

Image Classification by Reinforcement Learning with Two-State Q-Learning

Abdul Mueed Hafiz¹

¹Department of Electronics and Communication Engineering
Institute of Technology, University of Kashmir
Srinagar, J&K, India, 190006

ORC-ID: 0000-0002-2266-3708

Email: mueedhafiz@uok.edu.in

Abstract. In this paper, a simple and efficient Hybrid Classifier is presented which is based on deep learning and reinforcement learning. Here, Q-Learning has been used with two states and 'two or three' actions. Other techniques found in the literature use feature map extracted from Convolutional Neural Networks and use these in the Q-states along with past history. This leads to technical difficulties in these approaches because the number of states is high due to large dimensions of the feature map. Because the proposed technique uses only two Q-states it is straightforward and consequently has much lesser number of optimization parameters, and thus also has a simple reward function. Also, the proposed technique uses novel actions for processing images as compared to other techniques found in literature. The performance of the proposed technique is compared with other recent algorithms like ResNet50, InceptionV3, etc. on popular databases including ImageNet, *Cats and Dogs* Dataset, and Caltech-101 Dataset. The proposed approach outperforms others techniques on all the datasets used.

Keywords: Image Classification; ImageNet; Q-Learning; Reinforcement Learning; ResNet50; InceptionV3; Deep Learning;

1. Introduction

Reinforcement Learning (RL) [1-4] has garnered much attention [4-13]. In computer vision, good initial work [7,14,15,11,9,5,12,16-20] has been undertaken. In [14], the authors aim to reduce the large computational costs of using large images, by proposing a RL agent which adaptively selects the resolution of every image provided by the detector. They train the agent with double rewards by choosing lower resolution images for a coarse level detector in case of the image being dominated by large objects, and higher resolution images for a fine level detector in case of the image being dominated by small objects. In [20], the authors propose an **object detection technique based on reinforcement Q-learning** [21,22]. They use a policy search based on analytic gradient computation with continuous reward. They report almost two orders of magnitude speed-up over other popular techniques found in literature. In [23], an adaptive deep Q-learning technique has been used for improving and shortening computational time for digit recognition. They refer to their novel network as Q-learning deep belief network (Q-ADBN). This network **extracts features from a deep auto-encoder** [24], which are considered as current states of Q-learning algorithm. After conducting experiments on MNIST dataset [25], the authors of the above work claim that their technique is superior to other techniques in terms of accuracy and running time. In [26], the authors propose an object detection approach which uses zooming and translation for successively refining the bounding box for the objects. They use a VGG-16 Convolutional Neural Network (CNN) [27] and concatenate its feature map with a history vector, and this new map is fed to a Q-network for further processing, leading to interesting results. In [28], the authors propose a technique in which an agent learns to deform a bounding box using simple transformation actions, with the aim of obtaining specific location of objects by following top-down reasoning. The actions used are horizontal moves, vertical moves, scale changes and aspect ratio changes.

Taking a hint from the work in area of Maximum Power Point Tracking (MPPT) which is used in Photovoltaic Arrays[29], a simple and efficient technique has been proposed for image classification which gives high accuracy. It is based on deep learning as well as reinforcement learning. The technique involves using **feature maps obtained from a pre-trained CNN like ResNet50** [30], InceptionV3 [31], or AlexNet [32]. Next, reinforcement learning is used for **optimal action proposal generation** (rotation by a specific angle or translation) on the image. After application of the final action to the original test image, and obtaining feature map from the CNN [33-36,30,31,37], classification is done using a second classifying structure, like a Support Vector Machine (SVM) [38-40] or a Neural Network (NN) which has been trained on the CNN feature maps of training images.

Many reinforcement learning techniques [20,18,26,28] used for image classification use actions like zoom and translation according to visual detection in humans. Thus they miss the important action of rotating the field of view used in human visual image comprehension. In this paper rotation of image by specific angle(s) has been used which is novel in itself. Also Q-Learning has been used in reinforcement learning. Q-states which have been used in the other approaches of reinforcement learning based object detection, use **features with high dimensions combined with**

state history. This technique usually leads to large state-space, in turn leading to optimization problems. Addressing these problems was the motivation behind this work. The proposed technique uses only two states, and two or three actions. As a consequence of this strategy, the Q-table has two rows and two/three columns. To the best of available knowledge, this is the first technique using two Q-states, as well as using image rotation as an action. As a result, the overall task of using reinforcement learning in computer vision becomes simple and also becomes efficient. Better results are obtained in comparison to other techniques involving CNNs like ResNet50 [41,30,37], InceptionV3 [31], etc.

The major highlights of the paper are:

- A novel image recognition technique is proposed which is based on Reinforcement Learning. This is a first to the best of knowledge.
- Rotation of image has been used for image recognition, which is akin to tilting of vision. This is also a first to the best of knowledge.
- The proposed technique outperforms others on all the datasets used in the current work.

The rest of the paper is structured as follows. Section 2 discusses the proposed approach. Section 3 gives a brief description of the various datasets used in the study. Section 4 discusses the experimentation. Conclusion is presented in Section 5.

2. Proposed Approach

In this paper a hybrid approach of deep learning [42,7,14,43-46] and Reinforcement Learning (RL) is proposed. The CNNs used are ResNet50 [41,30,37], InceptionV3 [31], and AlexNet [32]. First, feature-map of the CNN is obtained for every training sample by feeding it to the CNN. Let the set of all of these feature maps be referred to as \mathbf{F}_{Train} . A secondary classifier like a Support Vector Machine, or a Neural Network is trained on \mathbf{F}_{Train} . For classifying a test sample, a filtering criterion for 'hard to classify' samples must be used. If the test sample is tagged as hard, it is classified by reinforcement learning. If not, then CNN is used for classification. This paper is not about filtering criteria, hence no such criterion has been used except that all test samples misclassified by the CNN are tagged as 'hard to classify.' Every 'hard' test sample is first fed to the CNN. Next, the feature map of the CNN, viz. \mathbf{F}_{Sample} for the test image is obtained. Reinforcement learning based classification is done as follows. A random action is selected from a bank of actions whose number does not exceed 3 in all the experiments. Next, the action permutation is applied to the test image. Then the new feature map (\mathbf{F}'_{Sample}) of the permuted image is obtained from the CNN. The action permutations used in this work are mainly image rotation with specific angles and sometimes diagonal translation. For reinforcement learning, Q-Learning with random policy is used. Two states ($n = 2$) and 'two or three' actions ($a = 2$ or $a = 3$) are used. The current state is decided after observing a metric viz. 'standard deviation' of the prediction scores (of the second classifier) before and after applying the permutation. Let \mathbf{M} be metric for original image and \mathbf{M}_1 be metric after applying current action. The new state is decided based on the criteria whether \mathbf{M}_1 is lesser than, equal to, or larger than \mathbf{M} respectively. The Q Table having 2 rows ($n=2$) and a columns ($a = 2$ or 3) is initialized to zero. Reward r is based on the comparison as shown below:

$$r = \begin{cases} +1, & \text{if } \mathbf{M}_1 > \mathbf{M} \\ 0, & \text{if } \mathbf{M}_1 = \mathbf{M} \\ -1, & \text{if } \mathbf{M}_1 < \mathbf{M} \end{cases} \quad (1)$$

Number of iterations for updating the Q Table is $N = a \times m$, where $a=2$ or 3 and m is a constant (usually 20). After each iteration, the Q value entry for the current 'state-action pair' with state s and action a i.e. $Q(s,a)$ present in the Q-Table, is updated as per the Q-Learning Update Rule:

$$Q(s,a) = Q(s,a) + \alpha [r + \gamma \max_{b \in A} Q(s',b) - Q(s,a)] \quad (2)$$

Where s' is new state, the learning rate $\alpha = 0.4$, and the discount rate $\gamma = 0.3$. Flowchart for the proposed RL algorithm is given in Figure 1. After completing N iterations of Q-Learning, optimal action is chosen as the action having highest value in Q-Table. Finally the optimal action is applied to the original sample/test image. Next the image is fed to the CNN giving its feature map. This feature map is fed to the second classifying structure for classification. Figure 2 shows the proposed approach in modular fashion.

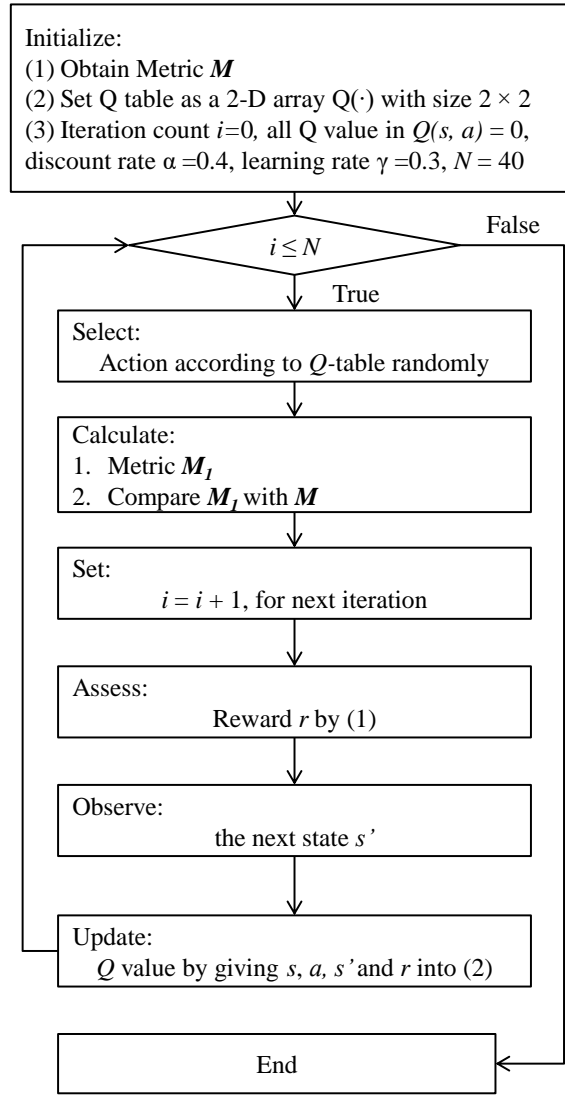


Figure 1. Flowchart of the proposed RL Algorithm

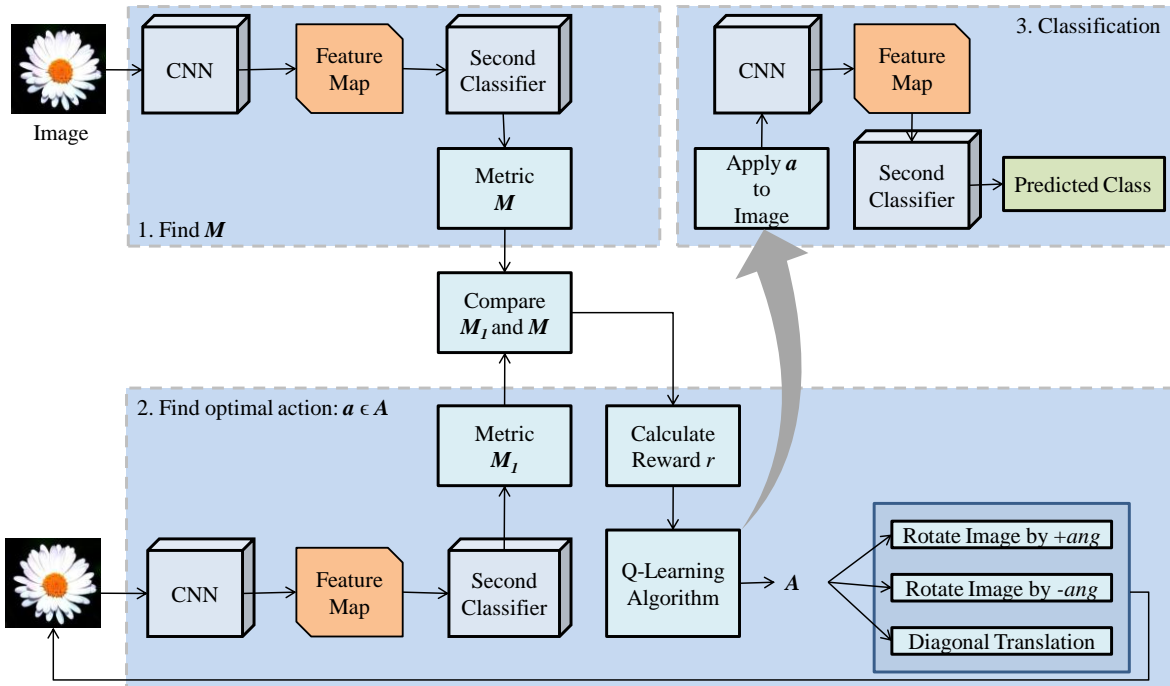


Figure 2. Proposed Technique

The NNs used on top of ResNet50 and InceptionV3 networks are shown in Figure 3(a) and 3(b), respectively.

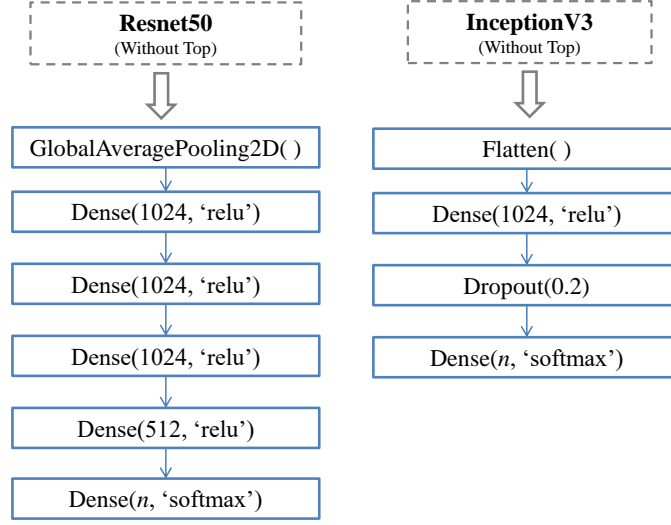


Figure 3. Shown are NNs used after the CNNs, viz. ResNet50 (left), and InceptionV3 (right) which are implemented in Tensorflow without top; n is number of classification categories

3. Datasets Used

Benchmarking has been done on three popular databases viz. ImageNet [47,48], *Cats and Dogs* Dataset [49], and Caltech-101 Dataset [50].

3.1 ImageNet

ImageNet[47] is a large image database for use in visual object recognition. In it, more than 14 million images are hand-annotated. Also at least in one million of the images, bounding boxes have also been provided. ImageNet has more than 20,000 categories with a each category typically consisting of hundreds of images. The annotation database of third-party image URLs is available directly from the ImageNet website. Since 2010, the ImageNet project runs an annual programming contest viz. the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). In this various algorithms compete for correct classification and detection of objects and scenes. This challenge uses 1000 non-overlapping object categories.

3.2 Cats and Dogs Dataset

Web services have often been protected with challenges that are relatively easier for people to solve as compared to computers, e.g. CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) or HIP (Human Interactive Proof). HIPs have been used for various purposes like email and blog spam reduction, and prevention of brute-force attacks on web sites having passwords. ASIRRA (Animal Species Image Recognition for Restricting Access) is a HIP that asks users to identify photographs of cats and dogs. The task is difficult for computers. However studies have shown that humans accomplish it quickly and accurately. ASIRRA has partnered with Petfinder.com, to provide Microsoft Research with more than three million images of cats and dogs, which have been manually classified by people. Kaggle offers a subset of this dataset [49] for research.

3.3 Caltech-101

Caltech-101 [50] is a database of images compiled by Fei-Fei Li, Marco Andreetto, Marc 'Aurelio Ranzato and Pietro Perona at the California Institute of Technology. It is intended to be used for computer vision research. It is mainly applicable to techniques for image recognition, classification and categorization. Caltech-101 contains 9,146 images, across 101 distinct object categories (e.g. faces, watches, ants, pianos, etc.). It also has a set of annotations which describe the outlines in each image.

The distribution of data amongst the experimental setups is shown in Table 1.

Table 1. Distribution of Data Experimentally

Dataset	Classes Used	Training Images	Validation Images	Testing Images
ImageNet [47]	4 (Bikes, Ships, Tractors, Wagons)	1531	788	745
Cats and Dogs [49]	2 (Cats, Dogs)	2000	500	500
Caltech-101 [51]	50	750	-	1250

4. Experimentation

Experiments were conducted on a machine having an *Intel® Xeon®* processor (with 2 Cores), 12.6 GB RAM and 12 GB GPU. For benchmarking of the performance of the proposed technique, its performance was compared with that of the pre-trained CNN used alone after being fine-tuned on the datasets. Tensorflow [52] has been used for implementing the CNNs (pre-trained, having ImageNet weights) and algorithms. For training the CNNs using transfer learning, 10 *training epochs* were used, with optimizer: *Adam*, Loss: *Categorical Crossentropy*, and Learning Rate: *0.001*. Benchmarking has been done on three popular databases viz. ImageNet [47,48], *Cats and Dogs* Dataset [49], and Caltech-101 Dataset [50].

Tables 2 to 4 show the benchmarking for the performance of the proposed approach on the datasets used. It should be noted that for conventional CNN usage no rotation is done during evaluation. It should be noted that for **Caltech-101**, a two action set comprising of angular rotation by 12.5° or by -12.5° gave best results. For **ImageNet** and **Cats and Dogs Dataset**, a three action set comprising of angular rotation by 90° , or by 180° , or *downward and rightward* diagonal translation by 15 pixels gave best results. The action sets gave best results, as compared to others including no-rotation action, etc. It should be noted that rotation permutation during evaluation leads to better results than the conventional CNNs which do not use rotation. This can be due to alignment between test image and previous training image (due to rotation during evaluation) in the CNN. One advantage of this technique is that training need not be done extensively, thus saving resources like memory and time considerably for systems having both offline and online training. This statement is subject to speculation and will be more revealed and explained in future work.

Table 2. Classification Accuracy of Various Approaches on ImageNet
(**Second Classifier Used:** NN; **Metric Used:** Std. Deviation of Softmax scores, **Feature Map Size:** 1x1024)

Approach	Secondary NN used on top of Layer	Image Size	Accuracy
ResNet50	#174: @(conv5_block3_out)	150x150x3	.8242
Proposed Approach using ResNet50	#174: @(conv5_block3_out)	150x150x3	.8309
Inception V3	#228: @(mixed7)	150x150x3	.8564
Proposed Approach using InceptionV3	#228: @(mixed7)	150x150x3	.8644

Table 3. Classification Accuracy of Various Approaches on *Cats and Dogs* Dataset (**Second Classifier Used:** NN; **Metric Used:** Std. Deviation of Softmax scores, **Feature Map Size:** 1x1024)

Approach	Secondary NN used on top of Layer	Image Size	Accuracy
ResNet50	#174: @(conv5_block3_out)	224x224x3	.9780
Proposed Approach using ResNet50	#174: @(conv5_block3_out)	224x224x3	.9860
Inception V3	#228: @(mixed7)	150x150x3	.9440
Proposed Approach using InceptionV3	#228: @(mixed7)	150x150x3	.9520

As is observed from Table 2, larger image sizes lead to higher accuracies. However, the training times also increase.

Table 4. Classification Accuracy of Various Approaches on Caltech-101 Dataset (**Second Classifier Used:** Binary-SVM Ensemble; **Metric Used:** Std. Deviation of SVM prediction scores, **Feature Map Size:** 1x4096)

Approach	Secondary SVM used on top of Layer	Image Size	Accuracy
AlexNet	-	227x227x3	.841
CNN-SVM Hybrid Approach [53]	#20: @(fc7)	227x227x3	.882
Proposed Approach using AlexNet	#20: @(fc7)	227x227x3	.898

As is observable from Table 2 to 4, the proposed approach outperforms other approaches on all the datasets used. This technique can pave the way for a novel image recognition approach using tilt of vision in classifiers, etc. Also, the use

of reinforcement learning in computer vision is a first, to the best of available knowledge. This technique can pave the way for a whole new generation of computer vision classifiers based on reinforcement learning. It should also be noted that in this study dimensional reduction [54] is not used. Also, that the proposed approach is instance-based. Thus the processing time is more as compared to other techniques in this area. This is one of the limitations which is intended to be addressed in future work. In future, more work will be done on using larger datasets. Also, work would be done on making the proposed approach faster by using techniques like dimensional reduction, or using smaller feature maps. Also, work would be done on using the proposed approach on other interesting computer vision tasks like instance segmentation [55], etc.

5. Conclusion and Future Work

In this paper, a straightforward and efficient learning system is investigated which combines deep learning with reinforcement learning. The proposed technique is simpler than other contemporary techniques found elsewhere. This is for the reason that others use high number of states while as the proposed approach uses only two states. Thus optimization is easy and the reward function is straightforward. Other approaches use visualization tasks like zoom and translation. A novel technique i.e. rotation has been used which is similar to tilt of visual field. Three databases have been used in the experimentation here. These are ImageNet, *Cats and Dogs* Dataset, and Caltech-101 Dataset. Benchmarking of the proposed classifier has been done. The proposed approach outperforms other approaches including ResNet50, InceptionV3, etc. on all the three datasets used.

Declarations

The authors declare no conflict of interest.

References

1. Arulkumaran, K., Deisenroth, M. P., Brundage, M., Bharath, A. A., A Brief Survey of Deep Reinforcement Learning, *arXiv:170805866v2*, 28 Sep 2017
2. Sutton, R. S., Barto, A. G., *Reinforcement Learning: An Introduction*. The MIT Press. Nov 5, 2017
3. Bernstein, A., Burnaev, E., Reinforcement learning in computer vision, in., April 2018.
4. Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., Hassabis, D., Reinforcement learning, fast and slow, *Trends in cognitive sciences*, 23, (5):408-422, 2019
5. Liu, Z., Wang, J., Gong, S., Lu, H., Tao, D., Deep reinforcement active learning for human-in-the-loop person re-identification, in: *Proceedings of the IEEE International Conference on Computer Vision*. pp 6122-6131, 2019.
6. Wirth, C., Akrou, R., Neumann, G., Fürnkranz, J., A Survey of Preference-Based Reinforcement Learning Methods, *Journal of Machine Learning Research*, 18:1-46, December, 2017
7. Gärtner, E., Pirinen, A., Sminchisescu, C., Deep Reinforcement Learning for Active Human Pose Estimation, in: *AAAI*. pp 10835-10844, 2020.
8. Wiering, M. A., Hasselt, H. v., Pietersma, A.-D., Schomaker, L., Reinforcement Learning Algorithms for solving Classification Problems,
9. Furuta, R., Inoue, N., Yamasaki, T., Fully convolutional network with multi-step reinforcement learning for image processing, in: *Proceedings of the AAAI Conference on Artificial Intelligence*. pp 3598-3605, 2019.
10. Lagoudakis, M. G., Parr, R., Reinforcement Learning as Classification: Leveraging Modern Classifiers, in: *ICML*. 2003.
11. Toromanoff, M., Wirbel, E., Moutarde, F., Deep Reinforcement Learning for autonomous driving, in: *Workshop on "CARLA Autonomous Driving challenge", IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019.
12. Jiang, M., Hai, T., Pan, Z., Wang, H., Jia, Y., Deng, C., Multi-agent deep reinforcement learning for multi-object tracker, *IEEE Access*, 7:32400-32407, 2019
13. Hafiz, A. M., Bhat, G. M., Deep Q-Network Based Multi-agent Reinforcement Learning with Binary Action Agents, *arXiv preprint arXiv: 200804109*, 2020
14. Uzkent, B., Yeh, C., Ermon, S., Efficient object detection in large images using deep reinforcement learning, in: *The IEEE Winter Conference on Applications of Computer Vision*. pp 1824-1833, 2020.
15. Zhang, D., Han, J., Zhao, L., Zhao, T., From discriminant to complete: Reinforcement searching-agent learning for weakly supervised object detection, *IEEE Transactions on Neural Networks and Learning Systems*, 2020
16. König, J., Malberg, S., Martens, M., Niehaus, S., Krohn-Grimberghe, A., Ramaswamy, A., Multi-stage Reinforcement Learning for Object Detection, in: *Science and Information Conference*. Springer, pp 178-191, 2019.
17. Pais, G. D., Dias, T. J., Nascimento, J. C., Miraldo, P., OmniDRL: Robust pedestrian detection using deep reinforcement learning on omnidirectional cameras, in: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, pp 4782-4789, 2019.
18. Pirinen, A., Sminchisescu, C., Deep Reinforcement Learning of Region Proposal Networks for Object Detection,

in: *CVPR*. 2018.

19. Hierarchical Object Detection with Deep Reinforcement Learning, in: *NIPS*. 2016.
20. Mathe, S., Pirinen, A., Reinforcement Learning for Visual Object Detection, in: *CVPR*. June 2016.
21. Watkins, C. J. C. H., Learning from Delayed Rewards. Ph.D. Thesis, University of Cambridge, 1989
22. Watkins, C. J. C. H., Dayan, P., Q-learning, *Machine Learning*, 8, (3-4):279-292, 1992
23. Qiao, J., Wanga, G., Li, W., Chen, M., An adaptive deep Q-learning strategy for handwritten digit recognition, *Neural Networks*, 107:61-71, 2018
24. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F. E., A survey of deep neural network architectures and their applications, *Neurocomputing*, 234:11-26, 2017
25. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86, (11):2278-2324, 1998
26. König, J., Malberg, S., Martens, M., Niehaus, S., Krohn-Grimberghe, A., Ramaswamy, A., Multi-stage Reinforcement Learning for Object Detection, *arXiv:181010325v2*, 26 Oct 2018
27. Simonyan, K., Zisserman, A., Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:14091556*, 2014
28. Caicedo, J. C., Lazebnik, S., Active Object Localization with Deep Reinforcement Learning,
29. Hsu, R. C., Liu, C.-T., Chen, W.-Y., Hsieh, H.-I., Wang, H.-L., A Reinforcement Learning-Based Maximum Power Point Tracking Method for Photovoltaic Array, *International Journal of Photoenergy*, 2015:12, 2015
30. He, K., Zhang, X., Ren, S., Sun, J., Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 770-778, 2016.
31. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., Rethinking the inception architecture for computer vision, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 2818-2826, 2016.
32. Krizhevsky, A., Sutskever, I., Hinton, G. E., Imagenet classification with deep convolutional neural networks, in: *Advances in neural information processing systems*. pp 1097-1105, 2012.
33. LeCun, Y., Bengio, Y., Hinton, G., Deep learning, *nature*, 521, (7553):436, 2015
34. Schmidhuber, J., Deep learning in neural networks: An overview, *Neural networks*, 61:85-117, 2015
35. Goodfellow, I., Bengio, Y., Courville, A., *Deep learning*. MIT press. 2016
36. Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R. M., Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, *IEEE transactions on medical imaging*, 35, (5):1285-1298, 2016
37. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., Aggregated Residual Transformations for Deep Neural Networks, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp 5987-5995, 21-26 July 2017 2017.
38. Cortes, C., Vapnik, V., Support vector machine, *Machine learning*, 20, (3):273-297, 1995
39. Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., Scholkopf, B., Support vector machines, *IEEE Intelligent Systems and their applications*, 13, (4):18-28, 1998
40. Chang, C. C., Hsu, C. W., Lin, C. J., Practical Guide to Support Vector Classification. 2009.
41. He, K., Zhang, X., Ren, S., Sun, J., Identity mappings in deep residual networks, in: *European conference on computer vision*. Springer, pp 630-645, 2016.
42. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M., Deep Learning for Generic Object Detection: A Survey, *International Journal of Computer Vision*, 128, (2):261-318, 2020
43. Hafiz, A. M., Bhat, G. M., A Survey of Deep Learning Techniques for Medical Diagnosis, in, Singapore, Information and Communication Technology for Sustainable Development. Springer Singapore, pp 161-170, 2020.
44. Hafiz, A. M., Bhat, G. M., Multiclass Classification with an Ensemble of Binary Classification Deep Networks, *arXiv preprint arXiv:200701192*, 2020
45. Hafiz, A. M., Bhat, G. M., Fast Training of Deep Networks with One-Class CNNs, *arXiv preprint arXiv:200700046*, 2020
46. Hafiz, A. M., Bhat, G. M., Deep Network Ensemble Learning applied to Image Classification using CNN Trees, *arXiv preprint arXiv:200800829*, 2020
47. Deng, J., Dong, W., Socher, R., Li, L., Kai, L., Li, F.-F., ImageNet: A large-scale hierarchical image database, in: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. pp 248-255, 20-25 June 2009 2009.
48. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., ImageNet Large Scale Visual Recognition Challenge, *International Journal of Computer Vision*, 115, (3):211-252, 2015
49. Parkhi, O. M., Vedaldi, A., Zisserman, A., Jawahar, C. V., Cats and dogs, in: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. pp 3498-3505, 16-21 June 2012 2012.
50. Fei-Fei, L., Fergus, R., Perona, P., Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, in: *2004 conference on computer vision and pattern recognition workshop*. IEEE, pp 178-178, 2004.
51. Li, F.-F., Andreetto, M., Ranzato, M. A. Caltech 101.2004
52. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Tensorflow: A system for large-scale machine learning, in: *12th {USENIX} symposium on operating systems design*

and implementation (*OSDI* 16). pp 265-283, 2016.

53. Tang, Y., Deep learning using linear support vector machines, *arXiv preprint arXiv:13060239*, 2013

54. Maaten, L. J. P. v. d., Hinton, G. E., Visualizing High-Dimensional Data Using t-SNE, *Journal of Machine Learning Research*, 9:2579-2605, 2008

55. Hafiz, A. M., Bhat, G. M., A survey on instance segmentation: state of the art, *International Journal of Multimedia Information Retrieval*, 9, (3):171-189, 2020