Bridging the Gap: Uniting Information Retrieval and Visual Question Answering

DEBALEEN DAS SPANDAN* and S.M. FAIAZ MURSALIN*, Missouri State University, USA

Visual Question Answering (VQA) is a challenging task that requires providing accurate natural language answers based on both an image and a textual question related to that image. Situated at the intersection of computer vision, natural language processing, and information retrieval, VQA tasks machines with understanding both textual queries and the visual content they reference. Traditionally, information retrieval involves text-based queries for finding relevant documents. In VQA, the query is more complex, comprising a textual question and an image. The system must comprehend both components to provide a meaningful response. Just as traditional systems rank documents based on relevance, VQA systems evaluate potential answers based on how well they address the question given the visual evidence. However, VQA introduces the challenge of multimodal integration, requiring effective fusion of textual and visual information to generate accurate answers. This demands sophisticated processing to align and interpret both modalities cohesively. In this project, we are trying to build a system for visual question answering. The key idea is to use the image features as a vector and the input question as another vector and to form a new vector which represents the answer. To this end, we explored training 6 different architectures on a large-scale standard VQA dataset. Initial results have highlighted the challenges of overfitting and generalization, indicating directions for future research in advanced regularization techniques and the integration of large-scale pre-trained models to enhance the system's performance and adaptability.

1 LITERATURE REVIEW

1.1 Overview of Visual Question Answering

Visual Question Answering (VQA) is an interdisciplinary field that synergizes the principles of computer vision, natural language processing, and information retrieval to interpret and answer questions about images. This dynamic field leverages advancements in machine learning, deep learning, and artificial intelligence to address complex visual and textual understanding tasks [6, 15].

1.2 Methods and Knowledge Integration in VQA

VQA research has predominantly utilized deep learning architectures due to their efficacy in handling large-scale data and extracting intricate features. Early approaches combined Convolutional Neural Networks (CNNs) for image understanding with Recurrent Neural Networks (RNNs) for processing textual input [15]. The field has seen significant integration of attention mechanisms, dynamically focusing on relevant parts of the image in response to the query, which enhances the model's interpretative accuracy [8, 10, 17].

Moreover, the expansion into knowledge-aware VQA introduces complex challenges that involve world knowledge about named entities present in images. The KVQA dataset, for instance, requires multi-entity, multi-relation, and multi-hop reasoning over knowledge graphs, making it necessary to understand both visual content and associated world knowledge [13]. This approach not only tackles conventional visual questions but also questions that necessitate deep, contextual understanding and reasoning beyond the visual elements [9, 18].

1.3 Challenges in VQA

Despite significant advancements, VQA faces numerous challenges that hinder its development. One of the primary challenges is the visual and linguistic ambiguity, which complicates the interpretation and correlation of questions with image contents [1, 4]. Furthermore, dataset biases and the lack of diversity in training data lead to overfitting, thereby limiting the generalization capabilities of VQA systems [10].

Authors' address: DEBALEEN DAS SPANDAN, debaleen0010@missouristate.edu; S.M. FAIAZ MURSALIN, sm943s@missouristate.edu, Missouri State University, Springfield, Missouri, USA.

1.4 Datasets for VQA

The evolution of datasets in VQA has played a crucial role in advancing the field. Initial datasets like DAQUAR were limited in scope and diversity. This was addressed by subsequent datasets such as the VQA dataset and COCO-QA, which expanded both the variety of images and complexity of questions [4]. More recently, datasets like GQA and Visual7W have introduced more structured and challenging setups to push the envelope of what VQA models can understand and respond to [6, 17]. The introduction of the KVQA dataset further extends this by requiring knowledge-based reasoning to answer questions about named entities [13].

1.5 Future Research Directions

Looking ahead, there are multiple promising directions for VQA research. Enhancing the model's ability to generalize to new, unseen datasets and reducing reliance on annotations through semi-supervised learning are pivotal [9, 15]. Additionally, integrating multimodal data, such as audio and text, to create more comprehensive models that mirror human sensory and cognitive abilities is another exciting frontier [9]. Finally, improving interpretability and trustworthiness of VQA systems by developing methods that provide explainable and justifiable answers is essential for real-world applications [1, 17].

As VQA continues to evolve, the integration of more sophisticated models, the development of less biased datasets, and the exploration of new applications are critical to overcoming current limitations and expanding the field's impact. In our work, we focus on building an information retrieval system using visual question answering, leveraging advances such as the Focused Dynamic Attention model and the KVQA dataset to enhance our understanding and interaction with visual and textual data. The proposed methodology is described in Section 2.

2 PROPOSED METHOD

Our Visual Question Answering (VQA) system is predicated upon the interplay between image processing and natural language understanding. The core methodology, depicted in Figure 1, involves the transformation of both

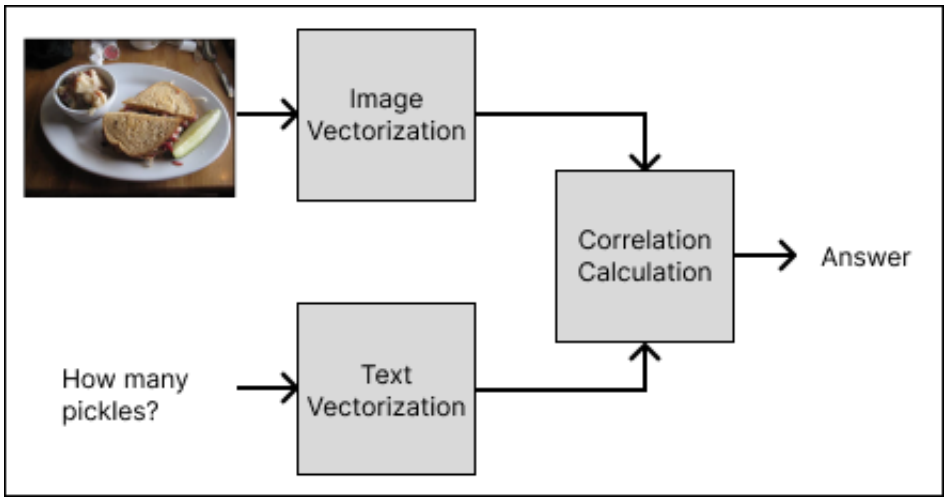


Fig. 1. Proposed Method

visual and textual inputs into vectorized forms, enabling a computational comparison and correlation. Below, we delineate the sequence of operations as illustrated in the figure.

2.1 Image Vectorization

The process begins by converting the input image into a vector representation. We propose the use Convolutional Neural Networks (CNNs) to extract the features form the image input and use the features as a vector representation. These networks excel in image analysis by capturing complex visual features through hierarchical layer structures. We utilize models such as VGGNet [14] and InceptionV3 [16], pre-trained on the extensive ImageNet dataset [3], to derive high-dimensional vectors from the images, encapsulating the critical visual elements in what we term the *"image vector"*.

2.2 Text Vectorization

In parallel, the textual question undergoes NLP to transform it into a *"question vector"*. Starting with tokenization and proceeding with embedding the tokens into a dense vector space using embeddings like Word2Vec [11] or GloVe [12], we capture the semantic meanings based on the context within a broad language corpus. Sequential processing of these vectors through Recurrent Neural network (RNN) architectures, such as Long Short-Term Memory (LSTM) [7] or Gated Recurrent Unit (GRU) [2], results in a composite vector that encapsulates the question's context and meaning.

2.3 Multimodal Correlation Calculation and Answer Prediction

With both the image and question cast as vectors, our system assesses their correlation to produce an "answer vector." This multimodal fusion is important, merging the vectors using various techniques to align the image features with the question's context. The answer vector is refined through fully connected layers, culminating in a softmax layer that outputs a probability distribution over potential answers.

Through the described operations, our system integrates the disciplines of computer vision and NLP to answer questions about the contents of images, showcasing a nuanced understanding of both visual perception and language interpretation.

3 IMPLEMENTATION

In this section, we elaborate on the implementation details of the methodology we proposed in Section 2. We employed the VQA dataset v2, a standard dataset that is openly available for Visual Question Answering tasks [5].

3.1 Image Vectorization

As previously discussed in Section 2.1, we used Convolutional Neural Networks (CNNs) to extract features from image inputs, which we then employed as vector representations. In this research, we utilized three distinct CNN architectures: the standard pre-trained VGG19 and InceptionV3, along with a proprietary architecture. Our architecture comprises of the encoder portion of an Autoencoder, as depicted in Figure 2. We use the output of this encoder as the vector representation of the images. The subsequent subsection details the text vectorization process that we developed to construct vectors for the questions and responses.

3.2 Text Vectorization

In this section, we detail the implementation of text vectorization process. For extracting *"question vectors"* from the input questions, we performed tokenization and subsequently embedded these tokens into a dense vector space. As we detailed in Section 2.2, we processed these vectors sequentially using established Recurrent Neural Network

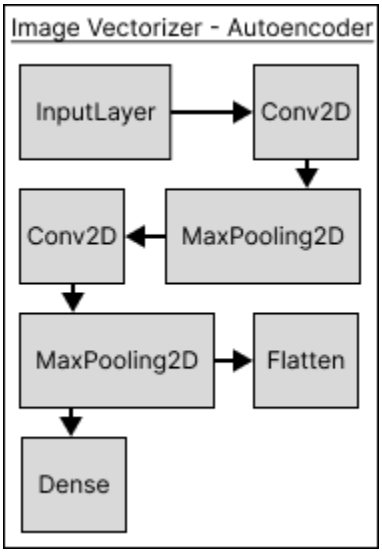
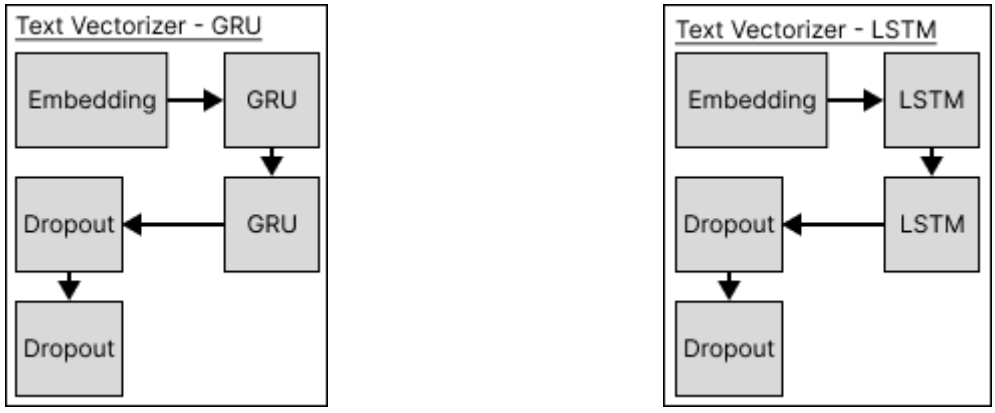


Fig. 2. Encoder from an Autoencoder



(a) GRU Vectorizer

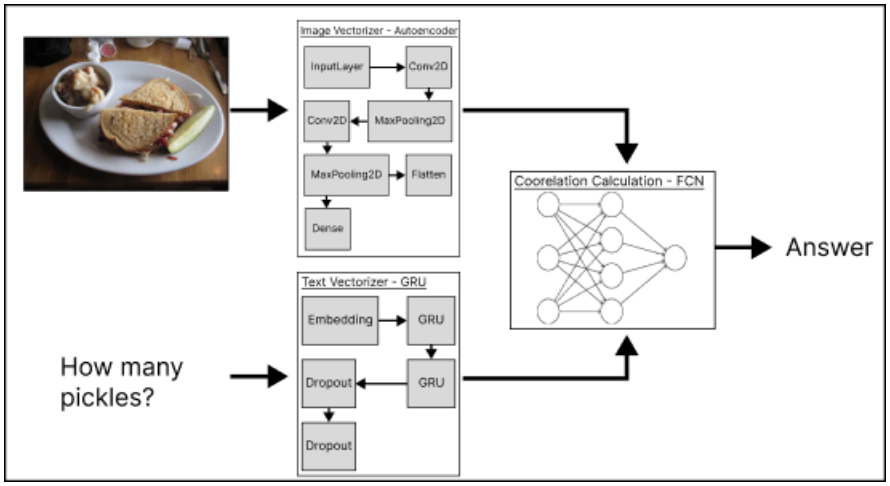
(b) LSTM Vectorizer

Fig. 3. Text Vectorizers

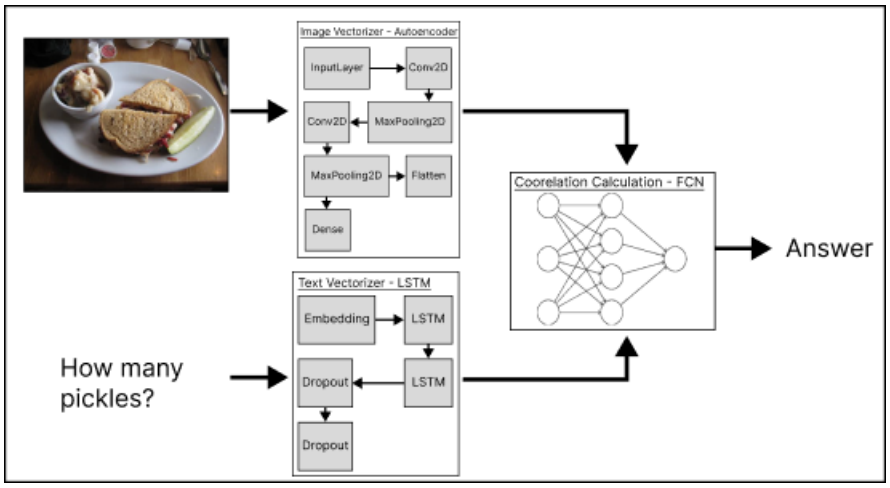
(RNN) architectures, specifically the GRU and LSTM. The architectures for our GRU and LSTM vectorizers, which receive tokens from the tokenizer, are shown in Fig 3. We will expound upon our methodology for calculating the correlations among *"image vectors"*, *"question vectors"*, and answers in the next section, describing it as the training process.

3.3 Training

Following our receipt of the *"image vector"* and the *"question vector"*, we concatenated these vectors and processed them through a dense neural network to correlate them with their respective answers. We divided the dataset into



(a) Autoencoder + GRU architecture



(b) Autoencoder + LSTM architecture

Fig. 4. Autoencoder architectures

batches of 32 and ran the training process over 200 epochs. We have used *nadam* function and *Stochastic Gradient Descent (SGD)* as our loss optimizer function. We focused on the increasing the accuracy metrics while reducing the loss. The architectures we trained included three image vectorizers and two text vectorizers, specifically: (1) Autoencoder + GRU (Figure 4a) (2) Autoencoder + LSTM (Figure 4b) (3) Inception V3 + GRU (4) Inception V3 + LSTM (5) VGG19 + GRU (6) VGG19 + LSTM

We will discuss the outcomes of these training exercises in the subsequent section.

4 RESULTS AND ANALYSIS

In this section, we will describe the training results of the different models we discussed earlier.

4.1 Autoencoder + GRU

As shown in Fig 5 both training and validation accuracies for our *Autoencoder + GRU* architecture start around 0.28 and reach a maximum of slightly over 0.30 before decreasing and re-stabilizing. The training loss starts very high (approximately 3.8) and then drops significantly to below 3 within the first few epochs. On the other hand, the validation loss decreases initially but then increases implying overfitting. Over the epochs we can see that both training and validation accuracy shows fluctuations, which indicates variance during the training process. Although a convergence pattern is shown, there is significant zigzag pattern implying potential issues with model architecture or training data.

4.2 Autoencoder + LSTM

For our *Autoencoder + LSTM* architecture, the training results are shown in Fig 6. The training accuracy shows a significant and consistent increase until around epoch 50, where it plateaus around 0.42. In contrast, the validation accuracy stabilizes at a much lower level, approximately 0.30, early in the training process and does not show any improvement with further epochs. The training loss decreases sharply and plateaus at a low level, which is typical and indicates effective learning from the training data. The validation loss decreases initially but then starts to increase slightly, showing fluctuations around a higher level compared to the training loss. The pattern from the accuracy curve of training and validation suggests that while the model is effectively learning features from the training data, it is not generalizing well to the validation data. The substantial gap between training and validation accuracy indicates a clear case of overfitting. This is further supported by the behavior of the validation loss, with its slight increase and fluctuations after initial decrease.

4.3 Inception v3 + GRU

For our *Inception v3 + GRU* architecture, the training results are shown in Fig 7. Both the training and validation accuracies start at a similar level, briefly reach a peak around epoch 10, and then show a notable decrease,

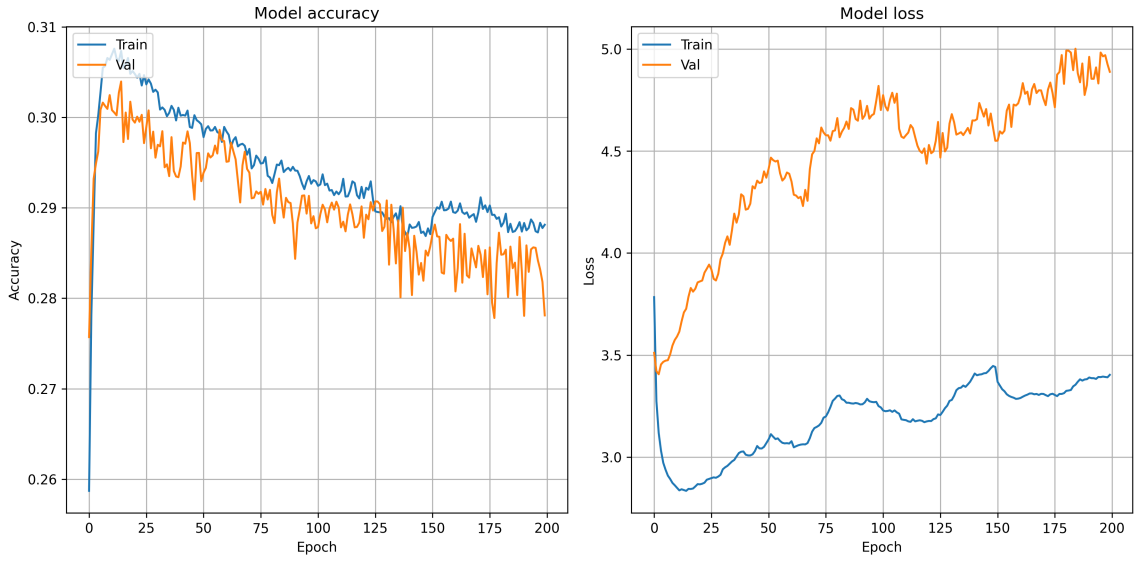


Fig. 5. Autoencoder + GRU: Training History

stabilizing at around 0.30 for training and slightly lower for validation. Post the initial drop, the accuracy exhibits fluctuations but largely remains stable throughout the remaining epochs. The training loss decreases sharply and stabilizes at a low level early in the training process, which is typical for model training. However, the validation

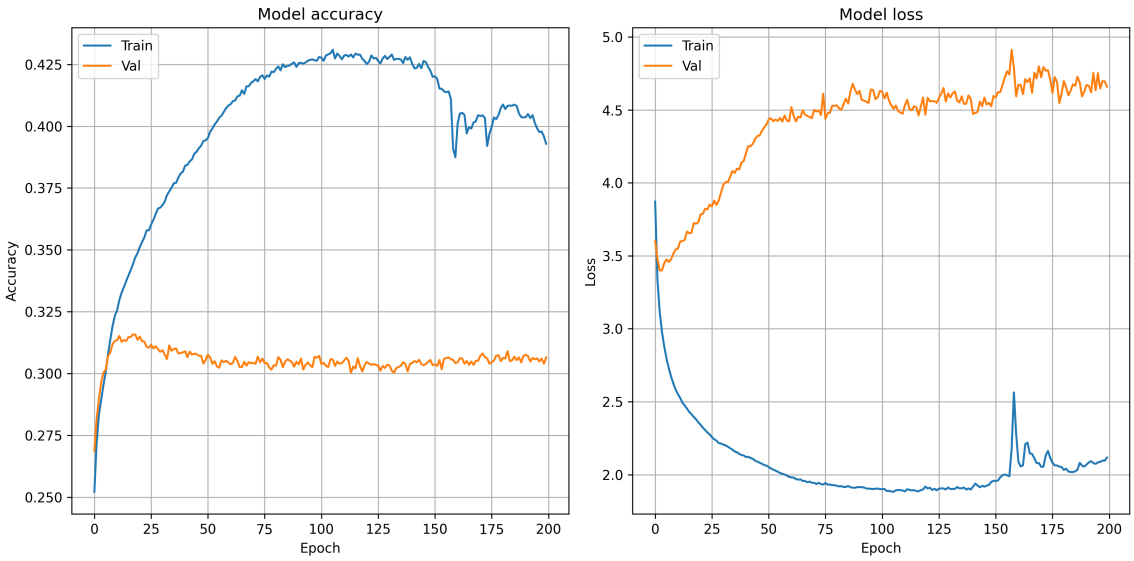


Fig. 6. Autoencoder + LSTM: Training History

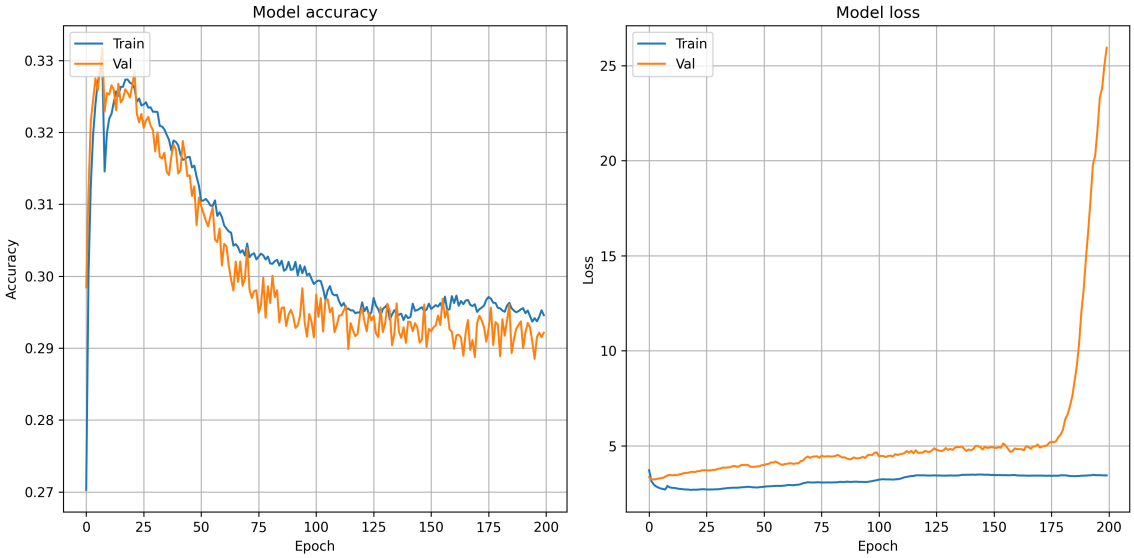


Fig. 7. Inception v3 + GRU: Training History

loss remains relatively flat and low throughout, then abruptly spikes around epoch 200. The initial peak in the accuracy followed by a drop could indicate the model rapidly fitting to nuances of the training data which do not generalize well, suggesting potential overfitting or ineffective learning dynamics.

4.4 Inception v3 + LSTM

For our *Inception v3 + LSTM* architecture, the training results are shown in Fig 8. The training accuracy shows a steady increase up until it peaks around epoch 100, maintaining high performance (slightly above 0.40) thereafter with minimal fluctuations. The validation accuracy, however, plateaus at a much lower level (around 0.31) and exhibits slight fluctuations throughout the training process. The training loss decreases sharply in the initial epochs and levels off, showing minor fluctuations but remaining at a low level. The validation loss, however, starts decreasing, then plateaus, and shows significant spikes towards the end of the training. The large and consistent gap between training and validation accuracy suggests that the model is overfitting the training data. The model learns and adapts well to the training dataset but struggles to generalize these learnings to the validation dataset, indicating a disparity in model performance across different data sets. We can also notice a significant fluctuation near epochs 125 and 180 which implies problems with the particular batch of the data.

4.5 VGG19 + GRU

For our *VGG19 + GRU* architecture, the training results are shown in Fig 9. Both training and validation accuracies start at a similar level around 0.27 and rapidly increase during the initial epochs. They peak at approximately 0.29-0.30 and then show significant fluctuation throughout the remaining epochs, though the training accuracy tends to remain slightly higher than the validation accuracy. The training loss declines sharply initially and then stabilizes around 3.5. However, the validation loss follows a similar sharp decrease but trends upward gradually over the rest of the epochs, indicating a slow divergence between training and validation loss. The convergence of training and validation accuracies with frequent fluctuations suggests the model is not highly stable but does

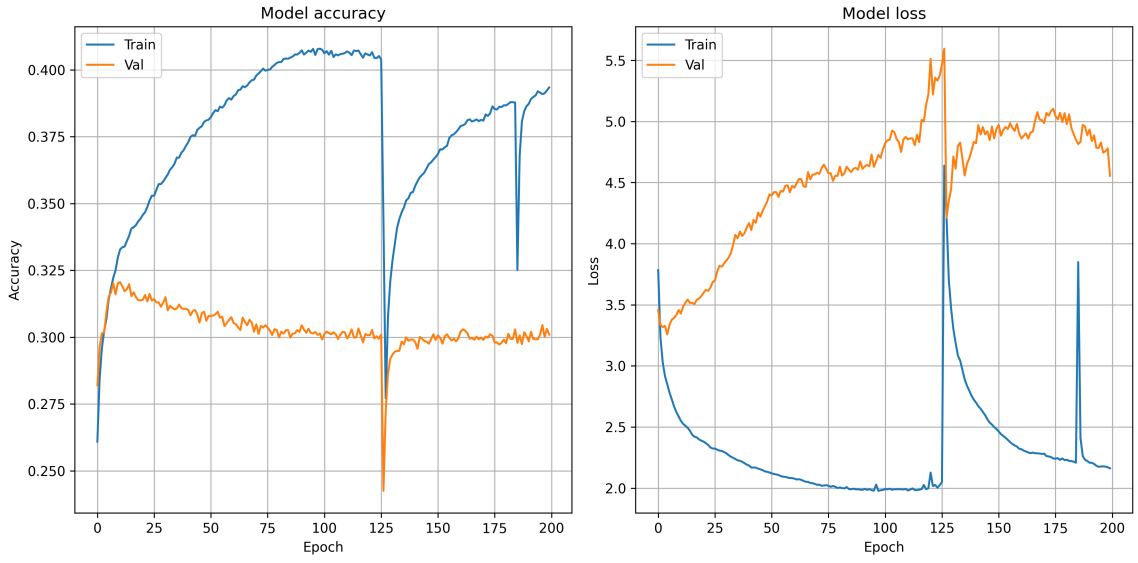


Fig. 8. Inception v3 + LSTM: Training History

manage to avoid significant overfitting. The fluctuations might be indicative of a learning rate that's too high or a dataset that's not sufficiently informative for the model to achieve higher accuracy.

4.6 VGG19 + LSTM

For our *VGG19 + LSTM* architecture, the training results are shown in Fig 10. The training accuracy shows a steep increase initially, peaking at nearly 0.70 and remaining fairly stable thereafter. In comparison, the validation accuracy increases much more slowly, stabilizing around 0.25 and showing no substantial improvement through the remaining epochs. The training loss decreases sharply in the early epochs and levels off at a very low value. Conversely, the validation loss starts high, decreases, but then increases slightly before plateauing. The validation loss shows an upward trend in the later epochs, further diverging from the training loss. The significant gap between the training and validation accuracies is a classic indication of overfitting. The model has learned to perform exceptionally well on the training data but fails to generalize this performance to the validation data.

4.7 Summary

Throughout our analysis of the various training and validation graphs for the architectures stated in Section 3, several key trends and challenges emerged that are crucial for optimizing the performance of our visual question answering system.

In most cases, there was a significant disparity between training and validation performance, with training accuracy being much higher than validation accuracy, and training loss much lower than validation loss. This suggests that the models are generally overfitting to the training data. Many of the models exhibited initial rapid improvements in both accuracy and loss, which then plateaued or deteriorated. This pattern indicates challenges in model generalization beyond the training set. Some models displayed considerable fluctuations in validation metrics, suggesting instability in the learning process which could be linked to high learning rates or inadequate model architectures for the task complexity.

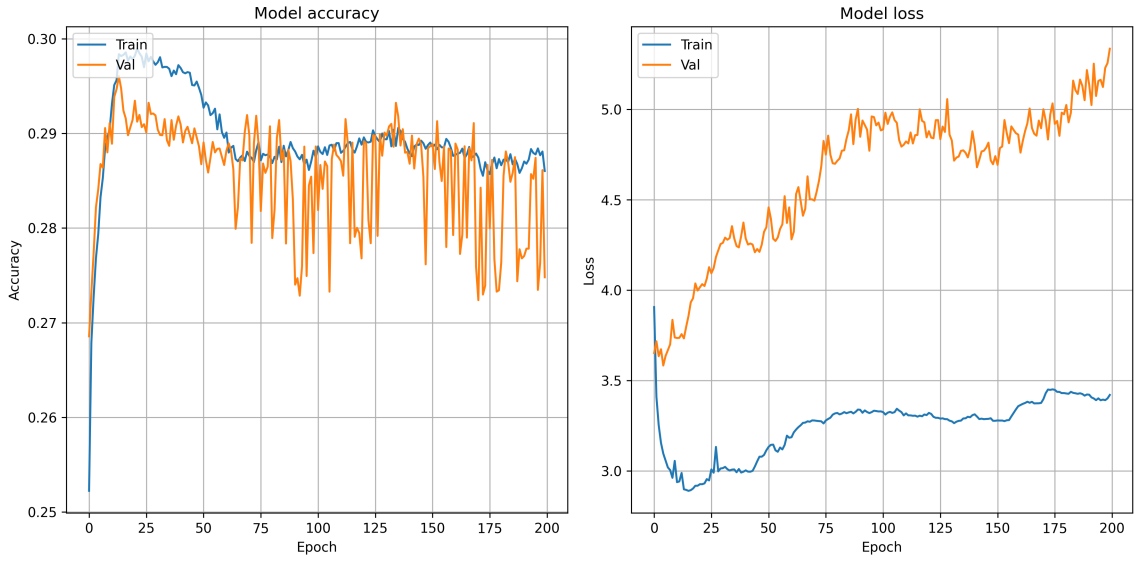


Fig. 9. VGG19 + GRU: Training History

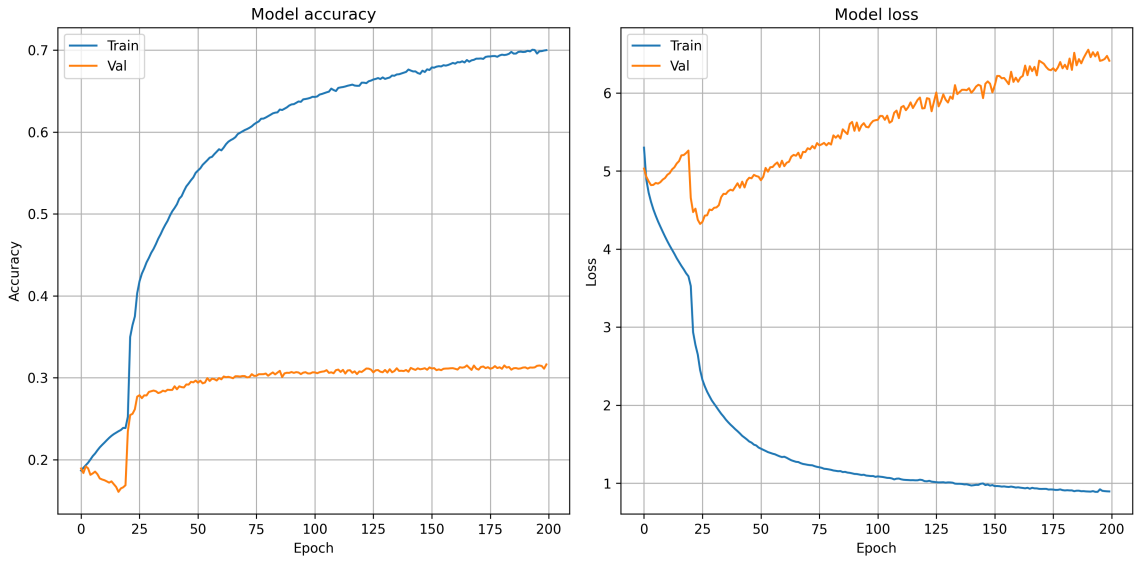


Fig. 10. VGG19 + LSTM: Training History

5 CONCLUSION

In this study, we explored development of a visual question answering system that integrates both image and textual data. Central to our approach is the representation of image features and input questions as vectors. Specifically, we extracted feature vectors from images using three distinct architectural models and represented questions using two different recurrent neural network architectures. We then combined these vectors to generate answers.

Initial training results across the combined architectures indicated a significant tendency for overfitting. To address this challenge, future work will focus on implementing strategies to mitigate overfitting, such as advanced regularization techniques, and exploring more dynamic learning rate adjustments. Additionally, we plan to investigate the potential of incorporating large-scale language models and vision models to enhance the system's ability to generalize and improve its accuracy on diverse datasets.

REFERENCES

[1] Ali Furkan Biten, Andres Mafla, Lluis Gomez, Dimosthenis Karatzas, and Marçal Rusinol. 2019. Scene Text Visual Question Answering. *ICCV* (2019).

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555 [cs.NE]

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[4] Yash Goyal, Kushal Kafle, Christopher Kanan, Devi Parikh, and Dhruv Batra. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[5] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Akshay Kumar Gupta. 2017. Survey of Visual Question Answering: Datasets and Techniques. *arXiv preprint arXiv:1705.03865* (May 2017).

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (nov 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[8] Ilija Ilievski, Shuicheng Yan, and Jiashi Feng. 2016. A Focused Dynamic Attention Model for Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[9] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. 2020. In Defense of Grid Features for Visual Question Answering. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[10] Kushal Kafle and Christopher Kanan. 2017. An Analysis of Visual Question Answering Algorithms. In *ICCV*.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs.CL]

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[13] Maitreya Shah, Anand Mishra, Narahari Yadati, and Partha Pratim Talukdar. 2019. KVQA: Knowledge-Aware Visual Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence* (2019).

[14] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs.CV]

[15] Yash Srivastava, Vaishnav Murali, Shiv Ram Dubey, and Snehasis Mukherjee. 2020. Visual Question Answering using Deep Learning: A Survey and Performance Analysis. *arXiv preprint arXiv:1909.01860* (Dec 2020).

[16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567 [cs.CV]

[17] Damien Teney, Peter Anderson, Xuming He, and Anton Van Den Hengel. 2017. Graph-Structured Representations for Visual Question Answering. *CVPR* (2017).

[18] Damien Teney and Anton van den Hengel. 2017. Zero-Shot Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.