Mathematical Problems of Computer Science 50, 5-14, 2018.

# Image Caption Generation Model Based on Object Detector

Aghasi S. Poghosyan and Hakob G. Sarukhanyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: agasy18@gmail.com, hakop@ipia.sci.am

**Abstract**

Automated semantic information extraction from the image is a difficult task. There are works, which can extract image caption or object names and their coordinates. This work presents object detection and automated caption generation implemented via a single model. We have built an image caption generation model on top of object detection model. We have added extra layers on object detector to increase caption generator performance. We have developed a single model that can detect objects, localize them and generate image caption via natural language.

**Keywords:** Neural networks, Image caption, Object detection, Deep learning, RNN, LSTM

## 1. Introduction

Automatic description of an image content is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. The progress of parallel computing is boosting the progress of machine learning. In the last years many working systems have been developed and many works published on computer vision topic. All these works enabled solution of hard problems such as image object classification, object detection, and localization, scene detection, semantic segmentation. In parallel machine translation approaches based on neural networks have been developed. These approaches have been applied on images, creating systems that can describe the image content via natural language sentences [1]-[4]. There is huge work to do because these systems are not complete, because they don't have human accuracy level. Nevertheless, generally they are working fast enough to have many applications in the real world. For example search engines, social networks are using the systems to annotate media content (images, videos, animations) with different annotating systems. In this case, it is very important that the annotating system will provide more annotation via single iteration. Therefore we have developed a single model that can detect objects, localize them and generate image caption via natural language.

Automatically generated image caption should contain the main object names, their properties, relations and actions. Moreover, the generated caption should be expressed through a natural language like English. There are a number of works approaching this problem. Some of them [3, 5, 6] offer combining existing image object detection and sentence

generation systems. But there is a more efficient solution [1] that offers a joint model. It takes an image and generates the caption, which describes the image adequately. The latest achievements in statistical machine translation were actively used in image caption generation tasks. The reason for this is mainly the proven achievement of greater results when using a powerful sequential model trained by maximizing the probability of the correct translation for the input sentence. These models [1, 7, 8] are based on Recurrent Neural Networks (RNNs). The model encodes the variable length input into the fixed length vector representation. This representation enables conversion of the input sentence into the target sentence or the input image into the target image caption.

Neural nets have become a leading method for high quality object detection in recent years. Modern object detectors based on Convolutional Neural Network (CNN) [9] networks, such as Faster Region-based Convolutional Neural Network (Faster R-CNN) [10], Region-based Fully Convolutional Network (R-FCN) [11], Multibox [12], Single Shot Detector (SSD) [13] and YOLO: Real-Time Object Detection [14], are now good enough to be deployed in consumer products (e.g., Google Photos, Pinterest Visual Search) and some of them have been proven to be fast enough to run on mobile devices. This work presents a merged model of object detection and automatic caption generation systems. We have developed and trained caption generation model based on pre-trained SSD model to minimize computation time by sharing image feature extraction step.

## 2. Model

Our merged model consists of object detection and automatic caption generation joint models.

### 2.1 Object Detection

The paper *Speed/accuracy trade-offs for modern convolutional object detectors* [15] makes a detailed comparison of the existing object detection algorithms, which have good performance and also give software framework with a unique interface to work with. It starts with The R-CNN paper [16], which was the first modern convolutional network-based detector. The R-CNN method took a straightforward approach of cropping externally computed box proposals out of an input image and running a neural net classifier on these crops. This approach can be expensive because of necessity to have many crops, that lead to significant duplicated computation for overlapping crops. Fast R-CNN [10] alleviated this problem by pushing the entire image once through a feature extractor then cropping from an intermediate layer so that crops share the computation load of feature extraction.

In the Faster R-CNN detection happens in two stages. In first stage, called the *region proposal network* (RPN), images are processed by a feature extractor, and features at some selected intermediate level are used to predict class-agnostic box proposals.

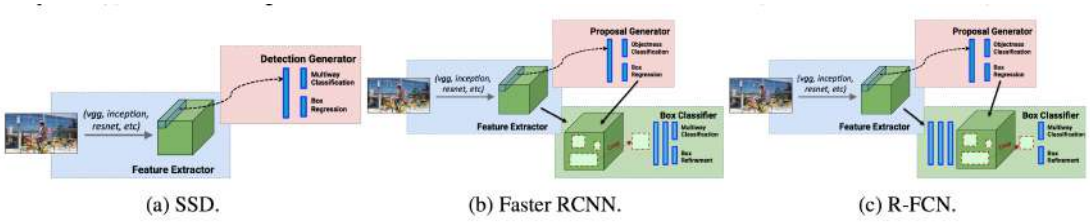
$$L(a, I; \theta) = \alpha \cdot 1[a \text{ is positive}] \cdot \ell_{loc}(\phi(b_a; a) - f_{loc}(I; a, \theta)) + \beta \cdot \ell_{cls}(y_a, f_{cls}(I; a, \theta)). \quad (1)$$

The loss function for this first stage takes the form of equation (1) [15] using a grid of anchors tiled in space, scale and aspect ratio. At the second stage, these (typically 300) box proposals are used to crop features from the same intermediate feature map which are subsequently fed to the remainder of the feature extractor in order to predict a class and

class-specific box refinement for each proposal. The loss function for this second stage box classifier takes the form of equation (1) using the proposals generated from the RPN as anchors. Notably, one does not crop proposals directly from the image and re-run crops through the feature extractor, which would be a duplicated computation. However, there is a part of computation that must be performed once per region, and, thus, the run time depends on the number of regions proposed by the RPN.

Determining classification and regression targets for each anchor requires matching anchors to groundtruth instances. Common approaches include greedy bipartite matching (e.g., based on Jaccard overlap) or many-to-one matching strategies, in which bipartiteness is not required, but matchings are discarded if Jaccard overlap between an anchor and groundtruth is too low. Paper [15] refers to these strategies as *Bipartite* or *Argmax*, respectively. Our model [15] uses *Argmax* matching along with thresholds set as suggested in the original paper [10]. After matching, there is typically a sampling procedure designed to bring the number of positive anchors and negative anchors to some desired ratio.

To encode a groundtruth box with respect to its matching anchor, the model uses the box encoding function $\phi(b_a; a) = [10 \cdot \frac{x_c}{w_a}, 10 \cdot \frac{y_c}{h_a}, 5 \cdot \log w, 5 \cdot \log h]$ (also used by [16, 10]). The scalar multipliers 10 and 5 are typically used in all of these prior works [15, 10, 16].




(a) SSD. (b) Faster RCNN. (c) R-FCN.

Fig 1. High level diagrams of the detection meta-architectures compared in [15].

Though the SSD paper [13] was published, [15] uses the term SSD to refer broadly to architectures that use a single feed-forward convolutional network to directly predict classes and anchor offsets without requiring a second stage per-proposal classification operation (Figure 2..1). Under this definition, the SSD metaarchitecture has been explored in a number of precursors to [13]. Both Multibox and the Region Proposal Network (RPN) stage of Faster R-CNN [12, 17] use this approach to predict class-agnostic box proposals.[14, 18, 19, 20] use SSD-like architectures to predict final (1 of K) class labels. And Poirson et al., [13] extended this idea to predict boxes, classes and pose. For this work we will use pretrained SSD [13] based on MobileNet [21] classifier as decribed in [15].

## 2.2 Caption Generaton

The model encodes the variable length input into the fixed length vector representation. This representation enables conversion of the input sentence into the target sentence or the input image into the target image caption. The last model was being trained to maximize $P(S|I)$ likelihood to generate the target sequence of words $S = \{S_1, S_2, \ldots\}$ for an input image $I$, where each word $S_t$ comes from a given dictionary, that describes the image adequately. Show and Tell [1] model can generate image descriptions with recurrent neural network. It

maximizes the probability of the correct caption for the given image,

$$\log p(S|I;\theta) = \sum_{t=0}^{N} \log p(S|I, S_0, \ldots, S_{t-1};\theta), \tag{2}$$

where $(S|I)$ is a training example pair. While training, we optimize the sum of the log probabilities for the whole training set using AdaGrad [22].

$p(S|I, S_0, \ldots, S_{t-1};\theta)$ probability will correspond to the $t$ step (iteration) of Recurrent Neural Network (RNN) based model. The variable number of words that are conditioned upon, up to $t-1$ is expressed by a fixed length hidden state or memory $h_t$. After every iteration for the new input, memory will be updated by using a non-linear function $f$.

$$f_{t+1} = f(h_t, x_t). \tag{3}$$

For $f$ from (3) we use a Long Short-Term Memory (LSTM) [23], which has shown state-of-the-art performance on sequence generation tasks, such as translation or image caption generation.

Long Short-Term Memory (LSTM) is an RNN cell. It helps in solving RNN training time problems like vanishing and exploding gradients [23], which is a significant problem for RNNs. LSTM is commonly used in machine translation, sequence generation and image description generation tasks. Paper [1] uses recurrent neural network with an LSTM cell to generate image caption. From a construction perspective, LSTM is a memory cell $c$ encoding knowledge at every iteration of what inputs have been seen up to this iteration. Later this knowledge is used for subsequent word generation (8, 9). Behavior of the cell is controlled by three gates: an *input gate*, an *output gate* and a *forget gate*. Each gate is a vector of real number elements ranging from 0 to 1. In particular, the *forget gate* is responsible for controlling whether to forget the cells old value, the *input gate* controls the permission for reading a new input value and finally the *output gate* controls the permission to output the new value from the cell. This is done by multiplying the given gate with the corresponding value (7, 8). The definition of the LSTM is as follows:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}), \tag{4}$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1}), \tag{5}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}), \tag{6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}), \tag{7}$$

$$m_t = o_t \odot c_t, \tag{8}$$

$$p_{t+1} = softmax(W_{pm} * m_t). \tag{9}$$

In (4)-(9) equations $i_t, o_t, f_t$ are *input*, *output* and *forget* gates, correspondingly, $c_t$ is a cell memory in step $t$ and $m_t$ is an output of the LSTM for step (iteration) $t$. $W_{ix}$, $W_{im}$, $W_{fx}$, $W_{fm}$, $W_{ox}$, $W_{om}$, $W_{cx}$, $W_{cm}$ are trainable parameters (variables) of the LSTM.$\odot$ represents the product with a gate value. Sigmoid $\sigma(\cdot)$ and $h(\cdot)$ hyperbolic tangent are nonlinearities of the LSTM. (9) will produce a probability distribution $p_{t+1}$ over all words in the dictionary, where $W_{pm}$ is a trainable parameter. We also have lookup embedding matrix $W_l \in R^{DxN_e}$, where $D$ is the number of dictionary words. Each row of the matrix represents a word embedding in image-word embedding space. Each $x_i$ (where $i \geq 0$) is the corresponding row at index $(S_i)$(equation (10)).

$$x_i = W_e^{S_i} \tag{10}$$

## 2.3 Image Embending for Caption Generation

We have tested many combinations to get feature layer for this object detector. The best one is to restore MobileNets [21]'s last layers (except softmax), which has been removed for object detector in [15]. These recovered layers (see Figure 2..3) will be trained during caption generation model training. These extra layers on object detector are increasing the caption generator accuracy, see Figure 3.

| Type / Stride | Filter Shape | Input Size |
|---|---|---|
| Conv / s2 | $3 \times 3 \times 3 \times 32$ | $224 \times 224 \times 3$ |
| Conv dw / s1 | $3 \times 3 \times 32$ dw | $112 \times 112 \times 32$ |
| Conv / s1 | $1 \times 1 \times 32 \times 64$ | $112 \times 112 \times 32$ |
| Conv dw / s2 | $3 \times 3 \times 64$ dw | $112 \times 112 \times 64$ |
| Conv / s1 | $1 \times 1 \times 64 \times 128$ | $56 \times 56 \times 64$ |
| Conv dw / s1 | $3 \times 3 \times 128$ dw | $56 \times 56 \times 128$ |
| Conv / s1 | $1 \times 1 \times 128 \times 128$ | $56 \times 56 \times 128$ |
| Conv dw / s2 | $3 \times 3 \times 128$ dw | $56 \times 56 \times 128$ |
| Conv / s1 | $1 \times 1 \times 128 \times 256$ | $28 \times 28 \times 128$ |
| Conv dw / s1 | $3 \times 3 \times 256$ dw | $28 \times 28 \times 256$ |
| Conv / s1 | $1 \times 1 \times 256 \times 256$ | $28 \times 28 \times 256$ |
| Conv dw / s2 | $3 \times 3 \times 256$ dw | $28 \times 28 \times 256$ |
| Conv / s1 | $1 \times 1 \times 256 \times 512$ | $14 \times 14 \times 256$ |
| $5\times$ Conv dw / s1 | $3 \times 3 \times 512$ dw | $14 \times 14 \times 512$ |
| Conv / s1 | $1 \times 1 \times 512 \times 512$ | $14 \times 14 \times 512$ |
| Conv dw / s2 | $3 \times 3 \times 512$ dw | $14 \times 14 \times 512$ |
| Conv / s1 | $1 \times 1 \times 512 \times 1024$ | $7 \times 7 \times 512$ |
| Conv dw / s2 | $3 \times 3 \times 1024$ dw | $7 \times 7 \times 1024$ |
| Conv / s1 | $1 \times 1 \times 1024 \times 1024$ | $7 \times 7 \times 1024$ |
| Avg Pool / s1 | Pool $7 \times 7$ | $7 \times 7 \times 1024$ |

**Recovered layers**

Fig 2. The image extractor of SSD object detector with recovered layers from MobileNet.

We will select the last layer ($Mnetfeatures$) from the construced layers and append *fully connected* neural layer with $N_e$ neurons, which will convert 4096-dimensional vector into $N_e$ dimensional vector. $N_e$ is an image-words embedding vectors dimensionality [24]. The output vector $x_1$ of a fully connected layer will be the first feed vector for RNN,

$$x_{-1} = Mnetfeatures * W_i + b_i \tag{11}$$

where $W_i \in R^{2048xN_e}$ and $b_i \in R^{N_e}$ are trainable parameters for image embedding.

## 3. Training

The LSTM model is trained to predict the probability for the next word of an image caption after it has observed all the previous words in the captions and image features [25]. For easier training LSTM is represented in unrolled form, which is a copy of the LSTM memory for the image and each word of the sentence. Also all LSTMs share the same parameters.

Thus, $x_{-1}$ is the first input for the first LSTM. Initial state of the LSTM is $c_{-1}$ zero-filled memory. For the next LSTMs, inputs correspond to the word embedded vectors. Also, all

recurrent connections are converted into feed-forward connections. Loss function will be sum of the negative log likelihood of the correct word at each step:

$$L(I, S) = -\sum_{t=1}^{N} \log p(S_t). \tag{12}$$

For training, we have used AdaGrad instead of multi-batch stochastic gradient descent. We have trained on *Microsoft Common Objects in Context* (MSCOCO) [26] image dataset and have less accuracy then [1], see Table 3.. To avoid overfitting we have used Perplexity (13), also we have applied different regularization techniques as dropout, weight decay and gradient clipping.

$$PP(S) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(S_i|S_1 \dots S_{i-1})}}. \tag{13}$$

Inference has been made via using Beam Search, which gives us variants for the best scored sentence after many predictions in Figure 3..

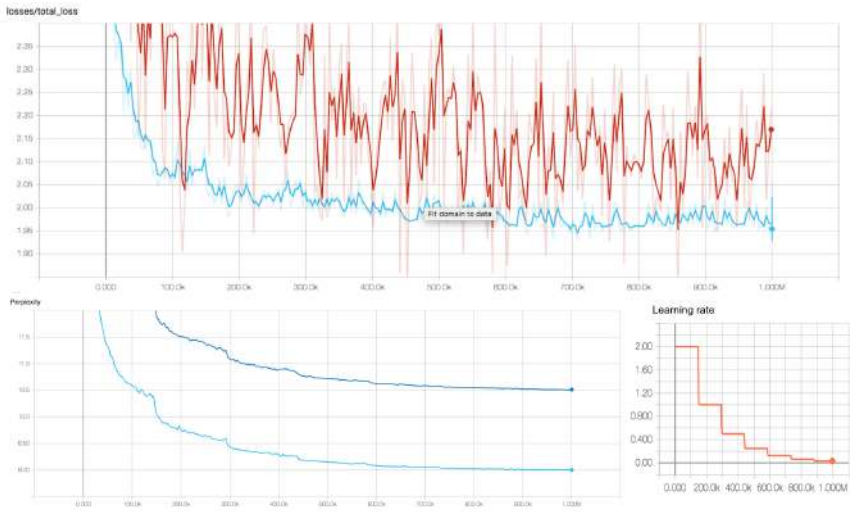



Fig 3. The training process comparation of SSD-MobileNet based caption generator with SSD-MobileNet based caption generator with recovered layers. The first graph is train loss comparation (red - *SSD-MobileNet*, blue - SSD-MobileNet + recovered), the second evaluation perplexity (dark blue - *SSD-MobileNet*, blue - *SSD-MobileNet + recovered*), the third is the learning rate graph.


Fig 4. Image caption generation and object detection via a single model.

Table 1: Comparison of SSD-MobileNet based caption generator (our model) with **Vinyals [1]**, **Human [26]**, **Random** - randomly picked words from the dictionary, **Shuffle** - correct caption with randomly shuffled words.

| Metric | BLEU 1 | BLEU 2 | BLEU 3 | BLEU 4 | METEOR | ROUGE-L | CIDEr-D |
|---|---|---|---|---|---|---|---|
| MNet based | 0.491 | 0.296 | 0.181 | 0.115 | 0.151 | 0.379 | 0.256 |
| Vinyals [1] | N/A | N/A | N/A | 0.277 | 0.237 | N/A | 0.855 |
| Human [26] | 0.663 | 0.469 | 0.321 | 0.217 | 0.252 | 0.484 | 0.854 |
| Random | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Shuffle | 1.000 | 0.404 | 0.139 | 0.050 | 0.415 | 0.469 | 1.102 |

## 4. Conclusion

We have built an image caption generation model on top of the object detection. That hepled us to reduce the computation time by more than 30% by sharing the image feature extraction step within the object detector and the caption generator. As we have not changed anything in the object detector it has the same accuracy as in [15]. But we have not achieved [1] accuracy in the caption generation, see Table 1. One problem that we have noted is the object detector word dictionary size. Our chosen object detector can detect objects from 80 categories. Moreover, our caption generator vocabulary consists of 11520 words. This causes the caption generator's inability to differ, for example, *men* from *women* because the object

detector has only the concept of *person*, see Figure 3.. One method to fix this issue is to train the system as a single model and that would require huge computational power.

## References

[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[2] A. Poghosyan and H. Sarukhanyan, "Short-term memory with read-only unit in neural image caption generator," *11-th International Conference Computer Science and Information Technologies, Revised Selected Papers, IEEE Xplore, 10.1109/CSITechnol.2017.8312163, Electronic ISBN: 978-1-5386-2830-0, Print on Demand(PoD) ISBN: 978-1-5386-2831-7*, pp. 162– 167, 2017.

[3] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137, 2015.

[4] A. Poghosyan and H. Sarukhanyan, "Rnn with additional constant memory for image caption," *International Academy Journal Web of Scholar*, vol. 1, no. 4(13), pp. 3–7, Jul. 2017.

[5] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *European conference on computer vision*. Springer, pp. 15–29, 2010.

[6] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Babytalk: Understanding and generating simple image descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2891–2903, 2013.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[8] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[9] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[11] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, pp. 379–387, 2016.

[12] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, high-quality object detection," *arXiv preprint arXiv:1412.1441*, 2014.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, pp. 21–37, 2016.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[15] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," *arXiv preprint arXiv:1611.10012*, 2016.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[17] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in neural information processing systems*, pp. 2553–2561, 2013.

[18] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[20] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[22] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[25] A. Poghosyan and H. Sarukhanyan, "Image visual similarity based on high level features of convolutional neural networks," *Mathematical Problems of Computer Science*, vol. 45, pp. 138–142, 2016.

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

**Submitted 10.06.2018, accepted 12.11.2018.**

# Պատկերների վերնագրերի գեներացում՝ օբյեկտներ հայտնաբերող մոդելի հիման վրա

Ա. Պողոսյան և Հ. Սարուխանյան

**Ամփոփում**

Մեր օրերում պատկերի սեմանտիկ ինֆորմացիայի ավտոմատացված հավաքագրումը ամենևին էլ հեշտ առաջադրանք չէ: Գոյություն ունեն աշխատություններ, որոնք ունակ են վերնագրել պատկերները, մինչդեռ մյուսները հնարավորություն ունեն տալ պատկերում առկա օբյեկտների անվանումներն իրենց կոորդինատներով: Այս աշխատության մեջ վերը նշված խնդրի լուծումը ներկայացված է ընդամենը մեկ մոդելի միջոցով: Մենք վերնագրերի գեներացման մոդելը նախագծել ենք օբյեկտներ հայտնաբերող մոդելի հիման վրա: Նաև ավելացրել ենք հավելյալ շերտեր օբյեկտների հայտնաբերման մոդելում՝ վերնագրերի գեներացման ճշտությունը բարձրացնելու նպատակով: Վերջնական մոդելի միջոցով կարող ենք ստանալ պատկերի վերնագիրը և պատկերում առկա օբյեկտների կոորդինատներն ու անվանումները:

# Генерация заголовков изображений на основе модели обнаружения объектов

А. Погосян и А. Саруханян

**Аннотация**

Автоматическое извлечение семантической информации из изображения является сложной задачей. Есть работы, которые могут извлечь заголовок изображения или названия объекта и их координаты. В данной работе вышеуказанная проблема представлена только через одну модель. Мы разработали модель генерации названий на основе модели обнаружения объектов. Мы также добавили дополнительные слои в модель обнаружения объектов, чтобы увеличить точность генерации названий. Окончательная модель позволяет получить заголовок изображения и координаты объектов и их имена.