Visual Question Answering

Shubhangi Upasani **Divyansh Kumar Roy** **Monica Gupta**

Georgia Institute of Technology
{shubhangi.upasani, droy41, mgupta334}@gatech.edu

Abstract

Visual Question Answering [1][2][3] is a multi-faceted problem wherein unified vision and language processing is applied to answer basic, common sense questions pertaining to an image. We have successfully achieved our goal of designing a VQA model that leverages both visual and textual cues as opposed to a model that predominantly relies on one of the input modalities. In doing so, we designed several model architectures, a fusion-based model that uses a novel self-conceived fusion strategy; a parallel and alternate co-attention based model and an architecture that combines these co-attention techniques with the self-conceived fusion technique. Our experiments with diverse attention mechanisms, neural network models and word embeddings have been fruitful. Our best model (alternate co-attention) gives a validation accuracy of 47% compared to the second best model (parallel co-attention) that gives a validation accuracy of 44% . These models perform way better than our hand-designed baseline model but fall short compared to our stronger baselines which have been able to achieve state-of-the-art performance in VQA. The reasons for this have been explained well in the following sections. The code for our project is hosted at the following link: https://github.com/DivyanshRoy/CS7650-project-vqa
The video for our project is hosted at the following link: https://drive.google.com/open?id=1vwYaNXBy7Gq5jYjZxcR8lEQujti8MLAb

1 Introduction and Related Work

VQA is essential in building truly intelligent AI. It has applications in scenarios where human-AI collaboration is required. VQA systems can potentially be used to aid visually impaired users, extract information from satellite data, interact with home robots and assist users in shopping online. The goal of our project is to design a VQA model that effectively leverages both textual and visual cues to answer questions about a given image accurately. The model would have to capture general knowledge and semantic understanding in order to correctly answer these questions. This has been achieved by using attention mechanisms that give appropriate weights to the images and questions based on their semantic relevance thereby producing coherent answers.

Related Work: The availability of relevant large-scale datasets ignited VQA research, which has accelerated in the past decade. VQA in its full-form is still an active research problem. Several existing works inspired the undertaking of our project. The EvalAI VQA challenge attracts top of the line VQA models every year. The winners of the 2019 challenge [8] proposed an interesting self and guided attention approach that used several stacked modular co-attention layers. This work advocates experimentation with different variations of attention and designing model architectures with mixed attention mechanisms. The MCAN implementation served as a guiding beacon for our work.

J. Lu et al. [4] proposed a novel hierarchical question-image co-attention approach for VQA. They applied co-attention mechanisms recursively on several levels of the question embedding namely, words, phrases and sentences. This was done in concurrence with the image features. The end-result was a top-down hierarchy of attention-weighted question and image features which were recursively combined to get the final answer predictions. They described two approaches for co-attention, specifically parallel and alternate co-attention which we have adapted for our project as well. Additionally, we implemented this approach in unison with our novel fusion technique for VQA in an attempt to address the gap between applying attention on images and questions independently versus using one as an attention-guide for another.

In addition, we followed [1] for our CNN+LSTM baseline model. In this research paper, S.Antol et al. propose a vanilla VQA model that uses a CNN model to obtain image features and an LSTM for question features. These features are then multiplied point-wise to transform them into a common feature space in order to predict the final answer. Such an approach has widely become a benchmark for VQA in recent times and we decided to base our baseline model on it. We also explored [9] as a possible avenue, wherein feature pyramid networks are introduced as an object detection routine capable of detecting objects across all scales and can be employed as a way of providing visual attention. [5][7][10][11] are also some of the related studies we went through for our project. Our models try to use both independent (wherein images and questions are treated separately and given attention) and co-dependent (wherein images and questions are given attention using each other as attention guides) regimes in unison and combine them using our novel fusion strategy. We refer to this model as Fusion+Co-Attention. We noticed a sharp increase in the learning capacity of this model as it gave the highest train accuracy relative to all models we trained. However, the validation accuracy for this model was not that great because of overfitting which was very difficult to control. Nonetheless, this experiment helped us in validating the usefulness of combining the two regimes. We feel that by fine-tuning our fusion strategy and using more data, we can get the best possible performance for VQA from the Fusion+Co-Attention model.

2 Methods

2.1 Data

We used the publicly available VQA 2.0 dataset. The imges included in the dataset come from Microsoft COCO dataset. On an average, three questions have been asked per image and the answers include one word answers like ’yes’, ’no’, ’1’, ’2’,’yellow’, ’red’ etc. The most common answers among ten answers with count above a certain threshold are selected as the ground truth. The table below summarizes some statistics for the VQA 2.0 dataset.

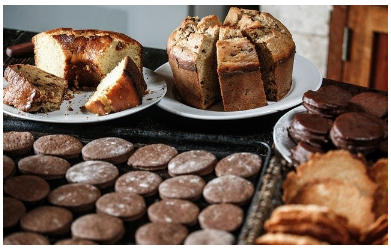
	Images	Questions	Answers
Train Split	82,783	443,757	4,437,570
Val Split	40,504	214,354	2,143,540
Test Split	81,434	447,793	NA

Table 1: Dataset description

Some examples form the dataset are shown below.

Question: Are these food items mini-pizzas?
Answer: Yes

Question: What are the walls made of?
Answer: Wood



Question: How many donuts are on the rack?
Answer: 0

Figure 1: Image, Question, Answer triplets from the dataset

Data Collection: After performing some analysis, we decided to create our own train/validation split from the VQA dataset described above. We created a subset of the VQA 2.0 dataset having 104 unique answer classes such that the number of examples (images) belonging to each class is at least 100. Additionaly, we made sure that each answer class had no less than 100 and no more than 1000 examples each. This helped us obtain a well-balanced dataset to combat the class-imbalance problem (illustrated below) that we faced during early phases of our project. The VQA dataset is such that Top-1000 answer classes alone constitute nearly 85% of the data. Getting a subset also helped us in getting a manageable amount of data that we could train our models with given the available resource and time.

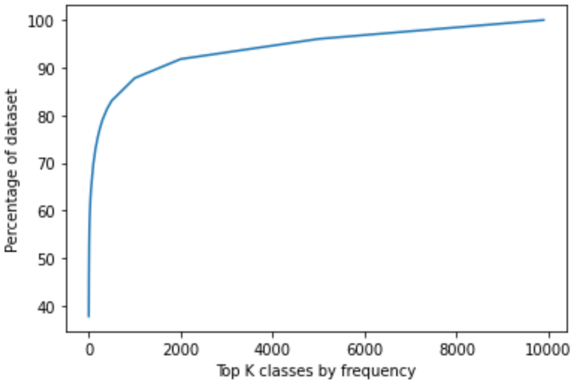


Figure 2: Percentage of dataset represented by Top-k classes sorted by frequency.

We significantly increased the scale of our project as compared to the midway report, wherein we used only 5000 training examples. The dataset now has 32498 examples in the training split and 20283 in the validation split. Each example in the data constitutes an image, question and answer triplet. The answer belongs to one of the 104 possible answer classes.

We resized all images to dimensions 224x224x3 prior to passing them through a ResNet18 to obtain image features. We also used a VGG-16 network to get spatial maps of images. All questions were padded to have a uniform length of 25 words.

We experimented with two different kinds of models (explained in the later sections). These models call for image features in different forms. Hence, we followed two different regimes for processing our image features. The first regime involves obtaining image vectors from a pre-trained ResNet18 network. Consequently, each image is represented by a vector of length 152. The second regime involved extracting spatial feature maps of images obtained from the last convolutional block of a pre-trained VGG-16 network. Every image is thereby represented by a 3D volume of dimensions 512x14x14. In other words, each image is represented using 512 spatial feature maps of size 14x14.

We converted the answers to class labels and in doing so, we converted our problem to that of multi-class classification. Each answer is assigned a number from 0 to 103, corresponding to the 104 unique answer classes.

2.2 Models and Analysis

Image features are obtained by passing the input images through a ResNet. Textual features from the questions are obtained using a combination of embeddings and LSTM. As mentioned above, we experimented with two different kinds of models and were able to create a third variant by combining them together. These models have been described in detail below.

2.2.1 Fusion model

Our fusion model merges image features obtained from an image classification model such as ResNet-18 with question features from an LSTM using a combination of feature space transformations and pointwise multiplications. Specifically, the image features (n, i) are transformed to the question feature space (n, q) and then it undergoes a pointwise multiplication with the question features. The result of these operations is transformed to a shared feature space (n, s). A similar set of operations are performed on the question features. The results of these transformations on the image and question features undergo pointwise multiplication to complete the fusion step.

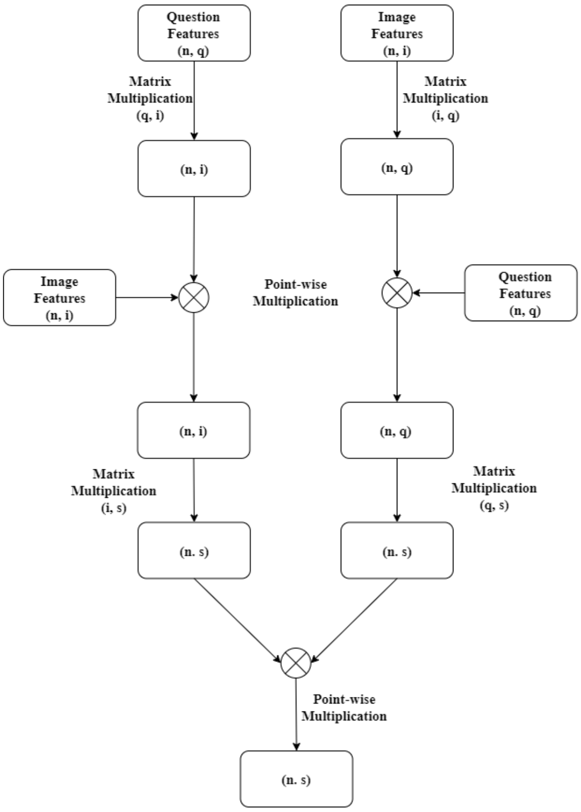

Question Features (n, q)
Image Features (n, i)
Matrix Multiplication (q, i)
Matrix Multiplication (i, q)
(n, i)
(n, q)
Image Features (n, i)
Point-wise Multiplication
Question Features (n, q)
(n, i)
(n, q)
Matrix Multiplication (i, s)
Matrix Multiplication (q, s)
(n. s)
(n. s)
Point-wise Multiplication
(n. s)

Figure 3: Fusion model which combined image and question features

The (i, q) matrix is significant because it can be used to represent the importance of an image feature with respect to a question feature. The rest of the steps are necessary for the fusion of features with different feature sizes.

2.2.2 Co-Attention models

These models are loosely based on [4]. The models focus or "co-attend" to both questions and images opposed to simply focusing on one input modality. They attempt to decide "where to look" in an image and "what to listen to" in a question to produce the answer. Two broad architectural forms for the co-attention model were adapted from [4] as discussed below. These models do not use feature hierarchy as described in the original work.

1. **Parallel Co-Attention:** This variant of the co-attention model builds attention over the image and the question concurrently. The image spatial maps and question embeddings are subjected to a series of transformations to obtain the final attention maps. The attentions weights are obtained and subsequently applied to the images and questions independently of each other.

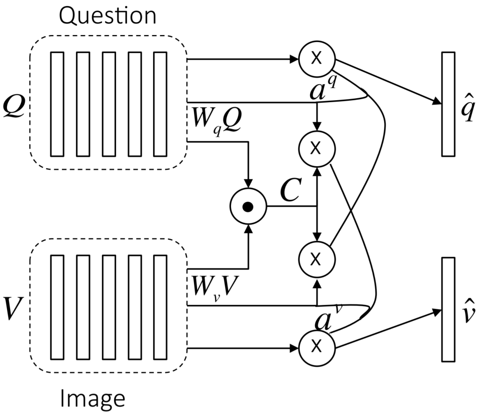


Figure 4: Parallel Co-Attention model (Source:[4])

2. **Alternate Co-Attention:** This variant alternates between paying attention to the image and the question. The underlying idea is to use one entity as the guiding factor to give attention to the other entity. First, the question is taken as the guiding factor to give attention to the image and then the images guides the attention given to the question. To summarise, attention weights for one entity are obtained using the other.

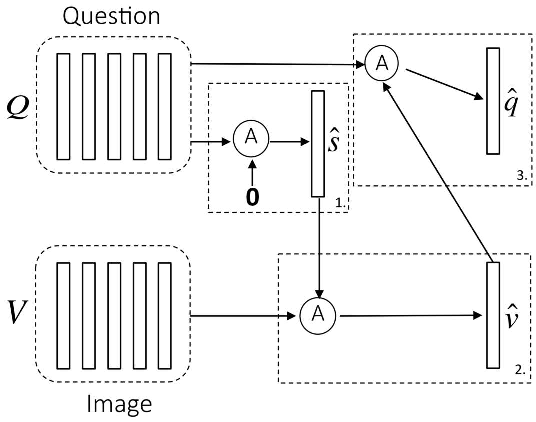


Figure 5: Alternate Co-Attention model (Source:[4])

Upon obtaining the attention weights, the input modalities (images and questions) are multiplied by their respective weights to get attention weighted modalities. These are then fed into an answer generation unit that combines them and passes the result through a multi layer perceptron to get the softmax predictions.

The parallel and alternate co-attention models were trained separately on the data and a comparison of their performance is included in the results section below.

2.2.3 Co-Attention+Fusion model

Our Co-attention+Fusion model engages the fusion strategy implemented in our fusion model, in the answer generation part of the co-attention models. The fusion mechanism is used to combine the attention weights for images and questions obtained from the parallel and alternate co-attention models. These combined weights of the two models are in turn fused again to obtain a single aggregated representation, which is used to predict answers.

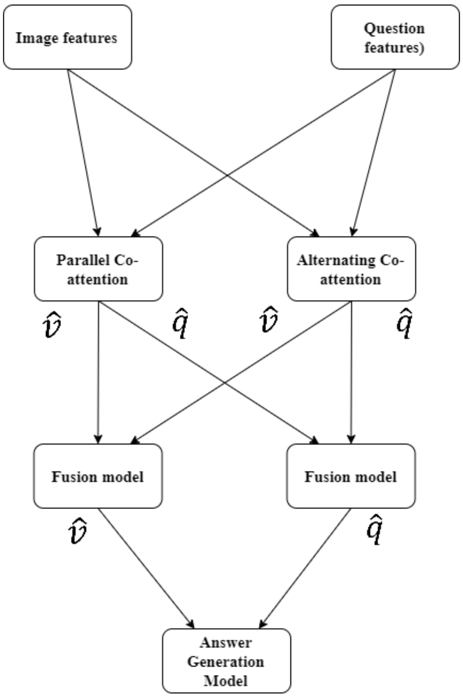


Figure 6: Fusion model combining features of Parallel and Alternate Co-Attention models

2.3 Baseline Models

The baseline used in our project is a simple CNN+LSTM baseline. The question embeddings are first passed through an LSTM. The question features obtained from these embeddings are multiplied in a pointwise fashion with image features. These image features are obtained using Resnet18 (as mentioned in the data collection section above). The combined feature vector is then fed into an MLP to get softmax predictions. [1]

We also use [4] and [8] as our baselines. These are eminent works in the field of VQA and serve as a tight baseline for any endeavour in this field. The results obtained from our models are compared with all three baselines.

3 Results

3.1 Experiment Setup

Our data, as mentioned before has about 32k training examples and 20k examples in the validation set. We attempted to make it well balanced by restricting the number classes to those that had at least 100 distinct examples (images) belonging to them. At the same time, we made sure that the number of examples that a class could have were no more than 1000. This helped in reducing the errors we were getting earlier due to class imbalance.

We experimented with five different kinds of models, namely the fusion model which uses our fusion strategy, co-attention models (both parallel and alternate co-attention that use a non-hierarchical representation of inputs), a combination of fusion, parallel and alternate co-attention models and our baseline model (CNN+LSTM). These models were implemented and trained from scratch and perform reasonably well, if not better, when compared to our stronger baselines.[4][8]

The experiments were conducted with a batch size of 1000, 3000 and 5000 on our training and validation data. Initially, we experimented with learning rates between 1e-3 and 4e-4. The fusion+attention model required a relatively lower learning rate than the fusion and attention models separately. We worked with different optimizers such as Adam and RMS Prop. We also used an LR reducer on Plateau with Patience=3, Cooldown=0 and decay rate as 0.3 while monitoring the validation loss. This helped in obtaining significant improvements over our previous results. For the loss functions, we switched between Cross entropy loss and Focal loss. A significant challenge while training was to control overfitting of the models. This was done using regularization techniques like dropout, larger batch sizes, adaptive learning rates and batch normalization. Further details about hyperparameters are mentioned below.

3.1.1 Focal Loss

In datasets with a large class imbalance the cross entropy loss contribution is largely comprised of negative examples from easily classified examples. Focal loss increases the focus on examples that are harder to train by down-weighting the loss from examples that are easy to classify. Focal loss adds a factor of $(1 - p_t)^\gamma$ where γ is a hyperparameter that can be tuned.

Focal Loss$(p_t) = -(1 - p_t)^\gamma \log(p_t)$

When $\gamma = 0$, focal loss becomes equal to the cross entropy loss.

3.1.2 Cross Entropy Loss

Cross Entropy loss is used for measuring the performance of a model for multi-class classification tasks. It takes softmax scores represented by probability values for each class in the range [0,1] and tries to maximize the score for the correct label. The loss increases as the predicted labels diverge from the true labels.

$$L(\hat{y}, y) = -\sum_i (y_i, \log \hat{y}_i)$$

3.2 Result Comparison

We compared our models based on the validation accuracy achieved on our validation set. The results are summarized below. All accuracy values are in percentages.

	Train Accuracy	Validation Accuracy
Fusion model	60.9	44.1
Parallel Co-Attention model	73.8	44.8
Alternate Co-Attention model	58.3	47.5
Fusion+Parallel+Alternate Co-Attention model	75.3	42.3
Baseline (CNN+LSTM) model	59.4	35.3

Table 2: Performance comparison of models

We observe that the best performance is achieved by the alternate co-attention model. This is because when giving attention to one entity (image or question), it uses the other entity as the guiding factor and is able to leverage the semantic relatedness between the image and question. The parallel co-attention and fusion model perform comparably. The parallel attention model is effective in giving attention but its limitation is that it considers the two entities separately while doing so. The fusion model is based on transforming the image and question to a shared space. In doing so, it considers the other entity as a guidance (just like alternate attention model). Interestingly, the performance of the fusion+parallel+alternate attention model is lower than that of the alternate and parallel attention models. This is because it began overfitting at a very early stage during training. However, it depicts the maximum learning capacity. Our baseline model is based on simple pointwise multiplication of the image and question embeddings (close to vanilla VQA model) and was thus expected to show limited performance.

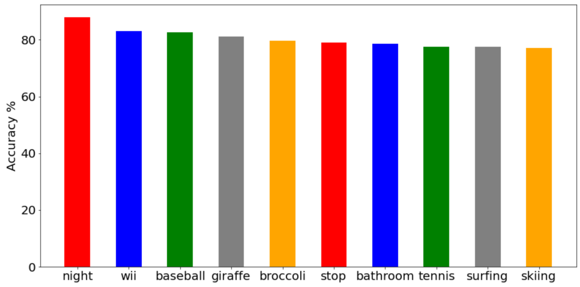
Our models performed better than our baseline i.e. CNN+LSTM model. However, our co-attention models didn't perform as well as the co-attention models described in [4]. This because they used a hierarchical representation of questions and images and applied attention on both images and questions at three different levels. The attention weights were then combined recursively to get the final weights. In addition to this, the model was trained on the entire train-dev split of VQA-v1 dataset. It was not possible for us to train our models on so much data due to time and computational limitations. The hierarchical model described in [4] achieved state of the art performance in 2016 by achieving 66.1% accuracy on the standard test split of the VQA challenge for multiple choice correct questions and 62.1% for open-ended questions. The MCAN implementation [8] achieved 70.63% accuracy in the 2019 VQA challenge on the test-dev set. They used stacked attention layers which had both self and guided attention mechanisms and used transformers for training their model. Training transformer models is a time consuming task and is heavy on compute. Therefore, we decided not to venture in this direction any further in the interest of time.

We tested various hyperparameters like the learning rate and learning rate decay, batch size, dropout, question embedding dimensions, image features sizes and types (spatial feature maps and vectors). We chose these parameters as they had the maximum influence on model performance and were of greatest importance to the model architecture. We also tested various loss functions and optimizers and played around with their hyperpaprameters like weight decay, momentum and other parameters central to the optimizers we were using (like α, β, γ values). In addition to this, we also experimented with 1-D batch normalization over embeddings to make the neural network more stable.

Our goal was to explore the importance of attention in VQA. We wanted to understand the interdependence of question and images input modalities and conduct experiments that provide evidence for our initial hypothesis that just focusing on the image or the question is not sufficient for VQA. The model must learn to consider both input modalities simultaneously in order to be able to produce accurate results. The results concluded above are a clear proof for our hypothesis. Our baseline model considers the question and image separately and does not try to capture the interdependence between them. In addition, it gives equal focus to all parts of the image and question. The attention models try and gauge the most relevant parts of the given inputs and perform better than the baseline. The results obtained by the attention model are not spectacular when compared to our stronger baselines [4][8] but they provide evidence for the fact that capturing the interdependence between question and image and focusing on their relevant parts is a more successful strategy for VQA.

A detailed analysis of our models made their inherent lack of generalizability apparent. The models were more or less memorizing the answers to the question and could generalize to unseen data only to a certain extent. They were leveraging the hidden priors in language and visual cues. Detailed diagnostics (loss and accuracy plots) for all models are included in the appendix. This is a well-known problem in VQA and is still an active research area. We feel that one way of adding more generalization power to the model is to try more elaborate and extended neural network structures like differential neural networks or deep modular co-attention networks that have greater learning capacity. Another way is try data augmentation techniques like generating (image, question, answer) triplets from the available data such that only the answer differs while the question and image is the same. This prevents the model from learning the input priors while at the same time being able to generalize better. [13]

We illustrate the top-10 and bottom-10 classes ranked by accuracy below:



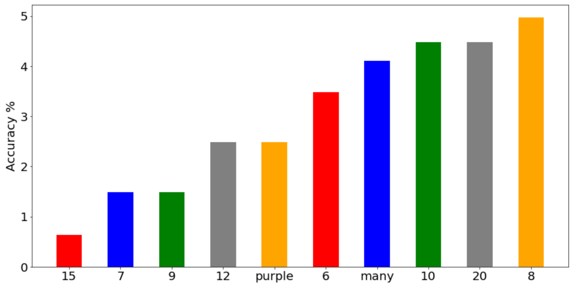


Figure 7: Top-10 and bottom-10 classes ranked by validation accuracy

The plots show that the model performs better on answers whose corresponding images have distinct visual features, implying that the model has a huge dependence on the vision module for its classifications. It can be seen that the model struggles to answer questions whose answers are large numbers so the model has a difficult time answering questions where it is expected to count objects in images.

3.3 Work Division

Divyansh: Created the larger version of the dataset, implemented and trained the fusion model and report preparation.
Shubhangi: Implemented and trained parallel and alternate co-attention models, trained fusion+co-attention model and report preparation.
Monica: Implemented and trained a hierarchical co-attention model, report and video preparation.

4 Conclusion

Some low-level inferences are discussed below:

1. The way question embeddings are learnt do not directly affect the performance of the system. We tried using pretrained Glove embeddings of varying sizes as well as generating embeddings from scratch. The model performance was not significantly affected by these methods of learning embeddings. This is probably because we trained the model for nearly 250-300 epochs which gave the model sufficient time to learn new and meaningful embeddings. Moreover, pretrained embeddings have a limited vocabulary, which led us to decide to learn our own embeddings from scratch.

2. Attention models are prone to overfitting. We had to constantly monitor the models to check for this. Several prior studies cement this conclusion [13]. They suggest that after training for a while, models begin to memorize language and visual priors and show a strong dependence on them for their performance. A well known example is when a VQA model is asked ”what is the color of the banana shown in the picture”, it would answer ”yellow” almost all the time, even if the banana in the image is, say green in color. The answer to this question is in fact ”yellow” in a majority of the cases. Training VQA models, especially attention-based VQA models require constant examinations for such cases and for their generalizability.

Some high-level conclusions that can be drawn from our work are as follows:

1. Attention mechanism considerably improve the performance of VQA models. They learn to focus on both image and questions and venture into learning how a human would answer a question about an image. They have the power to model human intuition while answering these questions.

2. However, these models are also prone to memorizing the answers and have to be equipped with generalizability. The intermediate representations of the models must be analysed to examine that they are indeed learning. One must make sure that they are learning the relevant parts of the inputs.

3. Overall if trained with abundant data and compute, these models do reflect the capability to make semantic sense from the inputs given. They have tremendous potential of being employed in numerous beneficial applications.

We find that the models we trained give a decent performance on our train/validation split. They are nowhere close to the established state of the art performance but they were able to beat our hand-designed baseline by a huge margin. A few reasons for that is firstly that the state of the art models have been trained on entire VQA and MS-COCO datasets. It was not possible for us to use so much data but we still utilized every ounce of computational resources we had to train our models on a considerably large subset of the data. Considering this, we conclude that our models perform reasonably well. Another reason for limited performance is the memorizing tendency of models. This has been discussed previously.

For future improvement, we can try expanding the scale of data used by using more existing data or adding to the existing data through augmentation. One can also experiment more with our fusion strategy. Instead of using pointwise multiplication, we can use other techniques like bilinear pooling [12] or convolution. Another way of extending the existing implementation is to use a hierarchy of input features that focus on both micro and macro details to harness them individually.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick and D. Parikh, “Vqa: Visual question answering,” in IEEE ICCV, 2015.

[2] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, “Stacked attention networks for image question answering,” in IEEE CVPR, 2016, pp. 21–29.

[3] D. Teney, P. Anderson, X. He, and A. van den Hengel, “Tips and tricks for visual question answering: Learnings from the 2017 challenge,” in IEEE CVPR, 2018, pp. 4223–4232

[4]J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In NIPS, 2016.

[5] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh, “Pythia v0. 1: the winning entry to the vqa challenge 2018,” arXiv preprint arXiv:1807.09956, 2018.

[6] J. Liang, L. Jiang, L. Cao, L.-J. Li, and A. G. Hauptmann, “Focal visualtext attention for visual question answering,” in IEEE CVPR, 2018, pp. 6135–6143

[7] C. Wu, J. Liu, X. Wang, and R. Li, “Differential networks for visual question answering,” AAAI 2019, 2019.

[8] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[9] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In CVPR, 2017.

[10] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In CVPR, 2017.

[11] Tan, Hao, and Mohit Bansal. ”Lxmert: Learning cross-modality encoder representations from transformers.”, arXiv preprint arXiv:1908.07490, 2019.

[12] Yu, Zhou, et al. ”Multi-modal factorized bilinear pooling with co-attention learning for visual question answering.” Proceedings of the IEEE international conference on computer vision. 2017.

13 R. Cadene, C. Dancette, H. Ben-younes, M. Cord, and D. Parikh. Rubi: Reducing unimodal biases in visual question answering. arXiv preprint arXiv:1906.10169, 2019.

Appendix

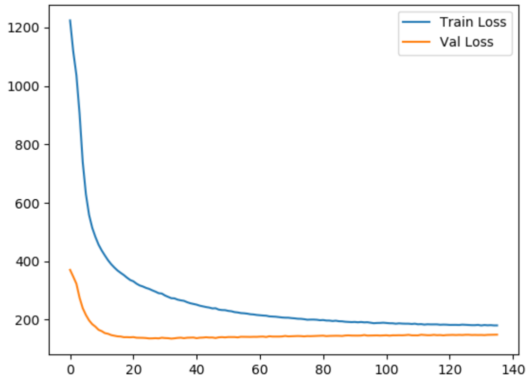


Figure 1: Fusion model Loss curve

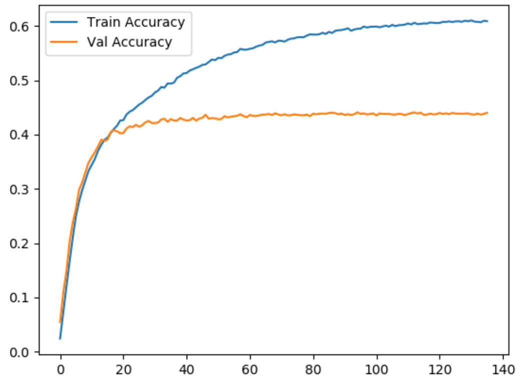


Figure 2: Fusion model Accuracy curve

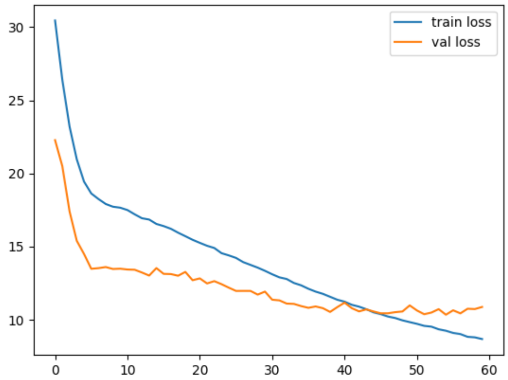


Figure 3: Alternate Co-Attention model Loss curve

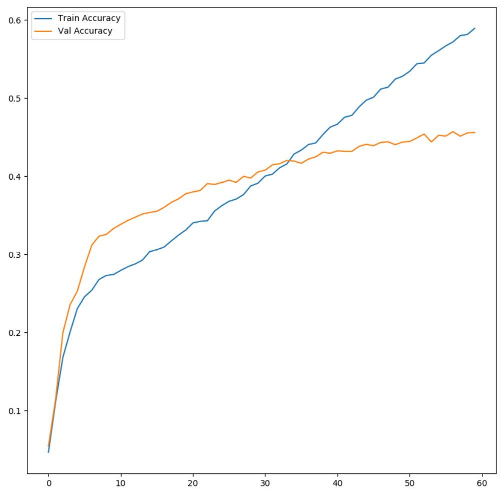


Figure 4: Alternate Co-Attention model Accuracy curve

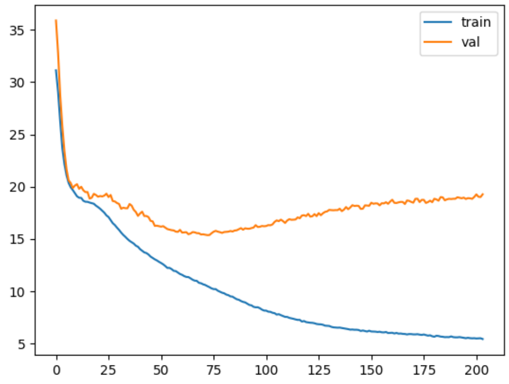


Figure 5: Parallel Co-Attention model Loss curve

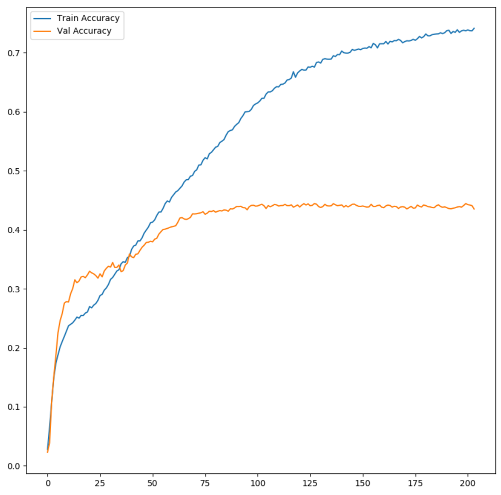


Figure 6: Parallel Co-Attention model Accuracy curve

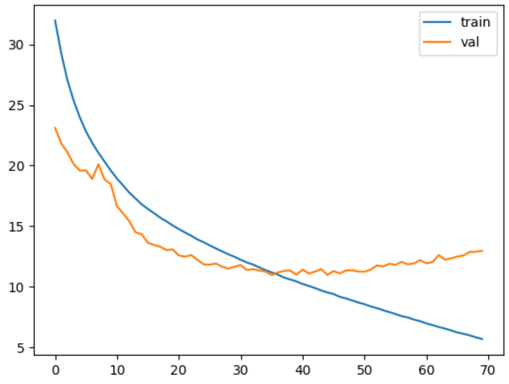


Figure 7: Fusion + Co-Attention model Loss curve

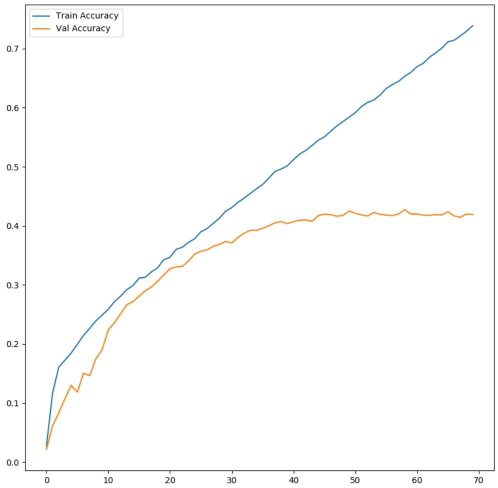


Figure 8: Fusion + Co-Attention model Accuracy curve

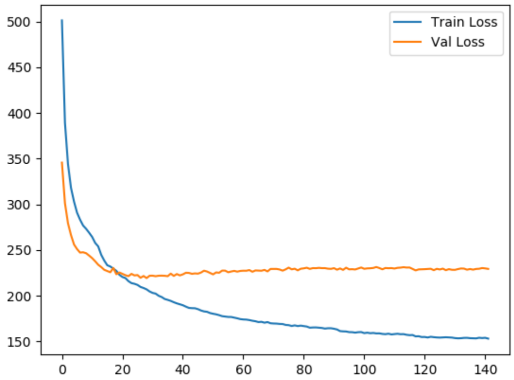


Figure 9: CNN + LSTM model Loss curve

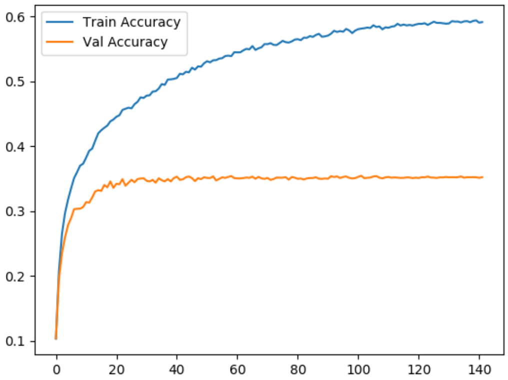


Figure 10: CNN + LSTM model Accuracy curve