1.5em 0pt

Visual Question Answering Using Convolutional Recurrent Neural Networks (CRNN)

1st Rupa Kandula
Amrita School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamilnadu, India
rupakandula21@gmail.com

2nd Kolli Dharani
Amrita School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamilnadu, India
dharaninaidu2006@gmail.com

3rd K Laxmi Vignesh
Amrita School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamilnadu, India
xyz@gmail.com

4rd Koruprolu Varalaxmi
Amrita School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamilnadu, India
varshakoruprolu19@gmail.com

5rd Lekshmi C.R.
Amrita School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamilnadu, India
cr_lekshmi@cb.amrita.edu

***Abstract*—Visual Question Answering (VQA) is a complex task that requires the interpretation of both visual content and natural language queries to generate meaningful responses. This study explores various neural network architectures for VQA, comparing their performances on standard datasets. The proposed model incorporates Convolutional Neural Networks (CNNs) for visual feature extraction and Recurrent Neural Networks (RNNs) for processing textual questions. Our approach achieves high accuracy and generalizes well to previously unseen questions. Furthermore, we discuss potential real-world applications of VQA, including assistive technologies, education, and automated customer support, underscoring its practical significance beyond theoretical research.**

***Index Terms*—Visual Question Answering, Convolutional Neural Networks, Recurrent Neural Networks**

I. Introduction

Visual Question Answering (VQA) is a new area of artificial intelligence that compares computer vision and natural language processing in order to comprehend and respond to human questions regarding images. VQA has been extensively debated in terms of its learning capacity, accessibility, and automatability. VQA models try to fill in the language comprehension and image comprehension gaps by pushing models to excel on visual and textual inputs.

VQA is a close cooperation between Natural Language Processing and Computer Vision communities, and other tasks like image captioning. VQA is a sub-task task involving question reasoning, object detection, and object recognition. A good VQA model should be capable of having a good comprehension of the questions and images in order to provide the right answers.

VQA is a multi-modal learning task in which a computer system needs to produce correct text output given an input image and a natural language input question. In visual question answering, the model is trained to produce a correct answer to a specific query question and image. Figure 2 shows an example of visual question answering. [ht] Particularly in this paper we see Convolutional Neural Networks (CNNs) for image feature extraction and Recurrent Neural Networks (RNNs) for processing the sequential nature of natural language queries.CNNs are the cornerstone of visual data processing in VQA systems.The goal is to generate accurate answers to natural language questions based on visual inputs, typically images. This complex task requires the model to understand and integrate information from both visual and textual modalities, making it a challenging yet crucial problem in artificial intelligence research.RNNs are employed to model the sequential dependencies inherent in natural language. Given that VQA questions often require understanding of syntax and contextual semantics,RNNs process words in a sequence, maintaining a hidden state that captures information from previous time steps.

Fig. 1: Example picture of VQA on images (Source — VQA: Visual Question Answering, visualqa.org)

II. Related work

Early approaches to Visual Question Answering (VQA) leveraged Stacked Attention Networks (SAN) , which employed iterative attention mechanisms to progressively refine focus on relevant image regions based on the posed question.

While SAN demonstrated significant improvements in visual reasoning by enabling multi-step attention, recent models have shifted towards more streamlined architectures integrating Convolutional Neural Networks (CNNs) for hierarchical visual feature extraction and Recurrent Neural Networks (RNNs) for sequential question encoding. In this paradigm,CNNs effectively capture spatial and semantic representations from images, while RNNs, particularly Long Short-Term Memory (LSTM) networks, model the temporal dependencies within natural language questions, facilitating efficient multimodal fusion for answer prediction.

A. Image Feature Extraction Approaches

Recent studies have utilized various CNN-based architectures for extracting high-level semantic features from images. Models like ResNet [6], DenseNet [7], and EfficientNet [10]have been widely used due to their ability to capture hierarchical visual representations. Transformer-based models, such as ViT [18]and Swin Transformer [12], have recently gained traction for their effectiveness in self-attention mechanisms for vision tasks.

B. Embedding Approaches

Traditional text embeddings such as Word2Vec [4], GloVe [15], and FastText [21] have been widely used in VQA. More recent approaches leverage contextual embeddings from transformers like BERT [9], RoBERTa [13], and GPT [17] to improve language understanding. These models generate word representations dynamically based on sentence context, enhancing model performance in multimodal tasks. Additionally, Embeddings from Language Models (ELMo) [16] provide deep contextualized word representations by leveraging character-level information, further enhancing model performance in multimodal tasks..

C. VQA Approaches

The advancement of VQA has been largely influenced by deep learning techniques, including attention mechanisms (e.g., stacked attention networks [20], multimodal transformers (e.g., ViLBERT [14], LXMERT [19], and graph-based reasoning methods (e.g., Relation-Aware Graph Attention Network [11]. The integration of knowledge graphs and reinforcement learning has also been explored to enhance reasoning capabilities in VQA models. Recent research has focused on improving VQA models by addressing issues like bias mitigation, multimodal learning, and enhanced reasoning capabilities. Jiang *et al.*. revisited grid-based convolution features for better VQA performance [8]. Chen *et al.* proposed a counterfactual sample synthesis framework to improve model robustness against language bias [3]. Agarwal *et al.*, [1] introduced causal VQA, which utilizes semantic image manipulation techniques to identify and reduce spurious correlations in models . Xi *et al.* developed a VQA model based on multi-object visual relationship detection, replacing traditional object features with appearance features for improved contextual understanding [1]. Yu *et al.* [2] surveyed VQA applications in the medical domain, highlighting the challenges and advancements in medical image-based question answering. Gokhale *et al.* [5] proposed a model for training out-of-distribution generalization in VQA by enforcing consistency constraints. These contributions collectively advance VQA by making models more robust, interpretable, and generalizable across various datasets and real-world applications.

Our approach focuses on refining the CNN-RNN framework for VQA, emphasizing efficient multimodal feature fusion techniques and optimizing recurrent architectures like GRUs for improved question understanding. By leveraging the strengths of CNNs for hierarchical visual representation and RNNs for capturing sequential dependencies in natural language, our model aims to bridge the gap between traditional deep learning methods and contemporary transformer-based solutions, offering a balance between computational efficiency and interpretability.

III. System Description

*A. **Convolutional Neural Networks***

A Convolutional Neural Network (CNN) is a deep learning model that is specifically used to process structured grid data like images. It is made up of several layers such as convolutional layers that use learnable filters to extract features, pooling layers that decrease the spatial dimensions with the preservation of relevant information, and fully connected layers that classify the features that are extracted. CNNs are particularly good at applications such as image classification, object detection, and medical image analysis because they can automatically learn spatial hierarchies of features. Their major strengths are automatic feature extraction, fewer parameters by weight sharing, and translation invariance, which make them a very useful tool in computer vision and beyond. In contrast to conventional machine learning methods that use hand-designed features, CNNs learn features from data themselves, allowing for strong pattern recognition. Architectures like VGG, ResNet, and EfficientNet have further refined CNNs by adding deeper networks, skip connections, and better efficiency. The combination of CNNs with other models, i.e., Recurrent Neural Networks (RNNs) in Visual Question Answering (VQA), improves multimodal learning by unifying visual and textual information. This renders CNNs essential in use cases such as autonomous driving, facial recognition, medical diagnosis, and anomaly detection in sophisticated systems.

*B. **Recurrent Neural Networks***

Recurrent Neural Network (RNN) is a form of deep learning network intended for sequential data, like time series, speech, and natural language processing. While regular neural networks lack a memory component, RNNs do, enabling them to carry forward information from earlier inputs and, therefore, perform tasks requiring relevance. It accomplishes this via recurrent connections, in which a neuron's output is cycled back into the network as the input for the subsequent time step. An RNN processes both the current input and the past hidden state

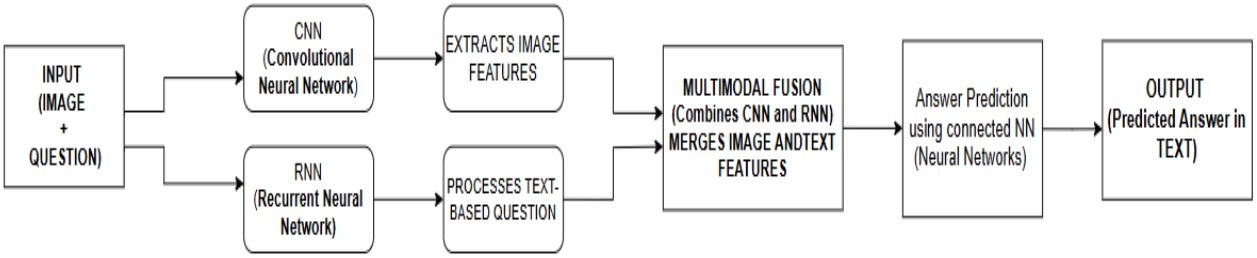

Fig. 2: Block diagram of proposed method of visual question answering

for each step, allowing it to learn temporal sequences. Standard RNNs do have vanishing and exploding gradients, though, which restrict the network from accessing long-term relationships. To overcome such constraints, recent architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) have emerged. LSTMs apply gate control (input, forget, and output gates) to modulate the information flow, enabling the network to learn critical data in longer sequences, whereas GRUs generalize this with update and reset gates by reducing the computation burden. The RNNs are trained via Backpropagation Through Time (BPTT), a generalization of backpropagation in which gradients are backpropagated through time steps, although cautionary optimization methods such as gradient clipping are commonly required in order to stabilize training. They are widely applied in tasks such as natural language processing (NLP) for sentiment analysis.

C. *Concatenation*

Concatenation at vector level, feature-by-feature, is a method that concatenates multiple feature representations—usually derived from different modalities like image and text—together into a unique, joined-up feature vector. In such an approach, the input first gets encoded in the form of a fixed-vector length (like employing a convolution neural network on an image and LSTM on a piece of text) and is subsequently concatenated side-by-side across the feature dimension (i.e., the final dimension). This type of concatenation retains all information from both inputs by preserving their original feature values, which enables the model to learn cross-modal relationships without data loss. It is referred to as ”vector-level” due to the fact that inputs being concatenated are one-dimensional vectors and ”feature-wise” because the concatenation places the two vectors’ features side by side in horizontal fashion, essentially doubling the feature space. This approach is applied extensively in multimodal problems such as Visual Question Answering (VQA), where it is necessary to jointly process both visual and textual information. Its simplicity, effectiveness, and capability to keep separate semantic information from both modalities render it a robust baseline for early fusion techniques in deep neural networks.

D. *Dataset Description*

Table I summarizes the key statistics of the CLEVR v1.0 dataset. It includes the number of images used for training, validation, and testing, along with the total number of images and questions. The dataset supports a variety of answer types, including Yes/No responses, numerical values, and categorical attributes such as colors and shapes. This structured composition enables comprehensive evaluation of visual reasoning capabilities in machine learning models.

Component	**Details**
Train Images	70,000
Validation Images	15,000
Test Images	15,000
Total Images	100,000
Questions	700,000
Answer Types	Yes/No, Numbers, Colors, Shapes

TABLE I: CLEVR v1.0 Dataset Summary

IV. Methodology

A. *Data Loading and Preparation*

The methodology begins by selectively loading and preprocessing a subset of the CLEVR v1.0 dataset for training and validation. Specifically, 30,000 samples are extracted from the training set and 5,000 from the validation set. For each sample, the code retrieves the associated image path, the corresponding natural language question, the ground-truth answer, and determines the type of question using a custom classification function. This function, `get_question_type()`, analyzes the question text to categorize it into predefined types: *count*, *color*, *shape*, *material*, or *other*. These types are not directly used during model training but are included to enable downstream analysis, such as evaluating model performance on different reasoning categories or implementing targeted improvements. The samples are stored in Pandas DataFrames,

which facilitate easy manipulation and integration into the data pipeline.

B. Text Tokenization and Encoding

In the tokenization and encoding phase, a vocabulary is constructed from all unique words present in both the questions and answers using the `Tokenizer` from the TensorFlow Datasets (TFDS) library. This ensures that the model has access to a comprehensive set of tokens during training. A `TokenTextEncoder` is then used to convert these text sequences into corresponding integer sequences. The same encoder is applied to both the input questions and the target answers, maintaining consistency in the text representation across the dataset.

C. Data Preprocessing Pipeline

The data preprocessing pipeline prepares both image and text inputs for model training. Each image is first read from disk, decoded from JPEG format, resized to a fixed dimension of 200×200 pixels, and normalized to have pixel values in the range [0, 1]. For the textual data, questions are tokenized using the previously constructed vocabulary and padded to a uniform length of 50 tokens to ensure consistent input size. Answers are also tokenized and treated as single-token sequences since the model is designed to output a single-word response. These preprocessed image-question-answer pairs are then organized into TensorFlow datasets using the `tf.data.Dataset` API, which enables efficient batching (with a batch size of 30) and mapping of the custom preprocessing function. This structured pipeline ensures that the data is uniformly processed and optimized for GPU training.

Step	Details
Image Size	200 × 200 pixels
Image Format	JPEG
Normalization	Pixel values in [0, 1]
Question Length	Padded to 50 tokens
Answer Format	Single-token sequence
Batch Size	30
Dataset API	tf.data.Dataset

TABLE II: Data Preprocessing Summary

Table II provides a summary of the preprocessing steps applied to both image and text data before training, including resizing, normalization, tokenization, and batching.

D. Model Architecture

The model architecture is constructed as a multi-modal neural network which combines both visual and textual information to execute Visual Question Answering (VQA). For visual input, an ImageNet pretrained Convolutional Neural Network (CNN) from the ResNet50 architecture is employed as the image encoder. ResNet50 is pre-trained on the ImageNet dataset and is provided without its top classification layers so that it can serve solely as a feature extractor. The output of the ResNet50 model is feature maps that are fed through a GlobalAveragePooling2D layer to compress the spatial dimensions and generate a compact, fixed-size feature vector. The feature vector represents the visual information that has been processed from the input image and forms part of the model's input used to make predictions.

The text input, i.e., the natural language question, is processed through a Recurrent Neural Network (RNN) structure based on Bidirectional Long Short-Term Memory (BiLSTM) layers. The question is initially fed through an embedding layer of size 256 to transform the tokenized input into dense vectors. This embedded sequence is then passed through three stacked BiLSTM layers: the first two contain 256 units and output full sequences, whereas the third contains 512 units and outputs only the last output vector. This last vector encapsulates the semantic meaning of the question. Outputs from both the visual (CNN) and textual (RNN) branches are concatenated to create a combined feature representation. This summed vector is then fed into a Dense layer with softmax activation, which gives a probability distribution over the whole vocabulary.

E. Training Setup

The model is trained using the `adam` optimizer, which is well-suited for handling sparse gradients and adaptive learning rates in deep learning tasks. The loss function used is `sparse_categorical_crossentropy`, which is appropriate for classification problems where the target labels are provided as integer indices rather than one-hot encoded vectors. Since the model is designed to predict a single answer token from a large vocabulary, this loss function efficiently computes the error between the predicted probability distribution and the true token index. To evaluate model performance, two metrics are used: `SparseCategoricalAccuracy`, which measures the frequency of exact matches between predicted and true answers, and top-5 Accuracy, which checks whether the correct answer is within the top five predictions. This combination of metrics allows for a more comprehensive assessment of both precision and model confidence. Training and validation are performed on the preprocessed datasets generated through the `tf.data` pipeline, ensuring efficient batching and data feeding during model optimization.

F. Saving the Model

Upon completion of training, the model and its associated components are preserved for future inference and evaluation. The trained multi-modal VQA model is serialized and saved in the HDF5 format as `vqa_with_qtype.h5`, encapsulating both the learned weights and the model architecture. This allows for seamless loading and reuse of the trained model without the need for redefinition or retraining. Additionally, the `TokenTextEncoder`, which was employed to convert textual data into integer sequences during preprocessing, is also saved using Python's `pickle` module in the file `encoder.pkl`. Preserving the exact encoder ensures consistency between training and inference phases, as it maintains the same vocabulary mapping and tokenization logic that the model was originally trained on. This step is essential for deploying the model reliably, as any deviation in the input

encoding could result in incorrect predictions or degraded performance.

Mathematical Formulation

The overall architecture integrates convolutional and recurrent components to perform Visual Question Answering (VQA) by combining image and text modalities. The input image $\mathbf{I} \in \mathbb{R}^{200\times200\times3}$ is passed through a ResNet50 backbone $f_{\text{CNN}}(\cdot)$ to extract a deep feature representation:

$$\mathbf{v} = f_{\text{CNN}}(\mathbf{I}) \in \mathbb{R}^{2048} \tag{1}$$

where $\mathbf{v}$ is the global average pooled feature vector from the final convolutional block.

The question text $Q = \{q_1, q_2, \ldots, q_T\}$ is tokenized and embedded using a learned embedding matrix $E \in \mathbb{R}^{|V|\times d}$, producing:

$$\mathbf{x}_t = E(q_t), \quad \forall t \in [1, T] \tag{2}$$

These embedded vectors are passed through a stack of Bidirectional LSTM layers. The final textual encoding $\mathbf{u}$ is obtained as:

$$\mathbf{u} = \text{BiLSTM}_3(\text{BiLSTM}_2(\text{BiLSTM}_1(\mathbf{x}_{1:T}))) \tag{3}$$

where $\mathbf{u} \in \mathbb{R}^{1024}$ is a fixed-length semantic representation of the question.

The visual and textual representations are concatenated to form a multimodal joint representation:

$$\mathbf{h} = [\mathbf{v}; \mathbf{u}] \in \mathbb{R}^{3072} \tag{4}$$

This joint vector is passed through a dense layer with softmax activation to predict the answer from the vocabulary space $\mathcal{A}$:

$$\hat{y} = \text{softmax}(W\mathbf{h} + b), \quad \hat{y} \in \mathbb{R}^{|\mathcal{A}|} \tag{5}$$

The model is trained using the sparse categorical cross-entropy loss function:

$$\mathcal{L} = -\log \hat{y}_y \tag{6}$$

where y is the true answer index and $\hat{y}_y$ is the predicted probability for the correct class.

Additionally, the top-k categorical accuracy is used as a performance metric during training:

$$\text{Top-}k \text{ Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[y_i \in \text{Top}_k(\hat{y}_i)\right] \tag{7}$$

Where:

- N is the total number of samples.
- y_i is the true label for the i^{th} sample.
- $\hat{y}_i$ is the predicted probability distribution for the i^{th} sample.
- $\text{Top}_k(\hat{y}_i)$ denotes the set of k class labels with the highest predicted probabilities in $\hat{y}_i$.
- $\mathbb{1}[\cdot]$ is the indicator function, which returns 1 if the condition is true, and 0 otherwise.

This metric measures the proportion of times the true label y_i appears within the top-k predicted labels, providing a more flexible evaluation than strict top-1 accuracy.

V. Results and Analysis

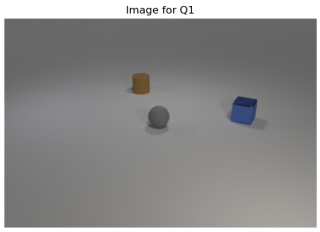

Fig. 3: Visual context for Questions from the test set.

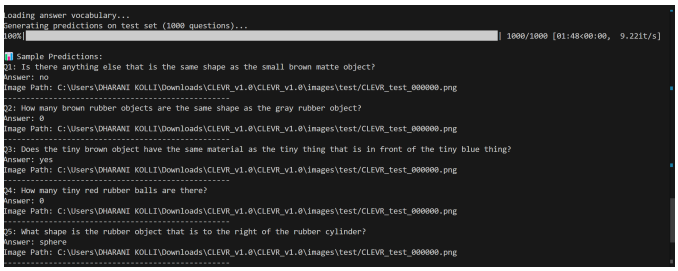

Fig. 4: Sample predictions from the test set with associated questions and answers.

The proposed Visual Question Answering (VQA) model, trained on the CLEVR dataset, demonstrates steady convergence and generalization across epochs, as evidenced by the training metrics shown in Figure 1. Over three epochs, the model achieves a final training loss of 1.1342 and a validation sparse categorical accuracy of 40.21 %, while maintaining a consistently high top-k categorical accuracy of approximately 95.94%. Despite the moderate primary accuracy, the top-k performance suggests the model is frequently placing the correct answer within its top predictions, indicating robust internal feature representation. It showcases qualitative evaluation results from the test set, comprising 1,000 CLEVR-style visual reasoning questions. Figure 4 showcases qualitative evaluation results from the test set, comprising 1,000 CLEVR-style visual reasoning questions. The model outputs semantically coherent answers aligned with visual semantics—demonstrating competence in attribute comparison, counting, and spatial reasoning. For instance, in Q1, it accurately responds *"no"* to a query involving shape matching across differently colored and sized objects. The image illustrates the spatial arrangement of these objects, validating the model's understanding of the visual context. Collectively, these results substantiate the model's efficacy in handling compositional reasoning tasks, validating both the architecture's learning capacity and its deployment viability in real-world VQA applications.

A. *Limitations*

Despite its promising performance, the proposed model exhibits several limitations. The reliance on a fixed-length

question encoding may hinder its ability to fully capture long or syntactically complex queries. Additionally, the absence of explicit attention mechanisms restricts the model's capacity to dynamically focus on relevant visual regions, potentially impacting its interpretability and precision in spatial reasoning tasks. Furthermore, the model is trained and evaluated solely on synthetic CLEVR data, which may not generalize well to real-world images with more visual noise, ambiguity, and linguistic variability.

VI. Conclusion

In conclusion, the proposed VQA framework effectively integrates convolutional neural representations with recurrent linguistic embeddings to perform multi-modal reasoning over synthetic scenes. By leveraging ResNet50 as a visual encoder and a hierarchical stack of Bidirectional LSTM layers for sequential language understanding, the architecture demonstrates substantial competency in learning compositional semantics and spatial relationships inherent in CLEVR-style queries. The model's ability to maintain high top-k accuracy, despite moderate sparse categorical performance, underscores its capacity to capture the latent structure of visual-question-answer triplets. The end-to-end trainable pipeline exhibits not only convergence stability but also resilience across diverse reasoning types, including attribute comparison, counting, and existential queries. These results validate the synergy between deep convolutional and sequential models in tackling the complexities of VQA tasks and pave the way for further research into attention-based mechanisms and fine-grained reasoning paradigms for real-world visual reasoning challenges.

References

[1] Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9690–9698, 2020.

[2] Asma Ben Abacha, Mourad Sarrouti, Dina Demner-Fushman, Sadid A Hasan, and Henning Müller. Overview of the vqa-med task at imageclef 2021: Visual question answering and generation in the medical domain. In *Proceedings of the CLEF 2021 Conference and Labs of the Evaluation Forum-working notes*. 21-24 September 2021, 2021.

[3] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10800–10809, 2020.

[4] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[5] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. Mutant: A training paradigm for out-of-distribution generalization in visual question answering. *arXiv preprint arXiv:2009.08566*, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[8] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10267–10276, 2020.

[9] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota, 2019.

[10] Brett Koonce and Brett Koonce. Efficientnet. *Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization*, pages 109–123, 2021.

[11] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10313–10322, 2019.

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[13] Zhuang Liu, Wayne Lin, Ya Shi, and Jun Zhao. A robustly optimized bert pre-training approach with post-training. In *China National Conference on Chinese Computational Linguistics*, pages 471–484. Springer, 2021.

[14] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[15] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[16] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. arxiv. *arXiv preprint arXiv:1802.05365*, 1802.

[17] Alec Radford. Improving language understanding by generative pre-training. 2018.

[18] Gilad Sharir, Asaf Noy, and Lihi Zelnik-Manor. An image is worth 16x16 words, what is a video worth? *arXiv preprint arXiv:2103.13915*, 2021.

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

[20] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[21] Tengjun Yao, Zhengang Zhai, and Bingtao Gao. Text classification model based on fasttext. In *2020 IEEE International conference on artificial intelligence and information systems (ICAIIS)*, pages 154–157. IEEE, 2020.