Visual Question Answering

Ahmed Nada
Computing and Software Systems
University of Washington Bothell
Bothell, USA
anada@uw.edu

Min Chen
Computing and Software Systems
University of Washington Bothell
Bothell, USA
minchen2@uw.edu

Abstract— **Visual question answering (VQA) is an artificial intelligence (AI) and computer vision (CV) comprehensive task to answer questions about the visual content of an image, such as "what color is the bus?" or "how many people are in the photo?" VQA has shown great potential and importance in various domains, ranging from medical imaging applications, autonomous driving, to virtual assistants and search engines. This study develops a framework to tackle VQA research challenges by adopting and extending recent breakthroughs in attention techniques, natural language processing, and image classification models. In addition, different from other previous work that uses static question embedding, we investigate how alternative dynamic embedding models enhance the effectiveness of VQA task. The work is evaluated using the latest developed VQA v2 dataset with a 9% improvement over the results obtained with static word embedding. We also deployed the model as a cloud based VQA system to facilitate VQA tasks in real-life applications.**

Keywords—deep learning, dynamic embedding, parallel co-attention, visual question answering

I. Introduction

Visual question answering (VQA) is a challenging task actively studied by computer vision (CV) and natural language processing (NLP) research communities [1] to understand the content of an image and answer questions in natural language. As a highly cognitive task, it requires an understanding of the interactions and relationships between objects as well as actions, events, object counts, and text bound within an image. According to [2], currently most VQA models perform inadequately to provide answers to visual questions and there is significant potential for research and room for improvement to achieve better results.

To address this need, we aim to improve the performance of VQA models by leveraging the latest research and findings in attention techniques, natural language processing, and image classification models. Specifically, to enable a model to answer natural language questions that rely on specific visual information in images, it must be able to identify the relevant areas of the image that pertain to the question and focus on these areas. This gave rise to the concept of "attention," which enables VQA models to enhance their performance by using attention mechanisms to determine "where to look" and incorporating this information into the model. In our research, we adopt and extend the visual co-attention techniques, and incorporate state-of-the-art dynamic word embedding and image classification models to improve the VQA performance. The model is then deployed to the cloud where the system will use reasoning over visual elements, in conjunction with cognitive understanding of input images, and natural language text to infer answers to questions.

II. Related Work

In this section, we will highlight the most used datasets in the VQA research and related research findings in VQA.

A. Datasets

The evolution of VQA datasets has been instrumental in advancing the research in this field. One of the first and most popular VQA datasets is the DAQUAR dataset [3]. It provides indoor images with associated natural language questions, answers, and bounding boxes, which served as a benchmark for early VQA models. However, its size was small with only 1,449 images, 6794 training and 5674 test question-answer pairs. Afterwards, COCO-QA dataset [4] and VQA v1 dataset [5] were introduced with more data. However, COCO-QA dataset limited each answer to be a single word and VQA v1 dataset covered only a limited set of question types and a limited range of visual concepts and attributes. VQA v2 dataset [6] was then released to address such limitations. It is a balanced dataset that adds more comprehensive real-life questions and annotations to cover everyday scenarios. Most existing state-of-the-art VQA models performed worse on VQA v2 dataset because of the dataset's more challenging and realistic setup [6].

B. Existing VQA Research

1) Attention in VQA: Attention was first proposed for the task of machine translation where the decoder could focus on relevant parts of the source language text as it generates the target-language text. Attention has emerged as the most widely used mechanism to address the VQA challenges [7] as different parts of the input image and question may be more relevant to find an answer. Co-attention was then proposed [8] to apply attention to the natural language question together with image attention to aid in answering the question.

2) Hierarchical Co-Attention: Question hierarchy is an important concept that can be used to direct attention to different levels of granularity in the question. One way to implement question hierarchy is by using unigrams, bigrams, and trigrams [9]. Hierarchical co-attention is a technique that uses this question hierarchy to train the model to extract more detailed and nuanced information about different parts of the question to improve the VQA performance. The hierarchical

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

co-attention techniques developed in [10] are still used on [2] as a standard benchmark and demo for VQA. However, the work in [10] used static word embedding (one-hot-encoding) to transform the words used in their questions into tokens to be fed to their network. One of the main drawbacks of using static word embeddings is that they may not be able to capture the nuances of language in a specific domain and words in a contextualized setting [9].

III. VQA Framework

As illustrated in Fig. 1, the VQA framework in our study mainly consists of data preprocessing on both questions and images, and VQA modeling process with the hierarchical co-attention technique.

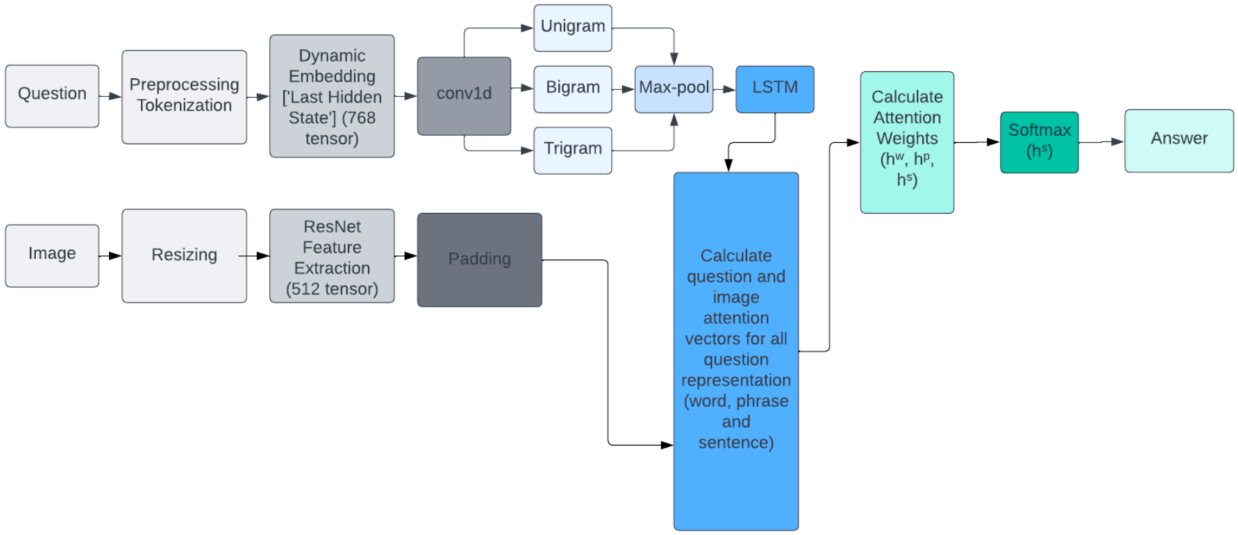
Question
Preprocessing Tokenization
Dynamic Embedding ['Last Hidden State'] (768 tensor)
conv1d
Unigram
Bigram
Trigram
Max-pool
LSTM
Image
Resizing
ResNet Feature Extraction (512 tensor)
Padding
Calculate question and image attention vectors for all question representation (word, phrase and sentence)
Calculate Attention Weights (h^w, h^p, h^s)
Softmax (h^s)
Answer

Fig. 1. Overall VQA framework

A. *Data preprocessing*

Data preprocessing is an important step that involves cleaning, transforming, and organizing the data to make it usable for further analysis. In this study, data preprocessing is conducted on the questions and images.

1) Question Preprocessing: The first several steps for question preprocessing are similar to those of most NLP tasks, i.e., segmenting the questions, eliminating the punctuation and converting all words to lowercase, and then applying tokenization to break down the text into smaller units (tokens). Afterward, word embedding technique is used to represent words in a numerical format such as arrays or tensors. Different from the previous work [10] that uses static embedding, this study investigates the effect of dynamic embedding on VQA performance. Dynamic word embeddings, also known as contextualized word embeddings, have emerged as a successor to static word embeddings in NLP because they are able to capture the meaning of a word in a specific context. A lot of dynamic word embeddings have emerged such as BERT [11] and its variants RoBERTa, ALBERT, XLMRoBERTa [12] as well as GPT [13]. However, dynamic word embeddings are more computationally expensive to generate and require large amounts of training data, making them more challenging to use. As discussed in [14], BERT's last hidden state contains the vast majority of valuable information needed. In our research, we utilized BERT's last hidden layer output to generate question embeddings while minimizing computation across all layers of the BERT model. In addition, we explore the impact of dynamic versus static embedding on model accuracy by applying a range of dynamic embedding models to the questions.

2) Image preprocessing: For image feature extraction, there are several existing pre-trained models that are commonly used including VGG [15], ResNet [16], and Inception [17]. These models have been trained on large datasets, such as ImageNet, and have been pre-trained to extract features from images. Previous VQA studies have used VGG and ResNet. Since ResNet has always been preferred and out-performed VGG [18], ResNet34 is used for image feature extraction in our study. ResNet was originally designed to accept input images with dimensions that are multiples of 112 or 224 [19]. However, larger image sizes require more memory and time to train. To balance performance with computational resources, we resized our images to 448×448. This resolution allows for strong performance while requiring a reasonable amount of memory.

B. *VQA Model*

Co-attention model was proposed in [10] for VQA that jointly reasons about image and question attention. Co-attention is about instructing the model "where to look" or "visual attention," and "what words to listen to" or "question attention." In our work, we adopted the parallel method since it yielded better results than the alternating method [10]. In parallel co-attention, the image representation is used to guide the question attention, and the question representation to guide image attention in parallel.

1) Question hierarchy: For the VQA model, the question is a sentence that the model needs to understand and extract the important features so it can generate an accurate answer. In this work, the model has a hierarchical architecture that co-attends to the image and question at three levels: 1) word-level; 2) phrase level; and 3) question level. At the word-level, we embed the words to a vector space through an embedding matrix. At the phrase level, 1-dimensional convolution neural networks are used to capture the information contained in unigrams, bigrams, and trigrams. Here, unigrams are for individual words in a sentence, bigrams for pairs of consecutive words, and trigrams for triples of consecutive words. These different levels of granularity are commonly used in NLP. Specifically, we convolve word representations with temporal filters of varying support and then combine the various n-gram responses by pooling them into a single phrase-level representation. Once the phrase-level embedding is obtained, it undergoes processing in a Long Short-Term Memory (LSTM). LSTM is a type of Recurrent Neural Networks (RNNs) designed to overcome the problem of vanishing gradients, which can occur when training RNNs on long sequences of data. However, despite their effectiveness, LSTMs can still suffer from forgetting, where previous information is lost as the network processes new input. Therefore, when using LSTMs to create sentence embeddings, n-gram phrase-level representation is used to overcome the problem of forgetting and ensure that important information from previous words at previous timesteps is retained.

2) Parallel co-attention: For each level of the question representation in the question hierarchy, we construct joint question and image co-attention maps, which are then combined recursively to ultimately predict a distribution over the answers. As discussed earlier, each question is tokenized in question preprocessing step and then passed to a dynamic word embedding model. The base model of the dynamic word embedding is used so each word of the question is represented as a 768-size tensor, by taking the output of the BERT model's last hidden layer. ResNet34 outputs 512 features for images so in this work we zero-padded the image tensor with 128 on both sides to match the question embedding. Parallel co-attention attends to the image and question simultaneously. In this work, we connect the image and question by calculating the similarity between image and question features at all pairs of image-locations and question-locations. The image and question attention vectors are calculated as follows:

Given an image V with N features, it is represented as:

$$V \in R^{d\times N} \tag{1}$$

Similarly, a question Q with T words is represented as:

$$Q \in R^{d\times T} \tag{2}$$

The affinity matrix $C \in R^{T\times N}$ is then calculated as:

$$C = \tanh\left(Q^T W_b V\right) \tag{3}$$

where $W_b \in R^{d\times d}$ represents the weights.

After computing this affinity matrix, one possible way of computing the image (or question) attention is to simply maximize out the affinity over the locations of another modality. We consider this affinity matrix as a feature and learn to predict image and question attention maps as shown in (4):

$$\begin{aligned} H^v &= \tanh\left(W_v V + \left(W_q Q\right)C\right) \\ H^q &= \tanh\left(W_q Q + (W_v V)C^T\right) \\ a^v &= \text{softmax}(W_{hv}^T H^v) \\ a^q &= \text{softmax}\left(W_{hq}^T H^q\right) \end{aligned} \tag{4}$$

Here, $W_v, W_q \in R^{k\times d}$ and $W_{hv}, W_{hq} \in R^k$ are the weight parameters; $a^v \in R^N$ and $a^q \in R^T$ are the attention probabilities of each image region v_n and word q_t, respectively. The affinity matrix C transforms question attention space to image attention space (vice versa for C^T). Based on the above attention weights, the image and question attention vectors are calculated as the weighted sum of the image features and question features. Finally, parallel image and question attention vectors are calculated in (5):

$$\hat{v} = \textstyle\sum_{n=1}^{N} a_n^v\, v_n;\ \hat{q} = \sum_{t=1}^{T} a_t^q\, q_t \tag{5}$$

The parallel co-attention is done at each level in the hierarchy, leading to $\hat{v}^r$ and $\hat{q}^r$ where $r \in \{w, p, s\}$ with w being the word, p the phrase and s the sentence. In other words, we predict the answer based on the co-attended features of the image and question from all three levels. To encode the attention features recursively, we utilize a multi-layer-perceptron (MLP) with a SoftMax transformation to the encoded features to obtain the probability matrix of all classes. A class with the highest probability is then chosen as the the predicted answer. Here, the probability is calculated in (6):

$$\begin{aligned} h^w &= \tanh\left(W_w(\hat{q}^w + \hat{v}^w)\right) \\ h^p &= \tanh\left(W_p[(\hat{q}^p + \hat{v}^p), h^w]\right) \\ h^s &= \tanh(W_s[(\hat{q}^s + \hat{v}^s), h^p]) \\ p &= \text{softmax}(W_h h^s) \end{aligned} \tag{6}$$

Here, W_w, W_p, W_s and W_h are the weight parameters, $[\cdot]$ is the concatenation operation on two vectors, and p is the probability of the final answer.

IV. Evaluation

The evaluation is centered on two fundamental components: the VQA model and the deployed web application. This section will synthesize the study's outcomes and evaluations of the model's elements.

A. Experimental dataset and evaluation metric

As discussed in Section II, VQA v2 dataset is used to develop and evaluate our VQA model because it offers a diverse range of images and questions from different sources, providing a comprehensive set for VQA training and evaluation, and is extensively utilized in the research community. Its training dataset has 443,757 questions, 4,437,570 annotations, and the validation has 214,354 questions and 2,143,540 annotations [6].

For model training, with the goal to obtain accurate ground truth answers for each question in the dataset, annotations with a confidence level of “yes” are retained as training data. This ensured that we captured all the possible answers that are with the highest confidence level for each question. As a result, such an extensive list of possible answers lead to the task that’s commonly referred as open-ended VQA. The open-ended nature means that the model must select one answer from a multitude of potential answers, making it more challenging to obtain accurate results. The complexity of this task is further compounded by the fact that the model must not only recognize and understand the image but also comprehend the question and its intent to provide an appropriate answer. A further investigation showed that while the number of answer classes in the VQA v2 dataset is approximately 120,000, among them, a few thousand answers are in fact used to cover the majority of the questions. For example, the top 1,000 most common answers in VQA v2 account for the answers to more than 90% of all questions in the dataset. This discovery allowed us to reframe the open-ended question answering challenge into a 1000-way multiple choice problem. This approach enables the model to potentially provide more accurate answers with a smaller model size. In addition, adopting this approach allowed us to convert the VQA task into a classification problem without losing significant information from the dataset [20].

For model evaluation, for each question, one ground truth answer with the top votes is labeled as the correct answer that will be used to compare with the model predicted answer. This is because simple accuracy is used as the primary evaluation metric for our VQA model. It quantifies the model’s performance by computing the ratio of correct predictions to the total number of predictions or questions as in (7).

$$accuracy = \frac{\#\, of\ questions\ answered\ correctly}{\#\, total\ questions} \tag{7}$$

Simple accuracy does have a potential drawback as it requires exact matches and overlooks factors such as the confidence level of the predictions. However, it is easy to comprehend and interpret, and evaluate model performance more strictly. It is therefore a metric commonly used for VQA model evaluation [2].

B. *VQA task evaluation*

1) Static vs dynamic embedding: The objective of this experiment is to evaluate the effect of static and dynamic word embedding techniques on VQA. One-hot-encoding static embedding used in previous VQA work [10] is compared with three dynamic embeddings including BERT, XLMRoBERTa [12] - a variant of BERT, and T5 [21]. The experiment was conducted using a batch size of 25. For dynamic embedding, the base model has an output size of 768, whereas static embedding utilizes a regular embedding layer with an output of 512. During training, both dynamic and static embeddings utilized ResNet18 for image encoding that has an output size of 512. Therefore, in models using dynamic word embedding, image tensors have to be zero-padded to match the size of question embedding.

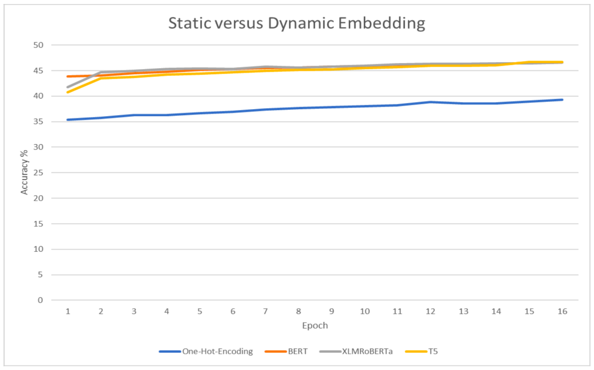

Fig. 2. Static vs dynamic embedding

As shown in Fig. 2, dynamic question embedding improves the accuracy of the VQA model by 9% as compared to the static embedding in previous research. The figure also demonstrates that XLMRoBERTa has a similar impact on accuracy as BERT and T5 [21]. These findings imply that different versions of BERT and T5 do not significantly affect the visual question answering task. We therefore use BERT as the question embedding method in our VQA model. It is also worth noting that the accuracy value (47%) is relatively low because we evaluate how accurately the predicted answer matches with the top 1 ground-truth answer labeled for each question as discussed earlier. Nevertheless, this level of simple accuracy rate is about 10% improvement in performance when compared to those of previous research. In the future, other performance evaluation metrics such as Wu-Palmer Similarity [22] may be used to take into account the semantic meanings and connotation between the model’s predicted answer and ground-truth answers ranked among the top instead of only the top 1 answer (e.g., a “dog” picture may have top-ranked labels including “dog,” “puppy,” “pet” that are semantically related).

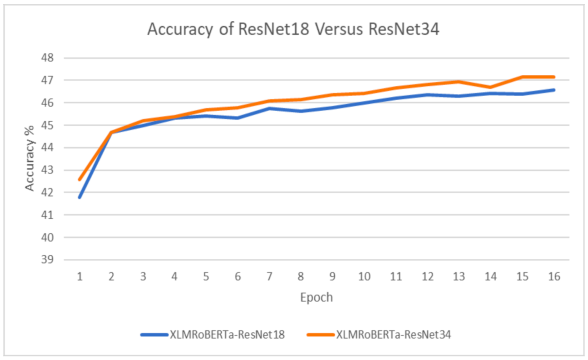

Fig. 3. ResNet18 versus ResNet34

2) ResNet18 vs ResNet34: ResNet18 and ResNet34 are both variants of the ResNet architecture [16] where ResNet34 is deeper and more complex (with 34 layers including 33 convolutional layers and 1 fully connected layer) than ResNet18 (18 layers with 17 convolutional layers and 1 fully connected layer). With BERT base model for question embedding, Fig. 3 reveals that ResNet34 had a slight improvement (approximately 0.5%) over ResNet18 on validation accuracy. However, ResNet34 requires more computational resources due to its increased complexity.

Model deployment: Our developed VQA model is saved and integrated into the Flask app backend. It is then deployed on the Google Cloud Platform service where its REST Web API provides a scalable web endpoint for online inference. We also deveoped a React-based website (https://vqa-react-app3.wm.r.appspot.com) for public access. To assess its usability, a survey based on the USE Questionnaire [23] was developed to solicit feedback from users. The questions cover four main dimensions: usefulness, satisfaction, ease of use, and ease of learning. The sixteen survey participants comprise a representative sample of users from different age groups ranging from 20 to over 40 and different occupation including students, software engineer, reseracher, and marketing. According to the survey findings, users encountered no difficulties in using the web application, and their overall level of satisfaction was rated as a 4 on a scale of 1 (dissatisfaction) to 5 (extremely satisfied). Note that due to the relatively small survey size, the feedback cannot be considered statistically solid. However, the responses are helpful for us to identify the strength of our work and the areas for future improvements.

V. Conclusions

In this work, we present a VQA framework that leverages the latest developments in computer vision and NLP. It uses dynamic word embedding for question encoding and ResNet for image encoding, and employs the parallel co-attention technique to deliver answers for visual questions. The model is trained and evaluated on the VQA v2 dataset with 9% performance improvement over previous research using static word embedding. A web application is also developed for public use and evaluation. In general, the work demonstrates promising outcomes in accurately answering questions based on visual inputs. In the future, performance may be further improved by integrating an optical character recognition (OCR) module to process image-embedded text and contextual data into the model. The model can also be trained with other recently developed image encoders such as U-NET or InceptionV3.

References

[1] K. Kafle and C. Kanan, “Visual question answering: datasets, algorithms, and future challenges,” Computer Vision and Image Understanding, 163, 2016.

[2] “VQA: visual question answering.” [Online]. Available: https://visualqa.org/evaluation.html (Accessed Mar. 13, 2022).

[3] N. Silberman, “NYU Depth Dataset V2,” [Online]. Available: https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html. (Accessed: Feb. 20, 2023).

[4] M. Ren, R. Kiros, and R. S. Zemel, “Exploring models and data for image question answering,” Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 2, pp. 2953–2961, 2015.

[5] S. Antol et al., “VQA: visual question answering,” Proceedings of IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 2425-2433, doi: 10.1109/ICCV.2015.279.

[6] Y. Goyal, T. Khot, A. Agrawal, et al., “Making the V in VQA matter: elevating the role of image understanding in visual question answering,” Int J Comput Vis, vol. 127, pp. 398–414, 2019.

[7] T. Rahman, S. -H. Chou, L. Sigal and G. Carenini, “An improved attention for visual question answering,” Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 2021, pp. 1653-1662.

[8] J. Wu, Z. Hu, and R. Mooney, “Generating question relevant captions to aid visual question answering,” Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3585–3594, 2019.

[9] J. Liu and M. Chen, “COVID-19 Fake News Detector,” Proceedings of 2023 IEEE International Conference on Computing, Networking and Communications, pp. 463-467, 2023.

[10] J. Lu, J. Yang, D. Batra, and D. Parikh, “Hierarchical question-image co-attention for visual question answering,” Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16), pp. 289–297, 2016.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186, 2019.

[12] A. Conneau et al., “Unsupervised cross-lingual representation learning at scale,” Proceedings of Annual Meeting of the Association for Computational Linguistics, pp. 8440-8451, 2020.

[13] T. Brown et al., “Language Models are Few-Shot Learners.” Proceedings of Conference on Neural Information Processing Systems, vol. 33, pp. 1877-1901, 2020.

[14] D. Takeuchi and K. Tran, “Improving SQUAD 2.0 performance using BERT + X,” Stanford, 2019.

[15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” Proceedings of 3rd International Conference on Learning Representations, Computational and Biological Learning Society, pp. 1–14, 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[17] C. Szegedy et al., “Going deeper with convolutions,” Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[18] R. Zhang, L. Du, Q. Xiao, and J. Liu, “Comparison of backbones for semantic segmentation network,” J. Phys.: Conf. Ser., vol. 1544, no. 1, 012196, May 2020, doi: 10.1088/1742-6596/1544/1/012196.

[19] S. Jaiswal, B. Fernando, and C. Tan, “TDAM: top-down attention module for contextually guided feature selection in CNNs,” Proceedings of 17th European Conference on Computer Vision, pp. 259–276, 2022.

[20] J. Couto, “Introduction to visual question answering: Datasets, approaches and evaluation,” Tryolabs, 01-Mar-2018. [Online]. Available: https://tryolabs.com/blog/2018/03/01/introduction-to-visual-question-answering. [Accessed: 10-Feb-2023].

[21] C. Raffel et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” J. Mach. Learn. Res., 21(1), Article 140, 2020.

[22] M. Malinowski and M. Fritz, “A multi-world approach to question answering about real-world scenes based on uncertain input,” Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. 1, pp. 1682–1690, 2014.

[23] A. Lund, “Measuring usability with the USE questionnaire,” Usability Interface, vol. 8, no. 2, pp. 3-6, January 2001.