

DocVQA: A Dataset for VQA on Document Images

Minesh Mathew¹ Dimosthenis Karatzas² C.V. Jawahar¹

¹CVIT, IIT Hyderabad, India ²Computer Vision Center, UAB, Spain

minesh.mathew@research.iit.ac.in, dimos@cvc.uab.es, jawahar@iit.ac.in

Abstract

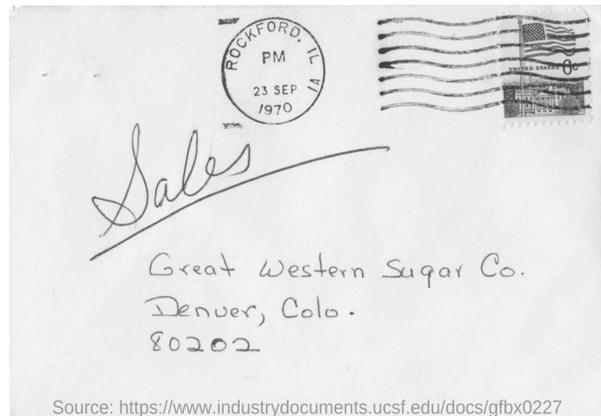
We present a new dataset for Visual Question Answering (VQA) on document images called DocVQA. The dataset consists of 50,000 questions defined on 12,000+ document images. Detailed analysis of the dataset in comparison with similar datasets for VQA and reading comprehension is presented. We report several baseline results by adopting existing VQA and reading comprehension models. Although the existing models perform reasonably well on certain types of questions, there is large performance gap compared to human performance (94.36% accuracy). The models need to improve specifically on questions where understanding structure of the document is crucial. The dataset, code and leaderboard are available at docvqa.org

1. Introduction

Research in Document Analysis and Recognition (DAR) is generally focused on information extraction tasks that aim to convert information in document images into machine readable form, such as character recognition [10], table extraction [22] or key-value pair extraction [30]. Such algorithms tend to be designed as task specific blocks, blind to the end-purpose the extracted information will be used for.

Progressing independently in such information extraction processes has been quite successful, although it is not necessarily true that holistic document image understanding can be achieved through a simple constructionist approach, building upon such modules. The scale and complexity of the task introduce difficulties that require a different point of view.

In this article we introduce Document Visual Question Answering (DocVQA), as a high-level task dynamically driving DAR algorithms to conditionally interpret document images. By doing so, we seek to inspire a “purpose-driven” point of view in DAR research. In case of Document VQA, as illustrated in Figure 1, an intelligent reading system is expected to respond to ad-hoc requests for information, expressed in natural language questions by human



Q: Mention the ZIP code written?

A: 80202

Q: What date is seen on the seal at the top of the letter?

A: 23 sep 1970

Q: Which company address is mentioned on the letter?

A: Great western sugar Co.

Figure 1: Example question-answer pairs from DocVQA. Answering questions in the new dataset require models not just to read text but interpret it within the layout/structure of the document.

users. To do so, reading systems should not only extract and interpret the textual (handwritten, typewritten or printed) content of the document images, but exploit numerous other visual cues including layout (page structure, forms, tables), non-textual elements (marks, tick boxes, separators, diagrams) and style (font, colours, highlighting), to mention just a few.

Departing from generic VQA [13] and Scene Text VQA [35, 5] approaches, the document images warrants a different approach to exploit all the above visual cues, making use of prior knowledge of the implicit written communication conventions used, and dealing with the high-density semantic information conveyed in such images. Answers in case of document VQA cannot be sourced from a closed dictionary, but they are inherently open ended.

Previous approaches on bringing VQA to the documents

domain have either focused on specific document elements such as data visualisations [19, 21] or on specific collections such as book covers [28]. In contrast to such approaches, we recast the problem to its generic form, and put forward a large scale, varied collection of real documents.

Main contributions of this work can be summarized as following:

- We introduce DocVQA, a large scale dataset of 12, 767 document images of varied types and content, over which we have defined 50, 000 questions and answers. The questions defined are categorised based on their reasoning requirements, allowing us to analyze how DocVQA methods fare for different question types.
- We define and evaluate various baseline methods over the DocVQA dataset, ranging from simple heuristic methods and human performance analysis that allow us to define upper performance bounds given different assumptions, to state of the art Scene Text VQA models and NLP models.

2. Related Datasets and Tasks

Machine reading comprehension (MRC) and open-domain question answering (QA) are two problems which are being actively pursued by Natural Language Processing (NLP) and Information Retrieval (IR) communities. In MRC the task is to answer a natural language question given a question and a paragraph (or a single document) as the context. In case of open domain QA, no specific context is given and answer need to be found from a large collection (say Wikipedia) or from Web. MRC is often modelled as an extractive QA problem where answer is defined as a span of the context on which the question is defined. Examples of datasets for extractive QA include SQuAD 1.1 [32], NewsQA [37] and Natural Questions [27]. MS MARCO [29] is an example of a QA dataset for abstractive QA where answers need to be generated not extracted. Recently Transformer based pretraining methods like Bidirectional Encoder Representations from Transformers (BERT) [9] and XLNet [41] have helped to build QA models outperforming Humans on reading comprehension on SQuAD [32]. In contrast to QA in NLP where context is given as computer readable strings, contexts in case of DocVQA are document images.

Visual Question Answering (VQA) aims to provide an accurate natural language answer given an image and a natural language question. VQA has attracted an intense research effort over the past few years [13, 1, 17]. Out of a large body of work on VQA, scene text VQA branch is the most related to our work. Scene text VQA refers to VQA systems aiming to deal with cases where understanding scene text instances is necessary to respond to the questions posed. The ST-VQA [5] and TextVQA [35] datasets were introduced in parallel in 2019 and were quickly fol-

lowed by more research [36, 11, 39].

The ST-VQA dataset [5] has 31,000+ questions over 23,000+ images collected from different public data sets. The TextVQA dataset [35] has 45,000+ questions over 28,000+ images sampled from specific categories of the OpenImages dataset [25] that are expected to contain text. Another dataset named OCR-VQA [28] comprises more than 1 million question-answer pairs over 207K+ images of book covers. The questions in this dataset are domain specific, generated based on template questions and answers extracted from available metadata.

Scene text VQA methods [16, 11, 35, 12] typically make use of pointer mechanisms in order to deal with out-of-vocabulary (OOV) words appearing in the image and provide the open answer space required. This goes hand in hand with the use of word embeddings capable of encoding OOV words into a pre-defined semantic space, such as Fast-Text [6] or BERT [9]. More recent, top-performing methods in this space include M4C [16] and MM-GNN [11] models.

Parallely there have been works on certain domain specific VQA tasks which require to read and understand text in the images. The DVQA dataset presented by Kafle *et al.* [20, 19] comprises synthetically generated images of bar charts and template questions defined automatically based on the bar chart metadata. The dataset contains more than three million question-answer pairs over 300,000 images.

FigureQA [21] comprises over one million yes or no questions, grounded on over 100,000 images. Three different types of charts are used: bar, pie and line charts. Similar to DVQA, images are synthetically generated and questions are generated from templates. Another related QA task is Textbook Question Answering (TQA) [23] where multiple choice questions are asked on multimodal context, including text, diagrams and images. Here textual information is provided in computer readable format.

Compared to these existing datasets either concerning VQA on real word images, or domain specific VQA for charts or book covers, the proposed DocVQA comprise document images. The dataset covers a multitude of different document types that include elements like tables, forms and figures, as well as a range of different textual, graphical and structural elements.

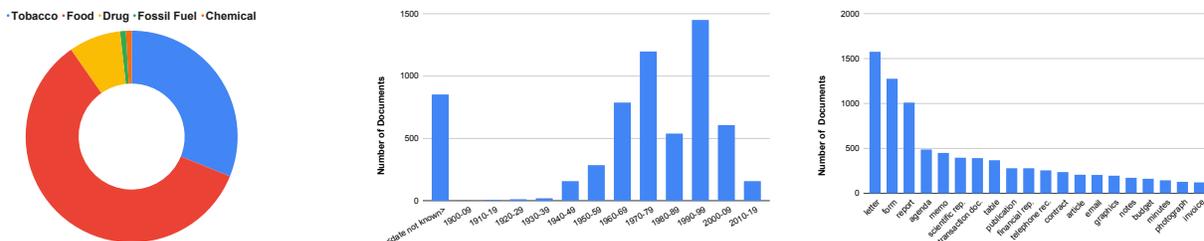
3. DocVQA

In this section we explain data collection and annotation process and present statistics and analysis of DocVQA.

3.1. Data Collection

Document Images: Images in the dataset are sourced from documents in UCSF Industry Documents Library¹. The documents are organized under different industries and

¹<https://www.industrydocuments.ucsf.edu/>



(a) Industry-wise distribution of the documents. (b) Year wise distribution of the documents. (c) Various types of documents used.

Figure 2: Document images we use in the dataset come from 6071 documents spanning many decades, of a variety of types, originating from 5 different industries. We use documents from UCSF Industry Documents Library.

further under different collections. We downloaded documents from different collections and hand picked pages from these documents for use in the dataset. Majority of documents in the library are binarized and the binarization has taken on a toll on the image quality. We tried to minimize binarized images in DocVQA since we did not want poor image quality to be a bottleneck for VQA. We also prioritized pages with tables, forms, lists and figures over pages which only have running text.

The final set of images in the dataset are drawn from pages of 6,071 industry documents. We made use of documents from as early as 1900 to as recent as 2018. (Figure 2b). Most of the documents are from the 1960-2000 period and they include typewritten, printed, handwritten and born-digital text. There are documents from all 5 major industries for which the library hosts documents — tobacco, food, drug, fossil fuel and chemical. We use many documents from food and nutrition related collections, as they have a good number of non-binarized images. See Figure 2a for industry wise distribution of the 6071 documents used. The documents comprise a wide variety of document types as shown in Figure 2c.

Questions and Answers: Questions and answers on the selected document images are collected with the help of remote workers, using a Web based annotation tool.

The annotation process was organized in three stages. In stage 1, workers were shown a document image and asked to define at most 10 question-answer pairs on it. We encouraged the workers to add more than one ground truth answer per question in cases where it is warranted.

Workers were instructed to ask questions which can be answered using text present in the image and to enter the answer verbatim from the document. This makes VQA on the DocVQA dataset an extractive QA problem similar to extractive QA tasks in NLP [32, 37] and VQA in case of ST-VQA [5]. The second annotation stage aims to verify the data collected in the first stage. Here a worker was shown an image and questions defined on it in the first stage (but not the answers from the first stage), and was required to enter

answers for the questions. In this stage workers were also required to assign one or more question types to each question. The different question types in DocVQA are discussed in subsection 3.2. During the second stage, if the worker finds a question inapt owing to language issues or ambiguity, an option to flag the question was provided. Such questions are not included in the dataset.

If none of the answers entered in the first stage match exactly with any of the answers from the second stage, the particular question is sent for review in a third stage. Here questions and answers are editable and the reviewer either accepts the question-answer (after editing if necessary) or ignores it. The third stage review is done by the authors themselves.

3.2. Statistics and Analysis

The DocVQA comprises 50,000 questions framed on 12,767 images. The data is split randomly in an 80–10–10 ratio to train, validation and test splits. The train split has 39,463 questions and 10,194 images, the validation split has 5,349 questions and 1,286 images and the test split has 5,188 questions and 1,287 images.

As mentioned before, questions are tagged with question type(s) during the second stage of the annotation pro-

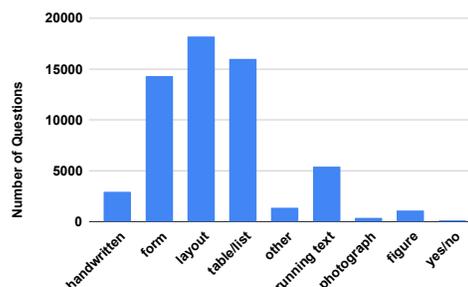


Figure 3: The 9 question types and share of questions in each type.

tected objects. Each OCR token recognized from the image is represented using (i) a pretrained word embedding (FastText), (ii) appearance feature of the token’s bounding box from the same Faster R-CNN which is used for appearance features of objects (iii) PHOC [2] representation of the token and (iv) bounding box coordinates of the token. Then these feature representations of the three entities (question tokens, objects and OCR tokens) are projected to a common, learned embedding space. Then a stack of Transformer [38] layers are applied over these features in the common embedding space. The multi-head self attention in transformers enable both inter-entity and intra-entity attention. Finally, answers are predicted through iterative decoding in an auto-regressive manner. Here the fixed vocabulary used for the closed answer space is made up of the most common answer words in the train split. Note that in this case the fixed vocabulary comprises of answer words, not answers itself as in the case of LoRRA. At each step in the decoding, the decoded word is either an OCR token from the image or a word from the fixed vocabulary of common answer words.

In our experiments we use original LoRRA and M4C models and few variants of these models. Document images in DocVQA usually contain higher number of text tokens compared to images in scene text VQA datasets. Hence we try out larger dynamic vocabularies (i.e. more OCR tokens are considered from the images) for both LoRRA and M4C. For both the models we also evaluate performance when no fixed vocabulary is used.

Since the notion of visual objects in real word images is not directly applicable in case of document images, we also try out variants of LoRRA and M4C where features of objects are omitted.

4.3. Reading Comprehension Models

In addition to the VQA models which can read text, we try out extractive question answering / reading comprehension models from NLP. In particular, we use BERT [9] question answering models. BERT is a method of pre-training language representations from unlabelled text using transformers [38]. These pretrained models can then be used for downstream tasks with just an additional output layer. In the case of extractive Question Answering, this is an output layer to predict start and end indices of the answer span.

5. Experiments

In this section we explain evaluation metrics and our experimental settings and report results of experiments.

5.1. Evaluation Metrics

Two evaluation metrics we use are Average Normalized Levenshtein Similarity (ANLS) and Accuracy (Acc.). ANLS was originally proposed for evaluation of VQA on

Baseline	val		test	
	ANLS	Acc.	ANLS	Acc.
Human	-	-	0.981	94.36
Random answer	0.003	0.00	0.003	0.00
Rnandom OCR token	0.013	0.52	0.014	0.58
Longest OCR token	0.002	0.05	0.003	0.07
Majority answer	0.017	0.90	0.017	0.89
Vocab UB	-	31.31	-	33.78
OCR substring UB	-	85.64	-	87.00
OCR subsequence UB	-	76.37	-	77.00

Table 1: Evaluation of different heuristics and upper bounds. Predicting random answers or majority answer do not even yield 1% accuracy. Answers are a substring of the serialized OCR output in more than 85% of the cases.

ST-VQA [4]. The Accuracy metric measures percentage of questions for which the predicted answer matches exactly with any of the target answers for the question. Accuracy metric awards a zero score even when the prediction is only a little different from the target answer. Since no OCR is perfect, we propose to use ANLS as our primary evaluation metric, so that minor answer mismatches stemming from OCR errors are not severely penalized.

5.2. Experimental setup

For measuring human performance , we collect answers for all questions in test split, with help a few volunteers from our institution.

In all our experiments including heuristics and trained baselines, OCR tokens we use are extracted using a commercial OCR application. For the heuristics and upper bounds we use a vocabulary 4,341 answers which occur more than once in the train split.

For LoRRA and M4C models we use official implementations available as part of the MMF framework [34]. The training settings and hyper parameters are same as the ones reported in the original works. The fixed vocabulary we use for LoRRA is same as the vocabulary we use for computing vocabulary based heuristics and upper bounds. For M4C the fixed vocabulary we use is a vocabulary of the 5,000 most frequent words from the answers in the train split.

For QA using BERT, three pre-trained BERT models² from the Transformers library [40] are used. The models we use are bert-base-uncased, bert-large-uncased-whole-word-masking and bert-large-uncased-whole-word-masking-finetuned-squad. We abbreviate the model names as bert-base, bert-large and bert-large-squad respectively. Among these, bert-large-squad is a pre-trained model which is also finetuned on SQuAD 1.1 for question answering. In case of extractive question answering or reading compre-

²https://huggingface.co/transformers/pretrained_models.html

Method	Objects' feature	Fixed vocab.	Dynamic vocab. size	val		test	
				ANLS	Acc.	ANLS	Acc.
LoRRA [35]	✓	✓	50	0.110	7.22	0.112	7.63
	✓	✗	50	0.041	2.64	0.037	2.58
	✗	✓	50	0.102	6.73	0.100	6.43
	✓	✓	150	0.101	7.09	0.102	7.22
	✓	✓	500	0.094	6.41	0.095	6.31
M4C [16]	✓	✓	50	0.292	18.34	0.306	18.75
	✓	✗	50	0.216	12.44	0.219	12.15
	✗	✓	50	0.294	18.75	0.310	18.92
	✗	✓	150	0.352	22.66	0.360	22.35
	✗	✓	300	0.367	23.99	0.375	23.90
	✗	✓	500	0.385	24.73	0.391	24.81

Table 2: Performance of the VQA models which are capable of reading text — LoRRA [35] and M4C [16]. Detection of visual objects and their features (bottom-up attention), which is a common practice in VQA is ineffective in case of DocVQA.

hension datasets ‘contexts’ on which questions are asked are passages of electronic text. But in DocVQA ‘contexts’ are document images. Hence to finetune the BERT QA models on DocVQA we need to prepare the data in SQuAD style format where the answer to a question is a ‘span’ of the context, defined by start and end indices of the answer. To this end we first serialize the OCR tokens recognized on the document images to a single string, separated by space, in top-left to bottom-right order. To approximate the answer spans we follow an approach proposed in TriviaQA [18], which is to find the first match of the answer string in the serialized OCR string.

The bert-base model is finetuned on DocVQA on 2 Nvidia GeForce 1080 Ti GPUs, for 2 epochs, with a batch size of 32. We use Adam optimizer [24] with a learning rate of $5e - 05$. The bert-large and bert-large-squad models are finetuned on 4 GPUs for 6 epochs with a batch size of 8, and a learning rate of $2e - 05$.

Pretrained model	DocVQA finetune	val		test	
		ANLS	Acc.	ANLS	Acc.
bert-base	✓	0.556	45.6	0.574	47.6
bert-large	✓	0.594	49.28	0.610	51.08
bert-large-squad	✗	0.462	36.72	0.475	38.26
bert-large-squad	✓	0.655	54.48	0.665	55.77

Table 3: Performance of BERT question answering models. A BERT_{LARGE} model which is fine tuned on both SQuAD 1.1 [32] and DocVQA performs the best.

5.3. Results

Results of all heuristic approaches and upper bounds are reported in Table 1. We can see that none of the heuristics get even a 1% accuracy on the validation or test splits.

OCR substring UB yields more than 85% accuracy on both validation and test splits. It has a downside that the substring match in all cases need not be an actual answer match. For example if the answer is “2” which is the most common answer in the dataset, it will match with a “2” in “2020” or a “2” in “2pac”. This is the reason why we evaluate the *OCR subsequence UB*. An answer is a sub sequence of the serialized OCR output for around 76% of the questions in both validation and test splits.

Results of our trained VQA baselines are shown in Table 2. First rows for both the methods report results of the original model proposed by the respective authors. In case of LoRRA the original setting proposed by the authors yields the best results compared to the variants we try out. With no fixed vocabulary, the performance of the model drops sharply suggesting that the model primarily relies on the fixed vocabulary to output answers. Larger dynamic vocabulary results in a slight performance drop suggesting that incorporating more OCR tokens from the document images does little help. Unlike LoRRA, M4C benefits from a larger dynamic vocabulary. Increasing the size of the dynamic vocabulary from 50 to 500 improves the ANLS by around 50%. And in case of M4C, the setting where features of objects are omitted, performs slightly better compared to the original setting.

Results of the BERT question answering models are reported in Table 3. We observe that all BERT models perform better than the best VQA baseline using M4C (last row in 2). The best performing model out of all the baselines analysed is the bert-large-squad model, finetuned on DocVQA. Answers predicted by this model match one of

References

- [1] Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. C-VQA: A compositional split of the visual question answering (VQA) v1.0 dataset. *arXiv preprint arXiv:1704.08243*, 2017.
- [2] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Word spotting and recognition with embedded attributes. *TPAMI*, 2014.
- [3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering, 2017.
- [4] Ali Furkan Biten, Rubèn Tito, Andrés Mafla, Lluís Gómez, Marçal Rusiñol, Minesh Mathew, C. V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. ICDAR 2019 competition on scene text visual question answering. *CoRR*, abs/1907.00490, 2019.
- [5] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluís Gomez, Marçal Rusiñol, Ernest Valveny, C.V. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, 2019.
- [6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5.
- [7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 2017.
- [8] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CPVR*, 2009.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.
- [10] David Doermann, Karl Tombre, et al. *Handbook of document image processing and recognition*. Springer, 2014.
- [11] Difei Gao, Ke Li, Ruiping Wang, Shiguang Shan, and Xilin Chen. Multi-modal graph neural network for joint reasoning on vision and scene text. In *CVPR*, 2020.
- [12] Lluís Gómez, Ali Furkan Biten, Rubèn Tito, Andrés Mafla, and Dimosthenis Karatzas. Multimodal grid features and cell pointers for scene text visual question answering. *arXiv preprint arXiv:2006.00923*, 2020.
- [13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering, 2016.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 1997.
- [16] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *CVPR*, 2020.
- [17] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 2901–2910, 2017.
- [18] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, 2017.
- [19] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *CVPR*, 2018.
- [20] Kushal Kafle, Robik Shrestha, Scott Cohen, Brian Price, and Christopher Kanan. Answering questions about data visualizations using efficient bimodal fusion. In *WACV*, 2020.
- [21] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2017.
- [22] Isaak Kavasidis, Carmelo Pino, Simone Palazzo, Francesco Rundo, Daniela Giordano, P Messina, and Concetto Spampinato. A saliency-based convolutional neural network for table and chart detection in digitized documents. In *ICIAP*, 2019.
- [23] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*, 2017.
- [24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [25] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Mallocci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from <https://storage.googleapis.com/openimages/web/index.html>*, 2017.
- [26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 2017.
- [27] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.
- [28] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. OCR-VQA: Visual question answering by reading text in images. In *ICDAR*, 2019.
- [29] Tri Nguyen et al. Ms marco: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268, 2016.

- [30] Rasmus Berg Palm, Ole Winther, and Florian Laws. Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. In *ICDAR*, 2017.
- [31] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.
- [32] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.
- [33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*. 2015.
- [34] Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Mmf: A multimodal framework for vision and language research. <https://github.com/facebookresearch/mmf>, 2020.
- [35] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019.
- [36] Ajeet Kumar Singh, Anand Mishra, Shashank Shekhar, and Anirban Chakraborty. From strings to things: Knowledge-enabled vqa model that can read and reason. In *ICCV*, 2019.
- [37] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordani, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830, 2016.
- [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*. 2017.
- [39] Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *CVPR*, 2020.
- [40] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.
- [41] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*. 2019.