Visual Question Answering using Saliency maps

Arpit Jain and Manush Gupta

Department of Computer Science
University of Massachusetts, Amherst

Abstract

Visual Question Answering or VQA is the task of answering questions in natural language by reasoning over the visual information given by an image. This problem is a supervised learning classification problem, where we are given labeled images with object instance information and we are also given free-form questions and answers to guide learning. The problem is fairly recent and has been receiving a lot of attention from computer vision, natural language processing, and deep learning communities. In our project, we build upon the language and image fusion models and dataset proposed in [1] and introduce the idea of saliency maps [2] to pre-process the input data to learn better feature representations of the important parts of the image. We make use of a blending mechanism to introduce the idea of a weighted dropout[3] for feature activation instead of just using saliency activation maps. We show that our method would work slightly better than the baseline model without other attention mechanisms involved during training. We make use of the natural bias in humans to ask questions about easily noticeable parts of the image because of how bottom's up saliency work cognitively, ie, our eyes tend to process high-frequency changes in an image first.

1 Introduction

Recent developments in computer vision and deep learning have significantly improved the performance in many computer vision tasks, such as image classification[4, 5], object detection [6, 7], activity recognition [8, 9, 10] etc. Deep convolutional neural networks (CNN) can rival the abilities of humans to do image classification given sufficient data is available[5] and with annotated datasets rapidly increasing in size similar outcomes can be anticipated for other focused computer vision problems. However, these problems are narrow in scope and do not require a holistic understanding of images. As humans, we can identify objects in an image, understand the spatial positions of these objects, infer their attributes and relationships to each other, and also reason about the purpose of each object given the surrounding context. We can ask arbitrary questions about images and also communicate the information gleaned from them. Over the last few years, the community interest in the problem of VQA has significantly increased.

The task requires the computer to answer a question about an image requiring the model to understand the content of the image and the question together. There are many potential applications for VQA. The most immediate is as an aid to blind and visually impaired individuals, enabling them to get information about images both on the web and in the real world. For example, as a blind user scrolls through their social media feed, a captioning system can describe the image and then the user could use VQA to query the image to get more insight about the scene. More generally, VQA could be used to improve human-computer interaction as a natural way to query visual content. A VQA system can also be used for image retrieval, without using image meta-data or tags. For example, to find all images taken in a rainy setting, we can simply ask 'Is it raining?' to all images in the dataset. Beyond applications, VQA is an important basic research problem. Because a good VQA system must be

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

Figure 1: VQA sample

able to solve many computer vision problems, it can be considered a component of a Turing Test for image understanding [11, 12]

The most commonly used deep learning approach is to extract a global image feature vector using a convolution neural network (CNN)[13] and encode the corresponding question as a feature vector using a long short-term memory network (LSTM) [14] and then combine them to infer the answer. This baseline model is only able to achieve an accuracy of around 54 percent on the testing dataset provided by the paper itself. Therefore there is a lot of scope for accuracy improvements in this field which remains the primary focus of our work.

By examining the VQA dataset we observe that many question concern the objects in the scene. The background does not provide information relevant to answering the questions. Therefore applying a saliency mask that can separate foreground objects from background information can help us learn better feature representation for the input images and boost the overall performance of the VQA task. This mimics an attention mechanism like pre-processing step that may be applied later on during training as explained in relative detail in the section 2 below containing related word. We experiment with only the baseline model to show the effectiveness of our idea, however, we think that similar improvements could be achieved in deep learning models which focus on attention mechanisms that solve this task as our idea complements the fact that the visual and question information guide human attention on the image and the questions.

We give a detailed explanation of our model in section 3. For our experiments, we were greatly limited with the computational resources at hand, therefore we chose to compare our model with the baseline model on a subset of the whole dataset, by learning on a lesser number of object categories. We explain these restrictions in section 4. We also consider other experiments involving tuning the hyperparameters of our blending mechanism including the corner cases in section 5. We are able to show relative improvements in accuracy results as detailed in section 6. Section 7 is a discussion of the problems we faced during this project and our conclusions from our results. We also propose other avenues worth looking at in terms of changing loss functions to learn better feature representations and further improving accuracy which is also motivated partly by classifying the foreground pixels more accurately.

2 Related Work

The VQA task gained popularity after deep learning approaches had already achieved significant backing due to their state-of-the-art performance on various vision and NLP tasks (Krizhevsky et al., 2012) (Bahdanau et al., 2014). As a result, almost all work on VQA in the literature involves deep learning approaches, as opposed to more classical approaches like graphical models. There are a couple of models which use a non-neural approach [15, 16]. [15] uses a Bayesian framework for VQA in which they predict the answer type of a question and use that to generate the answer. Deep learning models for VQA involve the use of convolutional neural networks(CNN) to embed the image and word embeddings such as Word2Vec or GloVe vectors along with Recurrent Neural Networks (RNNs) to embed the question which is then merged to learn a joint model. This model will serve as our baseline model for comparison purposes. [1] was the first to address the problem of

free-form, open-ended questions and answers provided by humans by forming their dataset called the VQA dataset.

They develop a 2-channel vision (image) + language (question) model that culminates with a softmax classifier.

The vision channel provides an embedding for the image which are activations from the last hidden layer of the VGG-net. The question channel provides an embedding for the question. Although several word embeddings are implemented, we use the LSTM model. An LSTM with one hidden layer is used. The embedding obtained from the LSTM is a concatenation of last cell state and last hidden state representations from the hidden layer of the LSTM.

Each question word is encoded with some embedding by a fully-connected layer and a non-linearity which is then fed to the LSTM. The input vocabulary to the embedding layer consists of all the question words seen in the training dataset.

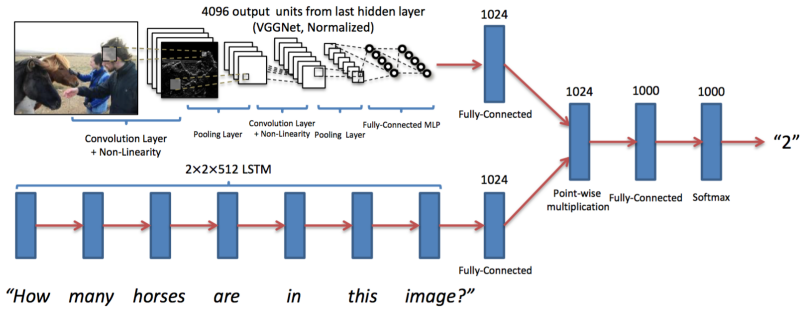

Figure 2: VQA baseline model architecture

The most popular techniques involve the use of attention based techniques [**Lu2016HierarchicalQC** , 17] . [18] is one of the state of the artworks in this field by modelling the visual attention and the newly introduced question attention model to jointly produce co-attention pairs. [17] proposes a VGGnet for encoding the image and concatenate the outputs of the final two layers of VGGnet to obtain image encoding. Question representation is obtained by averaging the word vectors of each word in the question. An attention vector is computed over the set of image features to decide which region in the image to give importance to. We were highly motivated by this idea to learn better features for our images.

We argue that we can introduce a saliency map mask to the input images to give more importance to the foreground of the image which will often have the information necessary for the task. While this could also be applied to both [17] and [18], it was out of scope for this paper's purposes. We implement our model over the baseline model and show that we can get better results by learning slightly better features with an induced saliency mask during pre-processing itself as compared to the baseline model.

3 Methodology

3.1 VQA baseline implementation method

As discussed previously we implement the baseline model from [1] paper which uses the VGG19 net merged with an LSTM to learn a join embedding between image and questions to answer the questions. We choose the top 1000 most frequent answers as possible outputs. This set of answers. The different components of this model are described below in detail:

3.1.1 Image Channel:

This channel provides an embedding for the image. The activations from the last hidden layer of VGGNet [48] are used as 4096-dim image embedding. These activations are 'l2 normalized

activations from the last hidden layer of VGGNet [48]. In addition to this, we implement the whole VGG19 model in front of the last hidden layer and load its pretrained weights. We play around with this model as part of our experiments as described later.

3.1.2 Question Channel:

This channel provides an embedding for the question. An LSTM with one hidden layer is used to obtain 1024-dim embedding for the question. The embedding obtained from the LSTM is a concatenation of last cell state and last hidden state representations (each being 512-dim) from the hidden layer of the LSTM. Each question word is encoded with 300-dim embedding by a fully-connected layer + tanh non-linearity which is then fed to the LSTM. The input vocabulary to the embedding layer consists of all the question words seen in the training dataset.

3.1.3 Multi-Layer Perceptron (MLP):

The image and question embeddings are combined to obtain a single embedding. The image embedding is first transformed to 1024-dim by a fully-connected layer + tanh non-linearity to match the LSTM embedding of the question. The transformed image and LSTM embeddings (being in a common space) are then fused via element-wise multiplication.

3.2 Our Model:

3.2.1 Core Idea:

Our core idea lies in implementing an intuitive weighted dropout technique wherein we try to multiply each pixel of the input image with a mask that determines its relative importance in view of answering a potential question. We argue that this relative importance of pixels can be represented as Saliency of the pixel and then we use the model proposed in [2] to obtain this mask which we explain below. We identify that questions are often always asked about foreground parts of the image rather than the background of the image so it would make sense to use a saliency mask to further support our idea.

In terms of the machine learning theory, we view the background information as true negatives that are easy to classify and contribute significantly to the loss despite being unimportant just because of the imbalance between background and foreground in an image(background usually being much larger than foreground).

We don't completely mask out the background because we allow for the possibility of questions on the background as well. We also try completely switching off the background portions of the image to see the relative performance on the question data. We control this by the use of a hyperparameter alpha which is described later in this section.

3.2.2 Saliency Map Model:

Our implementation of foreground and background segmentation is taken from [2]. This is a very classical approach to calculating saliency maps because it's not based on more recent deep learning based models for this task. We have used this method because of its easily implementable method which at the same time could help us demonstrate the effectiveness of our idea accurately.

The model proposed here (Fig.3) builds on a biologically-plausible architecture, proposed by Koch and Ullman [19] and at the basis of several models [20],[21].It is related to the so-called "feature integration theory", proposed to explain human visual search strategies.Visual input is first decomposed into a set of topographic feature maps. Different spatial locations then compete for saliency within each map, such that only locations which locally stand out from their surround can persist. All feature maps feed in a purely bottom-up manner, into a master "saliency map", which topographically codes for local conspicuity over the entire visual scene. This does not require any top-down guidance to shift attention.

The input is provided in terms of static color images. Nine spatial scales are created using dyadic Gaussian pyramids [11], which progressively low-pass filter and subsample the input image, yielding horizontal and vertical image reduction factors ranging from 1:1(scale 0) to 1:256(scale 8)in eight octaves.

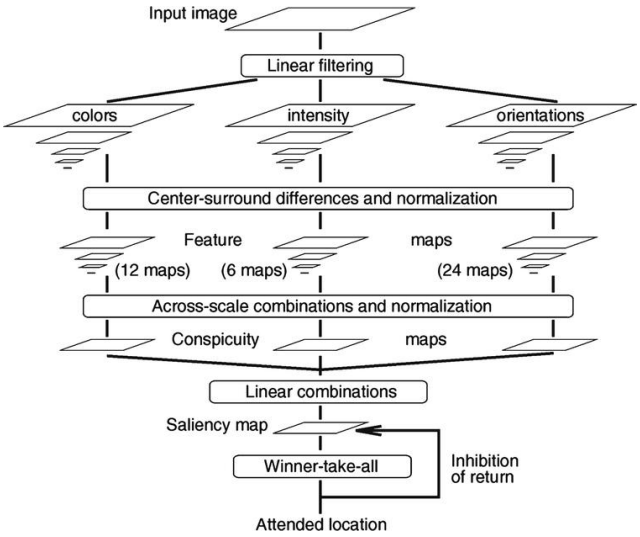

Figure 3: VQA sample

Each feature is computed by a set of linear 'center surround' operations similar to visual receptive fields. This is implemented as a difference between fine and coarse scales. With r,g, and b being the red, green and blue channels of the input image, an intensity image I is obtained as I =(r+g+b)/3. The first set of feature maps is concerned with intensity contrast between a fine-scaled image and a coarse scaled image.

The second set of maps is similarly constructed for the color channels, which in the cortex are represented using a so-called "color double-opponent"system. In the center of their receptive field, neurons are excited by one color(e.g., red)and inhibited by another (e.g., green), while the converse is true in the surround. Such spatial and chromatic opponency exists for the red/green, green/red, blue/yellow and yellow/blue color pairs in human primary visual cortex [22].

The third set of feature maps is obtained from local orientation information using oriented Gabor pyramids where the orientation is taken along every 45 degrees from 0 to 135 degrees.

The final saliency map is obtained after obtaining one conspicuity map for every type of feature set described above and averaging over all 3. The procedure for making a conspicuity map has been very well discussed in the original paper.

Fig 5 shows a few saliency maps obtained after this procedure.

Figure 4: Source images

Figure 5: Saliency maps

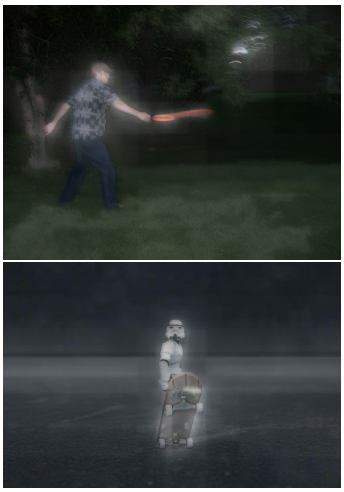

Figure 6: Blended images

3.2.3 Incorporating the Saliency map:

After having obtained the saliency map, we introduce the idea of making a blended image from the original image and the obtained grayscale saliency map to make a joint representation of the two, essentially doing what we called weighted dropout earlier. This involved scaling up the saliency activation map into an RGB image first, and then controlling the blend with a hyperparameter (alpha) as follows :

$$blended_image = \alpha * (mask_image) + (1 - \alpha) * (source_image)$$

We scale up the grayscale image to RGB because VGG weights are learned for 3-dimensional images and, in order to apply transfer learning effectively from the pretrained weights, we thought it best to stick to the original format of the images.

Fig. 6 some of the sample images that we obtain after the blending operation. As we can see, the salient regions have gotten lighter(ie have received a higher pixel value) and the nonsalient regions have gotten darker(ie have received a lower pixel value). This has effectively translated our earlier intuition of asserting more importance to the salient features of the image to arrive at a better convergence of the loss function given the task of answering questions.

4 Dataset

We follow the 2015 version 1 of the VQA dataset introduced by [vqa]. The original dataset included 82,783 training images taken from MSCOCO, 40,504 validation images, and 81,434 testing images. For every image, the dataset provided 3 free-form natural-language questions with 10 concise open-ended answers each. Datasets provide two formats of the VQA task: open-ended and multiple-choice Questions.

We chose this dataset because of the exhaustive data collected as part of it. This dataset was the first of its kind to introduce free form/open-ended questions made by someone who isn't asking the questions.

Midway through our experiments, we realized the scale of this task and aptly concluded that we would not have the computational resources to work on a dataset of this scale. In order to run our experiments and confirm our hypothesis, we introduced the following restrictions on the dataset :

-From the image dataset, we only chose images related to 3 object categories: person, dog, and skateboard instead of the total of 80 object categories provided by MS COCO.

-Also, we choose to take only 1000 images from each of these categories for training. After subtracting images common between all the 3 categories we were left with a total of 2905 images over which we wanted to transfer learn our weights in the VGG19 and the LSTM model.

- For testing, we take 500 images from each of the three categories of objects from the provided testing dataset of images.

- We included all the questions pertaining to each image, so we got a total of 8715 questions for training.

- For training, we obtained the most popular answer from the provided 10 answers for each question in the dataset.

In order to implement these restrictions, we went through APIs of both VQA and MS COCO dataset and represented many mappings of metadata according to these restrictions. We mapped each word in the question and answer to an integer number, by using a numeric id for each unique word in the training corpus of these answers. A similar process was followed for the question representation. The question representations were padded with zeroes to make the total length of questions 26 for each question.

5 Experiments

Inorder to run our experiments, we make separate containers for all our train images, test images, masked images from the saliency maps and the blended images with two different alpha values.

All of our experiments are done with batch size = 32, and the number of epochs = 10. The system on which these experiments were done had, 4GB RAM in GPU, using the Nvidia GeForce GTX 1050 with memory clock rate of 1.493 GHz.

We chose batch size as 32 because higher batch sizes caused memory allocation errors in the tensor, ie, our GPU was running out of memory.

We conduct the following experiments on the dataset.

1) We test the accuracy of our baseline implementation on the curated test dataset of 500 images from each category. That is,in this case our $\alpha = 0$

In order to do a fair comparison of test accuracy, we recalculate the test accuracy of the baseline model on our subset of the test data with our batch-size and number of epochs as opposed to the values used in the original paper.

2) We test the accuracy of our masked implementation using saliency maps as input for fine-tuning the model with VGG19 weights frozen on the curated test dataset. Ie, in this case, our $\alpha = 1$.

3) We test the accuracy of our masked implementation using the blended images with hyperparameter $\alpha = 0.7$, for fine-tuning the model with VGG19 weights frozen on the curated test dataset.

4) We test the accuracy of our masked implementation using the blended images with hyperparameter $\alpha = 0.5$, for fine-tuning the model with VGG19 weights frozen on the curated test dataset.

We also tried fine-tuning the full VGG model after loading the pretrained weights but we faced computational issues even with low batch sizes. Finally, we did train our VGG model for 4 epochs on CPU(which had higher memory space than GPU) but the tradeoff was with speed, which took us around 2 days to train although we could not obtain test data accuracy for this experiment.)

6 Results

These are the results we obtain for all our different models $\alpha = [0, 0.5, 0.7, 010]$:

Alpha	Model Accuracy
0	0.4523
0.5	0.4575
0.7	0.4647
1	0.4527

Figure 7: table showing accuracy across all models

From figure 7, we see that our models outperform the baseline VQA model [1].

From figure 8, we note that our model with $\alpha = 0.7$ converges to the least loss. We also see that $\alpha = 0.5$ follow a similar curve but converges to a slightly higher value. The most interesting result is with $\alpha = 1$ where we see that the loss function flat lines and stops learning abruptly. We suspect that this is because we have omitted essential background information in the input images and so the loss isn't able to converge properly.

Figure 9 shows our accuracy vs epochs curve for all the models. Again we see that the accuracy stops improving dramatically for the completely masked model but the decrease in improvement over accuracy is more gradual for the other two models.

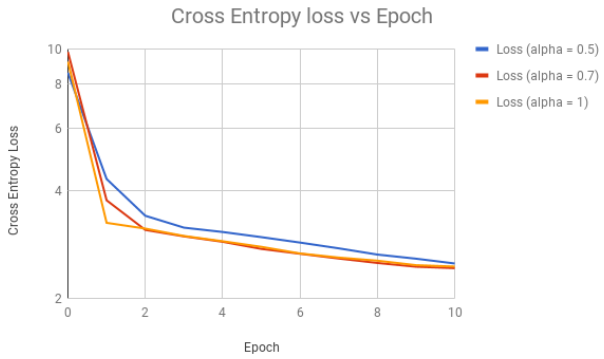

Figure 8: Cross Entropy Loss vs Epochs for all our models

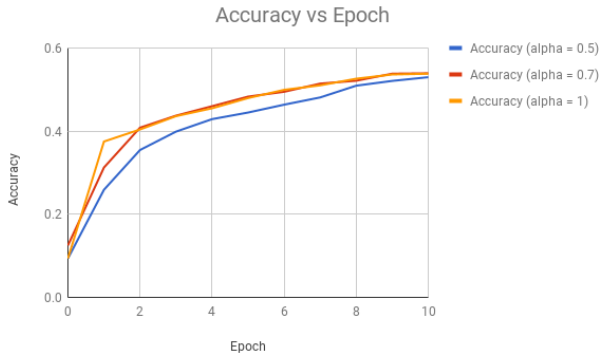

Figure 9: Accuracy vs Epochs for all our models

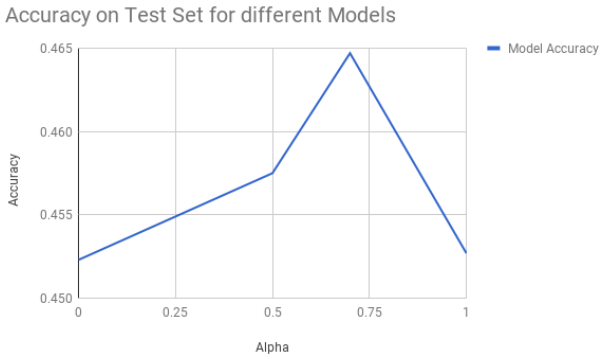

Figure 10: Visual comparison of accuracy across all the models

From figure 10, it is easily to visualize how the various hyperparameter settings of α worked out in terms of accuracy of our models. We provide a detailed discussion of the interpretations of this visualization in the next section.

7 Conclusions and Discussion

We show that our best model with $\alpha = 0.7$ performs better than the baseline model of [1]. We also show that the accuracy decreases from $\alpha = 0.7$ to $\alpha = 0.5$. This suggests that a stronger mask

on the foreground produces better results. This proves our hypothesis that salient attention during pre-processing helps in achieving better feature representation for images according to the relevant question data. However, we also see that $\alpha = 1$ performs worse than both of these but slightly better than $\alpha = 0$ (baseline). This suggests that a stronger mask may result in accuracy improvements but completely masking out background information gives a decrease in accuracy confirming our earlier hypothesis about using a weighted dropout-like method instead of a complete on-off dropout-like method for better results. We also get confirmation that using just a salient mask as input images still outperforms the baseline model, hence, proving that background information is relatively unimportant in answering questions.

For future efforts we would like to do the following:

- We would like to fine-tune our VGG19 model by freezing a few initial layers. We suspect that the image embeddings learned from this fine-tuning will far outperform our present best model. This is because we think that the image features represented by the VGG model do not aptly correspond to our blended images and require some fine-tuning.
- We would also like to see the effects of using a slightly more sophisticated model to produce saliency maps which employ the use of deep Convolutional Neural Networks.
- Instead of implementing our method on the baseline model, we would like to see the results of implementing our model over the state of the art model of [18].
- Finally, we would like to involve the use of the focal loss function provided by [23](Yet to be published). This loss function also tries to lessen the effect of an easily classified background portion of the image by introducing a modulating factor $(1 - \rho)^{\lambda}$ with a tunable focusing parameter λ . Therefore the loss function reads like :

  $$loss = -(1 - \rho)^{\lambda} * (log\rho)$$

  where $-log\rho$ is the cross-entropy loss.

References

[1] Stanislaw Antol et al. "VQA: Visual Question Answering". In: *The IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015.

[2] Laurent Itti, Christof Koch, and Ernst Niebur. "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 20.11 (Nov. 1998), pp. 1254–1259. ISSN: 0162-8828. DOI: 10.1109/34.730558. URL: http://dx.doi.org/10.1109/34.730558.

[3] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[4] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). arXiv: 1409.1556. URL: http://arxiv.org/abs/1409.1556.

[5] K. He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[6] J. Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[7] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes et al. Curran Associates, Inc., 2015, pp. 91–99. URL: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

[8] Jeff Donahue et al. "Long-term Recurrent Convolutional Networks for Visual Recognition and Description". In: *CVPR*. 2015.

[9] A. Karpathy et al. “Large-Scale Video Classification with Convolutional Neural Networks”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. June 2014, pp. 1725–1732. DOI: 10.1109/CVPR.2014.223.

[10] Karen Simonyan and Andrew Zisserman. “Two-Stream Convolutional Networks for Action Recognition in Videos”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 568–576. URL: http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf.

[11] Donald Geman et al. “Visual Turing test for computer vision systems.” In: *Proceedings of the National Academy of Sciences of the United States of America* 112 12 (2015), pp. 3618–23.

[12] Mateusz Malinowski and Mario Fritz. “Towards a Visual Turing Challenge”. In: *CoRR* abs/1410.8027 (2014). arXiv: 1410.8027. URL: http://arxiv.org/abs/1410.8027.

[13] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324. ISSN: 0018-9219. DOI: 10.1109/5.726791.

[14] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[15] Kushal Kafle and Christopher Kanan. “Answer-type prediction for visual question answering”. In: *CVPR*. 2016.

[16] Mateusz Malinowski and Mario Fritz. “A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 1682–1690. URL: http://papers.nips.cc/paper/5411-a-multi-world-approach-to-question-answering-about-real-world-scenes-based-on-uncertain-input.pdf.

[17] Kevin J. Shih, Saurabh Singh, and Derek Hoiem. “Where To Look: Focus Regions for Visual Question Answering”. In: *Computer Vision and Pattern Recognition*. 2016.

[18] Jiasen Lu et al. “Hierarchical Question-Image Co-Attention for Visual Question Answering”. In: *NIPS*. 2016.

[19] Christof Koch and Shimon Ullman. “Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry”. In: 4 (Feb. 1985), pp. 219–27.

[20] Thierry Pun Ruggero Milanese Sylvia Gil. “Attentive mechanisms for dynamic and static scene analysis”. In: *Optical Engineering* 34 (1995), pp. 34 - 34 - 7. DOI: 10.1117/12.205668. URL: http://dx.doi.org/10.1117/12.205668.

[21] Shumeet Baluja and Dean Pomerleau. “Expectation-based selective attention for visual monitoring and control of a robot vehicle”. In: 22.4-Mar (Dec. 1997), pp. 329–344.

[22] Stephen Engel, Xuemei Zhang, and Brian Wandell. “Colour tuning in human visual cortex measured with functional magnetic resonance imaging”. In: 388 (Aug. 1997), pp. 68–71.

[23] Tsung-Yi Lin et al. “Focal Loss for Dense Object Detection”. In: *CoRR* abs/1708.02002 (2017). arXiv: 1708.02002. URL: http://arxiv.org/abs/1708.02002.