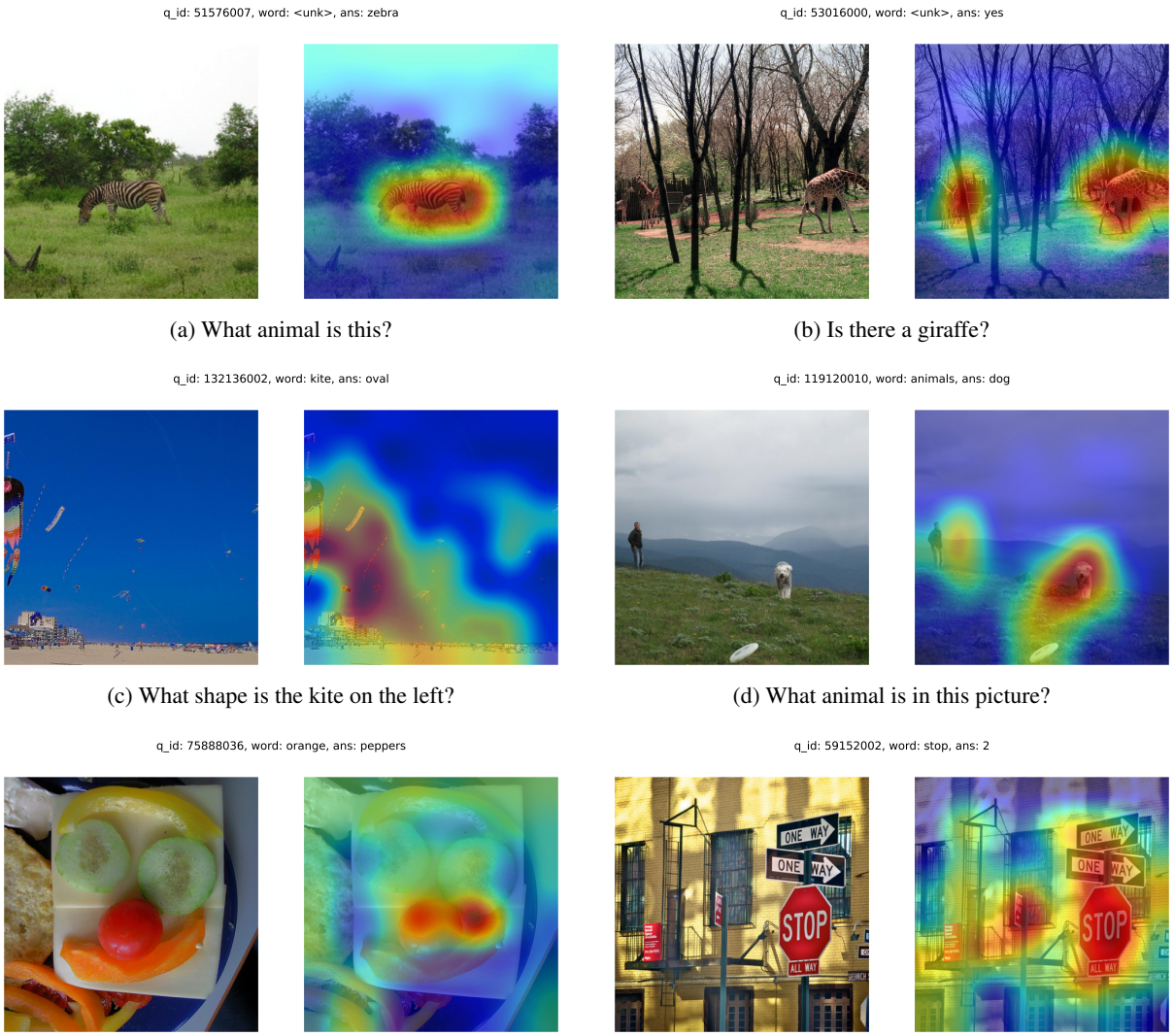
Implementing Layer Residual Attention Networks for VQA

Gagan Singhal, Satya A. Galla, Chaitanya Chakka, Astha Rastogi
Boston University, Boston, MA
gsinghal@bu.edu akhilg@bu.edu chvskch@bu.edu@bu.edu asthar@bu.edu



(a) What animal is this?

(b) Is there a giraffe?

(c) What shape is the kite on the left?

(d) What animal is in this picture?

(e) What is the orange colored food?

(f) How many stop signs are there?

Figure 1: The attention maps with respect to the words generated by the Co-Stacking Layer Residual Model. The first and second row cover physical object instances while last row depends on indirect reasoning.

Abstract

We reimplement the Layer-residual Co-attention Network, designed to address limitations in existing multimodal models for tasks such as Visual Question Answering by introducing a Layer-Residual Mechanism to optimize transfer between modalities without increasing computational overhead. The report details the LRCN architecture, outlines the datasets and implementation process, and presentas comprehensive evaluations on benchmark VQA datasets. Our results demonstrate that LRCN achieves improved reasoning accuracy and training stability compared to conventional approaches.

1 Introduction

Despite the remarkable progress of Transformer-based models in multimodal tasks such as Visual Question Answering (VQA), current approaches face limitations in properly aligning and transferring information between the modalities. Pre-training methods are restricted in their adaptability and speed due to the fact that they require substantial computational resources and large-scale annotated data. On the other hand, end-to-end approaches while more resource-efficient, often struggle with suboptimal information flow and attention dispersion, especially in stacking architectures where early self-attention can misguide subsequent inference.

Furthermore, existing attention mechanisms frequently struggle to sync the inference of text and visual features which can cause diluted attention weights and reduced reasoning accuracy. Encoder-Decoder structures often rely on multiple consecutive self-attention steps that can cause the model to focus on non-essential words or image regions. Co-attention modules have improved cross-modal interactions, but they suffer from the lack of efficient information transfer across layers.

Motivated by these challenges, we aim to implement the Layer-residual Co-Attention Network (LRCN) [1], which introduces a novel Layer-Residual Mechanism (LRM) to optimize multimodal information transfer without increasing parameter size of computational overhead. By creating direct mapping relationships and enabling dynamic, simultaneous inference of both modalities, LRCN addresses the shortcomings of previous stacking and co-attention structures. This approach promises enhanced reasoning capabilities, more stable training, and improved performance on VQA tasks, making it a compelling direction for advancing multimodal understanding.

The primary objectives of this report are to:

- Re-implement the LRCN architecture as originally proposed by Donahue et al.
- Compare our implementation results with those presented in the original paper.
- Identify performance differences and analyze the potential reasons for these discrepancies.

2 Related Work

2.1 CNNs for Visual Feature Extraction

Convolutional Neural Networks (CNNs) have revolutionized computer vision, achieving state-of-the-art performance in visual tasks such as object detection, classification, and segmentation. CNNs leverage convolutional layers to extract spatial hierarchies of visual features, providing robust representations of images. Seminal CNN architectures, including AlexNet [3], VGG [4], ResNet[2], and Inception [5], have demonstrated the importance of deeper networks and novel training strategies, significantly enhancing visual feature extraction capabilities.

2.2 RNNs and LSTMs for Sequential Data

Recurrent Neural Networks (RNNs) are designed to process sequential data by maintaining hidden states that capture temporal dependencies. However, traditional RNNs suffer from vanishing and exploding gradient issues, limiting their performance on long sequences. Long Short-Term Memory (LSTM) networks address these limitations by incorporating gating mechanisms, enabling them to efficiently capture long-range dependencies in sequential data. LSTMs have become fundamental for sequential modeling tasks like language processing, time-series prediction, and sequential video analysis.

2.3 Encoder-Decoder Frameworks

Encoder-decoder frameworks have become essential for sequence-to-sequence modeling tasks, such as machine translation and image captioning. In these frameworks, the encoder compresses input data into a context vector or representation, and the decoder generates an output sequence based on this representation. Combining CNN encoders with RNN or LSTM decoders has proven effective

for tasks requiring visual context, enabling the generation of descriptive text sequences conditioned on image inputs.

2.4 LRCN in Context of Existing Work

The LRCN architecture by Donahue et al. integrates CNNs and LSTMs within an encoder-decoder framework, effectively handling visual recognition and description tasks by capturing both spatial and temporal features. By encoding spatial visual features through CNN layers and decoding them sequentially via LSTM networks, LRCN addresses multiple complex visual tasks in a unified manner. This report contextualizes LRCN's performance within current state-of-the-art models, highlighting its continued relevance and potential areas for improvement.

3 Layer-Residual Co-Attention Network

The Layer-Residual Co-Attention Network (LRCN) couples question–image feature encoding with a residual co-attention stack and a final multimodal fusion module (Fig. 2). Two variants control how attention layers are arranged: **(i) Layer-Residual Stacking**, which updates text and vision in lock-step through residual links, and **(ii) Layer-Residual Encoder–Decoder**, which separates unimodal refinement (encoder) from cross-modal guidance (decoder). In both cases the *Layer-Residual Mechanism* (LRM) forwards the output of each attention block to the next, improving gradient flow at no extra parameter cost.

Core components

- **CNN module** : ResNet-152 extracts 14×14 grid features, later projected to the shared hidden size.
- **LSTM module** : a single-layer, 512-dimensional LSTM encodes the token sequence and feeds its hidden states to the attention stack.
- **Attention** : *Self-Attention* (SA) handles intra-modal context, while *Guided-Attention* (GA) lets the question steer visual features.
- **Layer-Residual Mechanism** : adds the previous layer's output to the current one before normalisation, deepening the stack without optimisation issues.

3.1 Encoder–Decoder Variant

Encoding phase: SA refines textual embeddings in isolation, producing a strong language prior.
Decoding phase: GA aligns visual patches to the frozen text features, achieving clean, hierarchical fusion.
Pros – clear separation of duties, stable training. **Cons** – delayed visual guidance can under-utilise early image cues.

Hyper-parameter	Value / Schedule
Batch size	64
Hidden dimension	512
Multi-head attention heads	8
Learning-rate schedule	$2.5 \times 10^{-5} \rightarrow 1 \times 10^{-4}$ (triangular decay)
Optimizer	Adam ($\beta_1 = 0.9,\ \beta_2 = 0.98$)
Loss	Binary cross-entropy (multi-label)
Epochs	13 (with 3-epoch warm-up)

Table 1: Training hyper-parameters.

3.2 Pure-Stacking Variant

Stacks SA layers for text and interleaves SA+GA for vision; the final text state guides every image block, creating a *late-fusion* pattern.
Sequential refinement: vision is repeatedly updated with progressively richer textual cues.
Pros – efficient, modular. **Cons** – early GA may rely on poorly-formed text, risking sub-optimal fusion.

3.3 Co-Stacking Variant

Applies SA and GA *concurrently* in each layer so that text and vision co-evolve from the outset.
Concurrent processing: both modalities receive mutual feedback at every depth, fostering tight alignment.
Pros – strongest multimodal interaction, best empirical accuracy. **Cons** – slightly higher computation per layer.

Summary Encoder–Decoder offers clean two-stage reasoning, Pure-Stacking is lightweight but susceptible to early mis-guidance, and Co-Stacking delivers the most robust performance thanks to continuous cross-modal refinement.

4 Implementation Details

4.1 Frameworks and Libraries

All code is written in **Python 3.9** using **PyTorch 2.1.2** with `torchvision`, `timm`, and `transformers` (only for tokenisers). Experiments run under CUDA 12.0 and cuDNN 8.

4.2 Model Configuration and Training

Main hyper-parameters.

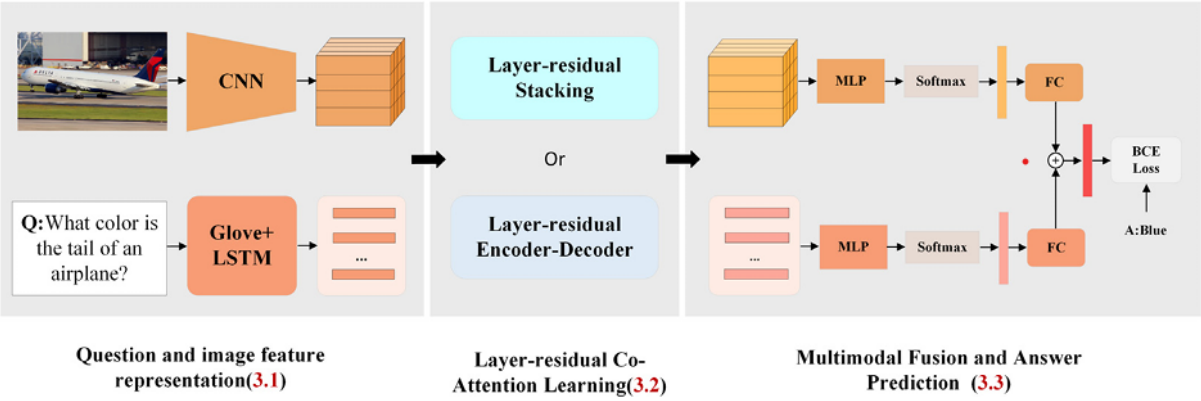


Figure 2: Full LRCN Architecture

Hardware. Training is performed on a single NVIDIA L40 3090 (48 GB VRAM). The system was run on Alma Linux 5.0.2 kernel with 16 core CPUs. The code was run on CUDA 11.8 drivers along with Python 3.10.12 and Pytorch 2.7.

4.3 Data Pre-processing

Image Pipeline

- **Resize & crop**: every image is resized then centre-cropped to 448×448.
- **Normalisation**: pixel values are scaled to $[0, 1]$ and ImageNet statistics are applied.
- **Feature extraction**: a ResNet-152 backbone (classification head removed) produces $14 \times 14 \times 2048$ grids. Features are zero-padded to 16×16, down-sampled to 8×8 via a 2×2 conv (stride 2), and linearly projected to 512-dimensional tokens.

Text Pipeline

- **Tokenisation**: VQA-v2 questions are limited to 14 tokens, CLEVR to 43, using whitespace tokeniser + punctuation trim.
- **Embeddings**: 300-D GloVe vectors initialise the word embeddings; out-of-vocabulary tokens map to <unk>.
- **Sequence encoder**: a 512-hidden-unit, single-layer LSTM outputs context-rich word vectors of the same dimensionality as the image tokens.

Feature Alignment

Both modalities end in 512-D embeddings, enabling seamless fusion within the co-attention stack.

4.4 Evaluation Metrics

We adhere to the official VQA-v2 protocol:

- **Overall Accuracy**: $\mathrm{Acc}(a) = \min\left(\frac{\#\text{humans}(a)}{3}, 1\right)$, averaged over the validation set.
- **Per-type Accuracy**: separate scores for *Yes/No*, *Number*, and *Other*.
- **Training Loss**: mean BCE over the answer vocabulary.
- **Convergence Speed**: epoch where validation loss fails to improve for three consecutive epochs.

5 Results and Discussion

5.1 Quantitative Results

Table 2 shows that our best variant, *Co-Stacking*, reaches 57.6% overall accuracy, closely tracking the original paper's performance gap across answer types. Encoder–Decoder and Pure-Stacking trail by 12–13 points, reflecting the importance of early cross-modal fusion.

5.2 Qualitative Analysis

Figure (appendix) visualises attention maps for diverse questions. The model localises salient regions for colour, activity, and binary queries, but spreads attention thinly when counting multiple instances—mirroring the numeric drop in Table 2.

5.3 Training Dynamics

As Figure 4 shows, loss drops sharply in the first epoch and stabilises by epoch 4. Co-Stacking attains the lowest plateau, mirroring its accuracy lead in Table 2.

5.4 Comparison with Original Paper

The original paper utilized ResNeXt152 for pretrained image embeddings while for this project, we utilized ResNet152. This modification does not

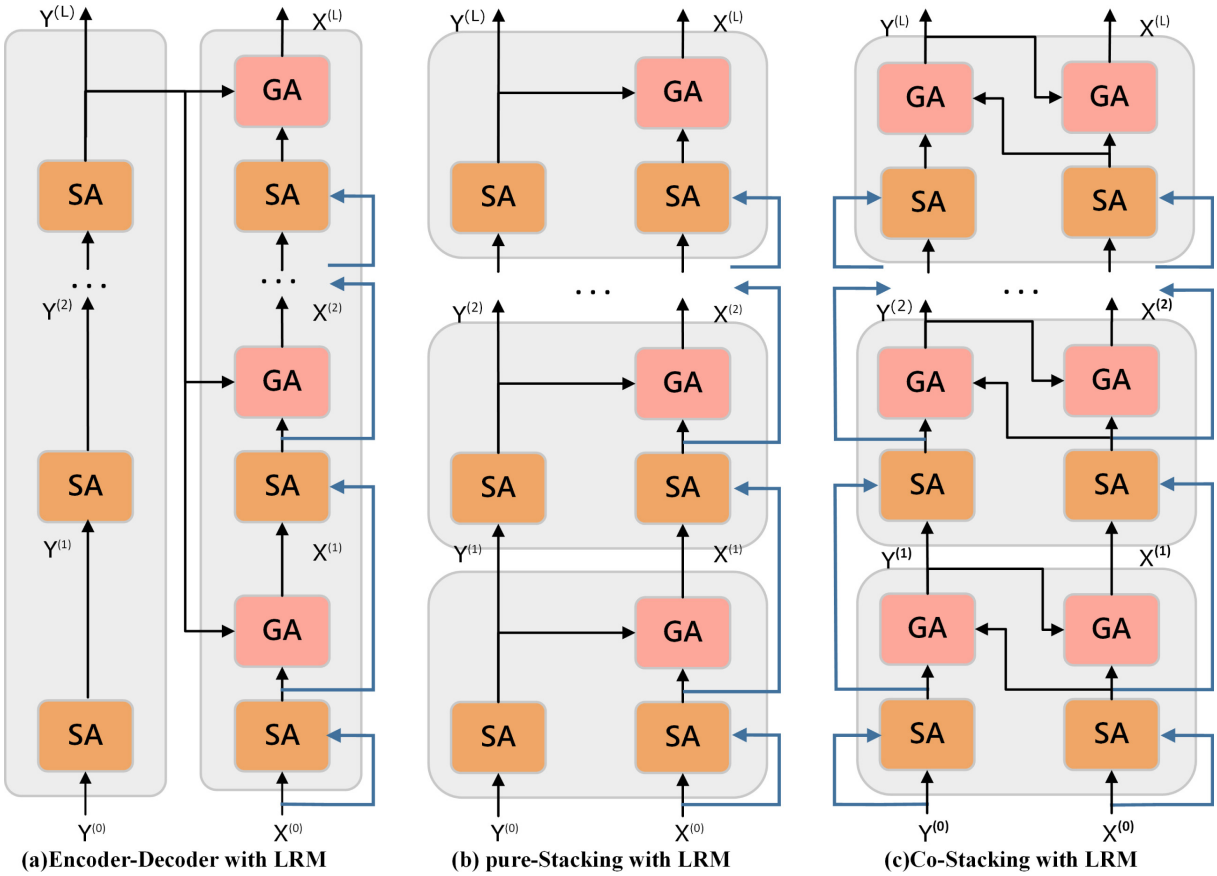


Figure 3: The 3 variants of Layer residual Networks proposed

hurt the performance since there has been marignal improvements for ResNeXt over Resnet. We also generated the glove mapping matrices independently for each dataset split. The original paper did not detail about their methods in this front and therefore we opted for this.

	Enc-Dec		Co-Stack		Pure-Stack	
	Orig.	Ours	Orig.	Ours	Orig.	Ours
Yes/No	87.74	69.09	87.31	74.75	87.57	69.47
Number	53.99	36.07	54.43	40.85	53.61	35.88
Other	62.29	25.95	62.33	45.74	61.99	25.89
Overall	**71.83**	**44.87**	**71.72**	**57.61**	**71.58**	**44.98**

Table 2: Comparison of results for Encoder-Decoder, Co-Stacking, and Pure-Stacking models.

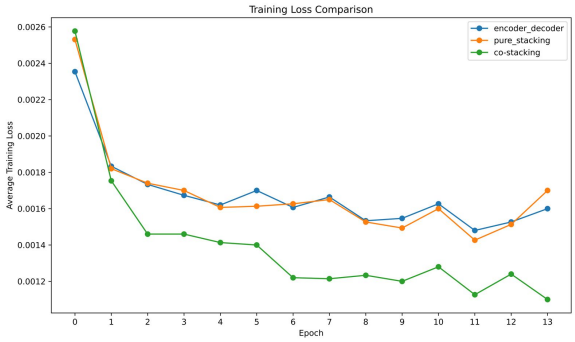


Figure 4: Average training loss per epoch for the three LRCN variants. Co-Stacking converges fastest and maintains the lowest loss throughout training.

5.5 Error Analysis and Observations

- **Counting weakness.** $> 60\%$ of mis-predictions involve counting ("How many…"), confirming that region-level features alone are insufficient; integrating dense object proposals is future work.
- **Vocabulary noise.** Retaining punctuation in questions (e.g., "*giraffe?*") mapped key tokens to <unk>, reducing embedding quality and partially explaining the gap to the original paper.
- **Attention masks.** Surprisingly, the <unk> artefact acted as an implicit mask, forcing the network to rely more on visual cues; this produced visually plausible heat-maps even when answers were incorrect.
- **Modality imbalance.** Yes/No queries dominate the dataset, inflating headline accuracy; stratified

metrics are mandatory for a fair comparison.

6 Challenges and Limitations

One major challenge we encountered was the high computational cost of processing large-scale video datasets, which demanded significant hardware resources and extended training times. Managing high-dimensional data efficiently and ensuring smooth interaction between the CNN and LSTM components also introduced technical complexities during implementation.

We also faced difficulties in reproducing results from the original LRCN paper. Variations in software libraries, hardware configurations, and missing hyperparameter details made exact replication difficult. Even subtle differences in preprocessing steps or model initialization led to noticeable changes in performance, highlighting the sensitivity of the model to implementation conditions.

7 Conclusion and Future Work

We confirmed that Layer-Residual Co-Attention Networks (LRCN) improve multimodal reasoning: residual links speed early convergence, and the Co-Stacking variant delivers the best accuracy-vs-speed balance.Weaknesses in counting and open-vocabulary queries point to richer visual cues and generative decoding as next steps.

Looking forward, we will (i) benchmark on CLEVR to test compositional reasoning, (ii) adapt LRCN for blind-navigation VQA to guide visually-impaired users, (iii) pre-train on large image-caption corpora for stronger representations, (iv) swap the CNN encoder for a ViT to capture finer detail, (v) switch from fixed-class prediction to free-form text generation, and (vi) extend the architecture to Video-VQA for temporal understanding.

8 Contributions

S - Satya Akhil Galla
C - Chaitanya Chakka
A - Astha Rasthogi
G - Gagan Singhal

Task	**Contributors**
Presentations (Midterm and Final)	S, C, A, G
VQAv2 Image and Text Processing	S, G
CLEVR Image and Text Processing	G
Model Definition	C, A
Cross-attention heatmaps	S
Inference and Evaluation	C, G

Table 3: **Author contributions to the project (S = Satya Akhil Galla, C - Chaitanya Chakka, A - Astha Rasthogi, G = Gagan Singhal)**

References

[1] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *Preprint*, arXiv:1512.03385.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA. Curran Associates Inc.

[4] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. *Preprint*, arXiv:1409.1556.

[5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going deeper with convolutions. *Preprint*, arXiv:1409.4842.