Pathophysiology, Clinical Presentation, Diagnosis And Nonthrombotic Pulmonary Embolism*. [online] PubMed Central (PMC). Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3718593

[7] F. A. Anderson, H. B. Wheeler, R. J. Goldberg, D. W. Hosmer, N. A. Patwardhan, B. Jovanovic, A. Forcier, and J. E. Dalen, “A population based perspective of the hospital incidence and case-fatality rates of deep vein thrombosis and pulmonary embolism: The worcester DVT study,” Arch. Intern. Med., vol. 151, pp. 933–938, 1991.

[8] Lee LC, Shah K. Clinical manifestation of pulmonary embolism. Emerg Med Clin N Am 2001;19:925-42.

[9] “RSNA STR Pulmonary Embolism Detection | Kaggle”, Classify Pulmonary Embolism cases in chest CT scans, Radiology Society of North America, Available at: https://www.kaggle.com/c/rsna-str-pulmonary-embolism-detection

[10] Y. Masutani, H. MacMahon, and K. Doi, “Computerized detection of pulmonary embolism in spiral CT angiography based on volumetric image analysis,” IEEE Trans. Med. Imag., vol. 21, no. 12, pp. 1517–1523, Dec. 2002.

[11] H. P. Chan, L. Hadjiiski, C. Zhou, and B. Sahiner, “Computer-aided diagnosis of lung cancer and pulmonary embolism in computed tomography— A review,” Acad. Radiol., vol. 15, pp. 535–555, 2008.

[12] "vtk · PyPI", an open-source toolkit for 3D computer graphics, image processing, and visualization, Available at: https://pypi.org/project/vtk/

[13] S. Ukil and J. M. Reinhardt, "Smoothing lung segmentation surfaces in 3-D X-ray CT images using anatomic guidance," in Proc. SPIE Med. Imag., 2004, vol. 5370, pp. 1066–1075.

[14] K. Janocha and W. M. Czarnecki. "On loss functions for deep neural networks in classification". Schedae Informaticae, 25:49–59, 2016.

[15] Ozkan, H., Osman, O. and Sahin, S., 2013. Computer aided detection of pulmonary embolism in Computed Tomography Angiography images. *2013 International Conference on Electronics, Computer and Computation (ICECCO)*