Privacy Preserving Visual Question Answering

Cristian-Paul Bara[*,1] **Qing Ping,** [2] **Abhinav Mathur,** [2] **Govind Thattai,** [2]
Rohith MV, [2] **Gaurav S. Sukhatme** [2,3]

[1]University of Michigan, [2]Amazon, [3]University of Southern California
cpbara@umich.edu, {pingqing,mathuabh,kurohith,thattg,sukhatme}@amazon.com

Abstract

We introduce a novel privacy-preserving methodology for performing Visual Question Answering on the edge. Our method constructs a symbolic representation of the visual scene, using a low-complexity computer vision model that jointly predicts classes, attributes and predicates. This symbolic representation is non-differentiable, which means it cannot be used to recover the original image, thereby keeping the original image private. Our proposed hybrid solution uses a vision model which is more than 25 times smaller than the current state-of-the-art (SOTA) vision models (Anderson et al. 2018), and 100 times smaller than end-to-end SOTA VQA models (Jiang et al. 2020). We report detailed error analysis and discuss the trade-offs of using a distilled vision model and a symbolic representation of the visual scene.

Introduction

The ubiquity of cameras in personal devices, monitoring devices, and appliances has advanced the skill-set of edge devices considerably, enabling them to answer questions through a combination of computer vision and natural language understanding. This is usually done by streaming video snippets to high-performance computing systems. However, as these images may contain private information (unlike a simple proximity sensor or an IR motion sensor), they can potentially compromise privacy. While some personal devices may alleviate these privacy concerns by performing image analysis, voice recognition, and semantic question-answering all on same device by incorporating expensive neural net hardware accelerators, it is impractical for most personal devices to be so equipped. In this paper, we explore whether accurate visual question answering (VQA) about a scene can be achieved in a cost-effective manner while preserving privacy, on devices that lack expensive edge-computation capabilities.

The problem of answering questions based on an image has been tackled either through a two-stage network that has a region or grid based network feeding into a vision language model downstream to answer the questions, or by using three-stage networks that explicitly generate a symbolic relationship graph from the scene in order to address biases observed in other models (Anderson et al. 2018; Jiang et al. 2020). While these methods are able to tackle a variety of questions, they have two shortcomings: (a) the model sizes are large - they typically have about 3-4 billion parameters which makes them unsuitable for realization in low-cost devices (b) though they can be split up into distinct image processing units (which could be distilled to run on the device) and question answering stages, the feature representation passed between the stages could potentially be decoded to reconstruct the image, thereby compromising privacy.

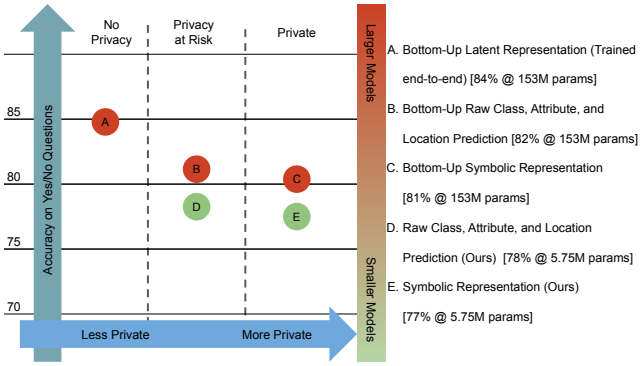

Figure 1: **VQA model tradeoffs** with respect to accuracy, privacy and size. We look at three privacy categories. Not Private where the image is sent to the cloud, at risk where the raw class, and attribute scores as well as the bounding boxes are sent to the cloud, and Private where only symbolic representations of the seen are sent to the cloud.

We look at three levels of privacy as shown in Figure 1. First is a setting where the model is trained end-to-end, and an intermediate feature representation is used as a means to bifurcate model complexity. We consider this not to be private since it is conceivable that an adversarial model can be trained to recover the original image. Second, where the model is trained in stages (as opposed to end-to-end), and the prediction distributions for both objects and attributes are used as intermediate representations. While this is a less descriptive representation of the input image than the end-to-end scenario, it still puts a privacy at risk since it is still a differentiable network that conceivably allows the original image to be recovered by model extraction methods. Third, we consider a symbolic representation of the scene. This ap-

[*]This work has been done during an internship at Amazon Alexa NLU

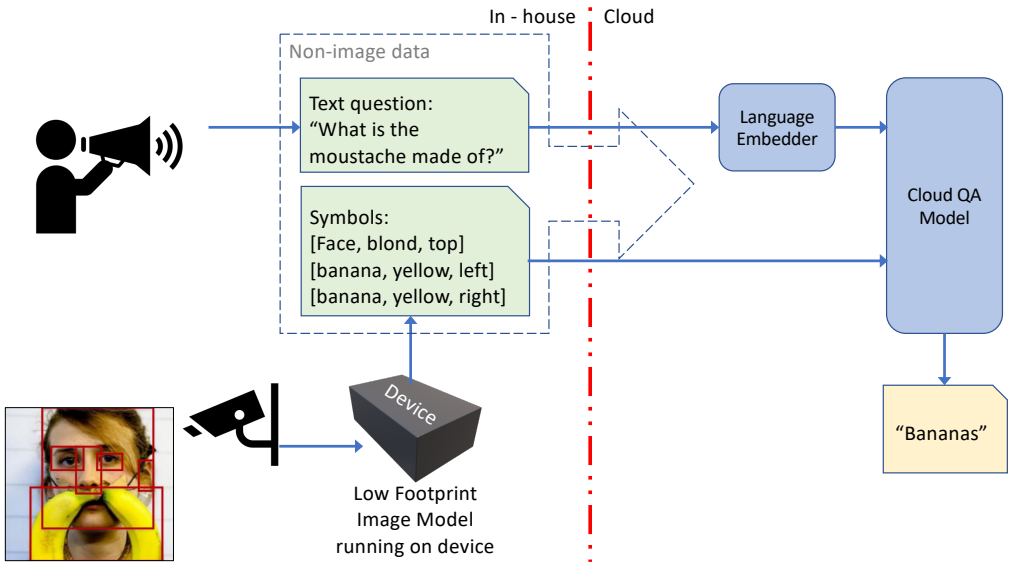

Figure 2: The interconnection of our hybrid model structure. Image adapted from (Yu et al. 2019).

proach makes the system non-differentiable, and makes it close to impossible to recover the original image, making the representation highly private.

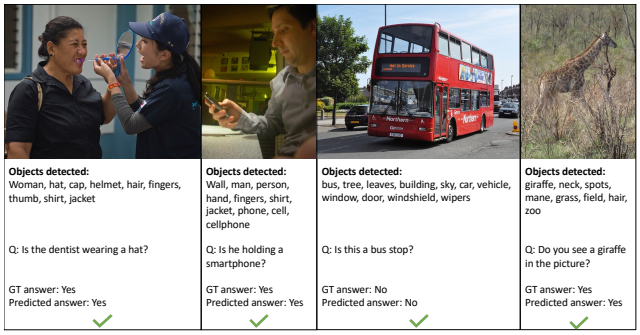

Figure 3: Examples of (a) scenes and questions where Bottom-Up answered incorrectly, but our model answered correctly and (b) scenes and questions where Bottom-Up answered correctly, but our model answered incorrectly.

The novel contributions of this paper are:

- A framework to enable privacy-preserving VQA with a small memory footprint edge-friendly model (reduction in footprint by $> 25\times$) (Anderson et al. 2018).
- A non-differentiable formulation using a symbolic representation to achieve competitive performance results on Yes/No type questions for VQA.
- An analysis of the trade-offs between privacy and performance.

Our results show that even severely limited information about the scene, enables answering a multitude of questions related to presence, attributes, and relationships of objects, while limiting the possibility of an adversarial system to reconstruct the scene. We see a 5% drop in performance from a visual model with a 25x smaller footprint (Anderson et al. 2018). The component running on the device is also 100x smaller than the original end to end model (Jiang et al. 2020). The performance drop is also be attributed to the fact that nearly half of the images in the dataset require information outside of what can be seen in the image, to generate accurate answers to the respective queries (Figure 3). This also raises questions about how SOTA models end-to-end (without external knowledge) were able to answer these queries. We examine the possibility of captioning systems towards creating a human-understandable level of abstraction of input images. This opens up the possibility of analyzing long-term activities, trends, and anomalies using natural language processing techniques in the future.

Related Work

Visual Question Answering

The problem of making sense of visual scenes is one that has been thoroughly pursued by both the Computer Science and Natural Language Processing fields (Wu et al. 2017; Sun et al. 2021). Visual Question Answering (VQA) is a

challenging task that has received increasing attention from both the computer vision and the natural language processing communities. Given an image and a question in natural language, it requires reasoning over visual elements of the image and general knowledge to infer the correct answer.

The intersection of vision and language is an important topic today. To this end there is a large body of work done to gather and annotate data for the creation of models that successfully fuse the two modalities. Out of these datasets we note that the Visual Genome dataset (Krishna et al. 2017) enables modeling of relationships between objects. It provides dense annotations, of objects, attributes, and relationships within each image containing over 100K images each having an average of 21 objects, 18 attributes, and 18 pairwise relationships between objects. These annotations are canonical in region descriptions and question answer pairs to WordNet synsets. Together, these annotations represent the densest and largest dataset of image descriptions, objects, attributes, relationships, and question answers. The image base for Visual Genome is the Microsoft COCO dataset (Lin et al. 2014). MS-COCO gathers images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentation to aid in precise object localization. In total there are 328k images with 2.5 million labeled instances on 91 object types in the COCO dataset. Another interesting dataset and benchmark is GQA (Hudson and Manning 2019). It is a dataset for real-world visual reasoning and compositional question answering, seeking to address key shortcomings of previous VQA datasets. They leverage Visual Genome scene graph structures to create 22M diverse reasoning questions, which all come with functional programs that represent their semantics. The programs are used to have control over biases and answer distributions.

There is a growing number of benchmarks related to VQA. We look at two of them: two stage and three stage models. Two-stage models usually first extract visual features from the image, and then perform cross-modality fusions between visual features and language features to predict final answers. Among this line of work there are task-specific VQA models that are designed and trained specifically on VQA datasets (Anderson et al. 2018; Jiang et al. 2018; Yu et al. 2019, 2020; Guo, Xu, and Tao 2021; Jiang et al. 2020). There are also joint vision-language models that are pre-trained on large vision-language datasets such as Visual Genome (Krishna et al. 2017) and Open Images (Kuznetsova et al. 2020) and then fine-tuned on VQA tasks (Tan and Bansal 2019; Lu et al. 2019; Li et al. 2020; Chen et al. 2020; Zhang et al. 2021). The three-stage models introduce an extra step of generating symbolic scene graph representation for images and symbolic neural module representations for questions, and perform symbolic reasoning between the two representations to predict final answers (Yang et al. 2020; Hu et al. 2017, 2019).

While much inspirations can be drawn from abovementioned work, these methods cannot be readily applied to our application which demands user information for privacy preservation. For most two-stage and three-stage models, the first step is usually to use a heavy-weight image feature extraction model to extract visual features. An example is the widely-used Bottom-Up model which is over 1GB footprint (Anderson et al. 2018) and thus cannot run on the device. In this case, the image has to be uploaded to a high-performance computation infrastructure, which imposes potential privacy risks as the image could be exposed to malicious usage.

One insight we learn from three-stage models is that they are separable modules. The QA models require the generation of a scene graph. Since symbolic outputs, similar in concept to scene graphs, are not differentiable they cannot be used to back-propagate through the model to recover the original image. Taking this into account there is no longer a need to run the whole VQA model on the edge or on dedicated computation infrastructure, but only up to a point where a symbolic representation can be generated on the device and used by downstream models. Towards this goal, if we consider deploying the image feature model on the edge, and uploading less private information to a computation infrastructure, the image feature model has to be lightweight, accurate in predicting symbolic representations such as object names, attributes and relations, and work well with downstream QA models. We consider that this addresses privacy, since it is non-differentiable information and thus cannot be used to reconstruct the original image.

Vision Model Compression

Another important aspect to consider is the compression of neural network models (Cheng et al. 2017; Choudhary et al. 2020; Feng et al. 2019; Lei et al. 2018). Due to the multi-stage nature of most VQA models, there are relatively rare work to compress or distill an entire VQA model as a whole. On the other hand, there are quite some small-footprint vision models such as YOLO(Bochkovskiy, Wang, and Liao 2020), EfficientDet(Tan, Pang, and Le 2020) and so on that are trained from scratch with a smaller neural architecture. Yet these models are usually trained on a limited number of object classes (for example, EfficientDet is trained on 90 object MSCOCO classes) and do not support predicting object attributes and relations.

We look at EfficientNet (Tan and Le 2019) and EfficientDet (Tan, Pang, and Le 2020) which are compressed substitutes to ResNet (He et al. 2016) and Faster-RCNN (Ren et al. 2015) respectively. Convolutional Neural Networks (CNNs) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available. EfficientNet proposes a uniform scaling of all dimensions of the model, i.e. depth, width, and resolution. They provide a family of models, EfficientNets, that outperform previous attempts. Their largest model achieves state of the art accuracy on ImageNet while being 8.4 times smaller and 6.1 times faster than its competitors. EfficientDet is an object detection network that builds upon an EfficientNet backbone. A key difference to competing large scale models, like Faster-RCNN, is that it replaces the region proposal network with a Bi-directional Feature Pyramid Network (BiFPN). This allows easy and fast multi-scale feature fusion. They harness EfficientNet's scaling technique to produce a family of models that, on the COCO dataset, achieves state of the art per-

formance at an up to 9 times smaller size and up to 42 time faster inference than other competing models. For our goal of developing a small-footprint image model that could predict symbolic information robustly, we propose to build a custom vision model that is trained on larger vocabulary of object classes and also predict object attribute information using EfficientDet as the model backbone.

Methodology

With user-facing VQA tasks there is always a concern with the privacy of the collected data. An obvious approach is to obfuscate the input streams. We take inspiration from the architecture of speech to downstream task architectures where there is conversion to a symbolic intermediate representation of the raw input audio measurement, i.e. audio to phonemes and subsequently words, before being sent to a language model for further processing. Current state-of-the-art vision models mostly extract visual features in continuous space and send these features to downstream VQA models. These features could be exploited to recover the original visual information and thus user privacy can be compromised. Instead, we propose to apply obfuscation at input and transform the visual image to symbolic representation first; consisting of object labels, attribute labels and so on, before sending them to downstream VQA models. This ensures that the intermediate symbolic information is not only addressing the privacy issue, but also more controllable in terms of what could be sent to downstream cloud QA models. We introduce this novel perspective in neural network model strategy. Since we implement raw to symbolic transformations for both speech and visual modalities, for the purpose of more control, over filtering out potentially private information, we can consider the symbolic representations as safe with regard to our initial privacy concerns. As such, it is only necessary to deploy the symbolic representation models on user devices and have the larger downstream language and visual QA models run on dedicated high-performance computation infrastructure.

Model Overview

We chose to split our model into two-stages, a perception stage and a question answering stage, see Figure 2. As opposed to other two-stage models like Bottom-Up (Anderson et al. 2018) or MiniVLM (Wang et al. 2020), we simplify the 3 stage graph generating approach where we rely on generating symbolic representations of the scene without going through the higher complexity relationship prediction stage. We use this symbolic output as a privacy granting representation. At a high level the two stages can be instances of any perception models and any downstream question answering model. This structure works as long as symbolic features can be extracted from the perception stage.

We continue with the description of our implementation composed of two stages: an on device image to symbolic representation models with a small footprint and a large scale QA model running on the cloud.

Image Model

The starting point for our image processing module is EfficientDet-D0 (Tan, Pang, and Le 2020) backbone, due to it's small size. The original EfficientDet's base network structure is made up of an EfficientNet processed by a bidirectional feature pyramid network (BiFPN) which is then passed to two output heads. Both of these output heads are a suite of stacked convolutional layers that output the object bounding boxes and the object classes. To note that EfficientDet does not have a region proposal network (RPM) which also accounts for its small footprint. The model will always output $48K$ objects and the relevant objects are picked if their object class prediction is over a threshold after non maximal suppression (at IoU default value 50%). There is no information exchanged between the output heads.

To suit our purposes, we made several modifications to the base EfficientDet-D0 network. First, and most importantly, we introduce an attribute prediction network that makes a multi-label classification on the attributes. As in the base model, there is no information exchanges between the three output heads, i.e. attributes are predicted independently of class, as opposed to MiniVLM (Wang et al. 2020) where attribute prediction is dependant.

We train our vision model on visual genome on the same set of classes and attributes as previously done (Anderson et al. 2018). We first train the object classification and localization heads without the attribute head until convergence. Once the classification and localization heads are trained we reintroduce the attribute head and continue training with all three heads till convergence.

Symbolic Representation

The symbolic representation of the visual scene uses different sections of $\mathbf{R}^{2048}$ vector to represent the various information categories. As previously stated, we use a set of 1600 object classes and 400 attributes. We use the output predictions for object classes, attributes, and bounding boxes and construct a symbolic representation of the image. For object classes, we select the top five class predictions (by confidence score) and concatenate the GloVe (Pennington, Socher, and Manning 2014) embeddings of the names of those classes which gives us a $\mathbf{R}^{1500}$ vector. For attributes we similarly take the top five attributes and compute the sum of the GloVe embeddings of the names of those attributes weighted by their output scores which gives us a $\mathbf{R}^{300}$ vector. For bounding boxes we make two representations - first, we normalize the bounding box to the original image and second we normalize the bounding boxes to the encompassing bounding box which gives us a $\mathbf{R}^{8}$ vector. We make two bounding box representations in order to transmit information global information about the scene as well as relative locations between the bounding boxes. We then concatenate the class, attribute, and bounding box representations together which gives us a $\mathbf{R}^{1808}$ vector. This representation is then padded to create a $\mathbf{R}^{2048}$ vector which is passed to MCAN. We also implement a BERT embedding structure where the question and the tuples of classes and attributes are all embedded by a pretrained BERT model. The classes

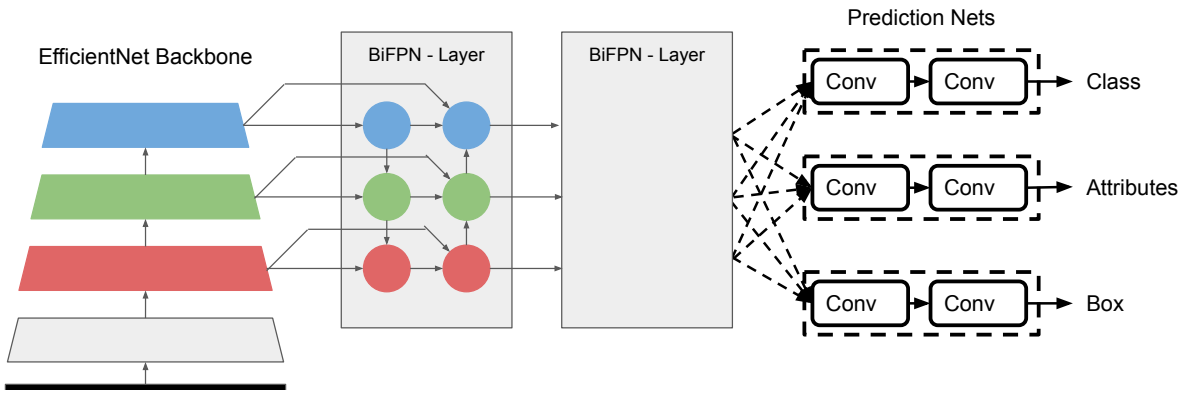

Figure 4: Diagram of EfficientDet and additions. Image adapted from (Tan, Pang, and Le 2020).

Name	Object Detection						Attribute Prediction			
	AP	AR	A	P	R	F1	A	P	R	F1
Bottom-Up	32.50	45.67	62.53	87.02	71.24	78.34	16.23	25.62	30.71	27.93
Ours	39.30	50.48	47.43	84.62	67.13	74.87	33.51	64.33	41.16	50.20

Table 1: Comparison of the performance on object detection with attribute prediction of Bottom-Up, and EfficientDet. We measure Class agnostic Average Precision (AP) class agnostic Average Recall (AR). For detected objects we also compute the Accuracy (A), Precision (P), Recall (R), and F1 score for both class and attribute prediction.

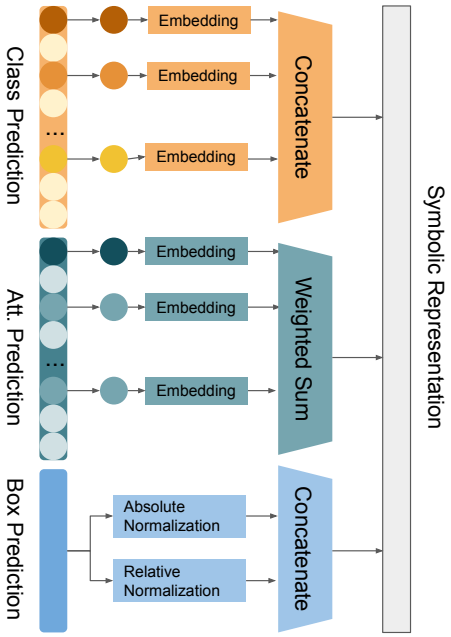

Figure 5: Privacy preserving symbolic representations. Top 5 class predictions are chosen and the GloVe embeddings of the class names are concatenated together. Top 5 attribute predictions are chosen and their GloVe embeddings are weighted by their confidence and summed. The top 5 wights are normalized. The bounding boxes are normalized to the image and to the enveloping bounding box. The Class, attribute, and bounding box representations are then concatenated and sent to MCAN.

and attributes are presented as a sequence of five words and processed to produce an embedding for each tuple, classes and attributes. The question is also embedded as a statement and presented to the downstream QA model.

QA Model

Our formulation of symbolic inputs could be applied in a plug-and-play manner, meaning all existing QA models that takes in image feature and text inputs could potentially work with our symbolic inputs.

We use the architecture of MCAN (Yu et al. 2019) to demonstrate our approach. While we retain the deep co-attention learning, and multimodal fusion and output classifier stages, the novelty of our approach is that we separate the input modality perception stages, specifically the vision model, and run them in a de-coupled fashion. Moreover, we replace the intermediate layer representation from the vision model, originally an adaptation of Faster-RCNN (Ren et al. 2015) to predict attributes and subsequently using Bottom-up Attention, with a symbolic representation of the scene. Given an output from a differentiable model it is conceptually possible to reverse-engineer and recreate the original image. The purpose of the symbolic representation is to break the differentiable link from the originally captured image to the input of MCAN, and hence preserve the privacy.

Experimental Results

Datasets

For the custom EfficientDet model, the Visual Genome dataset is used for pretraining with 1600 object classes and 400 attribute classes (Krishna et al. 2017; Anderson et al. 2018). For the VQA task, the VQA 2.0 dataset is used for evaluating our models (Antol et al. 2015). The VQA 2.0 dataset has 0.25M images, 0.76M questions, and 10M answers.

Experimental Settings

For the custom EfficientDet model, the adamw optimizer is used with an adaptive learning rate starting at 0.001. The model was trained on a 8 GPU P3.8-instance for 3 days for the object classification head, and then trained for 2 days for the attribute classification head. For the VQA model (MCAN), the adamw optimizer is used with an adaptive learning rate, starting at 0.01. The model is trained on a 1 GPU P2-instance for 24 hours.

Model Performance

Image Model To evaluate the change in performance of our vision model as compared to the reference model, Bottom-Up, we look at three aspects. First we look at class agnostic object detection where we measure Average Precision and Average Recall. Second, among the detected objects we evaluate the accuracy, precision, recall, and F1 score of the class prediction. Third, also for the detected objects, we compute accuracy, precision, recall, and F1 score for the attribute prediction task.

We conduct an evaluation of the vision model itself as compared to the reference model, Bottom-Up (Table-1). We notice the our model detects more objects although among the detected objects EfficientDet has a lower rate of successful classifications. On the contrary, the low footprint model has a much better success rate in predicting attributes.

Overall System Performance Since we use MCAN with minimal modifications we use the metrics described in (Yu et al. 2019). The experimental setting starts with exported output from a candidate visual network, i.e. EfficientDet with attributes or Bottom-Up. MCAN is trained alone on different types of representations. For Bottom-Up there are three representations. First, we use the intermediate representations as used in (Jiang et al. 2020). These were trained end to end with both Bottom-up and MCAN and used as provided by the authors. Second, we use the raw predictions of the model, i.e classes $\in \mathbf{R}^{1600}$ concatenated with attributes $\in \mathbf{R}^{400}$ and a global (to the image) and relative (among boxes) normalization $\in \mathbf{R}^{8}$ which is padded to a vector $\in \mathbf{R}^{2048}$ and presented to MCAN as the image representation. Third and finally, we use our novel symbolic representation. For classes we take the GloVe embeddings of the names of the top 5 class predictions and concatenate them into a vector $\in \mathbf{R}^{1500}$. For attributes we take the top 5 predictions and compute a sum, weighted by their confidence, of the GloVe embeddings of the names of the attributes. We also reproduce the same settings with our modification of EfficientDet-D0 with attribute prediction except for the intermediate layer. Since our goal is to break the back-propagation flow to make the representation private, there is no opportunity to train MCAN end to end with EfficientDet to generate the intermediate representations.

When using BERT the question is embedded as a single embedding. This replaces the LSTM in MCAN as well. We notice that the setting using BERT embedding instead of GloVe does not beat the performance of the standard MCAN implementation.

In Table-2 we compare the five settings described before to gauge the penalty for the two aspects in question. First, breaking the differentiability of the model, and second, the use of a small footprint model instead of the original state of the art. We start with the reference model which is the end to end trained model consisting of Bottom-Up and MCAN. This model plays the role of our upper bound on performance. The first comparison using less fine-tuned outputs, see Figure 2, yet still differentiable. We notice this drops the performance by 3.7% **overall** while on the specific tasks 3% on **Yes/No** questions, 5.2% on **count** and 4% on **other**. Next we move to using symbolic representations. When compared to the raw predictions we see a marginal drop in performance with the biggest drop of 1.71% on **count** and less than 1% for everything else and **overall**. Next we replace the Bottom-Up visual model with our modified EfficientDet-D0. For the raw prediction representation we see an **overall** drop of 6.64% drop from the equivalent Bottom-Up configuration. The **other** category sees the highest drop at 10.29% and the lowest drop for the **count** category with 1.24%. Our target model, distilled and private, as compared to the raw prediction distilled version also shows minimal loss in performance.

Using Captions

As mentioned previously, our setup is a middle ground between an end-to-end two stage model and a three stage graph generating model, where we only provide information about object class, attributes and location. Saying this we lose information about the holistic structure of the image. To this end we incorporate the use of captions in our model for two reasons. First, it implies a selection of the most relevant objects in the scene and second, captions imply relationships between these selected objects.

We also look into a hypothetical setting where we would have access to caption descriptions of the given scene. We present the results in Table 3. We use the ground truth captions provided by the MSCOCO dataset as a proxy for the ideal caption generation system. We go through the same comparisons as before though we will skip the intermediate representation configuration. We notice that the low footprint configurations using our modified EfficientDet has the most to benefit from this strategy giving more than a 4% gain overall and for the count category even beating the equivalent Bottom-Up configurations without captions.

Error Analysis

We run a series of error analyses where we compare the performance of the symbolic configurations using Bottom-Up and EfficientDet as the visual model. Specifically, we look at the object predictions on images which have associated Yes/No questions where the Bottom-Up based model answers correctly and the EfficientDet based model gives the wrong answer (see examples in Figure 3). The main observation is that Bottom-Up tends to detect fewer objects and that for a given object Bottom-Up is much more confident in its prediction. However, for the most part, EfficientDet correctly classifies the object and in other cases the correct class is in the top 5 predictions (see Figures 8, and 9). Another

Visual Model	Privacy Status	Deployability	Visual Model Num. Param.	Feature Type	Performance			
					Overall	Other	Yes/No	Count
Bottom-Up	Not Private	No	153M	Intermediate	67.08	58.41	84.73	49.19
Bottom-Up	At risk	No	153M	Raw	63.31	54.48	81.72	43.81
Bottom-Up	Private	No	153M	Symbolic (GloVe)	62.49	53.78	81.05	42.10
Ours	At Risk	Yes	5.75M	Raw	56.67	44.19	77.98	42.57
Ours	Private	Yes	5.75M	Symbolic (GloVe)	55.41	42.95	77.27	41.89
Ours	Private	Yes	5.75M	Symbolic (BERT)	54.17	72.10	75.25	39.16

Table 2: Performance of MCAN on VQA 2.0 using the output from Bottom-Up or EfficientDet with raw predictions representations or fully symbolic output. We also provide the performance of MCAN using the intermediate output provided by Bottom-Up as trained end-to-end as measured by us.

Visual Model	Privacy Status	Deployability	Visual Model Num. Param.	Feature Type	Performance			
					Overall	Other	Yes/No	Count
Bottom-Up	Not Private	No	153M	Raw	65.16	56.72	82.49	47.25
Bottom-Up	At risk	No	153M	Symbolic (GloVe)	64.58	56.32	81.91	46.01
Caption Only	Private	N/A	N/A	N/A	57.55	47.28	76.53	41.83
Ours	At risk	Yes	5.75M	Raw	60.10	49.42	79.26	45.32
Ours	Private	Yes	5.75M	Symbolic (GloVe)	59.76	49.02	79.14	44.62

Table 3: Performance of MCAN on VQA 2.0 using captions along side the output from Bottom-Up or EfficientDet with raw predictions representations or fully symbolic output. We also provide the performance of MCAN using captions alone.

observation is that among the questions that our model gets wrong, there are a large number of questions that require knowledge extrinsic to the image. We also provide comparisons when our model performs better than Bottom up in Figures 6, and 7.

Discussion

An important aspect to note is that the current symbolic approach proves to be feasible especially for Yes/No and counting questions. These types of questions are important because they could be used to break down a more complex query into more specific questions that can be processed later into an answer to the original question. It is also of note that even with low footprint models the drop in performance on simpler questions is acceptably small. Also, our tests of symbolic representations on larger, better performing models are available once the computational performance is available.

We also explored captions as a means to capture the relationships in a scene graph without explicitly and exhaustively generating them. This would bring further holistic information about the entire scene as opposed to the local information provided by object locations, classes, and attributes. Furthermore, captions also make an implicit selection of relevant objects and their separation of from the background. This was reflected in the increased accuracy when captions were used.

Conclusions

We introduced a flexible VQA architecture that, by the nature of its deployment on current low power devices, will maintain image privacy. The privacy maintaining strategy relies on a symbolic representation which eliminates the possibility of accurately reproducing the originally captured image. Our architecture had two major components. First, a low footprint visual perception model that produces a list of objects with their likely classes and attributes, as well as their locations in the scene. Second, a large scale powerful QA model that harnesses the full capabilities of modern cloud computing. Since the information provided to it cannot be used to recover sensitive information, it is privacy preserving as well.

Future Work

As we saw, there is a considerable benefit from incorporation of captions in our model. A logical next step is to design a caption generation module that could be merged into the low footprint vision model; so that it can produce the sentence-type output that gives a broader description of the scene.

Based on our error analysis, an alternate strategy for training EfficientDet would be to aim for matching the output of the Bottom-Up detector. This can be done without additional annotations, i.e. only train as a regression model and use Bottom-Up's output as the target. By priming the model to be as close as possible to Bottom-Up and afterwards to actually do the object recognition task. Finally, this can be done jointly by combining the regression and detection losses together in the same training schema.

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6077–6086.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Bochkovskiy, A.; Wang, C.-Y.; and Liao, H.-Y. M. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Chen, Y.-C.; Li, L.; Yu, L.; Kholy, A. E.; Ahmed, F.; Gan, Z.; Cheng, Y.; and Liu, J. 2020. Uniter: Universal image-text representation learning. In *ECCV*.

Cheng, Y.; Wang, D.; Zhou, P.; and Zhang, T. 2017. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*.

Choudhary, T.; Mishra, V.; Goswami, A.; and Sarangapani, J. 2020. A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review*, 53(7): 5113–5155.

Feng, X.; Jiang, Y.; Yang, X.; Du, M.; and Li, X. 2019. Computer vision algorithms and hardware implementations: A survey. *Integration*, 69: 309–320.

Guo, D.; Xu, C.; and Tao, D. 2021. Bilinear graph networks for visual question answering. *IEEE Transactions on Neural Networks and Learning Systems*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hu, R.; Andreas, J.; Rohrbach, M.; Darrell, T.; and Saenko, K. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, 804–813.

Hu, R.; Rohrbach, A.; Darrell, T.; and Saenko, K. 2019. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10294–10303.

Hudson, D. A.; and Manning, C. D. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6700–6709.

Jiang, H.; Misra, I.; Rohrbach, M.; Learned-Miller, E.; and Chen, X. 2020. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10267–10276.

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

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1): 32–73.

Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; et al. 2020. The open images dataset v4. *International Journal of Computer Vision*, 128(7): 1956–1981.

Lei, J.; Gao, X.; Song, J.; Wang, X.; and Song, M. 2018. Survey of deep neural network model compression. *Journal of Software*, 29(2): 251–266.

Li, X.; Yin, X.; Li, C.; Zhang, P.; Hu, X.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, 121–137. Springer.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Sun, G.; Liang, L.; Li, T.; Yu, B.; Wu, M.; and Zhang, B. 2021. Video question answering: a survey of models and datasets. *Mobile Networks and Applications*, 1–34.

Tan, H.; and Bansal, M. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 6105–6114. PMLR.

Tan, M.; Pang, R.; and Le, Q. V. 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10781–10790.

Wang, J.; Hu, X.; Zhang, P.; Li, X.; Wang, L.; Zhang, L.; Gao, J.; and Liu, Z. 2020. Minivlm: A smaller and faster vision-language model. *arXiv preprint arXiv:2012.06946*.

Wu, Q.; Teney, D.; Wang, P.; Shen, C.; Dick, A.; and van den Hengel, A. 2017. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163: 21–40.

Yang, X.; Lin, G.; Lv, F.; and Liu, F. 2020. TRRNet: Tiered Relation Reasoning for Compositional Visual Question Answering. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 414–430. Springer.

Yu, Z.; Cui, Y.; Yu, J.; Wang, M.; Tao, D.; and Tian, Q. 2020. Deep multimodal neural architecture search. In *Proceedings of the 28th ACM International Conference on Multimedia*, 3743–3752.

Yu, Z.; Yu, J.; Cui, Y.; Tao, D.; and Tian, Q. 2019. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6281–6290.

Zhang, P.; Li, X.; Hu, X.; Yang, J.; Zhang, L.; Wang, L.; Choi, Y.; and Gao, J. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5579–5588.

Appendix

Object and Attribute Prediction Comparison to BottomUp

In the following we show the class and attribute output for both EfficientDet and BottomUp in scenarios where EfficientDet provided output that produces correct results from MCAN but BottomUp did not, and vice-versa. The main observation here is that EfficientDet, although providing good top candidates for both class and attribute prediction, has low confidence when compared to BottomUp. Another observation would be that when EfficientDet provides outputs that lead to wrong answers the questions are likely to require information that is not present in the scene. It is important to remember that these comparable results were generated by a model with a much lower footprint.

We provide a list of the top objects considered in the scene along with their Top-5 class and attribute predictions for each object. The question and answers are also provided.

Is the dentist wearing a hat?

0	man(0.34)	person(0.15)	woman(0.10)	guy(0.05)	boy(0.05)	[smiling(0.09),white(0.07),black(0.07)]
1	hat(0.33)	cap(0.12)	helmet(0.06)	hair(0.06)	head(0.05)	[black(0.16),blue(0.10),red(0.07),white(0.07)]
2	man(0.27)	person(0.14)	woman(0.12)	boy(0.04)	guy(0.04)	[smiling(0.07),black(0.07),white(0.06)]
3	hand(0.25)	finger(0.12)	fingers(0.06)	thumb(0.06)	hands(0.05)	[white(0.13),black(0.08)]
4	shirt(0.24)	jacket(0.11)	man(0.08)	sweater(0.06)	suit(0.05)	[black(0.14),blue(0.12),white(0.11),gray(0.08)]
5	hair(0.21)	head(0.12)	man(0.09)	woman(0.07)	face(0.06)	[brown(0.11),black(0.08),blonde(0.07),short(0.06),dark(0.05)]
6	shirt(0.17)	jacket(0.14)	sweater(0.08)	man(0.08)	coat(0.05)	[black(0.14),white(0.11),blue(0.08),gray(0.07)]
7	man(0.16)	head(0.11)	woman(0.09)	person(0.08)	hair(0.07)	[smiling(0.08),black(0.07)]
8	head(0.15)	man(0.11)	hair(0.09)	person(0.07)	woman(0.07)	[black(0.06),smiling(0.05)]
9	head(0.15)	hat(0.09)	hair(0.07)	face(0.06)	man(0.05)	[black(0.11),red(0.09),blue(0.05)]
10	people(0.14)	man(0.13)	men(0.09)	wall(0.09)	person(0.05)	[white(0.06),black(0.06)]
11	finger(0.12)	hand(0.11)	thumb(0.07)	fingers(0.05)	hands(0.03)	[white(0.14),black(0.08)]
12	window(0.12)	wall(0.10)	building(0.06)	door(0.05)	mirror(0.04)	[white(0.10),black(0.06)]
13	hand(0.12)	wrist(0.07)	finger(0.06)	bracelet(0.06)	watch(0.06)	[white(0.16),black(0.06),red(0.06)]
14	arm(0.12)	sleeve(0.08)	hand(0.06)	strap(0.04)	shirt(0.03)	[black(0.17),white(0.17),gray(0.06)]
15	face(0.12)	head(0.11)	man(0.07)	hair(0.07)	woman(0.05)	[black(0.08)]
16	wall(0.11)	people(0.08)	man(0.07)	background(0.06)	men(0.05)	[white(0.07),blue(0.06),black(0.05)]
17	hand(0.11)	hands(0.04)	arm(0.04)	finger(0.03)	handle(0.03)	[white(0.11),black(0.10)]
18	face(0.10)	head(0.10)	hair(0.06)	hat(0.04)	man(0.04)	[black(0.12),white(0.06)]
19	ear(0.10)	face(0.04)	earring(0.03)	nose(0.03)	hand(0.03)	[black(0.12),white(0.07),brown(0.06)]
20	wall(0.10)	window(0.08)	building(0.04)	door(0.04)	mirror(0.03)	[white(0.12),black(0.06),brown(0.05)]
21	sunglasses(0.10)	glasses(0.08)	face(0.04)	hair(0.03)	head(0.03)	[black(0.15),white(0.06)]
22	scissors(0.10)	hand(0.07)	phone(0.04)	handle(0.04)	knife(0.03)	[white(0.09),black(0.09)]
23	face(0.10)	glasses(0.05)	head(0.04)	nose(0.04)	hair(0.03)	[black(0.10),brown(0.05)]
24	wrist(0.09)	hand(0.09)	bracelet(0.05)	watch(0.04)	tattoo(0.04)	[white(0.15),black(0.06),red(0.06)]
25	glasses(0.09)	sunglasses(0.08)	eyeglasses(0.04)	hair(0.03)	eyes(0.02)	[black(0.14),white(0.07)]
26	hair(0.08)	arm(0.05)	shirt(0.05)	head(0.03)	person(0.03)	[black(0.13),white(0.08),brown(0.07)]
27	hair(0.08)	bangs(0.03)	forehead(0.03)	glasses(0.02)	head(0.02)	[brown(0.12),blonde(0.10),short(0.08),black(0.08),dark(0.07)]
28	hand(0.08)	finger(0.04)	watch(0.04)	wrist(0.03)	phone(0.03)	[white(0.10),black(0.07)]
29	face(0.07)	ear(0.05)	head(0.05)	hair(0.04)	glasses(0.03)	[black(0.12),white(0.07)]
30	nose(0.07)	mouth(0.05)	glasses(0.04)	face(0.03)	hand(0.03)	[black(0.11),white(0.07)]
31	hand(0.07)	man(0.04)	hands(0.04)	camera(0.04)	phone(0.04)	[black(0.09),white(0.07)]
32	finger(0.07)	hand(0.05)	wrist(0.05)	thumb(0.04)	bracelet(0.03)	[white(0.15),black(0.07)]
33	bandana(0.07)	hat(0.07)	sunglasses(0.05)	hair(0.05)	visor(0.03)	[black(0.15),white(0.08),blue(0.08),red(0.07)]
34	ear(0.07)	face(0.04)	hand(0.04)	hair(0.03)	head(0.03)	[black(0.14),white(0.09)]
35	hand(0.07)	finger(0.06)	phone(0.03)	toothbrush(0.02)	handle(0.02)	[white(0.10),black(0.07)]

Is the dentist wearing a hat?, Pred:yes, GT:no

0	hair(0.81)	head(0.13)	woman(0.01)	face(0.01)	girl(0.00)	[dark(0.26),short(0.11),brown(0.28),black(0.17)]
1	shirt(0.71)	tshirt(0.10)	top(0.03)	jacket(0.02)	man(0.01)	[black(0.47),blue(0.32)]
2	bracelet(0.71)	bracelets(0.15)	wrist(0.04)	watch(0.02)	wristband(0.02)	[colorful(0.37),yellow(0.11),pink(0.15),multi colored(0.13)]
3	man(0.67)	woman(0.12)	person(0.04)	guy(0.02)	lady(0.02)	[standing(0.10),smiling(0.14),black(0.10)]
4	shirt(0.62)	tshirt(0.13)	woman(0.05)	top(0.05)	man(0.02)	[black(0.36),blue(0.30)]
5	woman(0.55)	man(0.10)	girl(0.08)	lady(0.06)	shirt(0.05)	[young(0.13),black(0.11),blue(0.13)]
6	cap(0.48)	hat(0.40)	baseball cap(0.05)	head(0.03)	beanie(0.01)	[black(0.11),blue(0.69)]
7	window(0.48)	door(0.05)	building(0.04)	wall(0.03)	frame(0.01)	[white(0.14)]
8	hat(0.44)	cap(0.37)	head(0.10)	baseball cap(0.04)	beanie(0.01)	[black(0.16),blue(0.63)]
9	head(0.36)	cap(0.27)	hat(0.25)	baseball cap(0.04)	woman(0.01)	[black(0.11),blue(0.64)]
10	hair(0.48)	head(0.32)	woman(0.05)	face(0.04)	man(0.02)	[dark(0.19),brown(0.26),black(0.16)]
11	man(0.30)	woman(0.11)	shirt(0.10)	wall(0.03)	jacket(0.03)	[black(0.13),blue(0.34)]
12	wall(0.27)	building(0.21)	background(0.05)	window(0.05)	door(0.02)	[white(0.45),gray(0.10)]
13	building(0.26)	wall(0.17)	man(0.09)	background(0.03)	woman(0.03)	[white(0.39),blue(0.16)]
14	man(0.35)	woman(0.22)	person(0.04)	wall(0.02)	hair(0.02)	[smiling(0.17)]
15	hair(0.21)	woman(0.19)	head(0.15)	girl(0.06)	hat(0.03)	[black(0.16),blue(0.19)]
16	head(0.27)	woman(0.21)	hair(0.05)	hat(0.04)	man(0.04)	[blue(0.40)]

Figure 6: Comparison on the output of EfficientDet on the bottom, left when rotated, and Bottom-Up on top, right when rotated. Our answers are correct and BottomUp's answers are incorrect. Answers: Ours: yes, BottomUp: no

Figure 7: Comparison on the output of EfficientDet on the bottom, left when rotated, and Bottom-Up on top, right when rotated. Our answers are correct and BottomUp's answers are incorrect. Answers: Ours: no, BottomUp: yes

Figure 8: Comparison on the output of EfficientDet on the bottom, left when rotated, and Bottom-Up on top, right when rotated. Our answers are incorrect and BottomUp's answers are correct. Answers: Ours:yes, BottomUp:no

Figure 9: Comparison on the output of EfficientDet on the bottom, left when rotated, and Bottom-Up on top, right when rotated. Our answers are incorrect and BottomUp's answers are correct. Answers: Ours:yes, BottomUp:no