Automated identification and transcription of chest pathology using CNN with Natural Language understanding

Neville Gai
ngai1@stanford.edu

Abstract

The work presented here deals with the topic of transcribing medical images in an automated fashion. In particular, a dataset of chest X-ray images with available radiological impressions was used to train two models, one based on RNN with GRU and another based on Transformer model. Image feature extraction was performed in two different ways – first using an available pretrained model and a second using training from hand-labeled multi-labels of the dataset under consideration. Comparisons were carried out using BLEU scores and it is shown that the transformer model with image features extracted from a trained model slightly outperformed the attention model. Mean BLEU scores around 0.4 indicate a reasonably good match between reference and predicted transcriptions, given a relatively small dataset. However, BLEU scores were not found to be reflective of the success of accurate transcription. Further improvements are possible by expanding the dataset and using a medical dictionary available for reference transcription along with better metrics to capture the accuracy of transcription.

1. Introduction

Chest radiography is the most performed diagnostic exam globally. The exam provides screening for several pathologies such as lung tumors, emphysema, pneumonia, and more recently, Covid-19. Typically, diagnosing chest pathology is a time-consuming task requiring considerable expertise by radiologists trained in thoracic imaging. Current state-of-the-art systems, including those based on artificial intelligence, cannot provide a substitute for a trained radiologist. Automatically, diagnosing chest images and transcribing them is an ambitious project with global impact. Such an application, if realized fully and accurately, can be deployed in remote or underserved areas where a trained radiologist may be hard to access.

Despite tremendous strides made in object detection and caption generation, application to medical diagnosis is a formidable challenge. This is mainly due to the open-ended image level anatomy and pathology labels provided by radiologists. This includes pathologies described using similar phrases as well as abstract and complex reasoning sentences than plain text. The second more pertinent problem is related to the size of the images (~1000x1000 or higher) and the relative extent of pathologies detected, which can range from a few voxels to several thousand pixels. Fully dense annotation of region-level bounding boxes normally needed for computer vision datasets remains non-viable for now. Another challenge is that ImageNet pre-trained deep CNN models usually serve as a good baseline for further model fine-tuning. However, this situation does not apply to the medical image diagnosis domain. Thus, we must learn the deep image recognition and localization models while constructing the weakly-labeled medical image database [1]. Under the circumstances, transcribing meaningful reads with high accuracy to datasets is still elusive.

In this work, different alternatives to handling this difficult problem are implemented and discussed. Although using a relatively smaller dataset, results are reasonably impressive while pointing to the correct approach to handling the problem. The image set available (discussed below) provided a common transcription for one or more images per patient. Feature extraction from the limited set of images was first carried out using a previously trained and available CheXnet model, which was then fed along with appropriately processed text to an RNN with GRU model as well as a transformer model. A second approach was to use the limited image set with hand-labeled multi-labels to train a DenseNet121 model and then extract image features from the trained model. In each case, transcriptions needed to be cleaned/parsed, tokenized, and embedded prior to feeding to the RNN or transformer model. A range of possible hyperparameters as well as two CNN models were considered for feature extraction. Output sequences of maximum fixed length were compared with reference transcriptions using BLEU scores. Steady improvement in scores was noticed from using pre-existing CNN + RNN to using trained CNN + transformer model. However, relatively higher Bleu scores do not necessarily reflect a better model, and other metrics need to be explored.

2. Related Work

Models such as ChexNet [2] and ChexPert [3] have achieved good classification accuracy and can serve as a basis for further exploration of human expert-like transcription of images. The problem was posed as a multi-class classification problem with 14 possible labels corresponding to different pathologies. The work achieved impressive results based on a large corpus of radiologist labeled images. However, NLP related transcription understanding and transcribing using a trained model was outside the scope of the work. Park *et al.* [4] used a Vision Transformer that utilizes low-level chest X-ray feature corpus obtained from a backbone network that extracts common CXR findings to model the severity quantification of Covid-19. Another work [5] used Vision Transformers for detecting tuberculosis on lateral chest X-ray images. Xue *et al.* [6] used CNN with LSTM in a multimodal model that combined encoding of the image and one generated sentence to construct an attention input to guide the generation of the next sentence, thus maintaining coherence. Chen *et al*. [7] generated radiology reports with memory-driven Transformer, where a relational memory is designed to record key information of the generation process and a memory-driven conditional layer normalization is applied to incorporating the memory into the decoder of Transformer. Zhang *et al.* [8] used a pre-constructed graph embedding module (modeled with a graph convolutional neural network) on multiple disease findings to assist the generation of reports. Liu *et al*. [9] performed domain-aware automatic chest X-ray radiology report generation system which first predicts what topics will be discussed in the report, then conditionally generates sentences corresponding to these topics. Chen *et al.* [10] used bi-directional mapping learning using RNN between images and their descriptions to automatically learn long-term visual concepts to aid in sentence generation given an image. Another recent work [11], applied a transformer model to chest X-ray data. While exact details on the methodology are not available, reported BLEU scores are below the ones achieved here. This is most likely related to the image feature extraction step which is more accurately captured by this work. Most of the works differed either in the dataset used or the technique employed and reported results although difficult to compare across different datasets, fell short of the scores achieved here.

3. Methods

The first part of the section deals with generation of image features fed as input to the encoder, while the second part deals with captioning related models. Implementation was carried out in TensorFlow 2.8.2 on Google Colab.

3.1 CNN for feature extraction

Each patient had an associated transcription, and each patient was associated with one or more images. The range for images per patient was 1 to 5, with frontal or lateral views (or both) present. Images associated with a transcription were first extracted from xml files. An example of a transcription xml file is given in Figure 1 (Appendix) with file names in bold. As can be seen, this patient had two associated images.

Preprocessing

Images were of different sizes with lateral images showing a more skewed aspect ratio. All images were resized to (224, 224, 3). Figure 2 (Appendix) shows the distribution of the iamges for 10 classes. This was a highly unbalanced data set with normal reads far outnumbering pathological reads. While this does reflect the clinical situation where normal X-rays are more likely to be a majority of cases, medical priorities dictate that pathological classes be given sufficient representation due to the importance of catching a disease state. Accordingly, images were resampled to increase representation of minority classes as seen in Figure 2 (Appendix). Post-resampling, data augmentation was done by randomly flipping images horizontally with probability of 0.5. Other augmentations were eschewed due to their unrealistic nature (vertical flipping) or after some preliminary analysis indicated adverse behavior (eg. distortion, zooming).

3.1.1 Pre-trained Model

For the first model, image feature extraction was performed by using a pre-trained network available from the CheXnet project. This model is based on a DenseNet121 architecture trained on 112,120 frontal-view x-ray images of 30,805 unique patients [1, 2]. The dataset was trained on 98637 images from 28744 patients, validated on 6351 images from 1672 patients and tested on 420 images from 389 patients, with no overlap between patients. The last few layers of the common model used for 3.1.1 and 3.1.2 are shown in Figure 1.

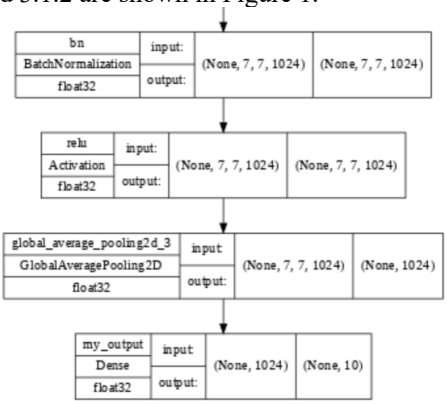


Figure 1: For both models, image features were extracted from the ReLU layer.

Note that the classification layer had 14 outputs for the pre-trained CheXnet model corresponding to 4 extra pathologies being present in that dataset. Image encodings were obtained with or without an optional 2D average pooling layer (not shown) which resulted in 3x3x1024 output encodings.

3.1.2 Training Model

Since the dataset used here differs from the CheXnet set in a couple of ways (only frontal views for CheXnet vs frontal and lateral views here, and 14 classes vs 10 here), training a separate model was an endeavor worth trying. Accordingly, two available models were explored - a ResNet50V2 model and a DenseNet121 model. Final classification layer used sigmoid activation similar to the CheXnet DenseNet121 model. After some preliminary training analysis, ResNet50V2 was abandoned in favor of the DenseNet121 model, especially since the DenseNet model could be initialized with weights from CheXnet model. The DenseNet121 model was then trained using a range of hyperparameter values, including learning rates and number of frozen layers. Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999. \epsilon = 10^{-7}$ was used. The best model was updated with each epoch.

Metrics used

Since this was a multi-class classification problem, F1-score was considered the best metric for accuracy since it reflects true positive (TP), false positive (FP) and false negative (FN) in its calculation. Two losses were considered for this accuracy metric: soft F1-loss and binary cross-entropy (which has also been employed for multi-class problems). Binary crossentropy loss showed behavior inconsistent with F1 accuracy while soft F1-loss was more consistent with F1 accuracy as could be suspected and was consequently used as the loss function. F1 soft loss is given by [12].

$$TP = \sum_i y_i \,.\, \hat{y_i} \quad TN = \sum_i (1 - y_i).(1 - \hat{y_i})$$

$$FP = \sum_i (1 - y_i).\hat{y_i} \quad FN = \sum_i y_i \,.\, (1 - \hat{y_i})$$

$$S = \frac{2 \times TP}{2 \times TP + FP + FN + \epsilon} \quad Loss = {}^{1}/_{L} \sum_l (1 - S_l)$$

where *i* refers to the batch index and *l* to the label index. F1 accuracy is given by $(\sum_l S_l)\,/L$.

3.2 Encoder-Decoder Models

Pre-processing

The following steps were taken to extract the impression field from the xml files:
1. Use regex to clean up text in xml files
2. Extract information for fields and store in dataframe.
3.Remove frame rows missing relevant information.
4. Add <start> and <end> tokens to impression string.
5. Tokenize words in the radiology impressions.
6. Restrict impressions to 90th percentile of all string lengths and pad if necessary.
7. Use Glove [13] with 42B tokens and embedding dimension 300 to create embedding matrix for words in vocabulary.

3.2.1 Attention Model

The model is shown graphically in Figure 2 and is based on [14, 15]. Note that input_2 goes to the decoder.

Encoder

Output from the CNN calculated image feature vector was reshaped and passed through a MLP layer of dimension 512. This was followed by batch normalization and a dropout layer (dropout_rate = 0.5, although other rates were tried).

Decoder

Output from the encoder was concatenated with the impression. At each decoder step, attention weights and context vectors were generated from encoder output and hidden state using $a(t) = softmax[v_a tanh(W_e E + W_h h(t)]$ and $c(t) = \sum_t a(t).E$, where W_e, W_h are weights corresponding to dense layers of dimension 512. The context vector was then concatenated with the embedded impression word at time *t* and passed through a GRU which provided the output word and the next hidden state. A schematic of the attention mechanism is provided in Figure 3 (Appendix).
Training was performed by using a custom learning rate scheduler typical of such models. Total trainable parameters for the model were 11,226,782 (Figure 4, Appendix). After initial warm-up steps, learning rate followed a cosine-based function (Figure 5, Appendix). SparseCategoricalCrossEntropy with a logical mask derived from the true impression was used to calculate loss.

3.2.2 Transformer Model

A standard transformer as described in [16] was implemented. Two-dimensional positional encoding was performed for the image vector while one dimensional encoding was done for the text sequence. Multi-head attention consisted of 8 heads.

Encoder

The encoder input was similar to the case of the attention model described above. The encoder is composed of a stack of N = 4 layers. All sublayers in the model as well as embedding layers produce output of dimension 512.

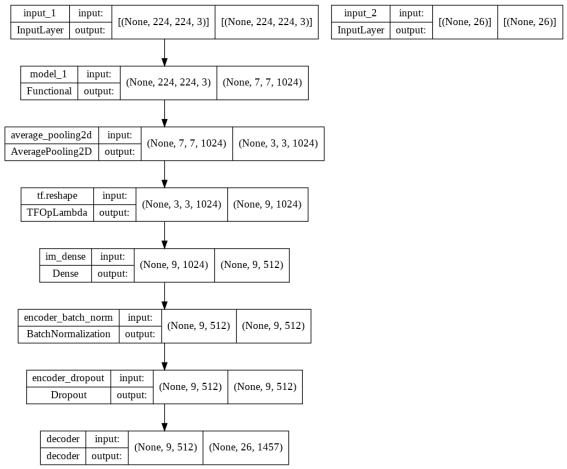


Figure 2: Layer graph of the first attention model.

Decoder

The decoder was also composed of stacks of N=4 identical layers. Other hyperparameters include inner-layer dimensionality for the feed-forward network of d_{ff}= 2048. Model was trained with the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999. \epsilon = 10^{-7}$. Total trainable parameters were 88,070,594 (Figure 6, Appendix), about 8 times larger than the attention model described in 3.2.1. Learning rate was set similar to [16] at lr = $d_{model}^{-0.5} \cdot$min(step_num$^{-0.5}$, step_num . warmup_steps$^{-1.5}$). Warm up steps set to 10000 gave better results than the default 4000 used in [16]. The final learning rate used is shown in Figure 7 (Appendix). Some of the other hyperparameters varied were the number of layers (4 and 6), number of heads (4 and 8) and the dropout rate (0.2 and 0.5).

3.3 BLEU score

To test the accuracy of the generated radiological report, the predicted and reference transcriptions were compared based on 1, 2, 3, and 4-gram BLEU scores. Both sequences were limited to a maximum of 33 words as referred to earlier based on the 90th percentile of the reference transcription lengths.

4. Dataset and Features

In this work, a smaller dataset [17] (the IU set) from the NIH website with detailed transcription available for each image in the dataset. The dataset consists of a total of 7471 images with 3955 reads [17]. Images are acquired in lateral and frontal views, and each read corresponds to one or more images in the data set. Labels for images for each patient were acquired (private communication with Dr. John Zech, Columbia U-Presbyterian Hospital) which allowed for training a model for image feature extraction. Note that this dataset has only 10 classes including normal reads, while the CheXnet set is based on 14 labels. Example of labels for the dataset is shown in Table 1.

pt_id	Cardiomegaly	Emphysema	Effusion	Hernia	Nodule	Atelectasis	Pneumonia	Edema	Consolidation
1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
5	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

Table 1: Nine pathological classes and a normal class resulting in 10 multi-label classifications.

The original image distribution corresponding to the 10 classes was [620. 114. 265. 84. 204. 589. 72. 86. 48. 5600.]. Data augmentation and resampling was done as described in the Methods section. After resampling, the distribution was [1367. 249. 612. 158. 444. 1279. 199. 167. 213. 4057.] Data were first split 85-15 between training and testing prior to resampling data. The remaining data were again split 85-15 between training and validation. Total training, validation and test examples were then 8041, 1420, and 1113 images with transcriptions. The dataset was augmented as described in the Methods section. The total vocabulary size after cleaning was 1476.

Schedule	Epochs	Layers frozen	Data	Learning rate	Training time	Validation accuracy
1	10	Conv base	Original	0.01*	-	0.1513
2	10	Conv base	Original + init.	0.01*	-	0.1582
3	20	Conv base	Aug. + init	0.01*	49 s/epoch	0.1905
4	20	Conv base	Aug. + init + flip	0.001*	49 s/epoch	0.2184
5	20	409 of 429	Aug. + init	0.001*	68 s/epoch	0.4562
6	20	409 of 429	Aug. + init + flip	0.001*	68 s/epoch	0.4273
7	100	409 of 429	Aug. + init + flip	0.001*	68 s/epoch	0.4387
8	20	None	Aug. + init + flip	0.001*	102 s/epoch	0.6832

*: Decay rate of 0.5/epoch.

5. Experiments, Results, Discussion

5.1 CNN Training Model

Loss and accuracy obtained with the ResNet50V2 and DenseNet121 model is shown in Figure 8 (Appendix). Both models behaved similarly without initialization with CheXnet weights. However, with CheXnet weights available as initial weights for the DenseNet121 model, it was easy to observe improvement in training and validation accuracy, and thus eliminate ResNet from contention. For the trained DenseNet121 model, several combinations of learning rate and frozen layers were tried out. Some of the results are shown in Figures 9-14 (Appendix) and captured in Table 2.

A learning rate of 0.001 with decay rate of 0.5/epoch provided smoother convergence. However, learning was slower, and although noisy, a fixed learning rate led to faster training. Since the best model weights were saved after each epoch, noisy performance while aesthetically less pleasing, proved more practical in this case.

Although schedule 5 showed higher validation accuracy than schedules 6 and 7, it performed relatively poorly on the test set. Final F1-score for schedule 8 was 0.5906.

5.2 Attention Model

Training and accuracy results are shown in Figures 15-16 (Appendix). Convergence was relatively quick with the CNN schedules considered from Table 2, with validation accuracy showing a slow decline with increasing epochs. BLEU scores on the test data for the first attention model using the pre-trained model, and trained model with schedules 7 and 8 are given in Table 3. When compared with scores derived using the pre-trained CheXnet model, an improvement was noticed using the trained models derived here. Generating scores took 2 hrs 23 min each for the three schedules.

	bleu1	bleu2	bleu3	bleu4
	0.207329	0.256134	0.328563	0.391791
	Bleu1	**Bleu2**	**Bleu3**	**Bleu4**
Test data	0.344757	0.347895	0.384872	0.432813
	Bleu1	**Bleu2**	**Bleu3**	**Bleu4**
Test data	0.380041	0.354422	0.383533	0.427885

Table 3: BLEU scores for the attention model and pre-trained CheXnet model (top), trained model (schedule 7, middle) and trained model (schedule 8, bottom).

Examples of the final predicted impression and the reference impression along with the corresponding images are shown in Figures 3 and 4. As is evident from Figure 4, the model learnt to express a sophisticated transcription although conflating “grossly clear left lung” with “grossly

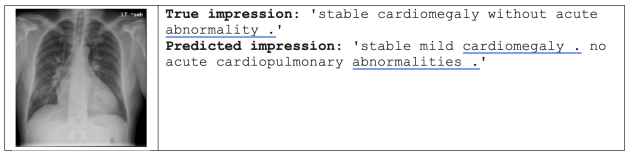


Figure 3: Example where true caption and predicted caption matched up well.

stable pleural effusion”. However, pleural effusion is seen in the right lung and prediction does describe a “partially

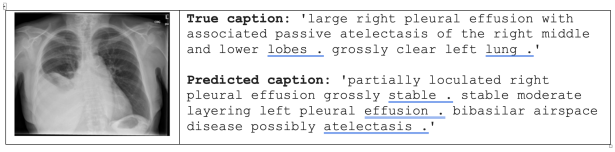


Figure 4: True and predicted captions showing good match.

loculated right pleural effusion” in addition to detecting “atelectasis”. This is quite remarkable considering that only 265 and 589 of the cases, respectively, belonged to these pathologies in the original data set.

5.3 Transformer Model

Training and accuracy over a range of hyperparameters for the Transformer model are shown in Figures 17-26 (Appendix). Compared to the attention model implemented above, the Transformer model proved difficult to train, despite a steep training and validation slope initially which petered out after around 30 epochs in each case. The validation accuracy plateaud between 0.35-0.4, regardless of the hyperparameters used to train the model. Minor differences were noticed in relation to the trained CNN model used, with schedule 8 providing the best results. BLEU scores for three cases are provided in Table 4.

	Bleu1	Bleu2	Bleu3	Bleu4
Test data	0.373848	0.348647	0.383988	0.418225
	Bleu1	**Bleu2**	**Bleu3**	**Bleu4**
Test data	0.368252	0.345675	0.38153	0.415511
	Bleu1	**Bleu2**	**Bleu3**	**Bleu4**
Test data	0.35629	0.353711	0.404249	0.440939

Table 4: BLEU scores for three cases of the trained Transformer model, correponding to training shown in Figures 21, 24 and 20 (Appendix).

Although the best Tranformer model showed better BLEU scores than the Attention model of 3.2.1, actual transcriptions were not as convincing as the Attention model generated ones, indicating that BLEU scores alone are not a surrogate for actual clinical accuracy. This could

be due to a multitude of reasons. For one, all BLEU scores reported here are in the 0.35-0.45 range, indicating only a modest agreement with the reference transcription. This is most likely a result of the larger training parameter space and the relatively smaller dataset employed here. In addition, normal reads make up a majority of cases in the original and resampled (augmented) dataset, indicating that if a trained model gravitated towards the most common diagnosis for all samples, it could still result in a higher BLEU score than another more diagnostically accurate model. Only BLEU scores with a larger difference (say, 0.4 vs 0.6) might have discriminative value. In many cases related to attention and transformer models, training accuracy was initially below validation accuracy. This could be because although the training and validation data split was stratified based on image label values, the unbalanced nature of the dataset could result in the anomaly. For example, the phrase "no acute cardiopulmonary findings" appeared repeatedly in various permutations. For a validation dataset which is known to contain such normal reads in a greater proportion than the augmented training set, initial training might gravitate towards labeling most sets with normal reads, resulting in better validation accuracy than training accuracy. However, in later epochs, training accuracy value did show higher value than validation accuracy, as the model learnt to identify more diverse transcription and pathological findings.

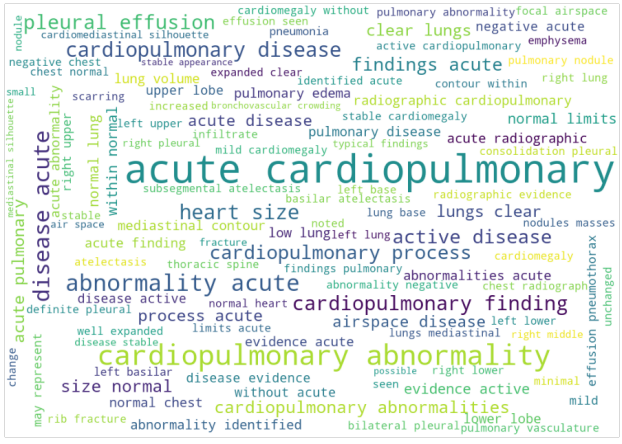


Figure 5: Wordcloud of the reference transcriptions showing a richer word space.

Figures 5, 6, and 7 show the Wordcloud [19] of reference and predicted transcriptions. The original transcription space looks much richer than the predicted transcription space. However, it does not necessarily reflect on the accuracy of the transcription. This is due to varied phrases used in the original transcription to describe the same pathology or lack of pathology. While the original transcription is richer in such variations, the predicted transcriptions likely precipitate towards a more common denominator of similar phrases. Prominently, phrases like

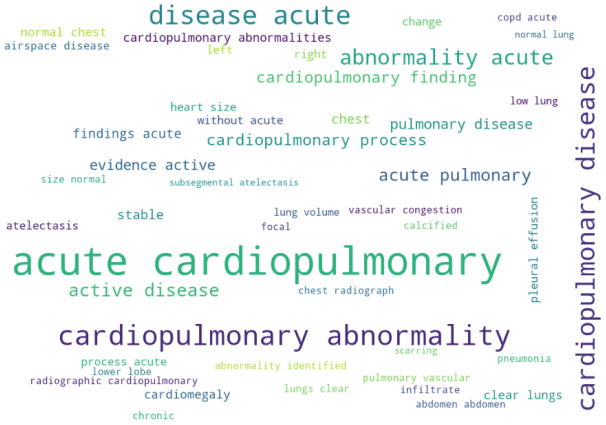


Figure 6: Wordcloud of predicted transcriptions obtained from the Attention model of 3.2.1.

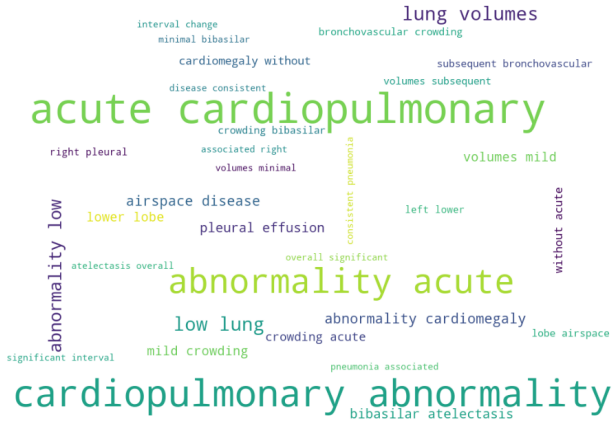


Figure 7: Wordcloud of predicted transcriptions obtained from the Transformer model of 3.2.2.

"acute cardiopulmonary', "cardiopulmonary abnormality" and "abnormality acute" occur in the Wordclouds indicating a certain confidence in the predictions. This is also seen in additional examples of predicted transcriptions provided in Figure 27 (Appendix).

The training time, where reported, varied based on the availability of GPU model Tesla T4 or P100 (for the most part) with T4 being faster by a factor of ~2.

6. Conclusion and Future Work

The current work exhibited the feasibility of deriving correct transcriptions on presented images, based on a limited dataset of images with transcriptions and labels, by using a trained model with weights initialized to those of a model trained on a much larger dataset without transcriptions. The Attention model exhibited better training and validation accuracy on the limited dataset, and had better success at transcribing accurate diagnosis when

compared to the Transformer model. This is due to the limited nature of the dataset and can be overcome with a larger, labeled, and transcribed dataset. BLEU scores were not found to directly eflect accuracy of transcriptions, and cannot be relied upon as a one-to-one scale discriminator of the ground truth.

A substantial amount of time was spent in dataframe manipulation and interfacing with the models. Training the Transformer model proved challenging with the limited dataset, but is likely to outperform the simpler attention model with a larger dataset. In addition, runtime time-outs considerably slowed down the process of finding optimal solutions in each case. As a result, training and validation consumed precious time, leaving less time for analyzing results in further depth. For example, mapping areas of the anatomy corresponding to particular phrases in the predicted transcriptions would be interesting (saliency maps), but would need a radiologist interpretation in most cases. X-ray images are difficult to interpret in comparison to tomographic images (from CT, MRI or PET), due to their projection of all anatomy into a single plane. Another area of interest would be to expand the reference vocabulary to include phrase mappings between equivalent terms. This would make BLEU scores more reflective of the accuracy of transcriptions. For example, phrases like "pulmonary edema", "wet lungs" and "subpleural edema" can be considered equivalent. Similarly, "pulmonary atelectasis" and "collapsed lung" have similar meanings. Using a dictionary of such equivalent terms to evaluate reference and predicted impressions should go a long way towards achieving accuracy in BLEU scores and other similar metrics.

References

[1] X. Wang, Y. Peng, L. Lu, Z. Lu et al. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. arXiv preprint arXiv:1705.02315, 2017.

[2] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang et al. CheXNet: Radiologist-level pneumonia detection on chest x-rays with deep learning, arXiv:1711.05225, 2017.

[3] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu et al. CheXpert: a large chest radiograph dataset with uncertainty labels and expert comparison, arXiv:1901.07031, 2019.

[4] S. Park, G. Kim, Y. Oh, J. B. Seo, S. M. Lee, J. Kim, S. Moon, J-K Lim, J. Ye. Vision Transformer using Low-level Chest X-ray Feature Corpus for COVID-19 Diagnosis and Severity Quantification arXiv:2104.07235v1, 2021.

[5] Rajaraman S, Zamzmi G, Folio L, Antani, S. Detecting Tuberculosis-Consistent Findings in Lateral Chest X-Rays Using an Ensemble of CNNs and Vision Transformers Using an Ensemble of CNNs and Vision Transformers. Front. Genet., 24 February 2022 | https://doi.org/10.3389/fgene.2022.864724

[6] Y. Xue, T. Xu, L. R. Long, Z. Xue, S. Antani, G. R. Thoma, and X. Huang. Multimodal Recurrent Model with Attention for Automated Radiology Report Generation. In: Frangi, A., Schnabel, J., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. MICCAI 2018. Lecture Noes in Computer Science(), vol 11070. Springer, Cham. https://doi.org/10.1007/978-3-030-00928-1_52.

[7] Z. Chen, Y. Song, T. H. Chang, X. Wan. Generating radiology reports via memory-driven transformer. arXiv preprint arXiv:2010.16056, 2020

[8] Y. Zhang, X. Wang, Z. Xu, Q. Yu, A. Yullie, D. Xu. When radiology report generation meets knowledge graph. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07), 12910-12917. https://doi.org/10.1609/aaai.v34i07.6989

[9] G. Liu, T-M. Hsu, M. McDermott, W. Boag, W-H Weng, P. Szolovits, M. Ghassemi. Clinically accurate chest X-ray report generation. Proceedings of the 4th Machine Learning for Healthcare Conference, PMLR 106:249-269, 2019.

[10] X. Chen, C. Zitnick. Learning a recurrent visual representation for image caption generation. arXiv preprint arXiv:1411.5654, 2014.

[11] A. B. Amjoud and M. Amrouch, "Automatic Generation of Chest X-ray Reports Using a Transformer-based Deep Learning Model," *2021* Fifth International Conference On Intelligent Computing in Data Sciences (ICDS), 2021, pp. 1-5, doi: 10.1109/ICDS53782.2021.9626725.

[12] https://www.kaggle.com/code/rejpalcz/best-loss-function-for-f1-score-metric/notebook

[13] J. Pennington, R. Socher, C. Manning. GloVe: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp1532–1543, 2014.

[14] K. Zhu, J-L. Ba, R. Kiros, K. Cho et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, arXiv:1502.03044, 2016.

[15]https://www.tensorflow.org/text/tutorials/nmt_with_attention

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin. Attention is all you need, arXiv: 1706.03762, 2017.

[17] https://openi.nlm.nih.gov/faq#collection

[18] https://www.datacamp.com/tutorial/wordcloud-python

Appendix

CXR open-access http://creativecommons.org/licenses/by-nc-nd/4.0/ byncnd 2013-08-01 XR. The data are drawn from multiple hospital systems. pulmonary diseases CXR 2013 08 01 Chest X-ray Collection None ,786.2 The lungs are clear. The heart and pulmonary XXXX are normal. The pleural spaces are clear. The mediastinal contours are normal. No acute cardiopulmonary disease Radiology Report 201308 01 Chest X-ray Collection normal F1 Xray Chest PA and Lateral /hadoop/storage/radiology/extract/**CXR96_IM-2450-2002.jpg** 7 f2p0k710 f1p0k137 f0p0k184 f4p0k2450 f3p0k181 F2 Xray Chest PA and Lateral /hadoop/storage/radiology/extract/**CXR96_IM-2450-3003.jpg** 7 f2p0k377 f1p0k36 f0p0k518f4p0k1155 f3p0k181

Figure 1: Text in an example xml transcription file.

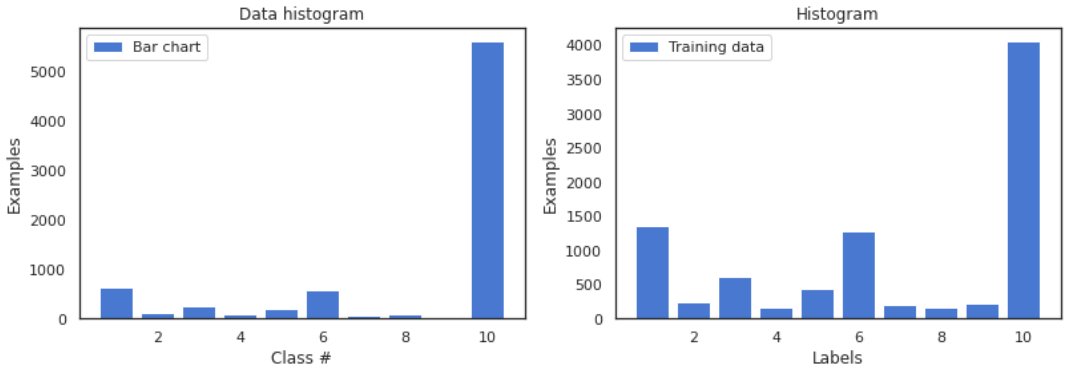


Figure 2: Original distribution across 10 classes and resampled image distribution. Note that class 10 corresponds to normal reads which far dominates other pathological classes.

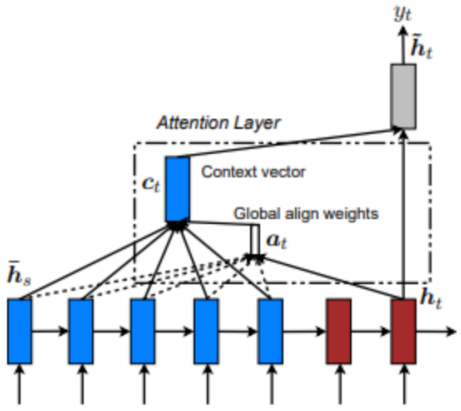


Figure 3: Schematic of attention mechanism used for the first model [15].

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	[]
model_1 (Functional)	(None, 7, 7, 1024)	7037504	['input_1[0][0]']
average_pooling2d (AveragePooling2D)	(None, 3, 3, 1024)	0	['model_1[0][0]']
tf.reshape (TFOpLambda)	(None, 9, 1024)	0	['average_pooling2d[0][0]']
im_dense (Dense)	(None, 9, 512)	524800	['tf.reshape[0][0]']
encoder_batch_norm (BatchNormalization)	(None, 9, 512)	2048	['im_dense[0][0]']
encoder_dropout (Dropout)	(None, 9, 512)	0	['encoder_batch_norm[0][0]']
input_2 (InputLayer)	[(None, 26)]	0	[]
decoder (decoder)	(None, 26, 1457)	3747102	['encoder_dropout[0][0]', 'input_2[0][0]']

Total params: 11,311,454
Trainable params: 11,226,782

Figure 4: Attention model layers and parameters.

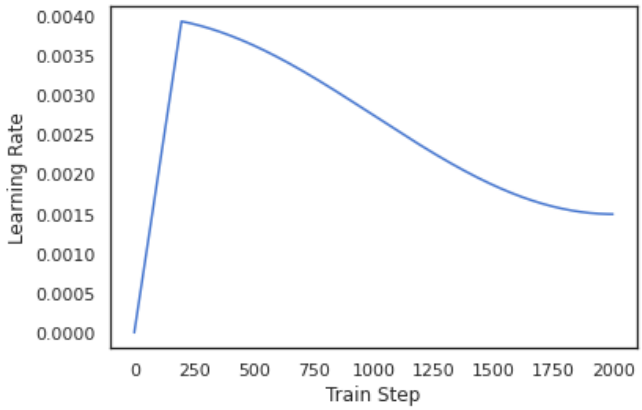


Figure 5: Learning rate schedule for the attention-based model.

```
Model: "transformer_3"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 encoder_3 (Encoder)          multiple                  34649088

 decoder_3 (Decoder)          multiple                  51910656

 dense_222 (Dense)            multiple                  1510850

=================================================================
Total params: 88,070,594
Trainable params: 88,070,594
Non-trainable params: 0
_________________________________________________________________
```

Figure 6: Transformer model with layers.

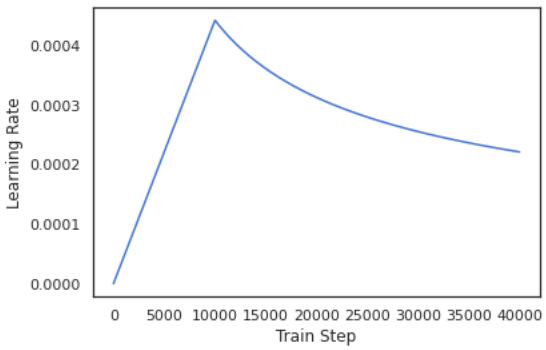
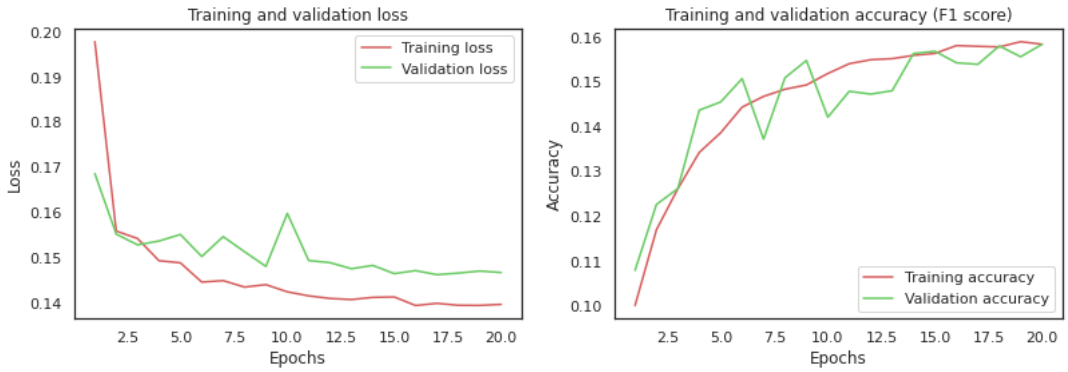


Figure 7: Final learning rate schedule used with transformer model.

CNN Model Training

(a) ResNet50V2 trained with CNN base frozen and with original data (lr = 0.01*).

(b) DenseNet121 with CNN base frozen and original data. No CheXnet weight initialization (lr = 0.01*)



(c) DenseNet121 with CNN base frozen and original data. CheXnet weight initilization (lr = 0.01*)

Figure 8: ResNet50V2 compared with DenseNet121 with and without CheXnet weight initialization.

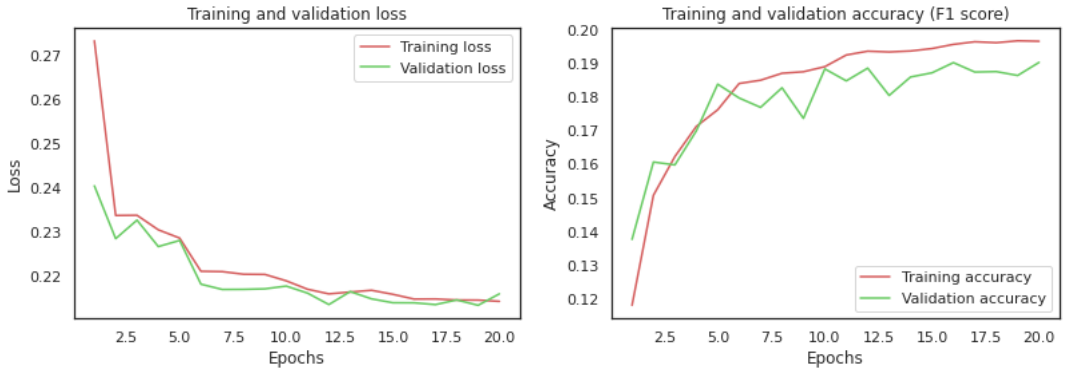


Figure 9: DenseNet121 with convolutional base frozen and augmented data (lr = 0.01*).

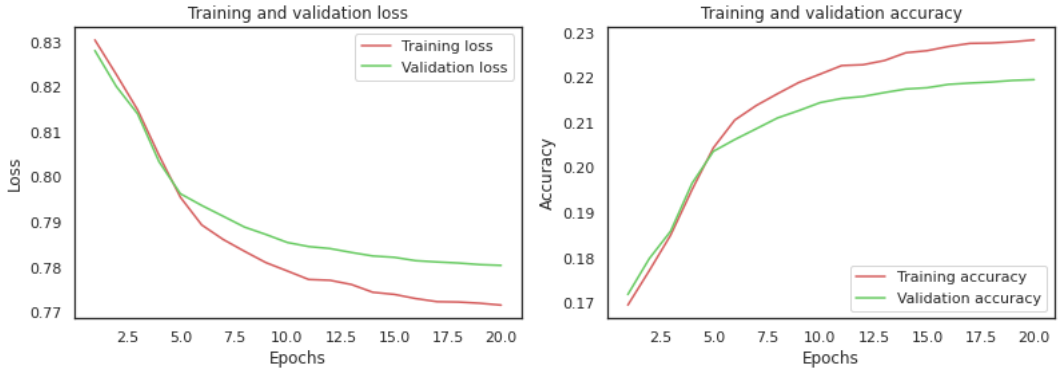


Figure 10: DenseNet121 with base frozen, resampled and flipped data (lr = 0.001*).

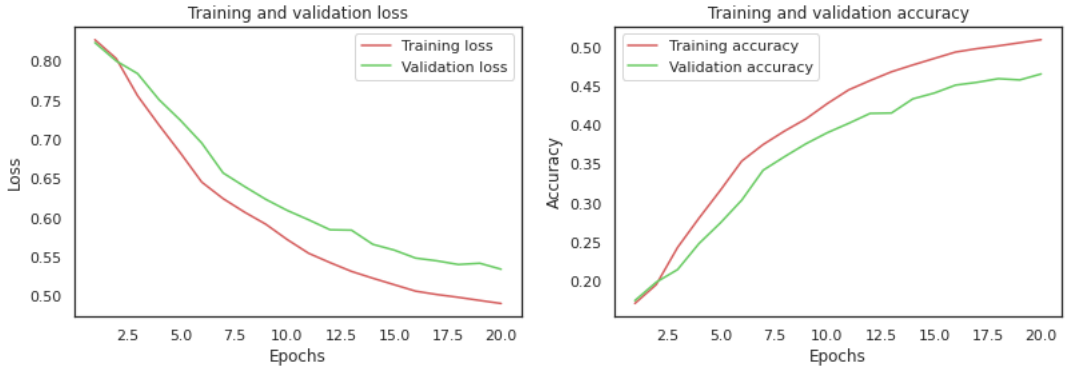


Figure 11: DenseNet121 with layers upto 409/429 frozen and resampled data (lr = 0.001*)

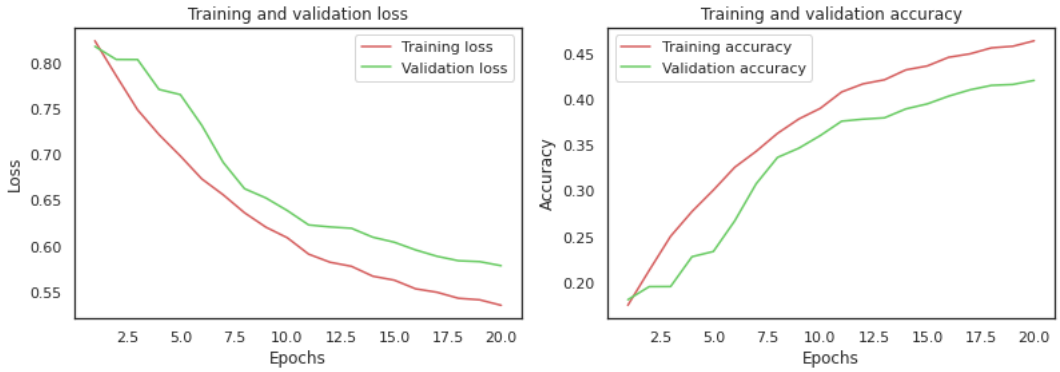


Figure 12: DenseNet121 with layers upto 409 frozen, resampled and flipped images (lr = 0.001*)

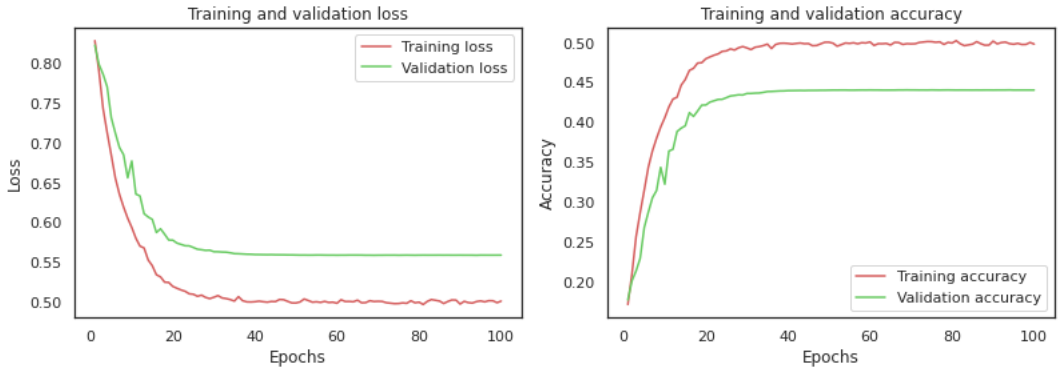


Figure 13: DenseNet121 with layers upto 409 frozen, resampled and flipped images (lr = 0.001*).

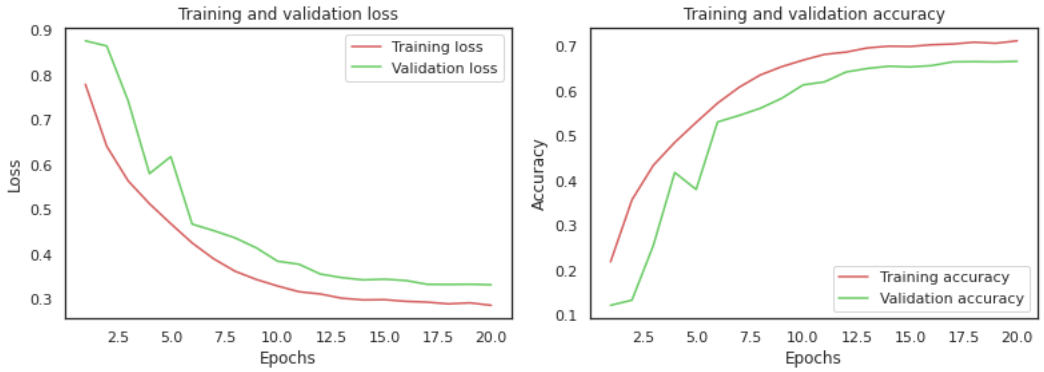


Figure 14: Densenet121 with all trainable layers and resampled and flipped images (lr = 0.001*).

Attention Model Training

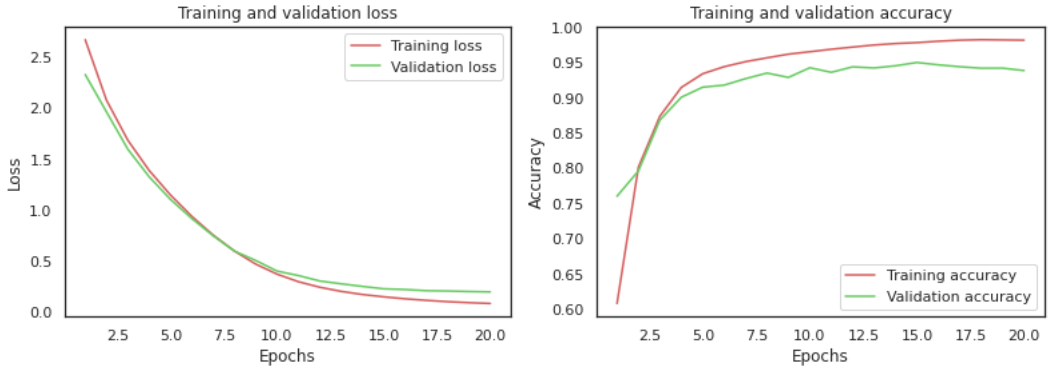


Figure 15: Attention loss and accuracy with schedule 7 of CNN model.

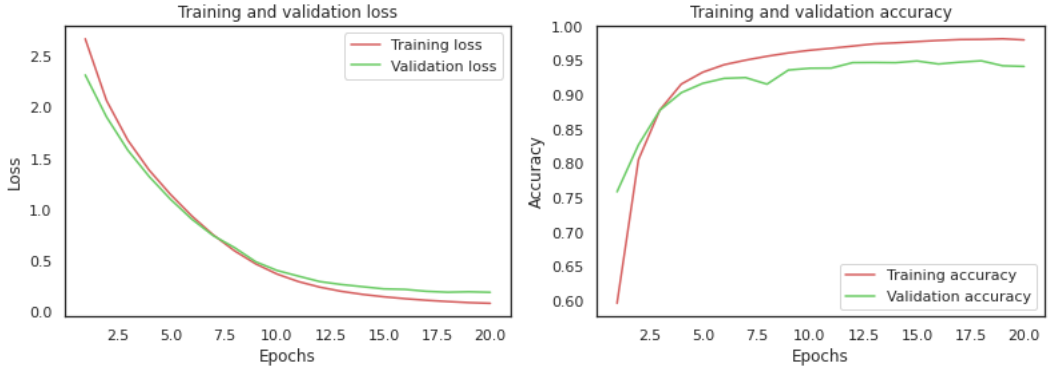


Figure 16: Attention loss and accuracy with schedule 8 of CNN model.

Transformer Model Traning

Figure 17: Transformer model with dropout = 0.5, num_layer = 4, d_model = 1024, 3x3x1024 image feature vector obtained from schedule 8.

Figure 18: Transformer model with dropout = 0.5, num_layer = 4, d_model = 512, 3x3x1024 image feature vector obtained from schedule 8.

Figure 19: Transformer model with dropout = 0.5, num_layer = 4, d_model = 512, 3x3x1024 image feature vector obtained from schedule 8 trained for 50 epochs.

Figure 20: Transformer model with dropout = 0.5, num_layer = 4, d_model = 512, 3x3x1024 image feature vector obtained from schedule 8 trained for 100 epochs.

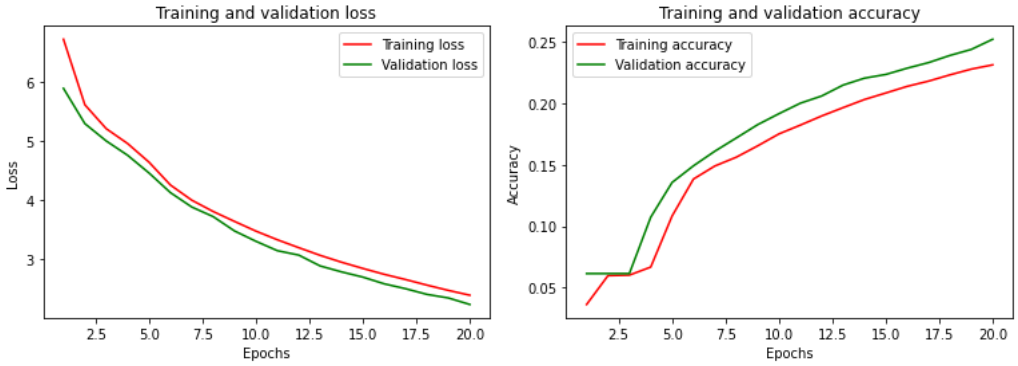


Figure 21: Model with dropout = 0.5, num_layer = 4, d_model = 512, but 7x7x1024 image feature vector.

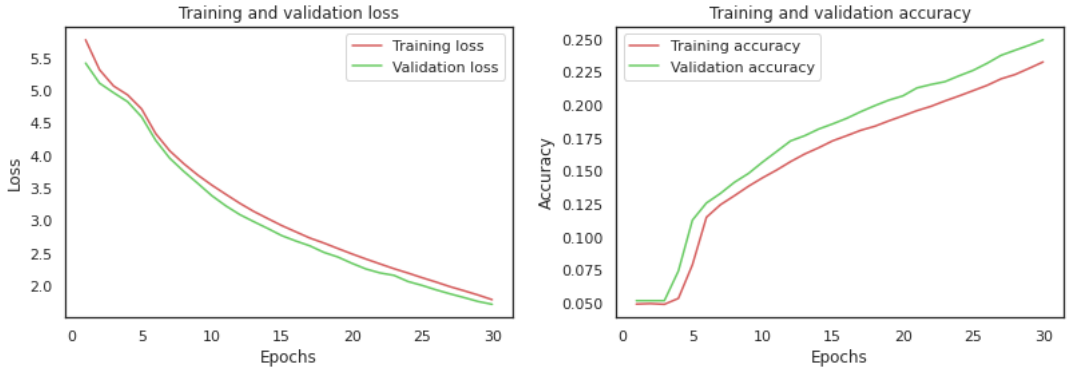


Figure 22: Transformer model dropout = 0.5, num_layer = 4, d_model = 512, image feature vectos 3x3x1024 (schedule 7)

Figure 23: Transformer model dropout = 0.5, num_layer = 4, d_model = 512, 1024x3x3, (schedule 7).

Figure 24: Transformer model with num_layer = 4, d_model = 512, 1024x3x3 image features, dropout = 0.2 (schedule 8).

Figure 25: Transformer model with num_layer = 6, d_model = 512, 1024x3x3 image features, dropout = 0.5 (schedule 8).

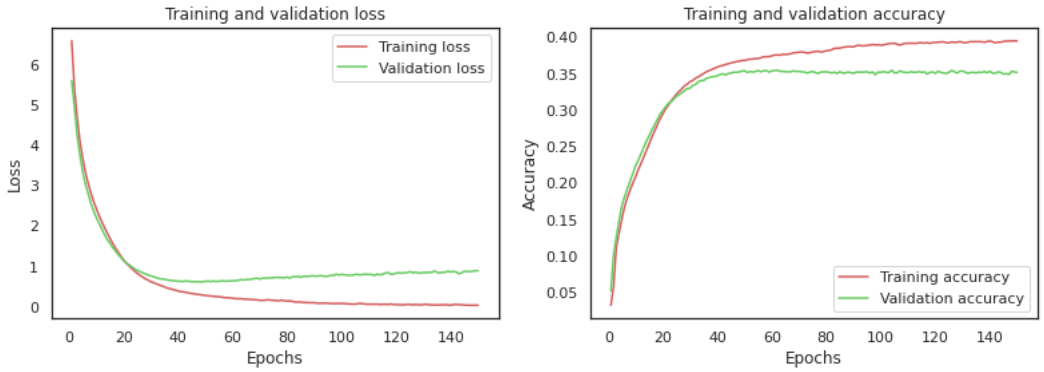
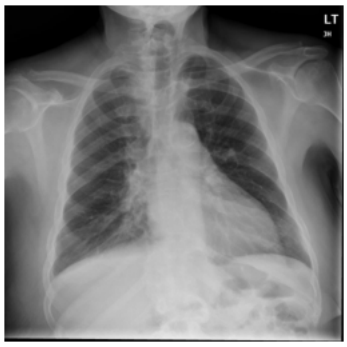
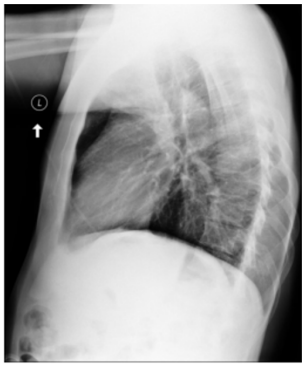


Figure 26: Transformer model with num_layer = 4, d_model = 512, 1024x3x3 image features, dropout = 0.5, trained for 150 epochs (schedule 8).

True impression: 'no acute cardiopulmonary disease .'
Predicted impression: 'no acute cardiopulmonary findings .'



True impression: 'central pulmonary vascular congestion without edema consolidation . bilateral pleural effusions .'
Predicted impression: 'stable cardiomegaly . improved aeration of lung bases with persistent left basilar effusion . prominent interstitium possibly due to mild volume overload .'



True **impression:** 'no acute cardiopulmonary abnormalities .'
Predicted impression: 'no acute cardiopulmonary abnormality .'

Figure 27: Further examples of transcriptions with close agreement between reference and predicted impressions.