Classification of Computer Generated and Natural Images based on Efficient Deep Convolutional Recurrent Attention Model

Diangarti Bhalang Tariang[a], Prithviraj Sengupta[b], Aniket Roy[c], Rajat Subhra Chakraborty[a], Ruchira Naskar[d]
[a]Indian Institute of Technology Kharagpur, [b]University of Illinois at Chicago
[c]Indian Statistical Institute, Kolkata, [d]National Instititute of Technology Rourkela
diazz.tariang@iitkgp.ac.in

Abstract

Most state-of-the-art techniques of distinguishing natural images and computer generated images based on hand-crafted feature and Convolutional Neural Network require processing of the entire input image pixels uniformly. As a result, such techniques usually require extensive computation time and memory, that scale linearly with the size of the input image in terms of number of pixels. In this paper, we deploy an efficient Deep Convolutional Recurrent Attention model with relatively less number of parameters, to distinguish between natural and computer generated images. The proposed model uses a glimpse network to locally process a sequence of selected image regions; hence, the number of parameters and computation time can be controlled effectively. We also adopt a local-to-global strategy by training image patches and classifying full-sized images using the simple majority voting rule. The proposed approach achieves superior classification accuracy compared to recently proposed approaches based on deep learning.

1. Introduction

In recent years, through the development of advanced 3D rendering software, it has become extremely convenient to create computer-generated (CG) images and scenes, with high levels of visual realism. This has wide applications in entertainment, gaming, architecture, publishing, advertising and marketing industries. 3D-rendering software tools are often utilized to combine CG images with natural images (images captured by a digital camera) to create life-like artificial scenes. However, such technical sophistication, although extremely useful in film and media industries, often pose practical threats. With the current levels of virtual realism achieved with CG images, it is extremely challenging to visually differentiate between a CG and a natural image (NI). This is many times exploited by intelligent adversaries to mislead forensic investigations. Differentiating between CG images and natural images has proven to be one of the biggest present-day challenges for the image forensics researcher community.

Several methods have been proposed in recent years to classify natural and CG images. The methods that rely on design and extraction of hand-crafted features from an image under test (or its pre-processed version) for classification, broadly employ features of the following types: (a) *statistical* [1–4]; (b) *textural* [5, 6], and (c) *physical* [7, 8]. More recently, deep learning has been adopted for CG identification [9–13].

1.1. Motivation

Convolutional neural network based architectures have recently achieved substantial success in tasks of visual recognition and classification [14]. However, such successful architectures come at the cost of high computational overhead, both while training and testing. Computational overhead grows linearly with the image resolution, as convolving filter maps using a sliding window mechanism are applied on the entire image. Most of these architectures impose constraints on the input image, by downsampling (resizing or cropping) it so as to reduce computational overhead [14]. This drawback has motivated recent researches in recurrent attention models [15–19] that take inspiration from the way human beings perform visual recognition tasks, specifically focusing on relevant areas as they progress through sequences. The attention mechanism allows the model to selectively focus its attention locally on some regions of an image, rather than sliding towards each region. Henceforth, the number of network parameters and computational power can be controlled independent of the image resolution.

Inspired by the model proposed by Ba et al. [16], in this paper we use a deep *Stochastic Convolutional Recurrent Attention Network* to distinguish between natural images and computer generated images. The network is trained to adaptively integrate features extracted from the sequence of selected patches of the input image, called *glimpses*, via a

Glimpse Network [15], and then feed it to the *Recurrent Neural Network* (RNN) for classification. Instead of processing an entire image, at each step, the model learns and selects the next location to attend to, based on past information.

The rest of this paper is organized as follows. Related works are briefly introduced in Section 2. The model is detailed in Section 3, followed by the experimental results in Section 4. Finally, conclusions are drawn in Section 5.

2. Related Work

Since natural images are generated by digital cameras, it is expected that the distinct physical image generation pipelines of cameras must introduce unique intrinsic characteristics into NI, which are absent in CG. Based on this assumption, some methods to distinguish between NI and CG have been reported. Ng et al. [1] studied three types of natural image statistics derived from the power spectrum, wavelet transform and local patch of images to distinguish CG from NI. Wang et al. [2] present a customized statistical model based on the homomorphic filter and use *Support Vector Machine* (SVM) as a classifier to distinguish CGs from NIs. In [3], the authors have used hand-crafted wavelet based features to distinguish CG images from natural images. Wu et al. [4] took several highest histogram bins of the difference images as features to carry out classification. Li et al. [5] present a multiresolution approach to distinguishing CGs from NIs based on local binary patterns (LBPs) features and an SVM classifier. Dirik et al. [8] developed two features that capture demosaicing features in camera image processing pipeline, and another feature to measure the sensor noise power changes all across the image for the classification. Yao et al. [12] proposed a method based on sensor pattern noise (SPN) and deep learning to distinguish CGs from NIs. Rahmouni et al. [11] proposed a custom pooling layer in CNN to optimize the features extracted in the best performing algorithms, then local estimates of class probabilities are computed and aggregated to predict the label of the whole picture. Yu et al. [9] and Quan et al. [10] investigated other kinds of shallow CNN architectures and achieved promising detection performance. However, these CNN based techniques involve processing of the entire image uniformly using classical sliding window paradigm, which incurs high computational cost to apply convolving filter on the entire image.

3. Deep Convolutional Recurrent Attention Model

3.1. The Model

Processing an image I with an attention based model is a sequential process with N steps. At each step n, the model receives a location L_n along with a glimpse observation I_n taken at that location. The model uses this observation to update its internal state and predicts the next location L_{n+1} to process in the next time-step. Usually the number of pixels in the glimpse I_n is much smaller than the number of pixels in the original image I, making processing of a single glimpse, independent of the entire image size. Inspired by the model proposed by Ba et al. [16], our network architecture can be decomposed into several subcomponents including a *CNN-based Glimpse Network*, a *Recurrent Network*, a *Classification Network* and an *Emission Network* as illustrated in Fig. 1.

CNN based Glimpse Network:

The glimpse network is a non-linear function that receives the current input image patch or glimpse, I_n, and its location tuple L_n, as inputs, where $L_n = I(x_n, y_n)$. It outputs a glimpse vector G_n. The job of the CNN based Glimpse Network is to extract a set of useful features from location L_n of the raw visual input. We use Gimage$(I_n|Wimage)$ to denote the output vector from function Gimage$(\cdot)$ that takes an image patch I_n and is parameterized by weights $Wimage$. In our implementation, Gimage$(\cdot)$ consists of two blocks of CONV–BN–ReLU–CONV–BN–ReLU–MAXPOOL (where CONV indicates a *Convolution layer* followed by *Batch Normalization* (BN) and *Rectified Linear Unit* (ReLU) as the activation function), followed by two fully-connected (FC) layers. Separately, the location tuple is mapped by Gloc$(L_n|Wloc)$ using a fully connected hidden layer and parameterized by weights $Wloc$. Both Gimage$(I_n|Wimage)$ and Gloc$(L_n|Wloc)$ have the same dimension. We combine the image information with the location tuple by multiplying the two vectors element-wise to get the final glimpse feature vector G_n, as follows:

$$G_n = \text{Gimage}(I_n|Wimage)\text{Gloc}(L_n|Wloc)$$

Recurrent Network:

The glimpse feature vector G_n from the glimpse network is supplied as input to the recurrent network at each time step. The recurrent network consists of two recurrent layers that contain stacked *Long-Short-Term Memory* (LSTM) units. The lower and upper recurrent layers are parameterized by weights Wr^1 and Wr^2, respectively. We define the two outputs of the recurrent layers as r^1 and r^2, where:

$$r^1_n = \text{LSTM}(G_n, r^1_{n-1}|Wr^1)$$
$$r^2_n = \text{LSTM}(r^1_n, r^2_{n-1}|Wr^2)$$

Emission Network:

The emission network E$(\cdot)$ parameterized by weights We, takes the current state of recurrent network as input and makes a prediction on where to extract the next image patch

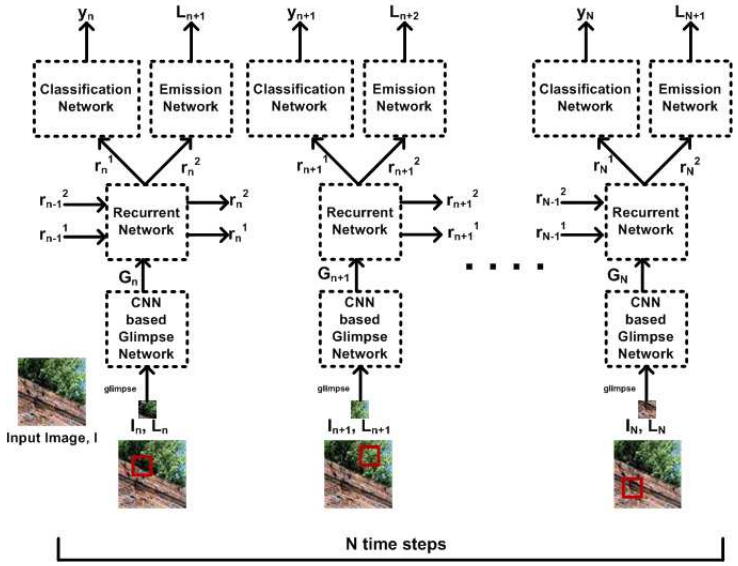

Figure 1. The Deep Convolutional Recurrent Attention Model. The input I_n is a patch from the input image I. At each time step n, the model focuses selectively on a given location l_n, extracts a feature vector G_n, updates its internal state and chooses the next location L_{n+1} to attend. The process is repeated for a fixed number of steps N, during which the model incrementally combines the information in a coherent manner to produce a final classification y_N.

for the glimpse network. It consists of a fully connected (FC) hidden layer that maps the feature vector, r_n^2 from the top recurrent layer to a coordinate tuple, and generates the location of the next location tuple:

$$L_{n+1} = \mathrm{E}(r_n^2|We)$$

Classification Network:

The classification network $O(\cdot)$ parameterized by weights Wo outputs a prediction for the class label y based on the final feature vector r_N^1 of the lower recurrent layer. The classification network has one fully connected hidden layer and a *softmax* output layer for the class y which outputs:

$$P(y|I) = \mathrm{O}(r_N^1|Wo)$$

Training and Optimization

As proposed by Mnih et al. [15], we used a hybrid supervised loss scheme in training to optimize the network. In particular, classification loss was defined as cross-entropy between the final prediction and the ground-truth label. However, the Emission Network has a non-differentiable transfer function, which means standard back-propagation techniques are not applicable. As proposed in [15, 16] we use a policy gradient method in the form of REINFORCE algorithm [20] to train this part of the model to select glimpse locations that lead to good classification results. Adam optimizer [21] is used as optimization method.

4. Experiments and Results

4.1. Dataset

In our experiments, 800 CG images were downloaded from the Columbia Dataset [22] while 1000 NIs were taken from the RAISE dataset [24]. All of these natural images were downloaded in TIFF format and converted to JPEG with a quality factor of 95. The images were divided into three subsets for training (70%), testing (20%) and validation (10%). We applied *patch augmentation* [14], i.e., we randomly cropped both the computer-generated graphics and the natural images into image patches sized at 30×30, 60×60, 120×120 and 240×240. Every patch was preprocessed by subtracting the per-pixel mean of all patches. In view of computational cost and diversity of image size, we adopted the local-to-global strategy [10] of training in small patches and classifying full-sized images using the simple majority voting rule. We also evaluated our work on the Rahmouni *et al.*'s dataset [11]. Rahmouni *et al.*'s CG images were downloaded from the Level-Design Reference Database [23] (a collection of video game screenshots) and PG images were taken from RAISE dataset.

4.2. Experimental Setup

All our experiments were conducted using *Chainer* [25], an open source deep learning framework and conducted on a GeForce GTX 1080 Ti GPU. At each time step, a glimpse from the input image is fed to the network and the model predicts parameters of extraction for the next iteration as well as a class label. Details of the glimpse size and the

Figure 2. Examples of computer generated images from (top row) Columbia Dataset [22], (middle row) Level-Design Reference Database [23] and (bottom row) natural images from RAISE Dataset [24]

.

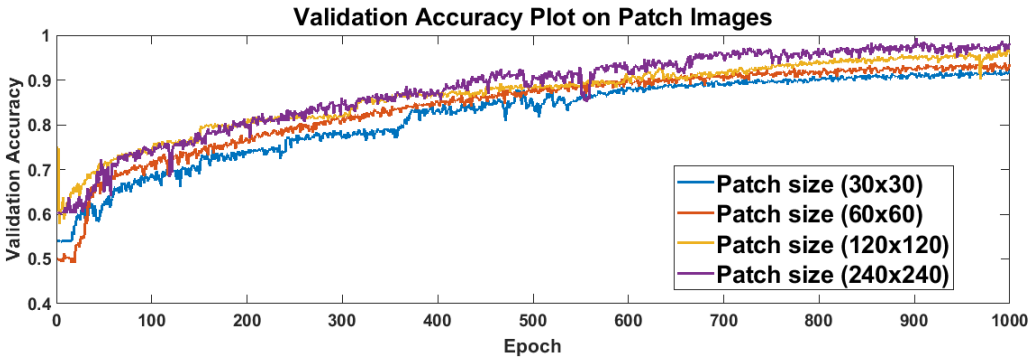

Figure 3. Validation accuracy during training for all patch size images

.

number of glimpses (the number of time steps = the number of glimpses per image) used for each patch size is given in Table 1. In this experiment, we use ReLU activation for all layers except the recurrent network, where standard tanh activation in LSTM units are employed. The size of filters in each convolution of the glimpse network is chosen to be 2×2 and the numbers of filters are 32 each. The max pooling layers are of size 2×2, and stride size of 2 pixels was used after second and fourth convolutional layers. There are 256 units in each LSTM layer and 256 hidden units in each fully-connected layer of the model.

During training, we set batch size to 128 images. The network parameters were optimized using the adaptive moment estimation (*Adam*) method [21] in a mini-batch manner with the size 128. We experimentally set the two different momentum values as $\beta 1 = 0.9$ and $\beta 2 = 0.999$ as in [21]. The learning rate was set to 0.0005 initially, and divided by 2 after every 300 epochs. Parameters in convolution layers and FC layers were initialized using the *Glorot uniform procedure* [26]. Biases in all layers are initialized to be zero. All weights in the recurrent network were initialized using normal distribution. Each experiment was conducted for 1000 epochs.

4.3. Experimental Results

In our experiments, all images in the testing dataset were clipped into image patches of size ranging from 30×30 to 240×240. These image patches were input to the trained model, and the prediction results for the image patches were obtained. The validation accuracy plot on different patch sizes is presented in Fig. 3. Our method is evaluated by computing the *Area Under Receiver Operating Characteristic* (ROC) curve (AUC) and the average classification accuracy. Based on the prediction results of the image patches, we deployed a majority vote scheme [10] to obtain the average classification accuracy for the full-size images. We compared our results with Quan et al. [10]. Quan et al. has proposed three different networks: NET-1 for patch size 240×240; NET-2 for patch size 60×60 and 120×120; and NET-3 for patch size 30×30. The average classification accuracies obtained by our method and that of Quan et al. [10] on patch-size images (ranging from 30×30 to

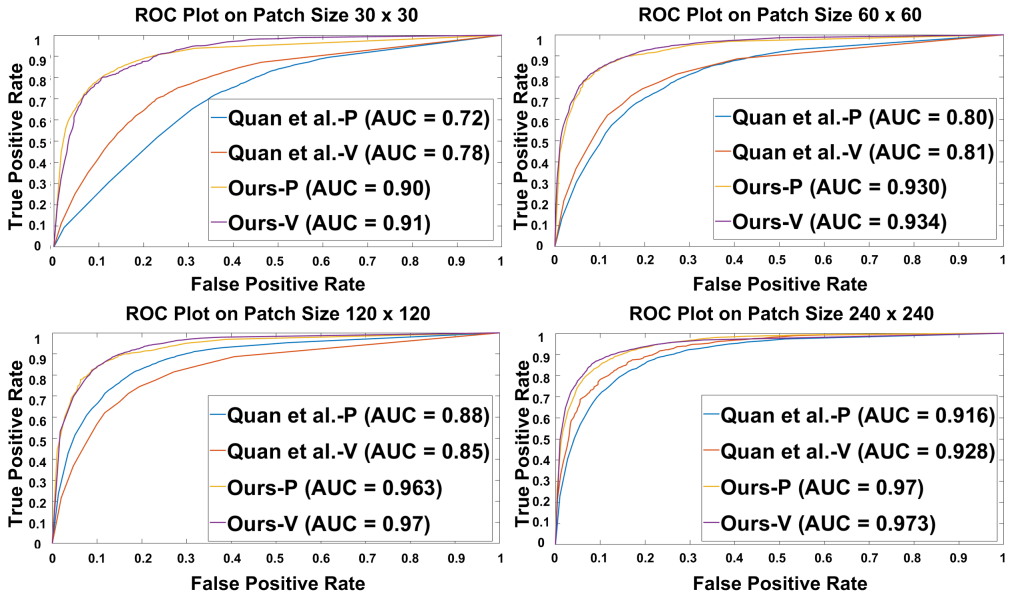

Figure 4. ROC curve comparison on the patch-size (P) testing data and full-size testing data after major voting (V) on the corresponding patch-size images

.

Table 1. Comparison of the average classification accuracy of the proposed technique with the CNN based classifier proposed in [10]

Methods	**Patch size (no. of glimpses)**	**Patch classification**	**Full size classification**
Proposed	30×30 (6 (8×8) glimpses)	90.90%	92.67%
Proposed	60×60 (6 (8×8) glimpses)	94.90%	94.55%
Proposed	120 × 120 (16 (8 × 8) glimpses)	96.90%	97.60%
Proposed	240×240 (16 (8×8) glimpses)	97.40%	97.20%
Quan et al. [10](NET-3)	30×30	73.90%	77.30%
Quan et al. [10](NET-2)	60×60	77.90%	84.30%
Quan et al. [10](NET-2)	120×120	89.00%	86.60%
Quan et al. [10](NET-1)	240×240	91.90%	92.00%

Table 2. Comparison of the average classification accuracy of the proposed technique with the CNN based classifier proposed in [11]

Dataset	**Methods**	**Patch classification**	**Full size classification**
RAISE vs. Columbia	Proposed	96.90%	97.60%
	Rahmouni *et al.* [11]	75.68%	86.49%
RAISE vs. Level-Design	Proposed	95.97%	98.76%
	Rahmouni *et al.* [11]	89.76%	99.30%

Table 3. Comparison of the number of network parameters of the proposed technique with the CNN based classifiers proposed in [10] and [11]

Methods	**No. of Parameters**
Proposed	1,200,901 (≈ 1.2M)
Quan *et al.* [10](NET-1)	17,265,458 (≈ 17.2M)
Quan *et al.* [10](NET-2 and NET-3)	4,440,485 (≈ 4.4M)
Rahmouni *et al.* [11]	2,840,34 (≈ 2.8M)

240 × 240 pixels) and full-size images are tabulated in Table 1. Also the number of network parameters are tabulated in Table 3. The total number of parameters was estimated by counting the total number of weight and bias parameters in the respective networks. It is observed that the accuracy improves when the patch size increases. Furthermore, ROC curves and AUC scores on patch-size and full-size images are depicted in Fig. 4. The average classification accuracy for both patch-size and full-size images of our method is always higher than the corresponding values of [10], while having substantially less number of parameters, as evident from Table 3. The number of parameters in the proposed model for patch size 30 × 30, 60 × 60 and 120 × 120 is 3.6 times less and for patch size 240 × 240 is 14.3 times less than that of [10].

Next, we also compared our work with that of Rahmouni

et al.'s method [11]. Rahmouni *et al.*'s dataset consists of 1800 CG images collected from the Level-Design Reference database [23] and 1800 PG images collected from RAISE database [24]. For comparison, images are divided into patches where for Rahmouni *et al.*'s method, test patch size images set to 100×100 are extracted as per their default setting described in [11] and we consider patch size 120×120 for our method. We compared our work not only on their dataset (RAISE vs. Level Design) but also on our test dataset (RAISE vs. Columbia). The average classification accuracies of both patch-size images and full-size images are tabulated in Table 2. Though Rahmouni *et al.* architecture is a three layer neural network with few number of parameters, but it can be observed from Table 2 that our proposed method achieved performance as good as that of Rahmouni *et al.*'s in case of RAISE vs. Level Design dataset but outperformed substantially in case of RAISE vs. Columbia dataset along with fewer number of parameters as evident from Table 3.

5. Conclusion

We have presented a Deep Convolutional Recurrent Attention Model that efficiently classifies computer generated images and natural images. The main ideas are the use of glimpse network to only process small patches of the input image, and the use of an attention mechanism to determine the next location for the image patch. The model outperforms the state-of-the-art works for both patch-size and full-size images, while having lesser number of parameters in the network model.

References

[1] Tian-Tsong Ng and Shih-Fu Chang, "Classifying photographic and photorealistic computer graphic images using natural image statistics," *Technical report, ADVENT Technical Report# 220–2006–6*, 2004. 1, 2

[2] Xiaofeng Wang, Yong Liu, Bingchao Xu, Lu Li, and Jianru Xue, "A statistical feature based approach to distinguish prcg from photographs," *Computer Vision and Image Understanding*, vol. 128, pp. 84–93, 2014. 1, 2

[3] Siwei Lyu and Hany Farid, "How realistic is photorealistic?," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 845–850, 2005. 1, 2

[4] Ruoyu Wu, Xiaolong Li, and Bin Yang, "Identifying computer generated graphics via histogram features," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 1933–1936. 1, 2

[5] Zhaohong Li, Jingyu Ye, and Yun Qing Shi, "Distinguishing computer graphics from photographic images using local binary patterns," in *The International Workshop on Digital Forensics and Watermarking 2012*. Springer, 2013, pp. 228–241. 1, 2

[6] Fei Peng, Jiao-ting Li, and Min Long, "Identification of natural images and computer-generated graphics based on statistical and textural features," *Journal of forensic sciences*, vol. 60, no. 2, pp. 435–443, 2015. 1

[7] Fei Peng and Die-lan Zhou, "Discriminating natural images and computer generated graphics based on the impact of cfa interpolation on the correlation of prnu," *Digital Investigation*, vol. 11, no. 2, pp. 111–119, 2014. 1

[8] Ahmet Emir Dirik and Nasir Memon, "Image tamper detection based on demosaicing artifacts," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 1497–1500. 1, 2

[9] Peisong He, Xinghao Jiang, Tanfeng Sun, and Haoliang Li, "Computer graphics identification combining convolutional and recurrent neural networks," *IEEE Signal Processing Letters*, vol. 25, no. 9, pp. 1369–1373, 2018. 1, 2

[10] Weize Quan, Kai Wang, Dong-Ming Yan, and Xiaopeng Zhang, "Distinguishing between natural and computer-generated images using convolutional neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2772–2787, 2018. 1, 2, 3, 4, 5

[11] Nicolas Rahmouni, Vincent Nozick, Junichi Yamagishi, and Isao Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *Information Forensics and Security (WIFS), 2017 IEEE Workshop on*. IEEE, 2017, pp. 1–6. 1, 2, 3, 5, 6

[12] Ye Yao, Weitong Hu, Wei Zhang, Ting Wu, and Yun-Qing Shi, "Distinguishing computer-generated graphics from natural images based on sensor pattern noise and deep learning," *Sensors*, vol. 18, no. 4, pp. 1296, 2018. 1, 2

[13] In-Jae Yu, Do-Guk Kim, Jin-Seok Park, Jong-Uk Hou, Sunghee Choi, and Heung-Kyu Lee, "Identifying photorealistic computer graphics using convolutional neural networks," in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4093–4097. 1

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. 1, 3

[15] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al., "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212. 1, 2, 3

[16] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv preprint arXiv:1412.7755*, 2014. 1, 2, 3

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al., "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025. 1

[18] Artsiom Ablavatski, Shijian Lu, and Jianfei Cai, "Enriched deep recurrent visual attention model for multiple object recognition," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017, pp. 971–978. 1

[19] Zijian Zhao, Xingming Wu, Peter CY Chen, and Weihai Chen, "General recurrent attention model for jointly multiple object recognition and weakly supervised localization," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 341–345. 1

[20] Ronald J Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992. 3

[21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 3, 4

[22] Tian-Tsong Ng, Shih-Fu Chang, Jessie Hsu, and Martin Pepeljugoski, "Columbia photographic images and photorealistic computer graphics dataset," *Columbia University, ADVENT Technical Report*, pp. 205–2004, 2005. 3, 4

[23] M. Piaskiewwicz, "Level-Design Reference Database," http://level-design.org/referencedb/, 2017. 3, 4, 6

[24] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato, "RAISE: a raw images dataset for digital image forensics," in *Proceedings of the 6th ACM Multimedia Systems Conference*. ACM, 2015, pp. 219–224. 3, 4, 6

[25] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, 2015, vol. 5, pp. 1–6. 3

[26] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256. 4