AI-driven Disease Detection from X-Ray Images

This project explores the development of an AI model capable of detecting various lung diseases from chest X-ray images and generating comprehensive reports.

Mostafa Fawzy
Abdelrahman Walid
Moataz nasr
Mahmoud naser
Yousef atef

Exploratory Data Analysis (EDA) for Images and Reports

1. **Image Analysis**

   Analyzing the distribution of different lung diseases, their prevalence, and potential correlations.

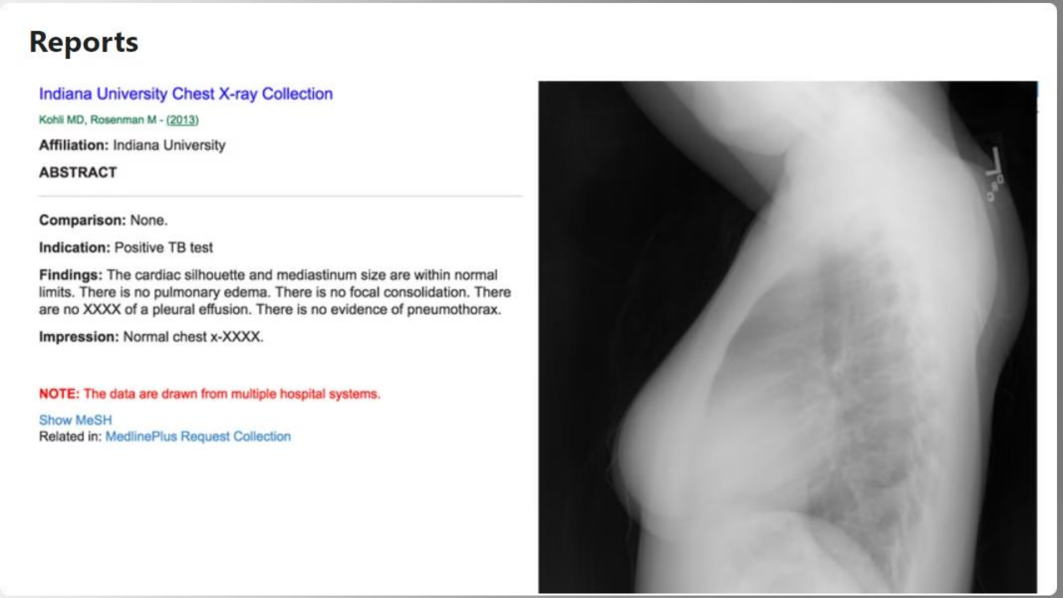
2. **Text Analysis**

   Identifying key terms, phrases, and patterns in the radiology reports that provide insights into diagnoses.

3. **Data Quality Assessment**

   Evaluating the quality and consistency of the data, addressing any inconsistencies or missing information.

The objective of this case study is to develop a deep learning model that automates the generation of the impression section of medical reports based on chest X-ray images, helping reduce the workload of medical professionals. Typically, writing these reports takes 5-10 minutes per case, and with doctors needing to generate reports for hundreds of images daily, automating the process can save significant time. The study uses a publicly available Indiana University Chest X-ray dataset, which includes both images and corresponding medical reports in XML format. The focus is on predicting the impressions from the X-ray images, aiming to assist doctors by automatically generating this critical part of the report.


Reports

Indiana University Chest X-ray Collection

Kohli MD, Rosenman M - (2013)

Affiliation: Indiana University

ABSTRACT

Comparison: None.

Indication: Positive TB test

Findings: The cardiac silhouette and mediastinum size are within normal limits. There is no pulmonary edema. There is no focal consolidation. There are no XXXX of a pleural effusion. There is no evidence of pneumothorax.

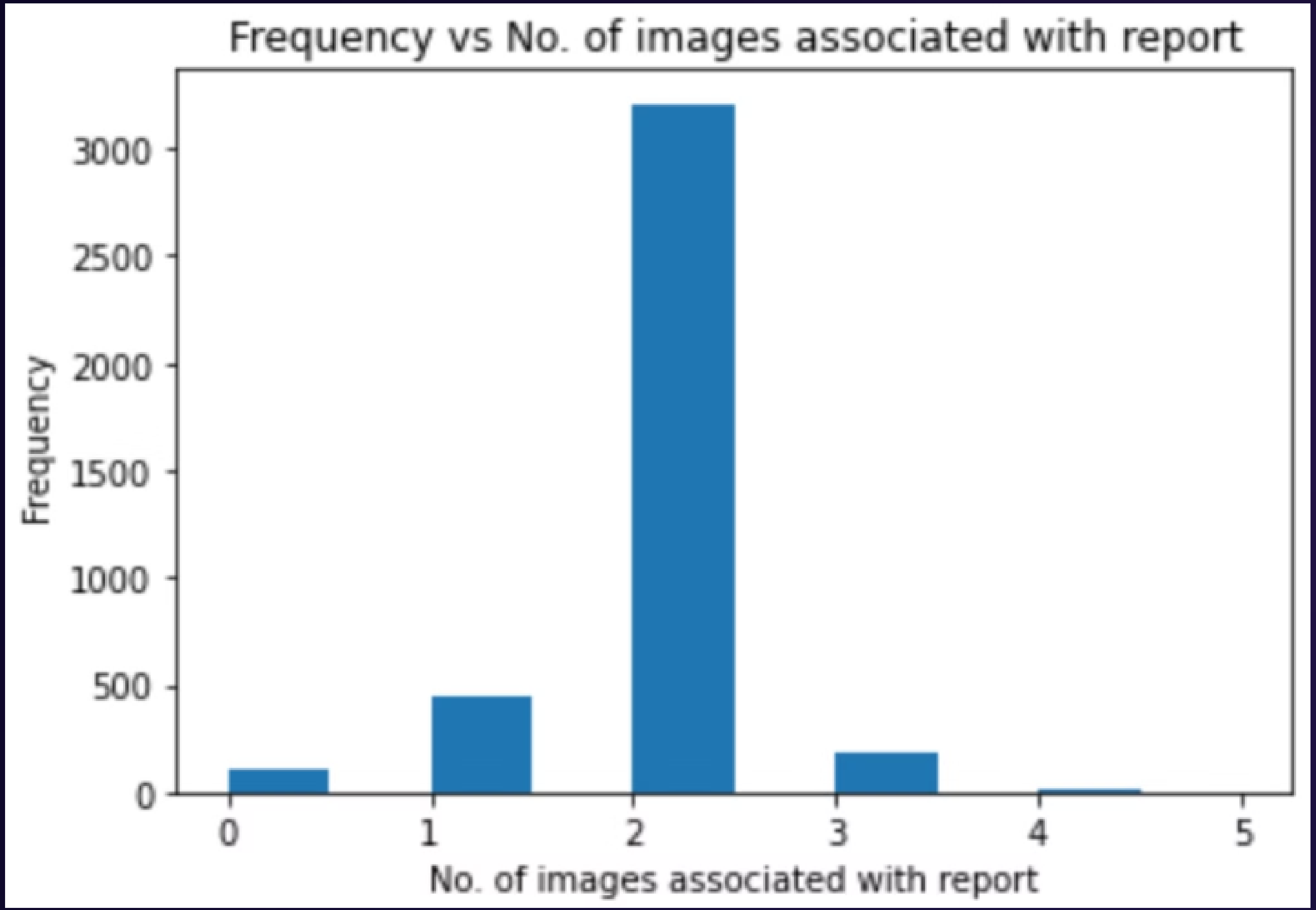
Impression: Normal chest x-XXXX.

NOTE: The data are drawn from multiple hospital systems.

Show MeSH

Related in: MedlinePlus Request Collection

We can see that the maximum number of images associated with a report can be 5 while the minimum is 0

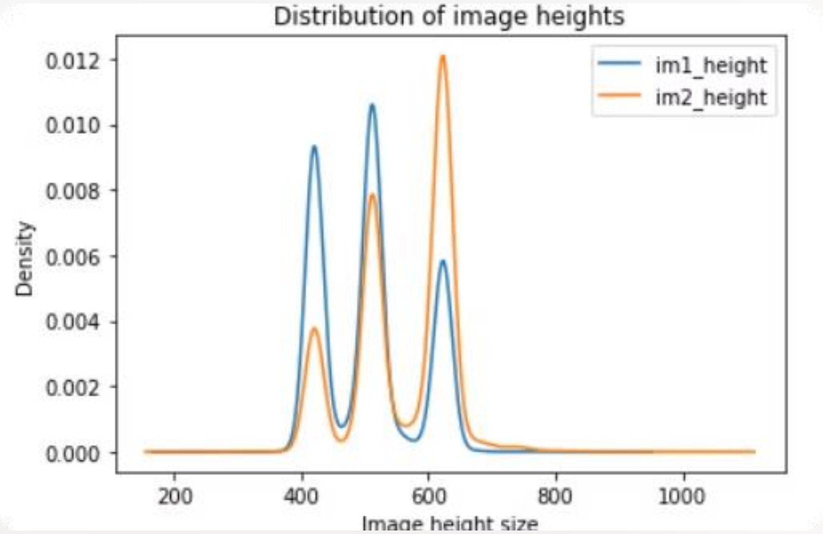


we will take 2 images as input as 2 images are highest frequency of being associated with a report

For 1-image report we will use the same image as second image

	image_1	image_2
0	CXR597_IM-2189-2001.png	NaN
1	CXR601_IM-2192-1001.png	CXR601_IM-2192-1002.png
2	CXR600_IM-2192-1001.png	CXR600_IM-2192-2001.png
3	CXR605_IM-2194-1001.png	CXR605_IM-2194-1002.png
4	CXR59_IM-2184-1001.png	CXR59_IM-2184-2001.png

Width for both of the images have only 1 unique value for all datapoints and that is 512. Since pretrained models are modelled for square-sized images we can choose 512*512 as the specified size of the image. Hence we can resize all images into 512*512 shape.



Leveraging ChexNet as Image Feature Extractor

Pre-trained Model

Utilizing a pre-trained ChexNet model, trained on a large dataset of chest X-rays for image feature extraction.

Feature Extraction

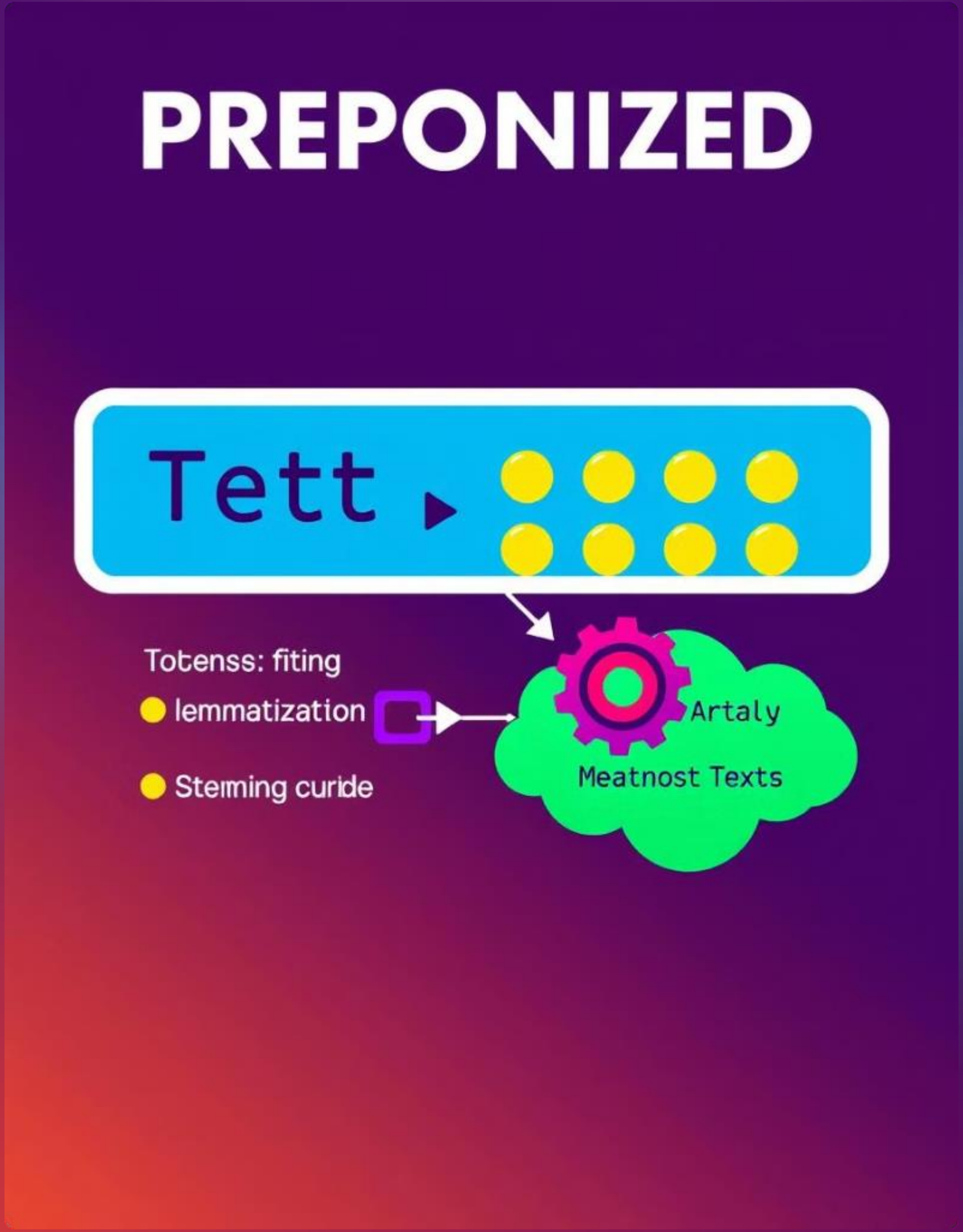
Extracting high-level features from input X-ray images, capturing crucial information for disease detection.

Transfer Learning

Leveraging the knowledge acquired by ChexNet on a large dataset to improve our model's performance on the specific task.

```
from sklearn.utils import resample
df_majority = train[train['impression_counts']>=100] #having value counts >=100
df_minority = train[train['impression_counts']<=5] #having value counts <=5
df_other = train[(train['impression_counts']>5)&(train['impression_counts']<100)] #value counts between 5 and 100
n1 = df_minority.shape[0]
n2 = df_majority.shape[0]
n3 = df_other.shape[0]
#we will upsample them to 30
df_minority_upsampled = resample(df_minority,
                                 replace = True,
                                 n_samples = 3*n1,
                                 random_state = 420)
df_majority_downsampled = resample(df_majority,
                                 replace = False,
                                 n_samples = n2//15,
                                 random_state = 420)
df_other_downsampled = resample(df_other,
                                 replace = False,
                                 n_samples = n3//10,
                                 random_state = 420)

train = pd.concat([df_majority_downsampled ,df_minority_upsampled,df_other_downsampled])
train = train.reset_index(drop=True)
# del df_minority_upsampled,df_minority,df_majority,df_other,df_other_downsampled
train.shape
```



Text Tokenization and Preprocessing

1 **Tokenization**

Splitting the text reports into individual words or tokens, creating a representation suitable for machine learning.

2 **Normalization**

Converting words to lowercase, removing punctuation, and applying other transformations for consistent representation.

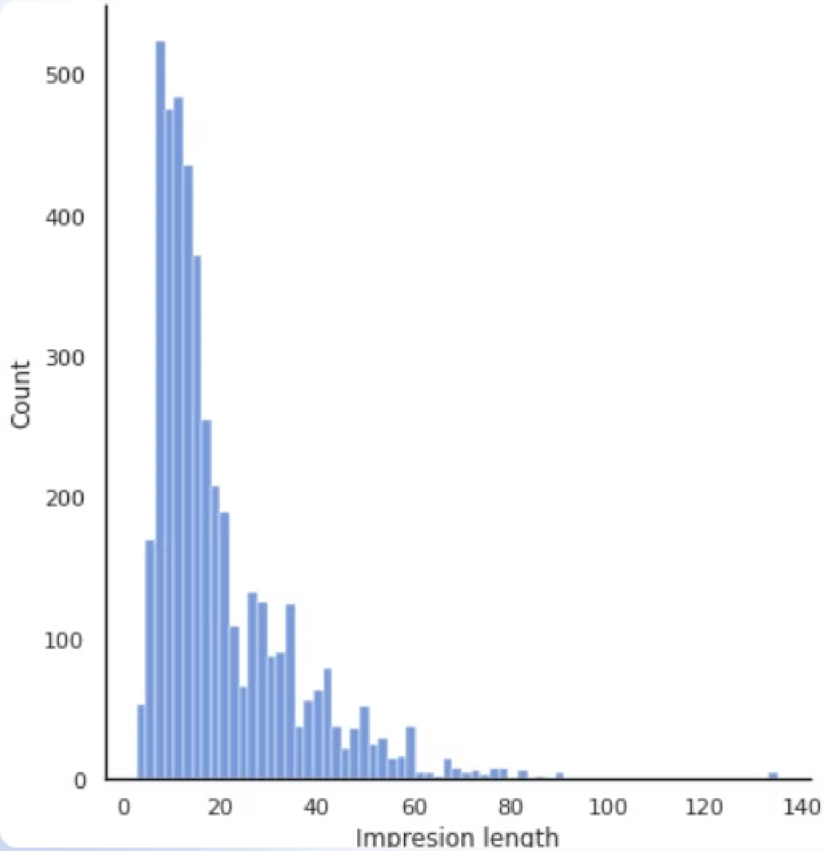
3 **Stemming/Lemmatization**

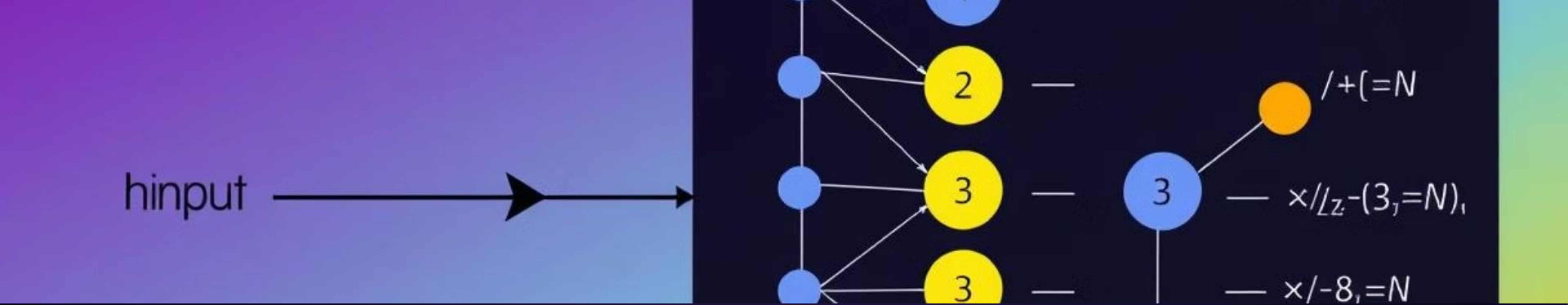
Reducing words to their root form, improving the model's ability to generalize across variations of words.

4 **Stop Word Removal**

Eliminating common words that hold little semantic value, focusing the model on meaningful terms.

The max and min value of "caption length" was found to be 135 and 3 respectively The 80 percentile value of caption_len which is 29 will be taken as the maximum padded value for each impression





Developing a Simple Classification Model

Input Layer

Takes the extracted image features and preprocessed text tokens as input.

Hidden Layers

Combines image and text information to learn complex relationships and patterns.

Output Layer

Generates probabilities for each disease class, indicating the model's prediction for the presence or absence of a disease.

Evaluating Model Performance

The BLEU score will be used as a metric to evaluate the model by comparing words in the predicted sentence to the reference sentence using n-grams. However, BLEU has limitations, as it penalizes similar words with the same meaning. To address this, in addition to n-gram BLEU scoring, a sample of the predicted captions will be manually compared to the original reference captions to provide a more accurate assessment of performance.

	bleu1	bleu2	bleu3	bleu4
greedy search	0.317412	0.308454	0.333496	0.366244

Introducing an Attention-based Model

1

Attention Mechanism

Focuses on specific parts of the input image and text that are most relevant to disease prediction.

2

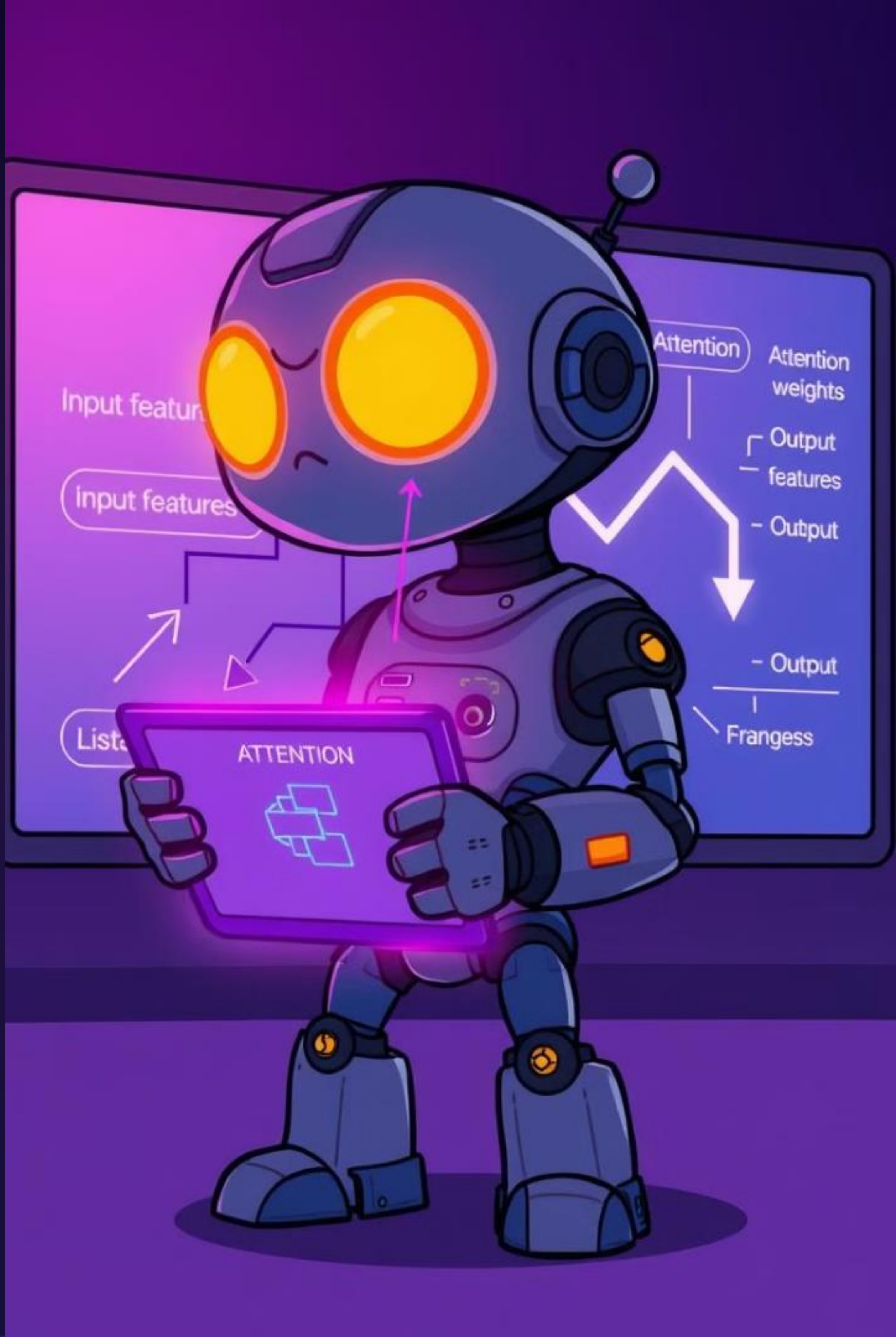
Weighting Importance

Assigns weights to different features, emphasizing those that contribute most to the final prediction.

3

Improved Accuracy

By focusing on relevant information, the model can achieve higher accuracy and better generalization.

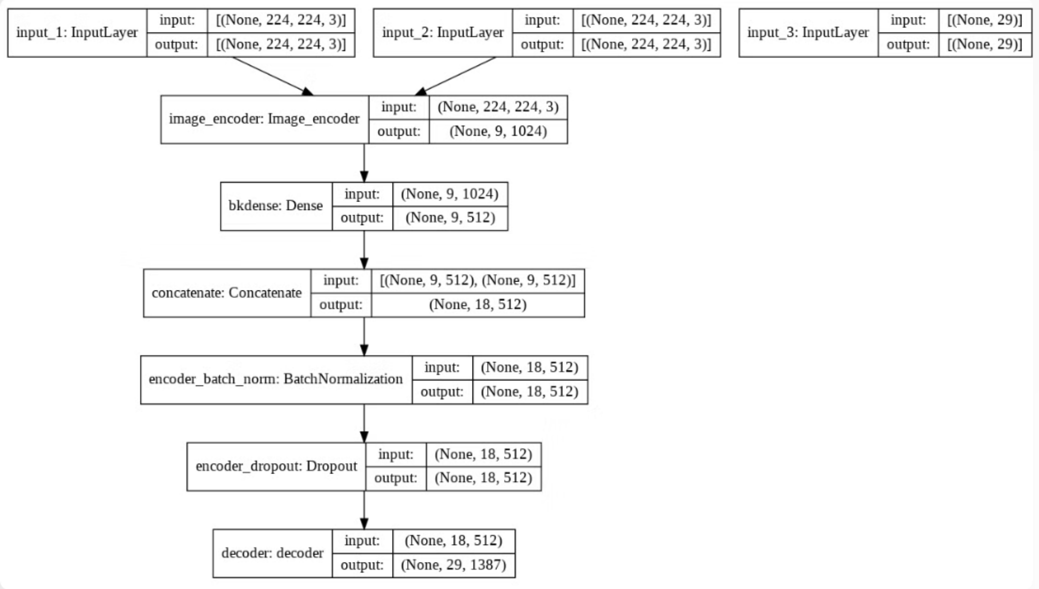


An attention model is used to enhance the performance of sequence-based tasks, like image captioning or translation, by allowing the model to focus on the most relevant parts of the input when generating each part of the output.

Global Attention equations

$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t ; \bar{\boldsymbol{h}}_s]\right) & \textit{concat} \end{cases}$$

Architecture of the AI model showcasing the greedy search mechanism. The model processes image inputs, encoding them into dense feature representations, followed by concatenation and batch normalization, before reaching the decoding stage, where the output is generated by selecting the optimal steps sequentially, based on immediate gains.



Testing and Comparing Model Outcomes

Performance Evaluation

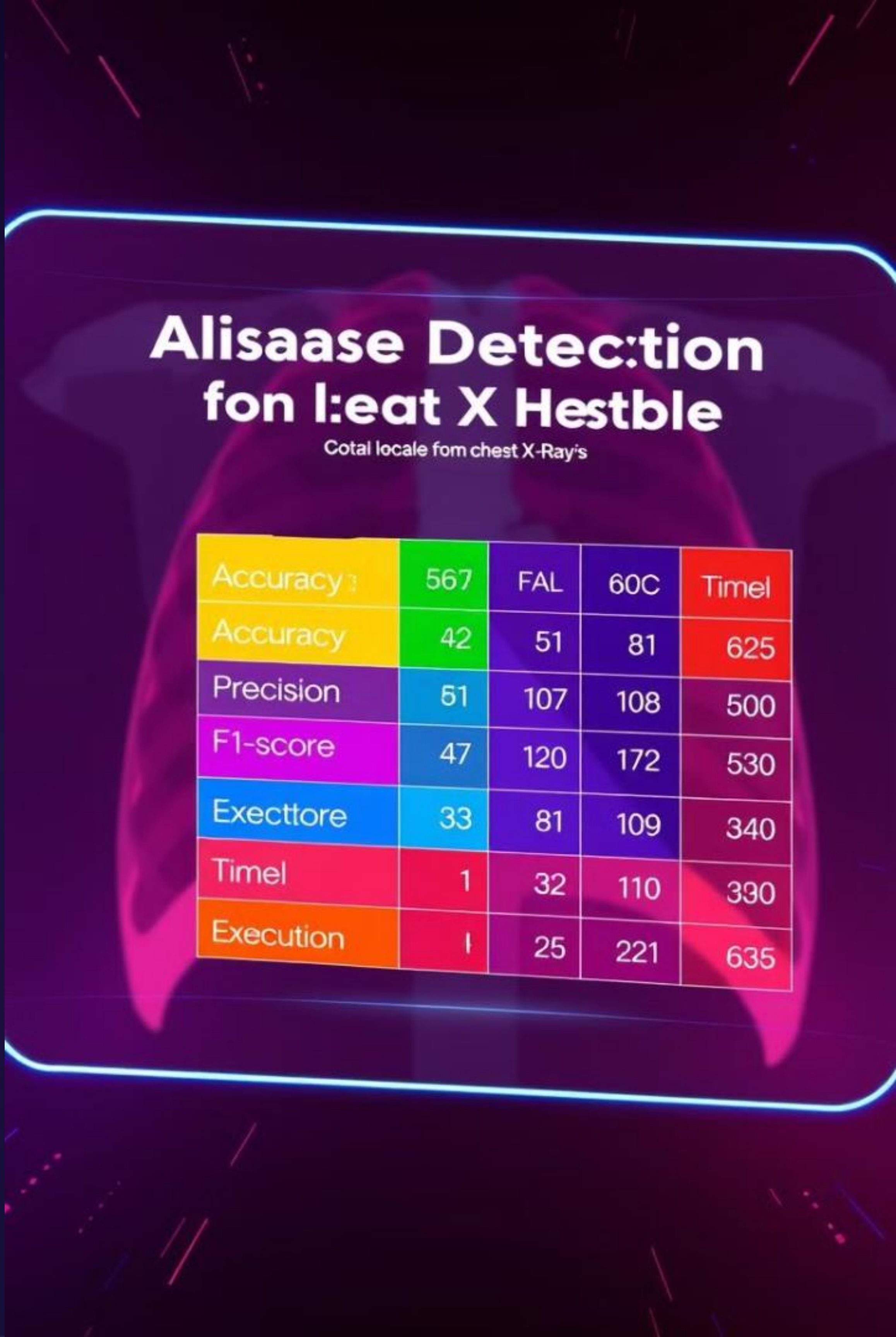
Evaluating the performance of both simple and attention-based models on a separate test set to assess generalization.

Model Comparison

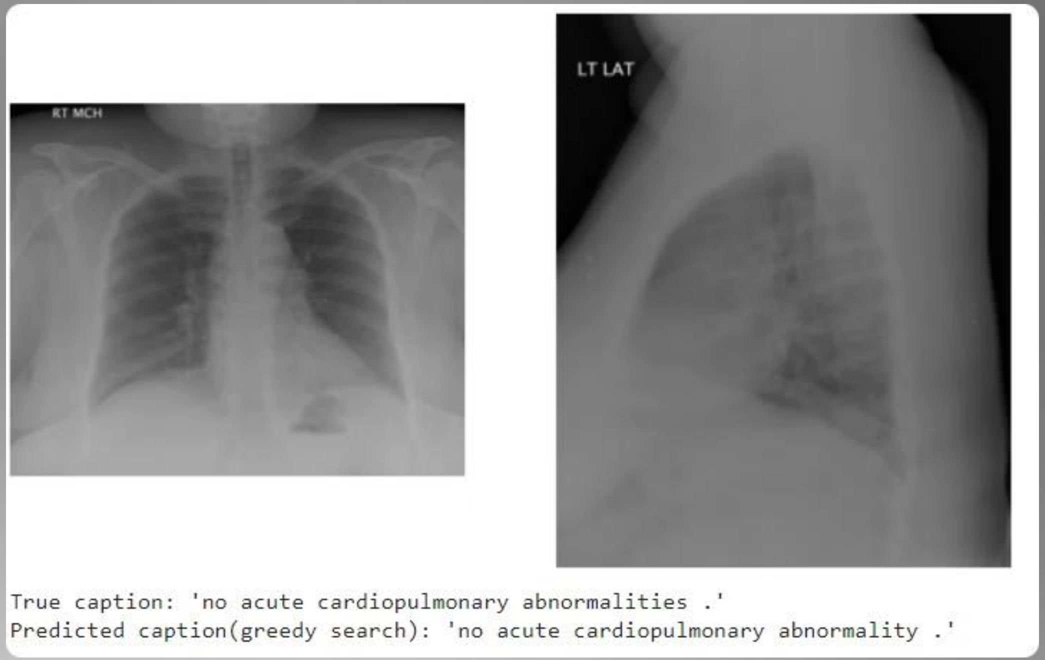
Comparing the accuracy, precision, recall, and other metrics of the models to determine which performs better.

Report Generation

Using the best-performing model to generate comprehensive reports for each X-ray image, providing detailed information about the detected diseases.

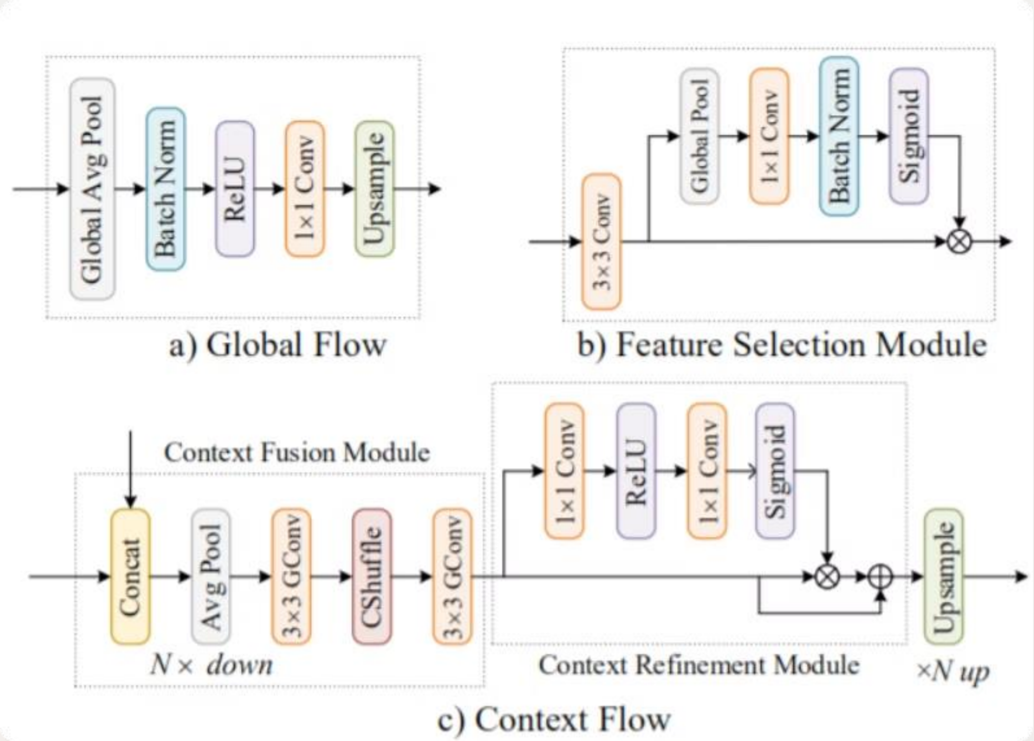


This model performed better than simple baseline model since it produced captions which had higher variablity and also remained lingusitically similiar. Even then we can see that most of datapoints were of normal chest or no disease category we need to collect more dataset which have x-rays of patients having diseases so that to improve the model's performance. Even the model predicted tough captions which had similiar meaning to the true ones. Beam search was found to take large amount of time in predicting even 1 caption so it was discarded.



Custom Final Model

This architecture implementation is taken from Attention guided chained context aggregation for image segmentation which was used for image segmentation but I will use it to extract image information. Here what I will do is that the outputs from Image encoder (ie chexnet) will be sent to global flow. Then outputs from both the chexnet and global flow will be concatted and sent to context flow. Here global flow extracts the global information of image while context flow will get the local features of the images. This will be then sent to decoder after reshaping, and applying batch norm and dropout.



Conclusion

This model performed better than simple baseline model since it produced captions which had higher variablity and also remained lingusitically similiar. Even then we can see that most of datapoints were of normal chest or no disease category we need to collect more dataset which have x-rays of patients having diseases so that to improve the model's performance. Even the model predicted tough captions which had similiar meaning to the true ones. Beam search was found to take large amount of time in predicting even 1 caption so it was discarded. An attention model is used to enhance the performance of sequence-based tasks, like image captioning or translation, by allowing the model to focus on the most relevant parts of the input when generating each part of the output.