IMAGECLEF2019 VQA in Medical Domain

CS4090 Project
Final Report

Submitted by

K A Siva Vardhan Reddy	**B160333CS**
P Satyanarayana	**B160340CS**
B Maheswara Rao	**B160349CS**
R Vamsi Krishna	**B160109CS**

Under the Guidance of

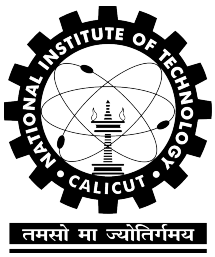
Ms. Lijiya A
Assistant Professor

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

June 13, 2020

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

Certified that this is a bonafide report of the project work titled

IMAGECLEF2019 VQA IN MEDICAL DOMAIN

done by

K A Siva Vardhan Reddy
P Satyanarayana
B Maheswara Rao
R Vamsi Krishna

of Eighth Semester B. Tech, during the Winter Semester 2019-'20, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology Calicut.

(Ms. Lijiya A)
(Assistant Professor)

June 13,2020 **Project Guide**

Dr. Saleena N
Head of the Department

DECLARATION

I hereby declare that the project titled, **ImageCLEF2019 VQA in Medical Domain**, is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place : NIT CALICUT
Date : June 13,2020

Signature : KASVR
Name : K A Siva Vardhan Reddy
Roll. No. : B160333CS

Signature : SATYA
Name : P Satyanarayana
Roll. No. : B160340CS

Signature: B.Maheswararao
Name : B Maheswara Rao
Roll. No. : B160349CS

Signature : VAMSI
Name : R Vamsi Krishna
Roll. No. : B160109CS

Abstract

Visual Question Answering is a model in which questions are answered based on the given image. It is highly evolved in case of general domain. Now in this project we will be focusing on medical domain. Automated systems could help clinicians cope with large amounts of images by answering questions about the image contents.
Our aim is to develop a model that can predict answer for the given question and its related image. For image processing we used pretrained VGG-Net model and for question processing we used LSTMs without pre-trained embeddings. After the implementation, we achieved an accuracy of 73.4% for plane category, 68.4% for organ category and 44.4% for modality category.

ACKNOWLEDGEMENT

Working in the field of Deep Learning is very interesting. During the course period of project, we learnt a lot on Deep Learning algorithms especially on Recurrent and Convolutional Neural Networks which may pave a path towards the scope of our future.

In this aspect we are very much thankful to our guide Ms Lijiya A. who gave enough freedom to chose this topic and also wide range of knowledge towards trending projects. And we would like to thank Lubna, a PhD Scholar who was very keen in explaining the doubts we faced and gave a direction to our work. Also we would like to thank the panel members Ms Anu Mary Chacko, Mr Sumesh T.A. and Dr. Saidalavi Kalady who guided us in the right track by evaluating us. Finally, We are thankful to the Department of CSE and faculty for giving this opportunity.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Visual Question Answering is a recent and exciting problem at the intersection between Computer Vision and Natural Language Processing, where the input is an image and a question related to it written in a natural language and the output is the correct answer to the question. The answer can be a simple yes/no, choosing one of several options, a single word, or a complete phrase of sentence.

From a first glance, the VQA problem seems like a very challenging one. The traditional CV techniques used for extracting useful information from images and the NLP techniques typically used for Question Answering are very far from each other and the interplay between them seem to be complex. Moreover the ability to construct a useful answer based on such multi-modal input adds to the complexity of the problem. Luckily, the recent advances in Deep Learning(DL) have paved the way to build more robust VQA techniques.

The VQA-Med task was introduced for the first time in 2018, inspired by the open-domain VQA challenges that started in 2015 [2]. Given a medical image and a natural language question about the image, participating systems are tasked with answering the question based on the visual image content. Three datasets were provided for training, validation and testing.

For image processing, we use pretrained VGG-Net model. For question processing, we use LSTMs without pre-trained embeddings. Question vectors are extracted from the final hidden layer of the LSTMs. These two are concatenated to get the result.

Chapter 2

Literature Survey

2.1 NLM at ImageCLEF 2018 VQA in the Medical Domain

This paper[2] describes the participation of the U.S. National Library of Medicine (NLM) in the Visual Question Answering task (VQAMed) of ImageCLEF 2018.They studied deep learning networks with state-of-the-art performance in open-domain VQA.They selected Stacked Attention Network (SAN) and Multimodal Compact Bilinear pooling (MCB) for their official runs.

2.1.1 Stacked Attention Network(SAN)

- The Stacked Attention Network (SAN) was proposed to allow multi-step reasoning for answer prediction. SAN includes three components: (i) the image model based on a CNN to extract high level image representations, (ii) the question model using an LSTM to extract a semantic vector of the question and (iii) the stacked attention model which locates the image regions that are relevant to answer the question.

- For the image model, they used the last pooling layer of VGG-16 pre-trained on imageNet as image features. For the question model, they used the last LSTM layer as question features. The image features and the question vector were used to generate the attention distribution over the regions of the image.

- The first attention layer of the SAN is then computed to capture the correlations between the tokens of the question and the regions in the image. Multimodal pooling is performed to generate a combined question and image vector that is then used as the query for the image in the next layer. They used two attention layers, as it showed better results in open-domain VQA. The last step is answer prediction. For a set of N answer candidates, the answer prediction task is modeled as N-class classification problem and performed using a one-layer neural network. Answers are predicted using Softmax probabilities.

2.1.2 Multimodal Compact Bilinear pooling(MCB)

- Multimodal Compact Bilinear pooling (MCB) is an attention mechanism that implicitly computes the outer product of visual and textual vectors.

- MCB architecture contains: (i) a CNN image model, (ii) an LSTM question model, and (iii) MCB pooling that first predicts the spatial attention and then combines the attention representation with the textual representation to predict the answers.

- For the image model, they used ResNet-152 and ResNet-50 pre-trained on imageNet. For the question model, a 2-layer (1024 units in each layer) LSTM model is used. Concatenated output from both layers (2048 units) forms the input to the next pooling layer. MCB pooling is then used to combine both image and textual vectors to produce

a multimodal representation. To incorporate attention, MCB pooling is used again to merge the multimodal representation with the textual representation for each spatial grid location. They also fine-tuned ResNet-50 on modality classification.

2.2 JUST at VQA-Med: A VGGSeq2Seq Model

The model[4] takes an image and a question as input and outputs the answer of this question based on fusing features extracted based on the image content with those extracted from the question itself.This model follows the encoder-decoder architecture.

- The encoder consists of two main components. The first component is a Long short term memory (LSTM) network with a pretrained word embedding layer which encodes the question into a vector representation, while the second component is a pretrained VGG network that takes the image as an input and extracts a vector representation for that image. The final state of the encoding, the outputs of the two components are concatenated together into one vector called thought vector.

- The decoder consists of LSTM network that takes the thought vector as initial state and start token as input in the first time step and try to predict the answer using softmax layer.

2.3 JUST at ImageCLEF 2019 Visual Question Answering in the Medical Domain

In this paper[9] JUST team used different models for different categories of data. The following subsections describe the models they build for each subcategory before describing how to combine them.

2.3.1 Plane Model

The questions format on this category are repetitive and all questions have the same meaning even if they use different words. So, it is expected that the questions would not contribute anything in answer predictions and only the image can determine the plane answer. Hence, they deal with this part as an image classification task. They used the pre-trained model VGG16 with the last layer (the Softmax layer) removed and all layers (except the last four) frozen. The output from this part is fed into two fully-connected layers with 1024 hidden nodes followed by a Softmax layer with 16 plane classes. The below figure shows the plane model architecture in details. Since the data is unbalanced, they used class weights in order to give the classes with smaller numbers of images higher weights.

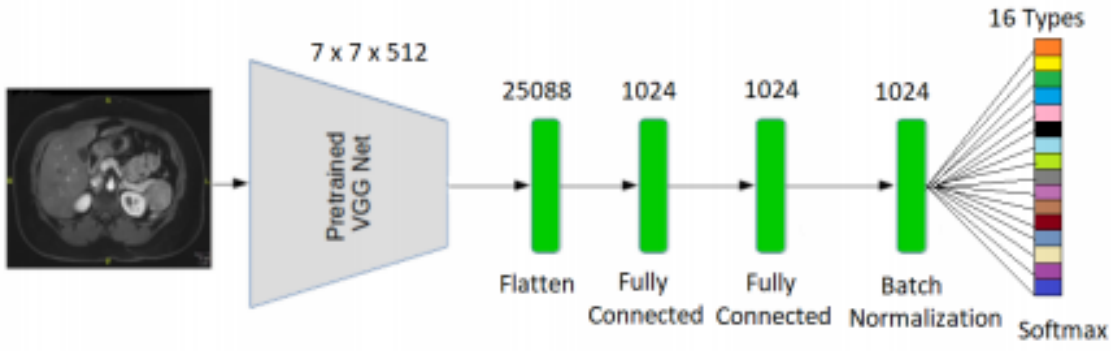


Figure 2.1: Plane Model architecture

2.3.2 Organ Model

The questions formats in organ system are also repetitive and have the same meaning. So, they rely on the images only to get the organ system answer, i.e., as an image classification task. They used the same model architecture for plane model except that the last layer, which is the Softmax layer, has the ten organ systems classes.

Chapter 3

Problem Definition

Given a medical image accompanied with a clinically relevant question, our desired model is supposed to be tasked with answering the question based on the visual image content. This may vary from simple problem such as classification of the image to a complex one such as answer generation.

3.1 Motivation

These are the various reasons to automate a job.

- Clinically relevant images are large in number.
- Training a person in answering such questions take huge time.
- There is a chance of mistakes being done by human beings.

Deep learning is trending and huge applications in various fields are being evolved. Also due to advancement in accurate recognition of objects in a given image and natural language processing techniques in understanding the semantics of a given sentence and answering according paved a way and motivated us in doing this project.

3.2 Input and Output

Clinical image and a medically relevant question accompanied with it is given as input.

Output is an answer(one word or a sentence) according to the given question and image.

Chapter 4

Methodology

4.1 Dataset Collection

Since Visual Question Answering is a large area which contains numerous types of images, countably many questions for each image and also many responses for each question, we restrict our project to medical domain. In the scope of the VQA-Med challenge, three datasets were provided:

- The training set contains 12792 question-answer pairs associated with 3200 training images.
- The validation set contains 2000 question-answer pairs associated with 500 validation images.
- The test set contains 500 questions associated with 500 test images.

The data is equally distributed over four categories based on the question types which are:

4.1.1 Plane Category

Question on planes come in one of the following formats: “in which plane”, “Which plane”, “what plane”, “in what plane”, “what is the plane”, “what

imaging plane is", and "what image plane". There are 16 different planes. Some of them are axial, sagittal, coronal, AP, lateral, frontal etc.

4.1.2 Organ Category

Question on organ systems come in one of the following formats: "what organ system is", "what part of the body is", "the ct/mri/ultrasound/x-ray scan shows what organ system", "which organ system is", "what organ system is", "what organ is this", etc. There are ten organ systems.

4.1.3 Modality Category

Question on organ systems come in one of the following formats: "what modality was used to take this image", "is this an mri image", "is this a t1 weighted, t2 weighted, or flair image", "what type of contrast did this patient have". There are eight main modality categories: XR, CT, MR, US, MA, GI, AG, and PT.

4.1.4 Abnormality Category

Question on organ systems come in one of the following formats: "what is the abnormality/wrong/alarming in this image", "is this image normal" or "is this image abnormal".

4.2 Design

Our inputs to the model are preprocessed representations of our images and questions. The architecture of our model is represented in the below figure. Our model consists of four modules: one for image, one for question, one to encode question image together and one for decoding to get answer.

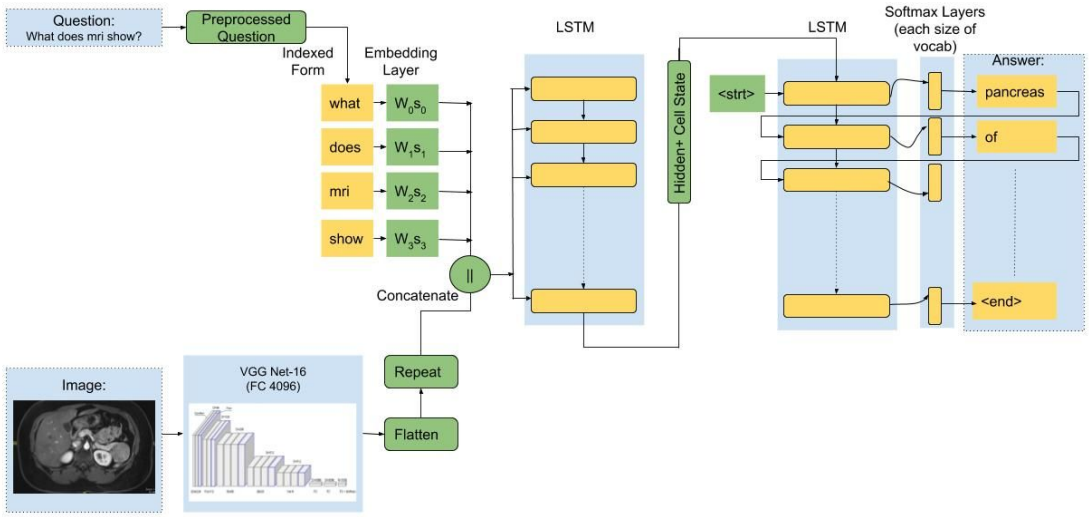


Figure 4.1: BaseLine architecture

4.2.1 Image Module

We have used pretrained VGG-Net model in order to extract image features. The input to the image module are features extracted using VGG-net. These features are of size (1,4096). We then use the operation repeat here to convert the features to size (max question length, 4096).

4.2.2 Question Module

We use an embedding layer that learns to map input words to dimensional features (or word-embeddings). Word-embeddings help us represent our words as vectors, where semantically similar have similar word vectors. We are basically using it to appropriately encode the words. Our input to this layer that is each question has dimensions (max question length,1). The output of this layer will be (max question length, embed layer size).

4.2.3 Encoder Module

The output of question module is concatenated with the output of the Image module. The idea of using repeat block for the features and merging features of the image with every word embedding of the question is to make the model learn which word corresponds to what part of the image. We then pass this merged output as input to LSTM.

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter Schmidhuber (1997). LSTMs are explicitly designed to avoid the long-term dependency problem.

LSTMs have this chain like structure, with four neural network layers, interacting in a very special way.

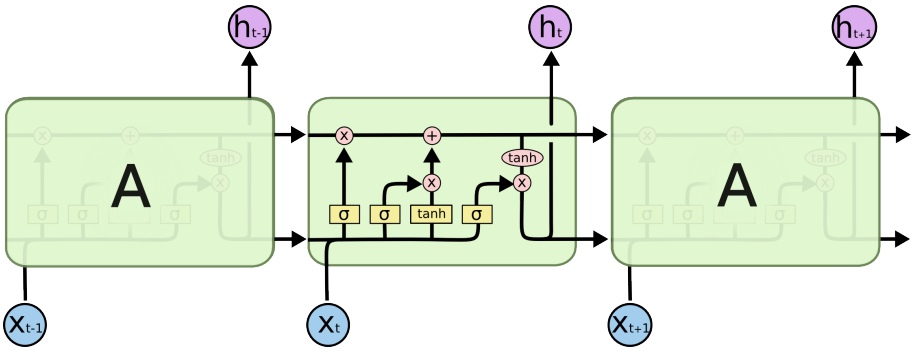


Figure 4.2: LSTM Networks

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer." It looks at ht-1 and xt, and outputs a number between 0 and 1 for each number in the cell state Ct-1. 1 represents "completely keep this" while 0 represents "completely get rid of this".

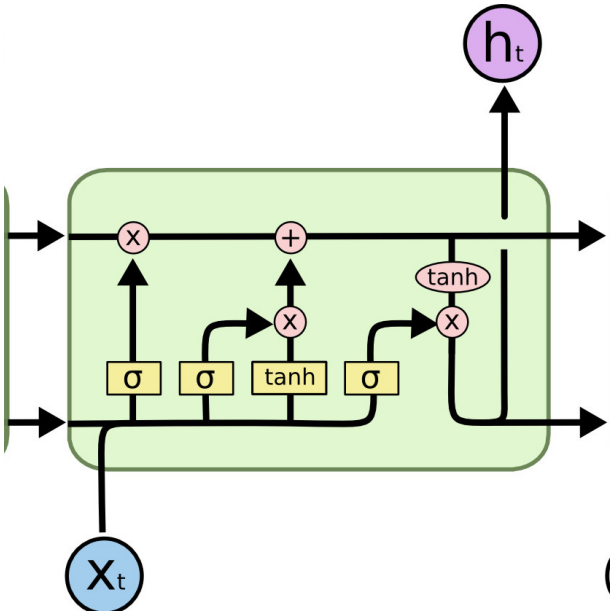


Figure 4.3: LSTM Networks

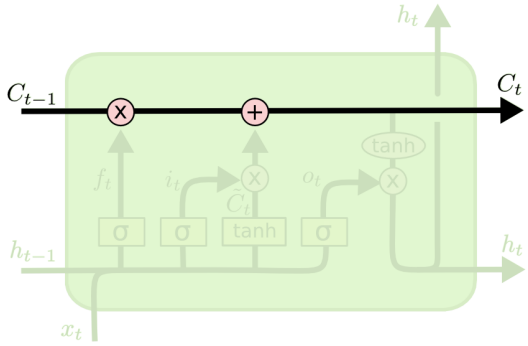
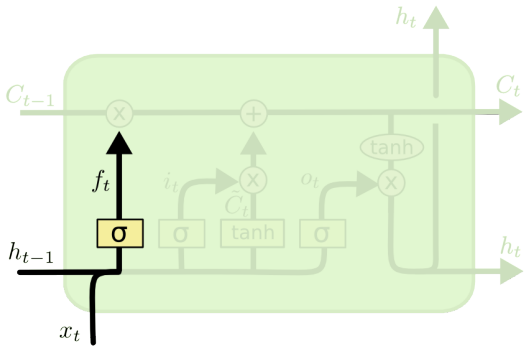


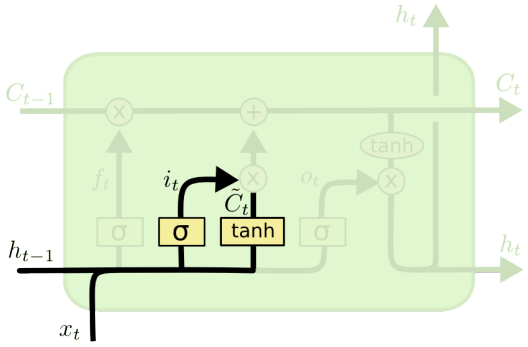
Figure 4.4: Cell State

The next step is to decide what new information we're going to store in the cell state. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a tanh layer creates a vector of new candidate values, C'‘t, that could be added to the state. In the next step, we'll combine these two to create an update to the state.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 4.5: first layer in LSTM



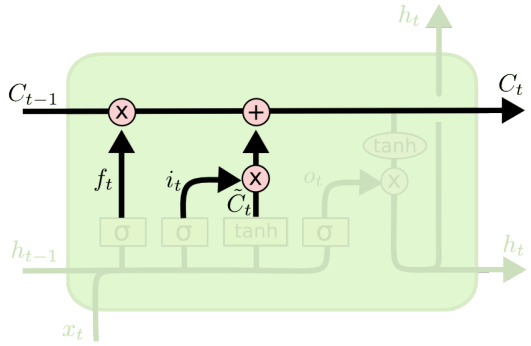
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 4.6: second layer in LSTM

It's now time to update the old cell state, Ct-1, into the new cell state Ct. The previous steps already decided what to do, we just need to actually do it.

We multiply the old state by ft, forgetting the things we decided to forget earlier. Then we add it*C‘t. This is the new candidate values, scaled by how much we decided to update each state value.

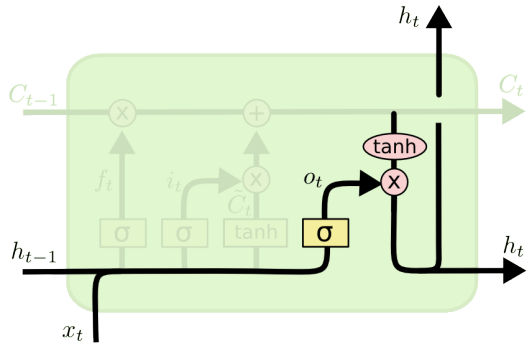
Finally, we need to decide what we're going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh (to push the values to be between -1 and 1) and multiply it by the output of the sigmoid gate, so that we only output



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 4.7: Calculating Ct in LSTM

the parts we decided to.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 4.8: Last layer in LSTM

4.2.4 Decoder Module

The output of the encoder module that is the (last hidden cell state) of the LSTM is passed as the initial state to another LSTM. During training, the outputs of the decoder of the LSTM are probability distributions (over all the words in the vocabulary) generated by the model for the next word in the sentence. The model is trained to minimize the negative sum of the log probabilities of each word. During inference part, to get the next word,

we pick the word with with maximum probability at each time step. The working of this part is clearly illustrated in the baseline architecture figure.

Chapter 5

Implementation and Results

5.1 Implementation

5.1.1 Image Preprocessing

- **Resize**
  Images are resized such that height and width of the target image is equal to 224.

- **Feature Extraction**
  Pretrained VGG-Net model is used for computing the image features. The activations from the last fully connected layer of the model are extracted and used as features . The features generated are of size 4096.

5.1.2 Question Preprocessing

- **Tokenize**
  Our task here is to convert text input sequence into numerical form so that each word in the sequence is represented by a number that can be used to index vocabulary. We start by first parsing the sentence to

remove punctuations, de-capitalize words etc. So, the input sentence of the form: “What does mri show?” -¿ “what does mri show”. Then, we tokenize this sequence to obtain: [“what”, “does”, “mri”, “show”].

- **Vocabulary Creation**
  Our next step is to create a vocabulary. We take the total number of unique words present in the questions and answers of the given dataset which turns out to be the size of 3499 in our case. We have chosen to taken all the words present in QA in the vocabulary instead of taking the most frequent ones. As, the answers contain medical terms, which have low frequency but are relevant in answering the question.

- **Additional Tokens**
  Also, we have added a 4 new tokens: [’¡NULL¿’, ’¡START¿’, ’¡END¿’, ’¡UNK¿’] that denotes “no word”, “start of the sequence”, “end of the sequence” and “unknown word” respectively to the vocabulary.

- **Fixed length sequences**
  Input sequences would be of variable lengths, so the task here is to transform all of them to fixed size. We took max question length and max answer length to be 16. The questions and answers with smaller length are padded with ¡NULL¿ token. So the tokenized input of the form [“what”, “does”, “mri”, “show”] gets converted to [“¡START¿”, “what”, “does”, “mri”, “show”, “¡END¿”, “¡NULL¿”, “¡NULL¿”, “¡NULL¿”, “¡NULL¿”] assuming max question length is 10.

- **Create two dictionaries**
  ‘word_to_idx’ and ‘idx_to_word’. As the name suggests, ‘word_to_idx’ dictionary maps words to index of that word in the vocabulary and ‘Idx_to_word’ does the opposite.

- **Change to numerical form**

Now, we have to transform our sequence to sequence of numbers where each number represents the index of that word in the dictionary. We can do that easily using our 'word_to_idx' dictionary. So our tokenized input ["¡START¿", "what", "does", "mri", "show", "¡END¿", "¡NULL¿", "¡NULL¿", "¡NULL¿", "¡NULL¿"] changes to say, [1, 7, 28, 55, 6, 2, 0, 0, 0, 0].

5.1.3 Training and Modelling parameters

- **Batch Norm:** Batch Normalisation is used to stabilize the network and it helps network converge faster. We applied Batch Normalisation in the encoder module LSTM that is after merging the outputs of question and image module.
- **Drop Out:** It is a regularization technique used to ensure that model does not overfit. The value of dropout parameter is 0.3.
- **Learning Rate:** Learning rate controls how model weights are adjusted with respect to the loss gradient. When learning rate is small, the model trained is more reliable but it takes time to get the optimal parameters as the parameters are updated by a smaller value. The optimiser used by us changes the learning rate dynamically during the training, but we can still set the initial learning rate.

Dropout is applied before getting outputs from the encoder and decoder module to avoid overfitting. Cross entrophy is used as loss function for training the model. Adam optimisation algorithm is used for updating the parameters of the model. TanH function is used as activation function in LSTM units. 20% of the training dataset is used as validation dataset.

The important model parameters and their values are used for the parameters

- Embedding size=150

- Dimension of hidden unit=150
- Maximum question answer length=21
- Learning rate=0.001
- Batch size=100
- No of epochs=10
- Dropout=0.3

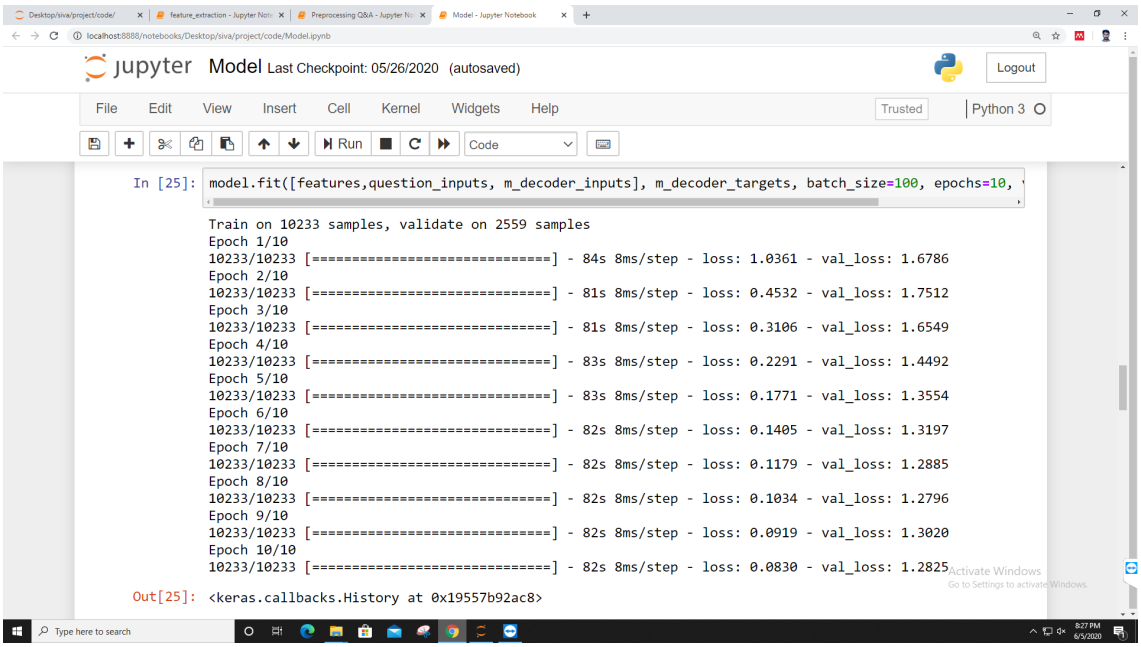


Figure 5.1: Training the model

5.2 Results

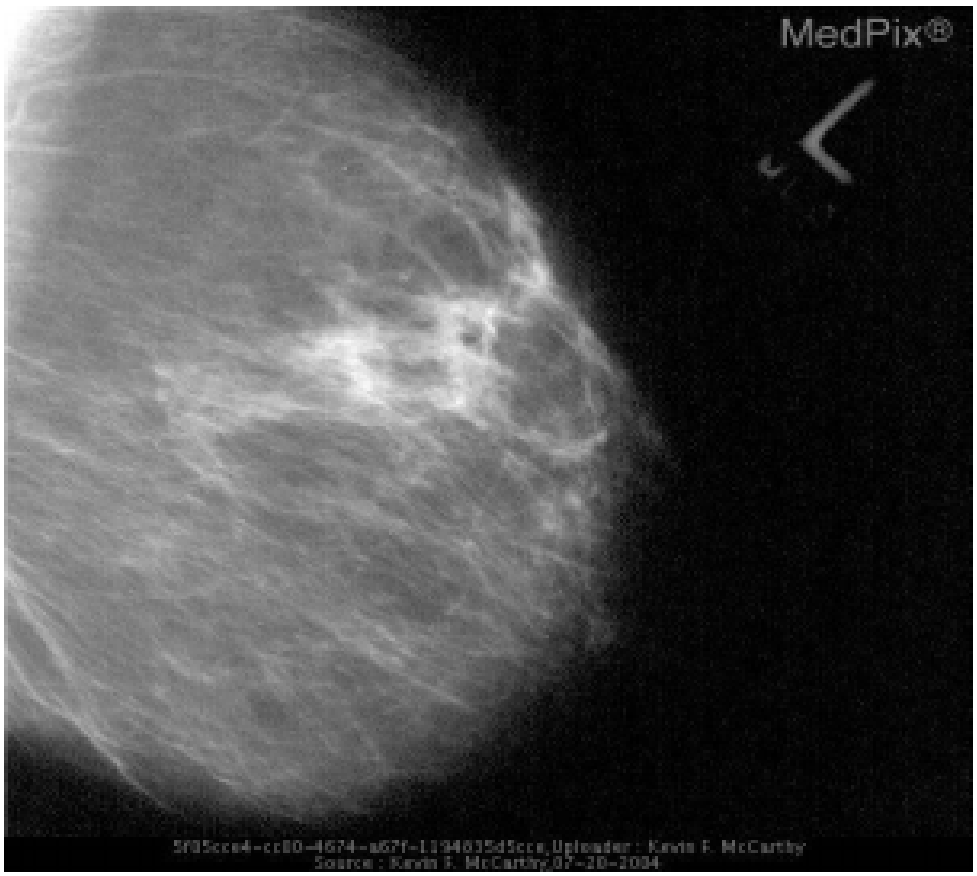
The results of the model are tabulated as follows:

Table 5.1: Results

Sl. No	***Category of data***	***Accuracy***
1	Modality	0.444
2	Plane	0.734
3	organ	0.684
4	Entire data	0.5155

Accuracy of the model increases with increase in the depth of the network. Some of the answers predicted by our model are as follows:

Input:



Question: What kind of image is this?

Output:

Mammograph

Figure 5.2: example1

Input2:

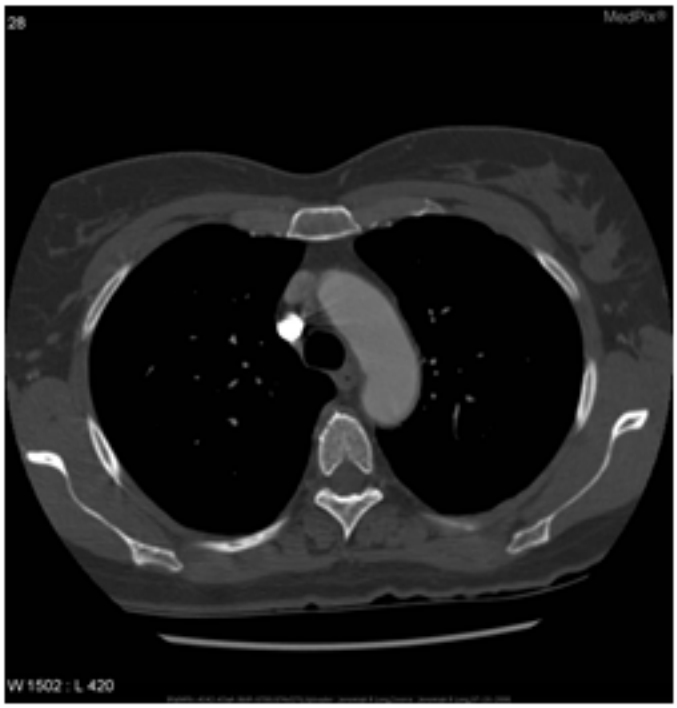
Question: What organ system is present in this image?

Output:

Musculoskeletal

Figure 5.3: example2

Input3:



Question: In what plane is this CT scan?

Output:

Axial

Figure 5.4: example3

Chapter 6

Conclusion and Future work

We reviewed popular methods in deep learning, and built a VQA model for ImageCLEF 2019. We adjusted some learning parameters in our model to increase accuracy. Despite shortcomings of current practices for both training and evaluating VQA systems, we identified a number of promising research avenues that could potentially bring future breakthroughs for both VQA and for the general objective of visual scene understanding. Since deep learning techniques are significantly improving, we can reasonably expect that VQA is going to be more and more accurate in the next years.

References

[1] Y. Zhou, X. Kang, and F. Ren, "Employing inception-resnet-v2 and bi-lstm for medical domain visual question answering"

[2] A. B. Abacha, S. Gayen, J. J. Lau, S. Rajaraman, and D. Demner-Fushman, "Nlm at imageclef 2018 visual question answering in the medical domain" 2018.

[3] J. J. Lau, S. Gayen, A. B. Abacha, and D. Demner-Fushman, "A dataset of clinically generated visual questions and answers about radiology images" Scientific data, vol. 5, p. 180251, 2018

[4] B. Talafha and M. Al-Ayyoub, "Just at vqa-med: A vgg-seq2seq model"

[5] Akira Fukui, Dong Huk Park, Daylen Yang and Anna Rohrbach "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding"

[6] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Smola. Stacked Attention Networks for Image Question Answering. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 21-29

[7] Damien Teney, Peter Anderson, Xiaodong He, Anton van den Hengel.Tips and Tricks for Visual Question Answering: Learnings From the 2017 Challenge. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4223-4232

[8] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel. The VQA-Machine: Learning How to Use Existing Vision Algorithms to Answer New Questions. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1173-1182

[9] Aisha Al-Sadi1, Bashar Talafha1, Mahmoud Al-Ayyoub1, Yaser Jararweh1 and Fumie Costen2 "JUST at ImageCLEF 2019 Visual Question Answering in the Medical Domain"

[10] Jiasen Lu, Jianwei Yang, Dhruv Batra, Devi Parikh "Hierarchical Question-Image Co-Attention for Visual Question Answering"

[11] https://colah.github.io/posts/2015-08-Understanding-LSTMs/ For Understanding LSTM and images used in LSTM.

[12] https://medium.com/ai2-blog/vanilla-vqa-adcaaaa94336 For Understanding Baseline architecture.