Identical Objects Recognition Based On Image and Textual Description

Andrei Aslanov[1][0000-0001-8052-6210] and Natalia Loukachevitch[1,2][0000-0002-1883-4121]

[1] Bauman Moscow State Technical University, Moscow, Russian Federation
andrew.aslanov@gmail.com
[2] Moscow Lomonosov State University, Moscow, Russian Federation
louk_nat@mail.ru

Abstract. Today image retrieval methods are rapidly growing but still, image as a type of information is not sufficient for some specific cases, especially when it comes to search for information in social networks. For this reason, we introduce a combined method including both text and image representations for identical object search. The task is solved with practical relevance to lost pets finding in social networks. The suggested method shows approximately 14.66% better quality result in comparison with the same method in which image retrieval technique reproduced only.

Keywords: dataset, object detection, neural network, deep learning, siamese network, joint representation.

1 Introduction

When solving different problems, a human uses different sources of information, including audio, video or text information. Multimodal task settings are intended to study approaches to exploit different modalities to improve the results of task solution. Cross-media (or multi-media) retrieval aims to search for information when queries and retrieval results are of different media types: text, image, or video [1–3].

Multimodal machine translation approaches consider techniques of translation with the use of available pictures [4, 5]. The main problem of all multimodal tasks is so-called “media gap”, which means that representations of different media types lie in different feature spaces.

Multimodal machine learning approaches aim to build models that can process and relate information from multiple modalities [6]. Authors of [7] list the following problems of multimodal machine learning: representation of multimodal data, translation (mapping) data from one modality to another, alignment between subelements of two or more modalities, fusion of information from different modalities, and co-learning.

In the current paper, this issue is studied for pets` search over the social network posts, where each post may include text (or image) only or image with its textual description. It can be useful in case a user could upload the photo or the textual portrait of his lost pet to a system and then get a response that contains the most similar recent posts aggregated from the social network.

2 Related work

In this section, we briefly discuss the related methods where text and image are both processed.

In [8] the authors introduced event photography method for automatic photos annotation. It's suggested to combine the image with its description and perform neural network in order to check how strongly text pertains to the corresponded photo. In [9] the authors proposed to represent image and text with 2 kinds of scene graphs: visual scene graph and textual scene graph. These parts jointly characterize objects and relationships between them. The image-text retrieval task is then naturally formulated as cross-modal scene graph matching.

In [11] the encoder is a convolutional neural network, and the features of the last fully connected layer or convolutional layer are extracted as features of the image. The decoder is a recurrent neural network, which is mainly used for image description generation. In [12] RNN was replaced with LSTM for vanishing gradient solving problem.

In [13] firstly attention mechanism was proposed to be applied. It allows the neural network to have the ability to focus on specific inputs or features. Attention mechanism is the following two aspects: the decision needs to pay attention to which part of the input; the allocation of limited information processing resources to the important part.

There are various techniques to calculate the attention distribution and "value" is used to generate the selected information. Experiments have proved that the attention mechanism is applied in abstract generation [14], visual captioning [15], and other issues.

3 Methods

We define *post* or *document* as an entity containing image or text or both image and text. We need to find images or texts where an identical object is mentioned. Formally, if we have training samples subset of texts or images, the task is to minimize the distance between documents if they contain the same object.

We propose two methods of identical objects search. The first one is represented in Section 4. For that case we consider a method based on image retrieval only. It includes image data collection (4.1), further inappropriate images filtration (4.2) and augmentation (4.3), object detection (4.4) and similarity evaluation (4.5). The second method is described in Section 5. It uses image-text joint representations. It contains collection of posts (image and corresponded description) (5.1), posts data transformation to joint feature vectors and vectors similarity evaluation (5.2). All steps for each of methods are shown in Fig. 1.

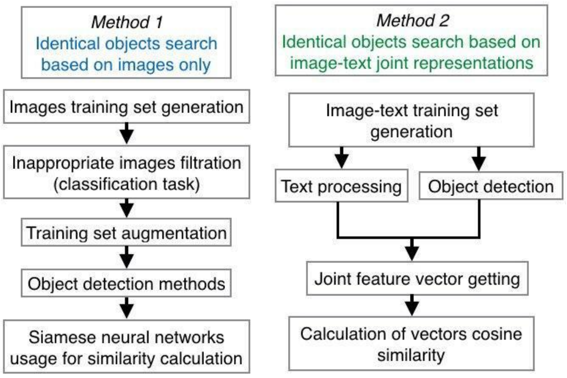

Method 1
Identical objects search based on images only
Images training set generation
Inappropriate images filtration (classification task)
Training set augmentation
Object detection methods
Siamese neural networks usage for similarity calculation
Method 2
Identical objects search based on image-text joint representations
Image-text training set generation
Text processing
Object detection
Joint feature vector getting
Calculation of vectors cosine similarity

Fig. 1. Researched steps

Also, results are valid for the configuration below:

1. *Processor*: Intel Core i9-9820X, 3.3 Ghz;
2. *Graphics*: nVidia 1080TI, 11 Gb GPU memory;
3. *Memory*: 64 Gb, 4.2 Ghz;
4. *OS*: Ubuntu 18.04 x64.

4 Method 1: Identical objects search based on images only

4.1 Data collection

Our goal is to build a dataset containing folders with images that show the same animal in each of them. For this purpose, we use crawling which is an automated data gathering method. In order to collect data, we should get all available images from the social networks and then filter out improper cases. To do that, we find Instagram and Flickr accounts that aggregate images of the desired animal (cat or dog), where for each post there is a text description with a link to the profile of the user who uploaded this image to a social network. The user profile specified in this post usually contains the desired identical animal images as expected (Fig. 2). The first 100 images are saved from every such a profile.

By means of offered algorithm, we extract 9502 cat accounts and 8070 dog accounts.

4.2 Inappropriate images filtration

For every account received from the previous step, it is necessary to filter the images that do not contain the desired object (cat or dog). To select a neural network architecture, which is most efficient for image filtering, we gather a test dataset, which is manually labeled.

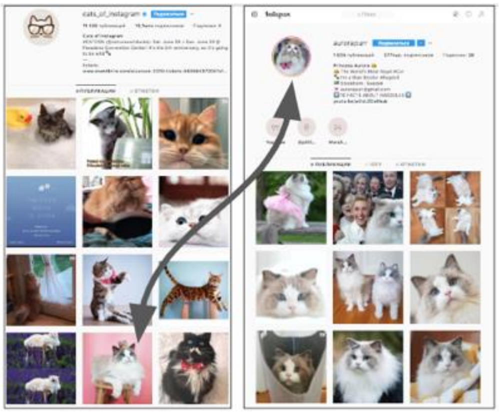


Fig. 2. Each post from aggregator account contains a link to the user profile with the desired identical images

The dataset includes:

- 1000 random images from the different Instagram accounts;
- 118 selected images representing complicated cases on which animal is situated but at the same time the image does not have any clear outlines or ambiguity of presence takes place (Fig. 3).

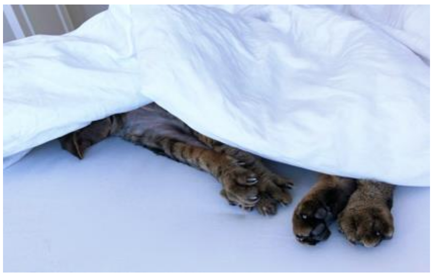


Fig. 3. The example of complicated image

Such images may contain disguise, difficult view (angle), merge with background, blur, close-up, graphical effects, and so on.

To meet the classification challenges, pretrained neural network models are used. It allows us not to spend time and computational resources as well as answer the question if there is a required animal on the image. After that, we are able to make a decision if it is necessary to filter out a specific image, with sufficient precision.

In this paper, we compare the following CNN architectures each of which has the unique properties for the classification tasks:

1. **VGG16/VGG19** [22]: unification of 16/19 convolution layers to a sequence of convolutions; reduction of filter size to *3*×*3*; rejection of local response normalization layer using;
2. **ResNet50** [23]: a high precision fixation on a current convolution layer and residual connections idea introduction;
3. **Inception v3** [24]: Inception module, Root Mean Square propagation (RMSprop) and batch normalization introduction; reduction of filter size to *3*×*3*; filter decomposition to a pair of *1*×*N* and *N*×*1* filters;
4. **Xception** [25]: spatial and channel feature separation, replacing Inception modules with depthwise convolutions.
5. **Inception ResNet v2** [26]: dimensionality reduction paradigm by *1*×*1* convolution using; adding shortcut connections; increase of hyperparameters number.
6. **NasNet Large** [27]: blocks or cells are searched by reinforcement learning; the number of initial convolutional filters are free parameters used for scaling. Only cells returning a feature map of the same dimension (or factored to 2) are searched by the recurrent neural network.

Table 1 and Table 2 show quality evaluation of binary classification (if the image contains a supposed class or not) among architectures. In terms of results, we can conclude that the NasNet Large architecture is most efficient for image filtration.

Table 1. Deep neural networks architecture quality evaluation when filtering random images (1000 pieces)

Architecture name	F1-cats	F1-dogs
VGG16	0.726	0.907
VGG19	0.720	0.919
ResNet50	0.694	0.907
Inception v3	0.873	0.929
Xception	0.913	0.924
Inception Resnet v2	0.898	**0.938**
NasNet Large	**0.916**	0.930

Table 2. Deep neural networks architecture quality evaluation when filtering complicated images (118 pieces)

Architecture name	F1-cats	F1-dogs
VGG16	0.569	0.880
VGG19	0.603	0.849
ResNet50	0.619	0.816
Inception v3	0.695	0.904
Xception	0.723	0.927
Inception Resnet v2	0.730	0.921
NasNet Large	**0.763**	**0.943**

The input of a neural network is an image with a fixed size. The output is a binary answer is there an object (cat or dog) on the image or not.

Table 3. Number of accounts before and after crawling and filtration steps

	Before filtration	After filtration
Cats	9502	6572
Dogs	8070	3552

Table 3 shows how many accounts remained after the crawling and filtration steps. Each account holds an average 42 photos of the identical animal.

4.3 Data augmentation

In case an account contains lower than 16 images, then the augmentation technique is applied. Based on available photos, some geometric operations (like Affine Transformation, brightness level changing, rotation, reflection and others) are performed. That is necessary in order to increase number of photos per account to ensure convergence of the subsequent algorithm.

4.4 Object detection

We use object detection to focus on an object instead of image background. As well as for the classification task, we use pretrained neural networks for this topic.

The following architectures were compared:

1. **Yolo v3** [20]: batch normalization and higher resolution classifier usage; multi-scale training and feature pyramid network introduction; darknet-53 feature extractor.
2. **Faster-RCNN** [18]: at the conceptual level, this architecture type is composed of 3 neural networks:

- *Feature Network* – pretrained image classification network without a few last layers;
- *Region Proposal Network* – purpose is to generate a number of bounding boxes that has a high probability of containing any object;
- *Detection Network* – takes input from both the Feature Network and RPN, and generates the final class and bounding box.

1. **Grid-RCNN** [21]: uses multi-point supervision formulation to encode more clues in order to reduce the impact of inaccurate prediction of specific points.
2. **RetinaNet** [17]: single, unified network composed of a backbone network (for computing a conv feature map) and two task-specific subnetworks. The first subnet performs classification on the backbones output; the second subnet performs convolution bounding box regression.
3. **CenterNet** [19]: uses centeredness information to perceive the visual patterns within each proposed region.

The obtained results of object detection are specified in Table 4 and Table 5. We use subset of 118 dog photos and 155 cat photos as a test set.

Table 4. Object detection results for subset containing 118 dog photos

Suggested method for object detection	mAP	*t*, ms	Framework
Faster-RCNN_FPN + ResNet50_1x	0.9814	79.49	pytorch
Grid-RCNN + GNHead_x101_32x4d_FPN_2x	0.9876	122.57	pytorch
Yolo3 + DarkNet53	0.5975	67.53	mxnet
RetinaNet_crop640_r50 + NasFPN_50e	0.9642	70.15	pytorch
RetinaNet_FreeAnchor_r101_FPN_1x	**0.9914**	102.54	pytorch
CenterNet + ResNet18_v1b	0.5078	**44.52**	mxnet

Table 5. Object detection results for subset containing 155 cat photos

Suggested method for object detection	mAP	*t*, ms	Framework
Faster-RCNN_FPN + ResNet50_1x	0.9711	81.02	pytorch
Grid-RCNN + GNHead_x101_32x4d_FPN_2x	0.9711	128.80	pytorch
Yolo3 + DarkNet53	0.8192	58.76	mxnet
RetinaNet_crop640_r50 + NasFPN_50e	**0.9886**	63.15	pytorch
RetinaNet_FreeAnchor_r101_FPN_1x	0.9872	112.96	pytorch
CenterNet + ResNet18_v1b	0.7189	**33.71**	mxnet

We have found the most qualitative results of the anchor-based RetinaNet neural network. We use RetinaNet + FreeAnchor neural network option for the dog and cat detection.

4.5 Similarity calculation with siamese neural network

Siamese neural networks are used to calculate the similarity between texts or images. We take a modified FaceNet version called EmbeddingNet[1] with triplet loss function [10] for our task. Our full dataset contains 6572 cat accounts and 3552 dog accounts. For the similarity calculation, we make a subset (training set) containing 1200 dog and cat accounts shuffled. It is done for computation reduction. Also, we make a subset (test set) containing 582 shuffled accounts. In this section, we provide hyperparameters used while training.

1. Initial image size: 512×512×3;
2. Margin: 0.4;
3. Loss function: triplet loss;
4. Learning rate: 0.0001;
5. Optimizer: radam;
6. Epochs: 500;
7. Maximum number of neighbors: 100.

If result has not changed more than 10 epochs in a row, then we conclude training is completed.

We use Instagram data for training and VK[2] posts for test. VK test set contains 582 photos. Training has taken 79 epochs. As a result, we have top-1 accuracy equals 54.10% and top-5 accuracy equals 71.81% for test data. Every account takes 56.62 ms. for processing. To compute similarity, we use nVidia 1080TI graphics processor configuration.

[1] https://github.com/RocketFlash/EmbeddingNet

[2] https://vk.com

5 Method 2: Identical objects search based on image-text joint representations

5.1 Data collection

To make identical objects search, much like the previous method we make automatic gathering firstly. This time, we collect specific VK social network posts and filter out inappropriate ones after that.

In general, the algorithm is comprised of the following stages:

1. *Data crawling from social network*. We make some groups crawling from VK. The first half of these groups (6 pieces) has different general-purpose topics and the last half (6 pieces) is relevant for animal shelter and volunteer particularities. We extract 39995 posts in total.
2. *Texts preprocessing*. We split and lemmatize all the words from the previous stage; also, we remove all punctuation marks. Words within hashtags commonly don't provide any added value therefore it should be removed primarily. After that, we remove numbers, proper names, English words (we work with the texts in Russian only), Latin symbols, and special symbols. We make basic stop-words processing (with default parameters) with Python nltk[3] framework.
3. *Tokens sorting*. We sort all the frequent words (unigrams) in descending order and then take the first 10%. Also, we consciously increase the frequencies number of some words in order to raise the priority of the locations' references.

As a result, we have k the most frequent meaningful words ($k = 370$ for our case). Further each unigram frequency is normalized in relation to the total number of frequencies with the following expression:

$$p_w = \frac{100 f_w}{\sum_{w=1}^{w=k} f_w}. \quad (1)$$

The obtained in (1) pw values are the unigrams tokens. The more frequently word is mentioned, the higher priority it has.

In (1): fw – frequency of the word w; pw – priority of the word w.

If we summarize all the priorities for every post, then we are able to calculate text c priority.

$$p_c = \frac{1}{l} \sum_{w=1}^{l} p_w. \quad (2)$$

In (2): *l* – post words number; p_c – priority of the text *c*.

If there is no any word in most frequent unigrams list, then the final priority is assigned to 0.

Text priority values are added to the array where the median is specified more precisely with every new priority. Median defined as a threshold: if priority is higher than

[3] https://nltk.org

the median value, then the text is referred to our specific (cats and dogs mentions) and the corresponded target is set to 1 for that reason. Otherwise, the target is set to 0.

Median value converges with a growing number of posts as shown in Fig. 4.

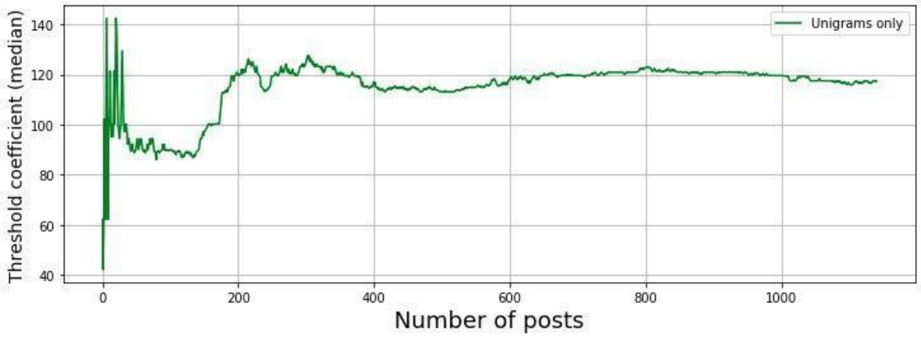


Fig. 4. Median value convergence

There are about 350-400 iterations for sufficient algorithm convergence. The final threshold after 1200 iterations equals 118,02.

We can calculate the current post error (residual) e_c as difference between the current text priority p_{ci} and median threshold value. Applying linear interpolation, we get confidence that current post is assigned to the correct class:

$$e_c = f(median(p_c) - p_{ci}). \tag{3}$$

In (3): $\mathbf{p_c}$ – vector containing all the priorities before current priority value; p_{ci} – current priority value; f – linear interpolation function.

We use scipy 1.5.0 of Python programming language for linear interpolation function computation.

Thus, in accordance with detection results, each image in the post is corresponded to mean average precision metric quality of the proper class (dog or cat).

As a result, we have the following full probability equation representing if post contains at least one mention of dog or cat:

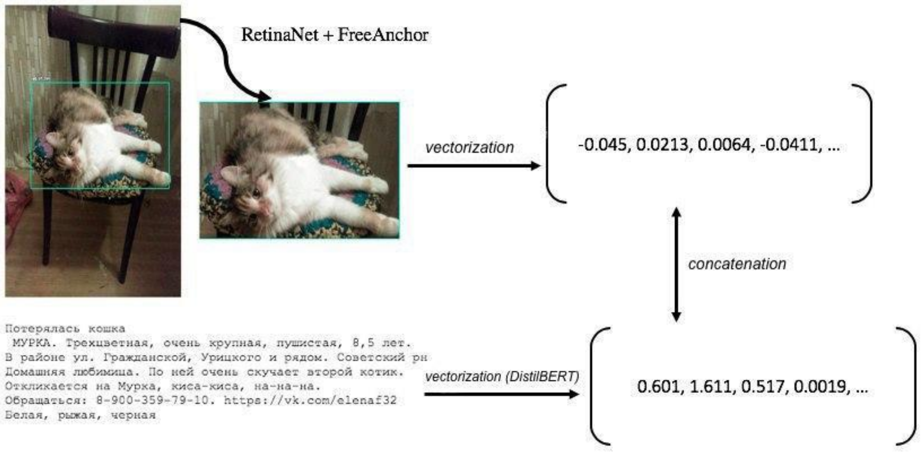
$$P_{full} = \frac{1}{4}P_{text} + \frac{3}{4}P_{image}. \tag{4}$$

In (4) we work on the image and text data in the post. For verification, we take 100 posts for algorithm testing and receive 94 posts of 100 are relevant.

As a result, we have a method capable to collect posts of required topics automatically. We collect topics with cats and dogs and achieve sufficient data precision for our case. We make a 7292 posts dataset totally.

5.2 Joint feature vector getting and similarity evaluation

We compare post vectors for object identity calculation. Both image and text were transformed into vector representations. We use pretrained DistilBERT [16] for text embedding getting. Also, we use img2vec[4] framework with ResNet50 backbone to obtain image embedding. After that, we make concatenation of these vectors to get a joint representation of the entire post. The schematic view of this algorithm is shown in Fig. 5.



```
Потерялась кошка
 МУРКА. Трехцветная, очень крупная, пушистая, 8,5 лет.
В районе ул. Гражданской, Урицкого и рядом. Советский рн
Домашняя любимица. По ней очень скучает второй котик.
Откликается на Мурка, киса-киса, на-на-на.
Обращаться: 8-900-359-79-10. https://vk.com/elenaf32
Белая, рыжая, черная
```

Fig. 5. Image and text joint vector representation creating

We compare received vectors with each other using cosine distance. We take subset including 582 test posts for the method verification as the previous way involves. The difference lies in additional text descriptions and social network choice. Also, objects identity on images inside of every post is not initially guaranteed. As a result, we have top-1 accuracy equals 69,09% and top-5 accuracy equals 86,15%. Every post takes 61.05 ms for processing.

Table 6. Metric results for suggested object identity methods

Suggested method	@1 acc	@5 acc	Time (ms)
Method 1	0.5410	0.7181	**56.62**
Method 2	**0.6909**	**0.8615**	61.05

The results of both methods are shown in Table 6.

Ultimately, in this article, we present new methods of training set creation. Further identical object recognition task solution is based on a search of similar objects on the images or image-text joint representations. Such representations consist of numerical vectors including both image and text features. We compare these representations to find out how close the similar objects are located in feature space after applied methods.

All the calculations are produced with Python 3.6.8 programming language and libraries: keras 2.3, tensorflow-gpu 1.15.0, mxnet+cu 1.5.1, CUDA 9.0, numpy 1.18.0, scipy 1.5.0, matplotlib 3.1.3, mmcv 0.5.4, torch 1.4.0.

[4] https://github.com/christiansafka/img2vec

As a result, we get text as an additional property has a significant positive impact. We obtain quality metrics increased approximately in 14.66%.

6 Conclusion

In this paper, we solve the problem of how to recognize identical objects in the social network. This issue is studied for lost pets' search. Two methods are suggested. In the first case, we collect a dataset of identical lost pet photos and then try to find the most similar objects of required class using a siamese neural network. The second case assumes we use combined technique including both text and image representations. As they are received, we try to transform them into the joint feature vector for further comparison with the cosine distance metric.

As the results demonstrate, approach including various data types is more preferable, it gives us more high result. We obtain 14.66% better metric quality in comparison with a method in which the image retrieval method presented only. Also, we provide Github repository[5] source code for reproducibility.

Potential future research will focus on designing a retrieval system that will employ image captioning additionally.

References

1. Peng, Y., Huang, X., Zhao, Y.: An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges. Transactions on circuits and systems for video technology, 2372–2385. IEEE (2017).
2. Jiang, B., Yang, J., Lv, Z., Tian, K., Meng, Q., Yan, Y.: Internet cross-media retrieval based on deep learning. Journal of Visual Communication and Image Representation, 356–366 (2017).
3. Wei, Y. et al: Modality-dependent cross-media retrieval. ACM Transactions on Intelligent Systems and Technology, 1–13 (2016).
4. Specia, L., Frank, S., Sima'an, K., Elliott, D.: A shared task on multimodal machine translation and crosslingual image description. In: Proceedings of the First Conference on Machine Translation, vol. 2: Shared Task Papers, 543–553 (2016).
5. Huang, P. Y., Liu, F., Shiang, S. R., Oh, J., Dyer, C.: Attention-based multimodal neural machine translation. In: Proceedings of the First Conference on Machine Translation, vol. 2: Shared Task Papers, 639–645 (2016).
6. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.: Multimodal deep learning. In: International Conference on Machine Learning (2011).
7. Baltrušaitis, T., Ahuja, C., Morency, L. P.: Multimodal machine learning: A survey and taxonomy. IEEE transactions on pattern analysis and machine intelligence, 423–443. IEEE (2018).
8. Postnikov, M., Dobrov, B.: News Stories Representation Using Event Photos. In: XIX International Conference on Data Analytics and Management in Data Intensive Domains, 359–366 (2017).

[5] https://github.com/andreqwert/lost_pets_search

9. Wang, S., Wang, R., Yao, Z., Shan, S., Chen, X.: Cross-modal Scene Graph Matching for Relationship-aware Image-Text Retrieval. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision. IEEE, USA (2020).
10. Yagfarov, R., Ostankovich, V., Akhmetzyanov, A.: Traffic Sign Classification Using Embedding Learning Approach for Self-driving Cars. In: International Conference of Human Interaction and Emerging Technologies, 180–184 (2020).
11. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3156–3164. IEEE, USA (2014).
12. Karpathy, A., Johnson, J., Li, F.-F.: Visualizing and understanding recurrent networks (2015).
13. Mnih, V., Heess, N., Graves, A.: Recurrent models of visual attention. Advances in Neural Information Processing Systems, 3, 2204–2212 (2014).
14. Allamanis, M., Peng, H., Sutton, C.: A convolutional attention network for extreme summarization of source code. In: Proceedings of the Thirty-Third International Conference on Machine Learning. New York, USA (2016).
15. Song, J., Guo, Y., Gao, L., Li, X., Hanjalic, A., Shen, H.: From deterministic to generative: multi-modal stochastic RNNS for video captioning. IEEE Transaction on Neural Networks and Learning System, 30(10), 3047–3058 (2018).
16. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (2019).
17. Lin, T., Goyal, P., Girshick, R, He, K., Doll`ar, P.: Focal loss for dense object detection. In: International Conference of Computer Vision, 2980–2988 (2017).
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1137–1149 (2017).
19. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE International Conference on Computer Vision, 6569–6578 (2019).
20. Redmon, J., Farhadi ,A.: Yolo v3: An incremental improvement. arXiv:1804.02767 (2018).
21. Lu, X., Li, B., Yue, Y., Li, Q., Yan, J.: Grid-rcnn. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7363–7372 (2019).
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks. Theory of Parsing, Translation and Compiling, vol. 1. Prentice-Hall, Englewood Cliffs, NJ (2014).
23. He, K., Zhang, X., Ren, S., Agarwal, P., Shroff, G.: Deep Reidual Learning for Image Recognition. arXiv:1512.03385 (2015).
24. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. In: The IEEE Conference on Computer Vision and Pattern Recognition, 2818–2826 (2016).
25. Chollet, F.: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1251–1258 (2017).
26. Szegedy, C., Ioffe, S., Vanhoucke, V., Alem, A.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 4278–4284 (2017).
27. Zoph, B., Vasudevan, V., Shlens, J., Quoc, V. le: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 8697–8710 (2018).